ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 4, Issue 3, March 2017

A New Parallel VLSI Architecture for Real-time Electrical Capacitance Tomography

[1] G.Dillirani, [2] A.Meghana
[1] Professor Dept.of.ECE, [2] M.Tech – ECE (VLSI-SD)
Chadalawada Ramanamma Engineering College,
Tirupati-517501 AndhraPradesh

Abstract: **-- This paper presents a fixed-point reconfigurable parallel VLSI hardware architecture for real-time Electrical Capacitance Tomography (ECT). It is modular and consists of a front-end module which performs precise capacitance measurements in a time multiplexed manner using Capacitance to Digital Converter (CDC) technique. Another FPGA module performs the inverse steps of the tomography algorithm. A dual port built-in memory banks store the sensitivity matrix, the actual value of the capacitances, and the actual image. A two dimensional (2D) core multi- processing elements (PE) engine intercommunicates with these memory banks via parallel buses. A Hardware-software codesign methodology was conducted using commercially available tools in order to concurrently tune the algorithms and hardware parameters. Hence, the hardware was designed down to the bit-level in order to reduce both the hardware cost and power consumption, while satisfying real-time constraint. Quantization errors were assessed against the image quality and bit-level simulations demonstrate the correctness of the design. Further simulations indicate that the proposed architecture achieves a speed-up of up to three orders of magnitude over the software version when the reconstruction algorithm runs on 2.53 GHZ-based Pentium processor or DSP Ti's Delphino TMS320F32837 processor. More specifically, a throughput of 17.241 Kframes/sec for both the Linear-Back Projection (LBP) and modified Landweber algorithms and 8.475 Kframes/sec for the Landweber algorithm with 200 iterations could be achieved. This performance was achieved using an array of [2×2] × [2×2] processing units. This satisfies the real-time constraint of many industrial applications. To the best of the authors' knowledge, this is the first embedded system which explores the intrinsic parallelism which is available in modern FPGA for ECT tomography.**

Keywords: **-- ECT, FPGA, matrix multiplication decomposition.**

I. INTRODUCTION

Electrical Capacitance Tomography (ECT) is an effective technique to measure a process non-intrusively by reconstructing the 2D or 3D dielectric distribution of its different constituencies [1][2][3]. This makes ECT a good candidate for several industrial applications such as two-phase flow monitoring, quality control in manufacturing industry, and corrosion detection [4]. In the last few years, several ECT systems have already been suggested [1]. Most of them used a desktop computer as a main processing unit and focused their research mainly on improving the accuracy and execution time of the algorithms on general purpose computers. Nevertheless, the tremendous computation complexity of the tomography algorithms let them to be executed still relatively slowly (in few seconds order), preventing them to sustain real-time applications. Hence, only few research works have been done on designing dedicated hardware architectures for ECT tomography using either DSP processor or FPGA technology. In [5], a dedicated architecture based on the DSP processor, TMS320C6701, operating at 133 MHz was suggested for the image reconstruction algorithm. The system consists of 12 electrodes and claimed to be able to achieve the image reconstruction of images of 480 pixels within a throughput of 135 frames/s, when using the Linear Back Project (LBP) algorithm. Similarly, in [6] a TMS3206416 processor was used in LBP-based ECT system with 16 electrodes to achieve a throughput of 200 frames/s.

The usage of FPGA instead of DSP processors by other researchers was motivated by the fact that FPGA can host a good portion of the control/data flow of the reconstruction algorithm, without the need of series of sequential memory accesses. Hence, recently, an FPGA-based Electrical Impendence Tomography (EIT) system was disclosed in [7]. The system comprises several FPGA-based Impedance Measurement Modules (IMM) comprising independent current sources and voltmeters intercommunicating through a reconfigurable FPGA-based Intra-network controller. Hence, the system could achieve a throughput of around 100 frames/second. The relatively low performance of the system can be

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 4, Issue 3, March 2017

justified by the fact that the IMMs modules are mainly involved in data acquisition and electric current generation for the electrodes, rather than in the image reconstruction task itself. Another similar FPGA-based ECT device was disclosed to digitally modulate (i.e. amplitude and phase modulations) the ac signal with various carrier frequencies and let the receivers to band-pass filter the transmitted signal to their allocated carrier frequency [8]. This allows simultaneous data acquisition as well as avoiding the usage of CMOS analog switches.

Other works were also suggested for FPGA-based image reconstruction targeting other tomography modalities such as computed tomography (CT) [9] and Single-Photon Emission Computed Tomography (SPECT) [10]. In [10], a dedicated architecture for SPECT imaging using a scalable multicore on- Chip Architecture for on-chip communication was proposed. The hardware which consists of 128 processing elements mapped onto an FPGA module could achieve a speed-up by a two orders of magnitude over 2 GHz Intel core 2 Duo Processor by exploring the intrinsic fine parallelism exhibited by the corresponding algorithm. A reconfigurable and modular interconnection switching network was used to dynamically and arbitrarily interconnect the processors array by eliminating data starvation. Other architectures using Graphics Processing Unit (GPU) for SPECT Tomography were also suggested by exploring their significant performance for arithmetically intensive algorithms that have structured data accesses and limited branching, which is the case of SPECT tomography algorithms [11]. However, compared to FPGA, their performance was lower because of the frequency and randomness of memory access which was off-chip. In [12], a Spartan 6 FPGA-based system for Dual Head Positron Emission Tomography (PET) was used to fully implement the constant fraction discriminator (CFD) algorithm to achieve time-resolution higher than the sampling period. The system consists of a 32-bit DSP RISC processor, running at 66.67 MHZ, interfaced with a custom core to perform event timing, energy determination-discrimination, position determination, and coincidence processing in real-time.

In this paper, we present a new architecture of ECT tomography algorithm using FPGA technology. The architecture is flexible enough to host three computationally intensive reconstruction algorithms, namely the LBP, Landweber, and modified-Landweber algorithms. It explores some of the new features of recent FPGA devices such as the DSP blocks and the Block RAM (BRAM). A simultaneous matrix-level and bit level designs was conducted to finely reduce the hardware cost and power consumption, while keeping the overall execution of the algorithms low.

To the best of the authors' knowledge, this is the first embedded system which explores the intrinsic parallelism which is available in modern FPGA for ECT tomography. Extensive simulations indicate that the proposed architecture achieves a speed-up of up to three orders of magnitude over the software version when the reconstruction algorithm runs on 2.53 GHZ-based Pentium processor. More specifically, a throughput of 17.241 Kframes/sec for both the Linear-Back Projection (LBP) and modified Landweber algorithms and 8.475 Kframes/sec for the Landweber algorithm with 200 iterations could be achieved. This performance was achieved using an array of [2×2] × [2×2] processing units. This satisfies the real-time constraint of many industrial applications.

II. SYSTEM OVERVIEW

Atypical ECT system consists of a set of electrodes which surround the process under investigation, a multiplexing unit to provide different combinations of capacitance channels, and a computing device (usually a general purpose computer) to run the tomography algorithms. However, as it will be justified in the next section, these reconstruction algorithms usually require significant computation time making dedicated hardware algorithms necessary to achieve real-time performance. This reason, in addition to the fact that a PC computer is not always suitable to be deployed in the field motivated us to suggest a new parallel and dedicated hardware architecture. The sequence of the data acquisition consists

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 4, Issue 3, March 2017

to enable one single pair of electrodes and read their corresponding capacitance value, while grounding all the other electrodes. This is done for all possible combinations of electrodes leading to 28 different capacitance values for an ECT system having 8 electrodes. Hence, for a system consisting of N electrodes, $M = N(N-1)/2$ different readings are required.

The linearized model of the forward problem of an ECT system, when noises are ignored can be expressed as follows:

$$C = S \times G \qquad (1)$$

Where: C is the capacitance value , G the target image matrix and, S the sensitivity matrix which can be defined for each element, k, of the pixels as follows:

$$S_{i,j}(k) = \frac{C^{h}_{i,j}(k) - C^{l}_{i,j}}{C^{h}_{i,j} - C^{l}_{i,j}} \frac{1}{\varepsilon_h - \varepsilon_l} \qquad (2)$$

The fact that the number of measurements is less than the number of pixels in the image make the problem ill posed and usually require several iterations of the inverse problem which can be defined as follows:

$$G^{*} = S^{*} \times C^{*} \qquad (3)$$

Solving Equation (3) above is done usually using either iterative or non-iterative algorithms. These algorithms usually exhibit accuracy-execution time tradeoffs and mainly require regular matrix multiplications. This makes them suitable for parallel VLSI hardware architectures. Linear BackProject (LBP), Landweber, and modified Landwber algorithms are practically the most used algorithms in process imaging. They are governed respectively by the following equations:

$$G = S^{T} \times C \qquad (4)$$

$$G_{k+1} = G_k - \lambda S^{T}(S \times G_k - C) \qquad (5), \text{ and}$$

$$D_{k+1} = (I - \lambda S^{T} S) D_k + \lambda S^{T} \qquad (6)$$

Where:

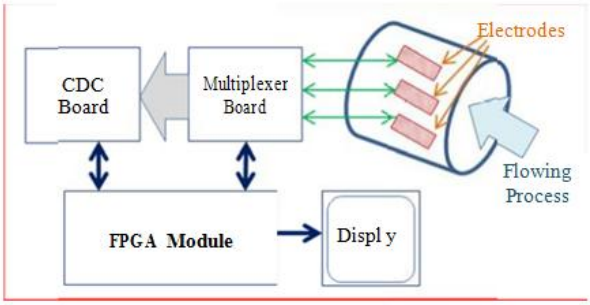
- λ is a positive scalar, which decide the k^{th} step size.
- $G_k = D_k \times C \qquad (0 \le k \le z)$

As it will be shown in Section 4, Landweber algorithm has the advantage to provide better accuracy than LBP algorithm, but at the expense of a higher time complexity, which increases linearly with the number of iterations, k, as $O(k \times M \times P \times P)$. The modified Landweber algorithm offers a compromise between the two algorithms by providing better S/N ration than LBP, but a same time complexity of $O(M \times P \times P)$. However, due to the large number of multiply and accumulate operations, all the three algorithms are not suitable to run on a serial general purpose machine to meet real-time requirements of process imaging. For instance, a desktop computer with a 2.53 GHz – intel CORE i5^{TM} processor and having 4 GB of RAM memory for a 32 x 32 image performs the LBP algorithm in more than 1.5 s.

III. ECT HARDWARE IMPLEMENTATION

Hardware Architecture Overview

Figure 1 shows the overall hardware architecture of the suggested system. It is modular and divided into five main modules: Capacitance electrodes, Multiplexer module, Capacitance to Digital Converter (CDC) board, FPGA module, and Display Interface module. These modules are interconnected to each other via various analog and digital buses.



(a)

Figure 1 Hardware system overview

The CDC module performs the analog to digital conversion of the capacitance, the electrodes of which are selected in time multiplexed manner by the FPGA module, via the Multiplexer board. The CDC module provides a very high S/N ratio and very high sensitivity, of aF order, which is usually required in ECT tomography.

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 4, Issue 3, March 2017

FPGA Module

This constitutes the core engine of our ECT system since it performs the image reconstruction task. Figure 2 shows the detailed VLSI architecture. It is modular and divided into four main modules: A parallel processing module, a data-variable input/output memories module to store the data and image matrices, a sequencer and memory controller module to schedule parallel memory accesses between the memory module and the parallel processing unit, and a post-processing module which performs image normalization and quantization. The design of the FPGA module considered two main issues: the execution speed and the hardware scalability. While the image reconstruction process needs to be achieved in real-time, the hardware should also accommodate various sizes of the hardware should also accommodate various sizes of matrixes to be processed.

The parallel processing module is a parallel-like architecture which is composed of several similar adder/multiplie processing units. Each of these units is scalable and consists of three pipelined stages: decomposition stage, basic unit operation stage, and composition stage. Figure 3 shows these stages at both matrix and bit-levels.

a. Decomposition Stage – Matrix Level

For the three afro-mentioned algorithms (i.e. LBP, Landweber, and Modified - Landwe be algorithms), the matrix decomposition consists to divide the two matrices to multiply into regular square blocks (q × q) which are then simultaneously processed by the parallel processing units using bit decomposition techniques. In case one of the two matrices has a number of columns (respectively rows) which are not multiply of q, then this matrix is filled with all zeros columns (respectively all zero rows) to generate a new matrix where both rows and columns are multiple of q. This does not engender further hardware cost or computation delay since only PEs with non-null input operands are provided in the architecture. For instance, without loss of generality, in case of the LBP algorithm the matrix product St × C is done by adding q-1 columns to matrix C since this latest is of size M × 1 (Figure 4).

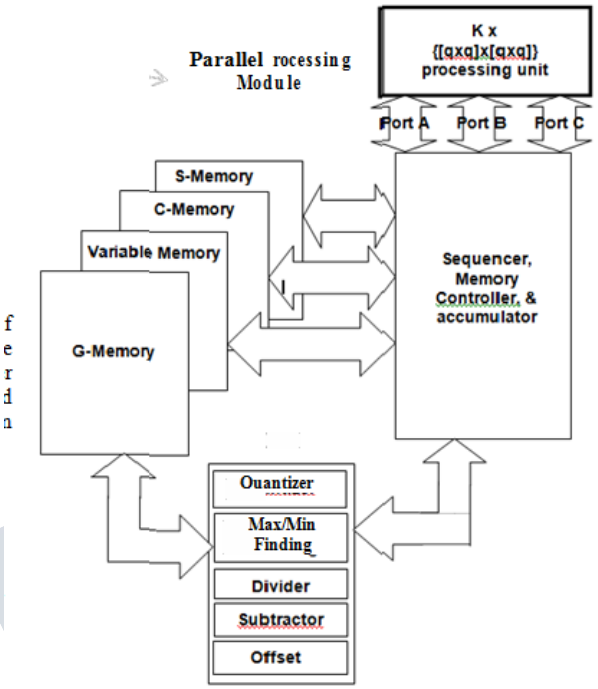


Figure 2 Image reconstruction implementation block diagram on FPGA

The parallel processing module then proceeds by executing simultaneously several (q × q) square matrix multiplications to obtain the values of several pixels within a same clock cycle. These matrices are stored into different Block RAM (BRAMs) controller module. Figure 5 shows the architecture of the (q × q) multiplier block. of the FPGA and are controlled by a dedicated address

b. Basic Unit Operation Stage – Matrix Level

The basic unit operator stage in matrix level covers decomposition stage in bit level, basic unit operator stage in bit level, and composition stage in bit level as described in following sections. Figure 6 shows the overall operations in basic unit operation stage. Ports A and port B are the input ports of the K-processing unit, supplied by the "Sequencer & Memory Controller" module as shown in Figure 2. These ports receive continuously the elements of the decomposed matrices S and C in every clock cycle. In Figure 6, the intermediary register files Rdecomp, Rcomp, and Rbasic are designed to operate in a parallel-like architecture in order to avoid stalling states in the overall process. Port

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 4, Issue 3, March 2017

C is the output port that provides the results back to the G memory (Figure 2). The next sections explain the details of each of the three modules composing the basic unit operation stage.

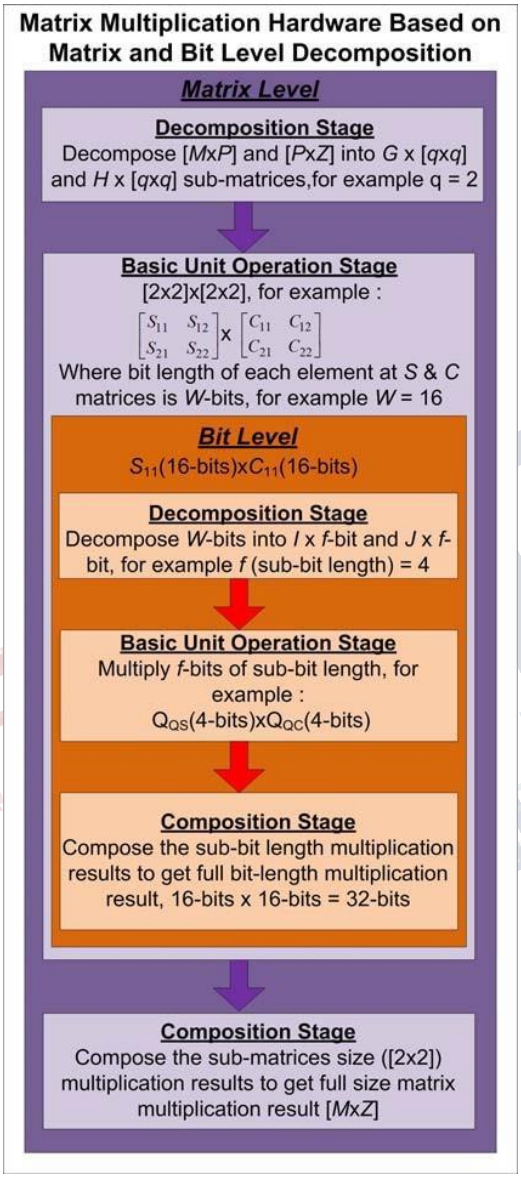


Figure 3 The decomposition, basic unit operation, and composition stages in matrix and bit level.

Decomposition Stage – Bit-Level

Following the matrix level decomposition, a bit level decomposition takes place. The inputs enter the decomposition unit in a pair of elements matrix S and C as (S11,C11), (S12,C12), (S21,C21), and (S22,C22). Two main task are assigned to the bit-length decomposition stage: First it decomposes the W-bit width elements of matrices into several f elements (e.g. W = 16 bits and f = 4 bits in case of Figure 7). The second one consists to order the elements of the [q×q] matrix to be processed by the basic unit operator in such a way that the output segments (e.g. Gij) are generated at the same rate as the input rate.

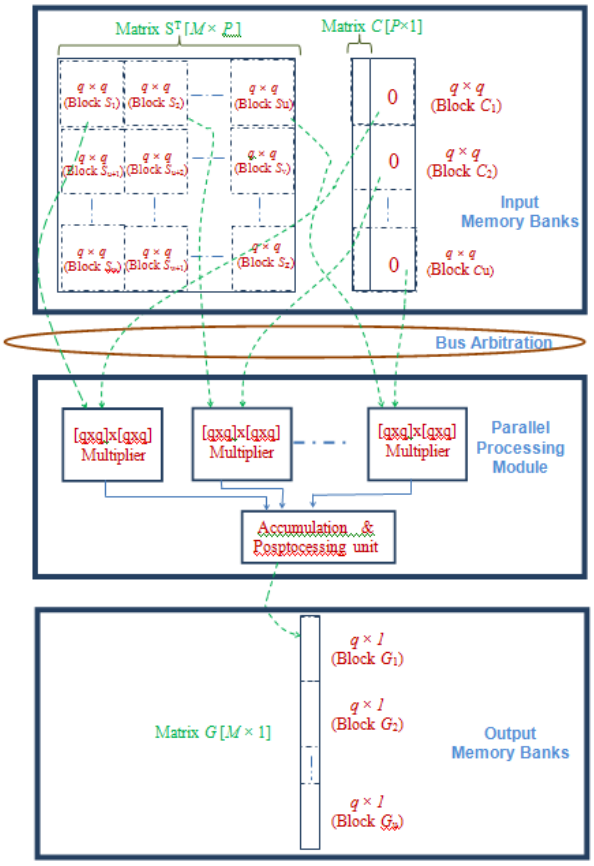


Figure 4 Matrix Decomposition Procedure for the LBP Algorithm. The Matrix ST decomposed row based and matrix C decomposed column based.

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 4, Issue 3, March 2017

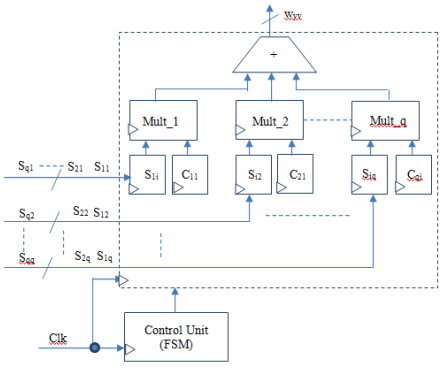


Figure 5 The architecture of the (q × q) multiplier block.

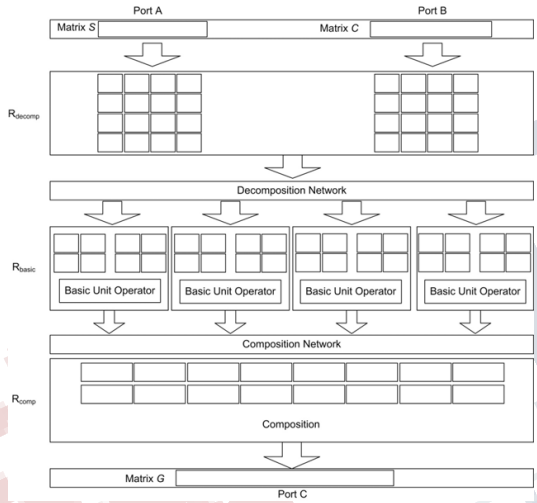


Figure 6 Overall architecture with three main units. Figure is illustrated with the number of basic operators, Nbasic, which equal to 4.

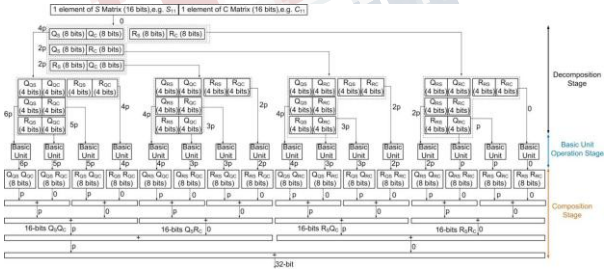


Figure 7 Bit-length decomposition, basic unit operation, and composition stages

The advantage of such decomposition is to let the hardware scalable to handle various bit widths operation, in addition to improve the overall throughput of the algorithm. Without loss of generality, Figure 7 represents the overall hardware multiplication of two elements of the matrices S and C (e.g. S11 and C11) in case they are both 16 bits-widths [13]. The principle of the bit decomposition consists to split the elements S11 and C11 into pairs (QS, RS) and (QC, RC) of 8 bits each respectively. This is followed by another 4 bits decomposition: (QQS, RQS), (QRS, RRS), (QQC, RQC), and (QRC, RRC). The 4 bits multiplications then take place concurrently within a 4 levels stages pipelining.

In our suggested architecture, a fixed point representation of numbers was considered since it is less hardware consuming; in addition all numbers to be handled are within a same range (i.e. 18 to 24). Also, each element of the matrices is represented as a signed variable using complement-base 2 representation. This is because the three considered ECT algorithms may handle negative and positive numbers alike. Figure 8 show the corresponding hardware implementation. Hence, an additional sign bit (s or s' bits in Figure 8) and zero bit (Z bit in Figure 8) are also associated to each of the decomposed f bits. The sign bit is added to the associated f bits to perform complement-base 2 operation. Note that the last segment (LSB part) does not require the addition to be performed. Nevertheless. for this segment, the sign bit is overwritten if the f-bit element is all zero. Also, in case the original decomposed bit-length number is positive the conversion is not needed. A multiplexor is used to select or not the adder output (Figure 8).

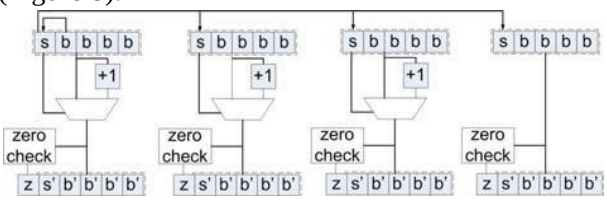


Figure 8 Illustration of the 2's compliment conversion with W = 16 and f =4 as an example

Note that the zero flag bit, Z bit, was associated to each f-bits digit in order to reduce the overall power consumption by avoiding the usage of unnecessary functional units in the basic unit operation stage. Hence, at the output of the

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 4, Issue 3, March 2017

multiplexor, each of the f-bits segments are checked if they are zero using NOR-gates and the corresponding Z bit is modified accordingly. In terms of hardware complexity, the total number of registers required in the matrix decomposition unit is:

$$N_{R.decomp} = L_d \times 2 \times \frac{W}{f} \times (f+2) \qquad (7)$$

where Ld represents the minimum depth of registers to support a continuous stream of [q×q] matrices. It is the sum of two factors, r and s (where r represents the data ready of the separation of data which will be processed by the basic unit operator that are needed to produce one output, while s represents the additional number of registers which are needed to keep the existing current elements). For example, to generate G11 from [2×2] × [2×2], S11, C11, S12, and C21 are needed where for example the C21 is separated from C11 by two input cycle for the degree of time multiplexing of 2. By introducing these additional registers, the basic unit operator can get continuously supplied with the input stream without overwriting the previous elements. Figure 9 shows the architecture of the 2's complement part, Rdecom, decomposition network, and the basic unit operator [13].

Basic Unit Operation Stage – Bit - Level

Figure 10 shows the basic unit operator architecture based on radix-f for q = 2. A basic operator consists of a set of registers, Rbasic, two (p + 1)-bit multipliers, one adder, and a multiplexor. Each operation takes four inputs. The outputs from the multipliers are added for the final output, (p + 2)-bit elements (one bit or sign extension and the other bit for the zero flag) of the working submatrix are loaded in order into registers. Once ordered, the data will propagate through a chain of register as shown in Figure 10. This ordering is established to produce

This design eliminates unnecessary switching which may occur within the multiplier and hence reduce the overall power consumption. The same concept is also considered for the right multiplier. The adder is active only when all elements going through the multipliers are nonzero. Four possible outputs, A + B, B, A, and 0, are selected by the multiplexor. This selection is controlled by the corresponding multiplexor select inputs zA and zB where zA = ZL1+ZL2=0 and zB = ZR1 + ZR2=0. It is worth noting here that in case of a pipeline implementation of the multipliers comprising the basic operator, the energy reduction scheme (i.e. zero flag bits, tri-state buffers, multiplexor) is less effective [13].

In terms of hardware complexity, for given values of W and f, the total number of basic operations is (W/f)2. Hence, we assume that the number of basic operators is multiple of W/f such that:

$$N_{basic} = k \times \frac{W}{f} \qquad (8)$$

Three-state buffers, controlled by zero flag bits (i.e. ZL1 and ZL2 for the left multiplier, and ZR1 and ZR2 for the right multiplier) are connected to the input of the multipliers and are active only when the corresponding data element is not zero. The implementation of these buffers into FPGA is done using SR latches of the FPGA chip. Hence, when either ZL1 or ZL2 control flags corresponding to the left multiplier is set, the multiplication is carried out and the zero output is selected at the multiplexor. where k is an integer constant, the value of which is chosen such that (W/f)/k is also an integer.

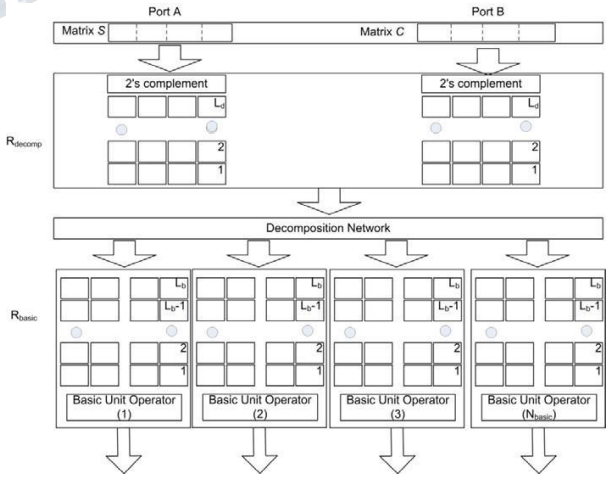


Figure 9 Illustration of the decomposition unit and interface (decomposition network) with the basic operators

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 4, Issue 3, March 2017

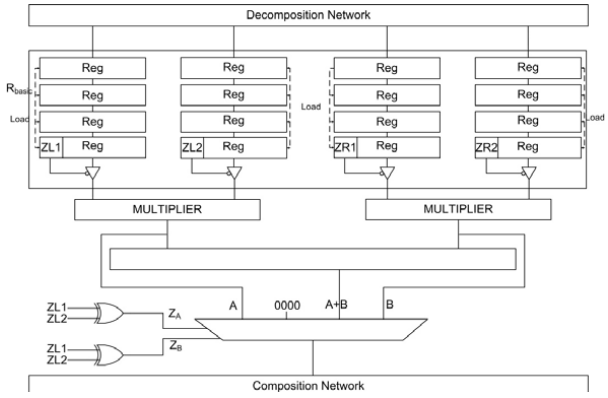


Figure 10 Illustration of the basic operator for submatrix with size q = 2. It consists of two multipliers and an adder. A multiplexor is used to minimize the switching

The ratio (W/f)/k represents the degree of time multiplexing. As an example, for W and f equal to 16 and 4 respectively, a total of 16 concurrent submatrices multiplications, with four basic operators executing in parallel are required. This leads to generate one output within 4 iterations. The decomposition clock period, Tdecomp, and the total number of registers in the basic operator uni, NR, basic are defined as follows:

$$T_{decomp} = \left(\frac{W}{f}\right)^2 \times \frac{1}{N_{basic}} \times T_{basic} \qquad (9)$$

$$N_{R,basic} = L_b \times 4 \times (p+2) \times N_{basic} \qquad (10)$$

where Lb is the register depth of a the basic operator, the value of which depends on the data transfer rate of both the decomposition unit and the basic operators. If we define T as the fetch rate from the registers in the decomposition to registers in the basic operator, then Nr,decomp and Nr,basic are affected by T. The registers in the basic operators are filled at the rate of Tq which is defined as follows:

$$Tq = (W/f)/k \times T_{basic} \qquad (11) \quad \text{and}$$

$$T_{basic} \le Tq \le T_{decomp} \qquad (12)$$

If $T_q = T_{decomp}$, more registers at the basic operators are needed (i.e., increase L_b). On the other hand, if $T_q = T_{basic}$, only register depth 1 is needed by the basic operators [13].

Composition Stage – Bit Level

Figure 11 shows the composition adder tree. The composition unit consists of registers and an adder tree with 0-mp adders. At the input of the composition unit, a finite set of data (i.e., equal to the number of basic operators) is available at each Tbasic. At the basic operators and the composition unit interface, the number of registers needed is equal to the total number of basic operators. It is worth to observe that the input o the 0-mp adders are skewed by mp-bits with respect to each other. Thus certain ranges of bits are known before the addition (i.e., zero bits at the low significant bits of one input and extended sign bits at the high significant bits of the other). Hence, the worst case delay is less than a normal adder with the same width. In the implementation, the adders are constructed with carry-select adders where each segment of adder is designed with a f-bit ripple carry adder [14].

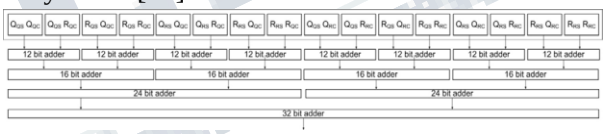


Figure 1 Illustration of the adder in composition stage in bit level.

In terms of hardware complexity, the number of registers required for composition is given as follows

c. Execution Throughput and Latency

The throughput is defined as a rate of completion of W-bit [q×q] matrix multiplication per second. This rate ultimately depends on the clock period of the decomposition unit, Tdecomp.

$$N_{R,comp} = \left(\frac{W}{f}\right)^2 \times (2f+1) \qquad (13)$$

Hence, the throughput Tthroughput, can be defined as follows: The output from each operator is stored in Rcomp through the composition network which depends on the number of basic operators. The bit-width of each register is (2f + 1) bits, among which

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 4, Issue 3, March 2017

one bit is for the sign bit. The values stored in these registers represent the partial results [13]. At the end of the adder tree, the output word-width is 2W-bits. It is clear that the composition adder tree is critical for high throughput operations such as ECT. The minimization of the critical path of this unit while maintaining high throughput was considered by inserting several pipeline stages. Thus, the target of Tcomp should be made equal to Tdecomp through pipelining. The pipelining of the adder tree is greatly influenced by the data ordering. Hence, the data is ordered at the decomposition unit and executed at the basic operators in a such a way that the partial results for lower significant values are generated first. In case there is

$$T_{throughput} = 4 \times T_{decomp} = 4 \times m \times T_{basic} \quad (14)$$

Where the constant coefficient, 4, represents the number of elements in [q×q] matrices (i.e. in case q = 2) and m represents the degree of time multiplexing of the basic operators. Thus, doubling the number of basic operators reduce the decomposition unit period by half. Figure 12 illustrates the relative timing of each unit. Note that the rates of both the decomposition and composition units are equal whereas the rate of the basic operators is two times faster than both units[13].

The initial latency of the architecture is the sum of latencies in the decomposition, basic operator, and composition units. Thus, the total latency is given as follows: multiplexing, the speed of basic operators must be faster by a factor equal to the degree of time multiplexing. The pipelining depends on the speed of the basic operators and the desired speed of the composition unit [13].

$$T_{latency} = r \times T_{decomp} + T_{basic} \times \left(\frac{W}{r}\right)^2 \times \frac{1}{N_{basic}} + pipe_b \times T_{basic} + pipe_c \times T_{comp} \quad (15)$$

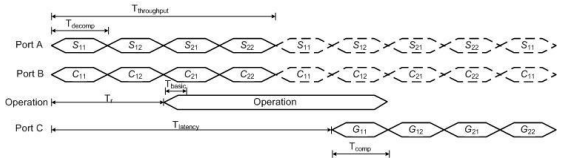


Figure 2 Illustration of execution timing and latency of the proposed matrix multiplier. The figure is illustrated for the degree of time multiplexing (i.e., $T_{basic} = 1/2T_{decomp}$ and $T_{decomp} == T_{comp}$)

where the first component of the total latency is due to the data ready factor in the decomposition unit and this value is the same for all implementations (i.e., a actor s does not influence the latency). The second component is due to the multiplexing of the basic operators. The third and the fourth components are due to pipeline of the basic operators and composition unit, where pipeb and pipec are the numbers of pipeline stages in the basic operators and the composition unit, respectively.

d. Composition Stage – Matrix Level

The composition stage in the matrix level accumulates the [q×q] × [q×q] result to produce a final matrix multiplication result. The final matrix multiplication result is formed by accumulating the whole multiplication result [q×q] between the operand matrix. For example, in Figure 4, to produce the [q×q] matrix of matrix G at (1,1), numbering of location refer to the matrix location system where we treat [q×q] matrix as if a single element of matrix G, P number of [q×q] × [q×q] result needs to be accumulated where the left-hand operand matrix come from the first row of matrix ST and the right had operand matrix come from the first column of matrix C.

IV. DESIGN VERIFICATION AND RESULTS

The design flow used various commercially-available software tools, namely SolidWork, Comsol, Livelink, and Matlab for high level simulation and Modelsim and Quartus II for hardware level simulation (Figure 13). The capacitance values of all possible combinations and the output image were assigned to one input and one output test vector respectively. Prior to verilog implementation, bit level programming was done in Matlab

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 4, Issue 3, March 2017

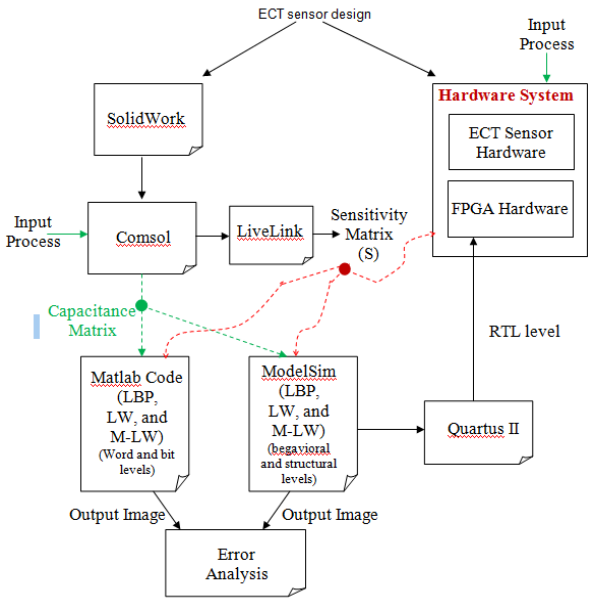
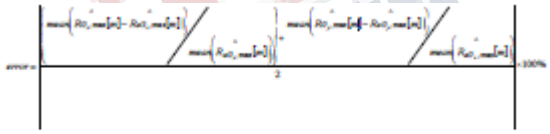


Figure 3 Overview of the Hardware/Software Codesign Methodology

Matlab Design Verification and Result

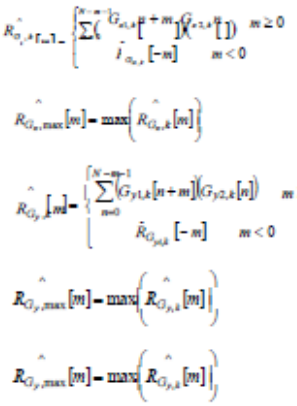
In order to assess the suggested hardware architecture, several simulations were conducted for different bit widths for each of the three reconstruction algorithms. Hence, in addition to quality assessment using image visualization, a new metric using cross-correlation is suggested in this paper which takes care of eventual pixel shifts which may be caused by discretization errors:



(21)

Figure 14 shows the results obtained for three input images and the corresponding results with different number of bit widths using the three reconstruction algorithms (LBP, LW, and MLW algorithms). In the Figure, each elements of the matrix is split into three parts: 1 bit sign, m integer bits, and n decimal bits.

Where:



In addition:

k : Row or column index, $k = 1 - 32$

$G_{x1,k}$: The pixel of original image of row k from column 1-32

$G_{x2,k}$: The pixel of reconstructed image of row k from column 1-32

$\hat{R}_{G_x,k}[m]$: Cross correlation result of $G_{x1,k}$ and $G_{x2,k}$

$G_{y1,k}$: The pixel of original image of column k from row 1-32

$G_{y2,k}$: The pixel of reconstructed image of column k from row 1-32

$\hat{R}_{G_y,k}[m]$: Cross correlation result of $G_{y1,k}$ and $G_{y2,k}$

$\hat{R}_{aG_x,k}[m]$: Auto correlation of original image of row k from column 1-32

$\hat{R}_{G_x,k}[m]$: Cross correlation result of $G_{x1,k}$ and $G_{x2,k}$

$G_{y1,k}$: The pixel of original image of column k from row 1-32

$G_{y2,k}$: The pixel of reconstructed image of column k from row 1-32

$\hat{R}_{G_y,k}[m]$: Cross correlation result of $G_{y1,k}$ and $G_{y2,k}$

$\hat{R}_{aG_x,k}[m]$: Auto correlation of original image of row k from column 1-32

As shown in Figure 15 (a) until (c), the effect of increasing the length of the fraction part may not be beneficial since it gets

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 4, Issue 3, March 2017

saturated after few iterations. Nevertheless, for Case 2 - LBP, the longest fractional bits didn't lead to the best reconstructed images. It can also be noticed that the LBP algorithm generates images without pixel shifts, however as it can be seen for images 2 and 3 the reconstructed images are not solid. On the other hand, both the LW and the MLW could reconstruct more complete images, with phantoms similar to the ones in the reference image but with some shifts. Table 1 shows the hardware resource requirements for each unit in case the target hardware is Xilinx Virtex-II XCV1500. The resource requirement for each unit is represented in the number of CLBs used in implementation. The basic operators are implemented with and without the pipeline stages. Two 5-bit embedded multipliers are incorporated in each basic operator.

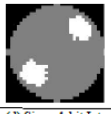
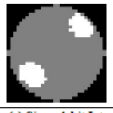
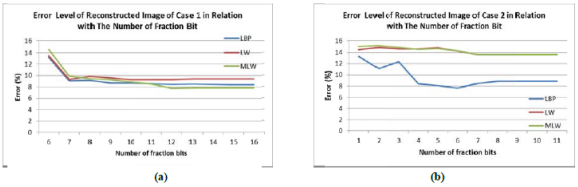
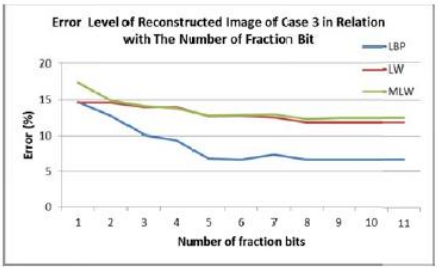
Case	Original Image	LBP	LW	MLW
		(a) Sign : 1-bit Int : 1-bit, Dec : 16 bits	(b) Sign : 1-bit Int : 8-bit, Dec : 10 bits	(c) Sign : 1-bit Int : 1-bit, Dec: 12 bits
1				
		(d) Sign : 1-bit Int : 1-bit, Dec : 11 bits	(e) Sign : 1-bit Int : 8-bit, Dec : 12 bits	(f) Sign : 1-bit Int : 1-bit, Dec :12 bits
2				
		(g) Sign : 1-bit Int : 1-bit, Dec :16 bits	(h) Sign : 1-bit Int : 8-bit, Dec : 13 bits	(i) Sign : 1-bit Int : 1-bit, Dec :13 bits
3				

Figure 4 Reconstructed images for LBP, LW, and MLW for various bit width. Bit consists of [sign bit][integer bits][decimal bits]



(a) (b)



(c)

Figure 5 Errors for (a) Image 1 (b) Image 2, and (c) Image 3

Table 1 Resource of each unit for $W = 16$ and $p = 4$. Logic resource in measured in the number of CLBs

Component	Logic (CLBs)	Delay (*nsec*)
Basic Operator (no pipeline)	7	10.43
Basic Operator (pipeline)	11	3.186
Decomposition	23	2.458
Composition (full word-width, Nbasic = 4)	24	7.457
Composition (full word-width, Nbasic = 8)	27	5.40
Composition (full word-width, Nbasic = 16)	52	3.302

Verilog Design Verification and Result

Table 2 shows the throughput in terms of the number of matrix multiplications per second with the fastest basic operators (i.e. The DSP pro essor was selected based on the fact that it is one pipelined version) with degrees of time multiplexing of 1, 2, of the most recent DSP processors available in the arket, in and 4 respectively. The execution time represents the amount of time involved in executing one matrix multiplication for a given size of matrices. Data for higher order matrix multiplications are derived from the [2×2] × [2×2] scheme. The inverse of the execution time represents the throughput represented in the number of matrix computation per second addition of having suitable features for matrix intensive multiplications such as Duo core DSP engine, large on-chip memory (large enough to store the sensitivity matrix as well as the image frames), Viterbi complex unit for fa t matrix multiplication and inversion, as well as featuring an onboard eal-time op rating system. Note that with regard to the [13]. The execution time for matrix multiplication larger than proposed parallel architecture, the

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 4, Issue 3, March 2017

Landweber algorithm [2×2] × [2×2] (for example [4×4] × [4×4]), can be improved by implementing multiple [2×2] × [2×2] processing unit. For example, the fastest execution of the [4×4] × [4×4] operation requires eight (8) ([2×2] × [2×2]) processing units. Using two (2) ([2×2] × [2×2]) processing units, leads to an execution time of 0.054 μsec, which is half the execution time shown in the last row of Table 2. Considering the data shown in Tables 1 and 2, the overall throughput, execution time, and speed up factor (relatively to computer based system- 2.53 GHZ-based Pentium processor and a DSP, TI's Delphino TMS320F2837 processor[15]) is illustrated in Table 3 for each image reconstruction algorithm equires two size of matrix multiplicati n operation which are 1024×28] × [28×21] and [28×1024] × [1024×28]. This is the eason why this algorithm requires two types of matrix decomposition and composition whic are based on 1024 [2×2] × [2 2]) processing units and 28 ([2×2] × [2×2]) processing units respectively. As can be seen in Table 2 below, the proposed parallel architecture outperformed the two other aforemention d platforms y a factor of at least 5.

Table 2 Throughput and execution time for executing larger matrix multiplication using the proposed [2×2] × [2×2] multiplier for W =16 and degree of time multiplexing of 1, 2, and 4.

	[2×2] × [2×2]	[4×4] × [4×4]	[8×8] × [8×8]	[16×16] × [16×16]
Throughput (4) (Mops)	19.617	2.452	0.307	0.038
Execution Time (4) (μsec)	0.051	0.408	3.262	26.10
Throughput (2) (Mops)	39.234	4.904	0.613	0.077
Execution Time (2) (μsec)	0.025	0.204	1.631	13.05
Throughput (1) (Mops)	71.428	9.259	1.153	0.144
Execution Time (1) (μsec)	0.014	0.108	0.867	6.935

Table 3 Throughput and execution time for executing larger matrix multiplication using the proposed [2×2] × [2×2] multiplier for W =16 and degree of time multiplexin of 1, 2, and 4.

	[2×2] × [2×2]	[4×4] × [4×4]	[8×8] × [8×8]	[16×16] × [16×16]
Throughput (4) (Mops)	19.617	2.452	0.307	0.038
Execution Time (4) (μsec)	0.051	0.408	3.262	26.10
Throughput (2) (Mops)	39.234	4.904	0.613	0.077
Execution Time (2) (μsec)	0.025	0.204	1.631	13.05
Throughput (1) (Mops)	71.428	9.259	1.153	0.144
Execution Time (1) (μsec)	0.014	0.108	0.867	6.935

V. CONCLUSION

This paper presented a VLSI implementation of the ECT algorithm using FPGA chip (Virtex-II XCV1500). An analysis of our implementation regarding the quantization effect caused by finite bit widths indicates that 18 - 24 bits are enough to generate less than 15 % relative error. The hardware explored the implicit parallelism for matrix multiplication. Its advantage is that it is flexible to accommodate various bit-widths and image sizes, in addition to consume less power since the most significant bits of the sensitivity matrix are usually equal zero. In order to achieve 1 pixel per clock cycle throughput, a new pipelined architecture was suggested. Extensive simulations indicate that the proposed architecture achieves a speed-up of up to three orders of magnitude over the software version when the reconstruction algorithm runs on 2.53 GHZ-based Pentium processor. More specifically, a throughput of 17.241 Kframes/sec for both the Linear-Back Projection (LBP) and modified Landweber algorithms and 8.475 Kframes/sec for the Landweber algorithm with 200 iterations could be achieved. This performance was achieved using an array of [2×2] × [2×2] processing units. This satisfies the real-time constraint of many industrial applications. To the best of the authors' knowledge, this is the first embedded system which explores the intrinsic parallelism which is available in modern FPGA for ECT tomography.

REFERENCES

[1] Yang, WQ and Peng, Lihui "Review Article : Image Reconstruction Algorithms for Electrical Capacitance Tomography", IOP Measurement Science and Technology, 2003

[2] Marashdeh, Q. and Teixeira, F. L., "Sensitivity Matrix Calculation for Fast 3-D Electrical Capacitance Tomography (ECT) of Flow Systems", IEEE Transaction on Magnetics, Vol. 40, NO. 2, 2004.

[3] S. Teniou and M. Meribout, "Multiphase Flow Meters: Principles and Applications: A Review", Canadian Journal on Scientific and Industrial Research, Vol. 2, No 8, 2011.

[4] Dikcin, F.J., Hunt, A., Hoyle, B.S., Huang, S.M., Ilyas, Lenn, C., Waterfall, R.C., Williams, R.A., Xie, C.G., and Beck, M.S.: 'Tomographic imaging of industrial process equipment - Review of needs and methods', IEE Proc. G, Electron. Circuits & Syst., 1992, 139, (I), pp. 72- 82

[5] D F Garcia-Nocetti, J C Gamio, "Parallel Realization of the Linear Back Projection Algorithm for Capacitance Tomography Using TMS320C6701 Digital Signal Processor", 3rd World Congress on Industrial process Tomography, Banff, Canada, 2003.

[6] Wang H, Xin S, and Zhang X, "New Progress of the Digital Electrical Capacitance Tomography System for Gas/Liquid Two Phase Flow", International Workshop on Imaging Systems and Techniques, China, 2009.

[7] H. Sohal1, H. Wi1, A. Lee, E. Je Woo1 and T. In Oh1, "Electrical impedance imaging system using FPGAs for flexibility and interoperability.", Biomedical Engineering Online, Vol. 13, NO 126, 2014.

[8] L. Xu, H. Zhou, Z. Cao, and W. Yang, "A Digital Switching Demodulator for Electrical Capacitance Tomography", IEEE Transactions on Instrumentations & Measurements, Vol. 63, NO 5, May 2013

[9] I. Kim, Z. Zhang, and J. Fessler, "Hardware Acceleration of Iterative Image reconstruction of X-Ray Computed Tomography", Proceedings, International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 1697-1700, 2011.

[10] P. Kinsman and N. Nicolici, "NOC-Based FPGA Acceleration for Monte Carlo Simulations with Applications to SPECT Imaging", IEEE Transactions on Computers, Col. 62, NO. 3, March 2013.

[11] L. Xu, M. Taufer, S. Collins, and D.G. Vlachos, "Parallelization of Tau-Leap Coarse-Grained Monte Carlo Simulations on GPUs," Proc. IEEE Int'l Symp. Parallel Distributed Processing (IPDPS), pp. 1-9, Apr. 2010.

[12] E. Fysikopoulos, M. Georgiou, N. Efthimiou, S David, G. Loudos, and G. Mastopoulos, "Full Digital FPGA-Based Data Acquisition System for Dual Head PET Detectors", IEEE Transactions on Nuclear Science, Vol. 61, NO. 5, Oct. 2014.

[13] Hong, S. Park, K-S, and Mun, J-H, "Design and Implementation of a High-Speed Matrix Multiplier Based on Word-Width Decomposition", IEEE Transaction on Very Large Scale Integration (VLSI) System, Vol. 14, No. 4, 2006.

[14] J. M. Rabaey, A. Chandrakasan, and B. Nikolic, "Digital Integrated Circuit : A Design Perspective, 2nd ed.", Englewood Cliffs, NJ : Prentice-Hall, 2003.

[15] Datasheet of Delphino DSP Processor TMS320F2837. URL: http://www.ti.com/lit/ds/symlink/tms320f28377d.pdf