

Research Article

Model-Based Hardware-Software Codesign of ECT Digital Processing Unit

Atef Allam ^{1,2} and Wael Deabes ^{3,4}

¹Electrical Engineering Department, Assiut University, Assiut, Egypt

²Computer Engineering Department, Taibah University, Medina, Saudi Arabia

³Computer Science Department, University College, Umm Al-Qura University, Meccah, Saudi Arabia

⁴Department of Computer and System Engineering, Mansoura University, Mansoura, Egypt

Correspondence should be addressed to Wael Deabes; wadeabes@uqu.edu.sa

Received 13 June 2019; Accepted 17 March 2021; Published 31 March 2021

Academic Editor: Jing-song Hong

Copyright © 2021 Atef Allam and Wael Deabes. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Image reconstruction algorithm and its controller constitute the main modules of the electrical capacitance tomography (ECT) system; in order to achieve the trade-off between the attainable performance and the flexibility of the image reconstruction and control design of the ECT system, hardware-software codesign of a digital processing unit (DPU) targeting FPGA system-on-chip (SoC) is presented. Design and implementation of software and hardware components of the ECT-DPU and their integration and verification based on the model-based design (MBD) paradigm are proposed. The inner-product of large vectors constitutes the core of the majority of these ECT image reconstruction algorithms. Full parallel implementation of large vector multiplication on FPGA consumes a huge number of resources and incurs long combinational path delay. The proposed MBD of the ECT-DPU tackles this problem by crafting a parametric segmented parallel inner-product architecture so as to work as the shared hardware core unit for the parallel matrix multiplication in the image reconstruction and control of the ECT system. This allowed the parameterized core unit to be configured at system-level to tackle large matrices with the segment length working as a design degree of freedom. It allows the trade-off between performance and resource usage and determines the level of computation parallelism. Using MBD with the proposed segmented architecture, the system design can be flexibly tailored to the designer specifications to fulfill the required performance while meeting the resources constraint. In the linear-back projection image reconstruction algorithm, the segmentation scheme has exhibited high resource saving of 43% and 71% for a small degradation in a frame rate of 3% and 14%, respectively.

1. Introduction

Electrical capacitance tomography (ECT) is an industrial process tomography technique for imaging materials distributions inside a certain interest area [1, 2]. Visualizing the multiphases flow such as the gas/oil in oil pipes is one of the most significant applications of the ECT [3]. The ECT system consists of three main components, capacitance sensors, data acquisition unit, and ECT digital processing unit (ECT-DPU) as shown in Figure 1 [4]. Measured capacitance data are sent wirelessly to a base station attached to ECT-DPU where an image reconstruction algorithm is imple-

mented to produce an image describing the material distribution inside the imaging area [5, 6].

An ECT image reconstruction algorithm is realized as a software on a general-purpose processor [7], but in stringent time constraints, a dedicated hardware can be used [6] to achieve the real-time operation. The most important driving factors in embedded system design are the performance and flexibility. While high flexibility and low design effort can be gained by applying the software implementation, its performance gain is low. In contrast, the hardware intrinsic parallelism realizes excessive system performance, but its design complexity overhead is high. Hardware-software (HW/SW)

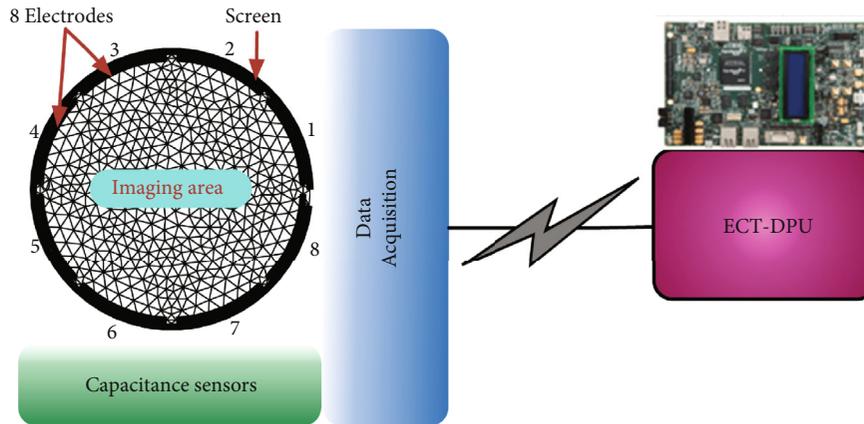


FIGURE 1: Wireless electrical capacitance tomography system.

codesign of the ECT digital processing unit proposed in this paper allows the trade-off between attainable performance and flexibility.

Recently, the FPGA SoC becomes an appropriate embedded system hardware-software implementation platform; and it turns to be the proper candidate platform for ECT-DPU realization [8, 9]. Traditional design of the embedded SoC carries out hardware and software components in two different branches in the design flow and uses different tool-sets [10, 11]. The hardware part is modeled and simulated based on a handwritten HDL code [12], and the software is modeled and cross-compiled using a different set of tools. This traditional design and implementation of ECT-DPU software and hardware components, their integration and verification require a great effort and are error-prone. These issues can be managed using model-based design approach.

Model-based design (MBD) is a model-centric approach widely used in the embedded system design [13, 14]. It enables the usage of an executable system model throughout the whole design cycle spanning from the system-level down to the implementation. The MBD is a system-level approach that applies refinements and transformation on the abstract system model used for algorithm design and simulation in system-level to HW/SW partitioning, automatic code generation for both SW processor as well as HW implementation, and test and verification, in a single integrated platform. Refinements and transformation processes are achieved by applying the designated tool in the MBD tool-chain [15].

The model-based design has been used extensively in the implementation of software defined radio (SDR) systems on FPGA [16–18], in embedded control hardware/software codesign and realization on FPGA [19], and in image processing algorithms design and implementation on FPGA [20].

The image reconstruction is the main constituent module of the ECT digital processing unit. The inner-product constitutes the kernel operation of the matrix multiplication of numerous image and signal processing algorithms [21, 22] and cryptography [23]. It is the core operation of the matrix-vector multiplication (MVM) used in linear-back projection (LBP) and the Landweber image reconstruction

algorithms of the electrical capacitance tomography system [24–26].

Matrix-vector multiplication is a core macrooperation in most of the ECT image reconstruction algorithms. In this research, iterative linear-back projection (iLBP) [27] is used as the image reconstruction algorithm. Mathematically, matrix-vector multiplication (MVM) constitutes the pivotal computation structure in the iLBP image reconstruction algorithm, while the inner-product constitutes the kernel operation of the MVM. The inner-product as well as the matrix-vector multiplication possesses inherent parallelism that makes it to be executed in parallel over general-purpose graphics processing units [28] and multicore processors [29]. On the other hand, the FPGA intrinsic parallel structure makes it a promising viable platform for hardware implementation of the inner-product and the matrix-vector multiplication.

The FPGA realization of the matrix-vector multiplication algorithm has been tackled by many research work at the algorithmic level as well as at the bit-manipulation level [30–32]. Most of the FPGA implementation of these proposed parallel structure of the matrix multiplication at an algorithmic level are for small matrix dimensions. Full FPGA parallel implementation of large matrix-vector multiplication consumes a huge number of FPGA resources and incurs long combinational path latency. Careful setting of the degree of parallelism as well as the design of parallel structure is essential to meet the stringent embedded system performance and complies with the available FPGA resources.

This paper proposes a model-based hardware-software codesign flow of the digital processing unit for realization of the image reconstruction and control module of the ECT system on the FPGA SoC platform. Model-based design is proposed to fully automate and tune the design, and implementation of software and hardware components of the ECT-DPU and their integration and verification. Another contribution of this paper is that it presents a parametric segmented parallel inner-product architecture to work as a shared hardware core unit for parallel matrix multiplication in the image reconstruction, control of the ECT system, and similar matrix-vector multiplication-based embedded system algorithms. This segmentation approach allows the

designer to use the MBD to tune the design process to fulfill the required performance while meeting the FPGA resources constraint. In each design cycle, the ECT-DPU is simulated, tested, and verified, and the code is generated for both FPGA fabric and the attached ARM processor in the FPGA SoC platform. System design using MBD with the proposed segmented architecture allows the system to be flexibly tailored to fulfill the required performance while meeting the resource constraint. The proposed segmented architecture modeling equations can be used to rapidly generate an estimate of the execution time and required resources at the system-level. Using MBD can greatly minimize the development time and reduces the design cycle as well as alleviates remodeling the system in each design cycle.

Our proposed solution of the image reconstruction and control module of the ECT system is different than that introduced in the previous work in [25, 33]. The image reconstruction, FPGA module in [25], is totally a hardware system built around the matrix decomposition at bit level, while our SW/HW system's hardware module of image reconstruction is built around the proposed shared parallel segmented inner-product architecture. In addition, our parameterized MVM core unit is adjustable at the system-level to tackle large matrices, with the segment length set by the designer to fulfill the required performance while meeting the FPGA resource constraint.

The rest of this paper is organized as follows: "Section 2" explains the details of the ECT-DPU model, while "Section 3" introduces the formulation of the matrix-vector multiplication problem. The modeling and the implementation of the proposed system are presented in "Section 4". Finally, experiments will be carried out to validate the proposed method.

2. Digital Processing Unit (ECT-DPU)

2.1. Image Reconstruction Algorithm. Measuring the capacitance is carried out in sequence by making one electrode as a transmitter and the rest as receivers then consecutively changing second electrode to receiver [26]. Therefore, the number of independent measurements for 8-electrode ECT system is 28 computed from

$$M = \frac{n(n-1)}{2}, \quad (1)$$

where M is the total collected capacitances and n represents electrodes' number. The linear forward model of the ECT is expressed as

$$C_{M \times 1} = S_{M \times N} G_{N \times 1}, \quad (2)$$

where C is the measurements, G the image matrix, N the number of images' pixels which is around 256 pixels for a 16x16 image, and S the sensitivity matrix defined for each element k as follows:

$$S_{i,j}(k) = \frac{C_{i,j}(k) - C_{i,j}^l}{C_{i,j}^h - C_{i,j}^l}, \quad (3)$$

where $C_{i,j}^l$ is capacitance vector when the imaging region is full by a low permittivity material and $C_{i,j}^h$ is capacitance vector when filled by the high permittivity one. As shown in Equation (2), the number of the image pixels is much larger than the measured data; therefore, the problem is ill-posed and any small change in the measurements can cause a big difference in the image. Moreover, the sensitivity matrix is not a square matrix, and the reconstructed image cannot be computed by using S^{-1} [34]. Hence, the reconstruction algorithms are classified into two types: noniterative and iterative algorithms. Linear-back projection (LBP), Equation (4), is one of the noniterative algorithms which usually creates blurred images, but applies low computations.

$$G = S^T C \quad (4)$$

While iterative algorithms such as iterative linear-back projection (iLBP), shown in Eq. (5), provide more accurate images, its time complexity is high and linearly proportional with the number of iteration k .

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

where λ is the relaxation parameter, $S G_k$ is the forward problem solution, and k is the iteration number [34].

Typically, these algorithms involve a large number of matrix operations; therefore, implementing it on a parallel processing platform rather than a sequential execution on PC is crucial. For example, the LBP algorithm implemented on a 2.53 GHz-i5 PC with 4 GB RAM generates a 32x32 ($G = 1024$ elements) image in more than 1.5 s.

The solution of Equation (5) can be summarized in the following steps and described by a flowchart in Figure 2:

1. An initial image is obtained by the LBP algorithm Equation (4) using sensitivity in Equation (3)
2. The forward problem Equation (2) is solved to calculate a vector of capacitance measurements
3. Differences between the calculated and the actual measurements are multiplied by S^T to calculate pixels errors
4. The difference between the previous image and the pixel errors represents the new image

- (5) The termination is reached when the difference in step 3 reaches a certain acceptable value.

2.2. System Architecture. The ECT-DPU unit is responsible for image reconstruction and control of the ECT system. It consists of the image reconstruction subsystem (IR-unit) and the main DPU controller (DPU-C) as shown in Figure 3. The image reconstruction subsystem consists of the image reconstruction algorithm, the image reconstruction controller, and the associated memory and buffering blocks.

The core of the image reconstruction algorithm module (IR-alg) is a matrix processing realizing the three iLBP algorithm steps in Equation (5). Memory and buffering blocks required to store the input measured the electrode capacitance, the constant sensitivity matrix, and the computed

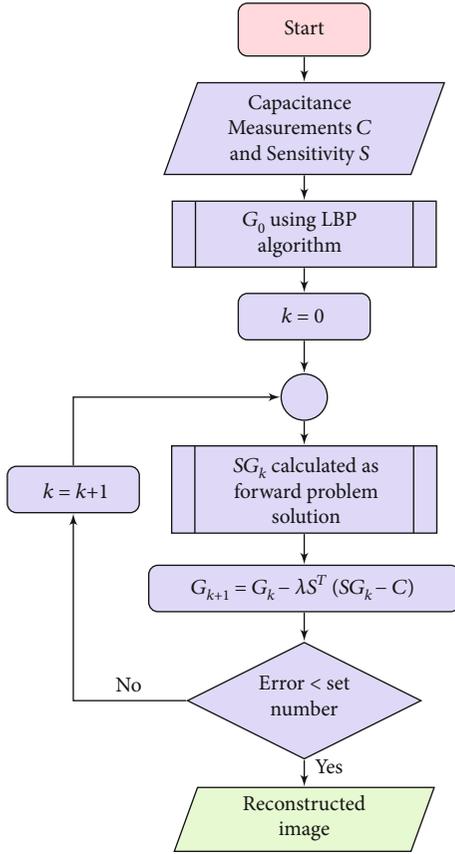


FIGURE 2: The iLBP image reconstruction algorithm in ECT.

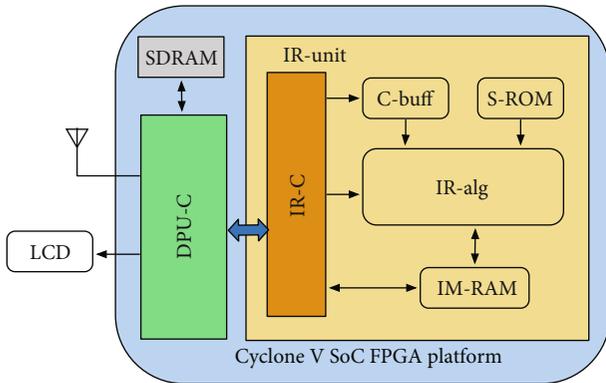


FIGURE 3: ECT digital processing unit.

image pixels that constitute the memory subsystem in the IR-unit. They are designated as the C Buffer, S-ROM, and IM-RAM block in Figure 3, respectively. The image reconstruction controller (IR-C) controls the IR-alg processing and coordinates the flow of data to and from the memory subsystem. It works as an interface between the image reconstruction subsystem and the main DPU controller.

The main DPU controller (DPU-C) is the interface to the external LCD and the wireless base station peripherals connected to the ECT-DPU system. It wirelessly collects the received data of the measured electrode capacitance and sends it to the image reconstruction subsystem. At the end

of frame processing, it collects the image-pixel vectors, stores it in the attached SDRAM, and displays it to the LCD.

2.3. Partitioning. The most important driving factors in embedded system design are the performance and flexibility. While high flexibility and low design effort can be gained by applying the software implementation, its performance gain is low. In contrast, the hardware intrinsic parallelism realizes excessive system performance, but its design complexity overhead is high.

Typically, embedded system is designed following the system-on-chip approach, where the whole-system components such as the processor, memory, dedicated hardware coprocessor, and the input-output peripherals are integrated in a single chip. The hardware-software (HW/SW) codesign of the embedded SoC allows the trade-off between attainable performance and flexibility. The hardware-software partitioning of an application in terms of software and hardware components is a key step in the embedded SoC HW/SW codesign [35].

Quantitative design metrics of the system building blocks are required to drive the partitioning process. These quantitative values such as latency (execution time), area, and power can be acquired using profiling, simulation, and static analysis of the system. The executable model and automatic code generation in the MBD allows handy verification and profiling data collection in order to assist in HW/SW partitioning decision. Analysis of the image reconstruction system exposes the computation intensiveness of the iLBP algorithm. Its core is a computation intensive process of repeated matrix multiplication and addition in large loop iterations, which makes it a viable candidate for the hardware realization on the FPGA fabric. Thus, the required performance gain can be achieved. The image reconstruction controller (IR-C) controls the flow of data to and from the Block-RAM housing of the sensitivity, capacitance, image matrices, and IR-alg module. Its communication is cost-effective to map it to the FPGA fabric rather than to the HPS software side.

On the other hand, the functionality of the DPU-C block as a control flow intensive state machine makes it a perfect candidate for the software mapping on the ARM processor inside the FPGA (SoC) platform. In addition, software implementation of the DPU-C block allows to exploit the legacy software drivers for these peripherals. The IR-alg has to be fed with the input-data from the sensitivity matrix, S , as well as the capacitance vector, C . Because the sensitivity matrix is very large in the ECT system, a careful system-level mapping decision has to be taken into consideration for the FPGA implementation. Since the sensitivity matrix has fixed constant elements, it can be hard-coded as part of the IR-alg module to be modeled as a single MATLAB function block in the MBD approach. In this approach, the sensitivity matrix is left to the synthesis tools to map it to registers scattered inside the FPGA fabric. Connecting such large sensitivity matrix to the synthesized computation elements of the IR-alg module in this way consumes a huge number of FPGA routing resources and suffers from long routing paths that might violate the timing constraints.

Using modular-based system approach to separate the memory requirement and its internal organization from the processing structure of the IR-alg module itself, the sensitivity matrix, S , is mapped to the FPGA Block-RAM, while the IR-alg algorithm is considered a separate module that can be modeled as a MATLAB function block in the MBD approach. In this case, an entire block of S matrix has to be ready and fed to the IR-alg module in each computation cycle. This model offers deterministic timing requirements and a small number of FPGA routing resources. The sensitivity matrix is modeled using this approach in our ECT-DPU system.

Based on the above reasoning as well as the collected profiling data, the ECT-DPU system is partitioned as illustrated in Figure 3. It shows the mapping of the ECT digital processing system to the hardware and software sides of the Cyclone V SoC FPGA platform.

2.4. Fixed-Point Representation and Word Length. A fixed-point version of the iLBP algorithm has to be generated for low-cost hardware implementation as well as high-performance gain and energy efficiency. Although the fixed-point word length can be set manually inside the MATLAB code of the IR-alg module, it is more appropriate to be generated automatically from the floating-point model with aid of the fixed-point conversion tool as part of the MBD workflow. Since increasing the word length consumes more hardware resources, the fixed-point conversion tool can be guided by the designer to set the word length in the fixed-point version of the iLBP algorithm to preserve similar precision of its floating-point counterpart.

3. Matrix-Vector Multiplication Segmentation Scheme

3.1. Matrix-Vector Multiplication. The matrix-vector multiplication (MVM) constitutes the pivotal computation structure in the LBP and iLBP image reconstruction algorithms in Equations (4) and (5), respectively, while the inner-product constitutes the kernel operation of the MVM. This section introduces the segmented inner-product architecture to serve as the core unit for resource-sharing-based approach of each matrix-vector multiplication stage of the image reconstruction algorithm. It is required to design and implement an efficient FPGA hardware architecture for the large matrix-vector multiplication to meet the real-time performance requirements without violating the hardware resource constraints. The proposed solution is to build each matrix-vector multiplication stage around a shared segmented parallel inner-product architecture to achieve the performance/resource-usage trade-off.

In this section, for the generic MVM problem, we will follow the generic XY notation. Then, the matrix and vector names, as well as their indices, are replaced with the corresponding notation of each stage in the iterative linear-back projection (iLBP) algorithm in Equation (5).

Let Y and X be one-dimensional fixed-point data vectors with sizes of $M \times 1$ and $N \times 1$, respectively. The MVM, $Y =$

AX , is represented as

$$y_i = \sum_{j=1}^N a_{ij}x_j \forall i = 1 \text{ to } M, \quad (6)$$

where

$$A = A_{M \times N} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M1} & a_{M2} & \cdots & a_{MN} \end{bmatrix}. \quad (7)$$

3.2. Multicycle Path Quantification. Define the computation cycle as the atomic processing of multiplication and addition operations applied to data from the two input vectors of the inner-product. Let the combinational path delay, dcp , indicate the delay required to propagate the signals through the combinational path of the inner-product hardware unit in a single computation cycle. The inner-product hardware unit requires a computation time, CT , equals to its dcp for combinational data-processing in each computation cycle. Thus, the computation time of the combinational path can be realized in a single clock-cycle with a period equal to dcp , or in Kc , multiple clock-cycles equal to $\lceil dcp/tm \rceil$ with clock period tm . Multicycle path realization is achieved via enabling the combinational path result to be written to the storage element at the end of the computation cycle, at clock-cycle Kc , and incurs a computation time of

$$CT = Kc.tm; Kc = \lceil dcp/tm \rceil. \quad (8)$$

Thus, the system design runs in an operating frequency with period, tck , equals to tm in case of multicycle path realization and equals to dcp in case of single-cycle path realization. The multicycle combinational path realization can also be set by the FPGA SDC timing tools (c.f. Altera TimeQuest [10] Analysis tool). Data elements of the inner-product vectors have to be ready to the computation hardware via reading from their memory locations such as FPGA Block-RAM.

3.3. Parallel Inner-Product. The fully serial realization of MVM can be implemented in hardware using a single multiply-accumulate unit with a controller to generate the row and column indices in a nested-loop similar to software implementation, as has been introduced in [12]. Although this serial implementation requires a single multiplier and single adder, it suffers from long computation time of MN computation cycles. On the other hand, fully parallel implementation of MVM can achieve high performance and be realized in a single computation cycle [36], but on the cost of a huge number of FPGA resources that requires MN multiplier and $M(N-1)$ adders. A performance/resource-usage trade-off is a viable approach to meet the embedded system time constraint and/or achieving high performance while still within the available FPGA resources. Building the matrix-vector multiplication around a shared parallel inner-product architecture can fulfill this trade-off.

Exploiting the intrinsic parallelism among multiplication operations of the inner-product procedure to build a parallel inner-product architecture will increase the performance in the cost of increasing the required resources. It consists of N multiplier and $(N - 1)$ adders in order to achieve the inner-product between two vectors of length N in a single computation cycle. Combinational path delay can be shortened by performing the multiplication operation in parallel for all pairs of elements of the inner-product input vectors; then, the multiplication results are summed with a set of adders to produce the final inner-product result, as shown in Figure 4.

The parallel inner-product architecture with multicyle path realization of its combinational path requires input-data to be read from their memory locations in the FPGA Block-RAM with a read-time, RT , as in Equation (9), while its single-cycle realization incurs input-data read-time as in Equation (10). This contrasts the reduction of the input-data read-time with a factor of $Kc = \lceil dcp/tm \rceil$ in case of the multicyle path as opposed to the single-cycle path realization. On the other hand, the computation time is almost similar that is $\lceil dcp/tm \rceil \cdot tm$, and dcp , in case of the multi- and single-cycle path realization, respectively.

$$RT = N \cdot tm = N \cdot \lceil dcp/tm \rceil, \quad (9)$$

$$RT = N \cdot dcp. \quad (10)$$

3.4. Segmented Inner-Product Architecture. Combinational path delay tends to be long for large vectors, which leads to long computation cycle delay and a large number of resources. The resource usage, as well as combinational path delay, can be greatly shortened through segmenting the inner-product input vectors to multiple segments of length SL , and the complete-vector inner-product is completed in $Ns = \lceil N/SL \rceil$ computation cycles for length N vectors, where Ns denotes the number of segments. In each computation cycle, the resultant segmented inner-product is added to the previously buffered partial inner-product, as it is expressed by Equation (13).

Following the notation in Equation (6), where A_i and X are $1 \times N$ and $N \times 1$ vectors, respectively, the inner-product can be written as

$$y_i = A_i X. \quad (11)$$

Then, the partial inner-product of a single segment, s of length SL , is written as

$$y_{is} = \sum_{j=(s-1)SL+1}^{s \cdot SL} A_{ij} X_j. \quad (12)$$

Thus, the inner-product can be written in the segmented form as

$$y_i = \sum_{s=1}^{Ns} y_i s = y_{i1} + y_{i2} + \hat{a}, \quad (13)$$

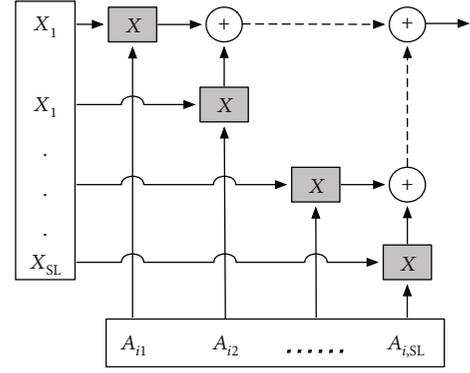


FIGURE 4: Parallel inner-product architecture.

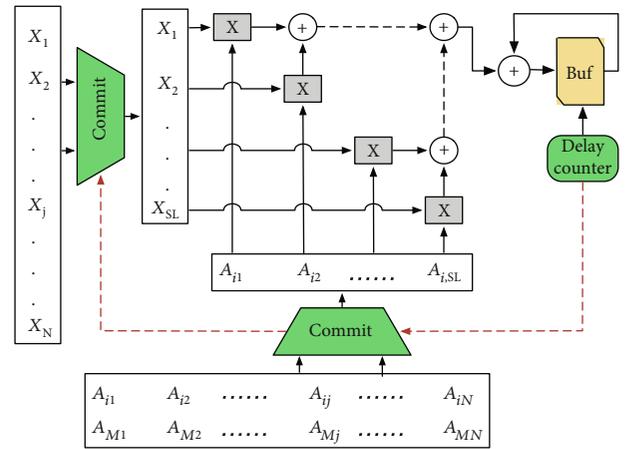


FIGURE 5: Segmented inner-product architecture.

where the partial inner-product of each segment is computed in a single computation cycle.

In each computation cycle, it is required to feed the segmented inner-product unit with only a segment of length S from both input vectors, instead of the whole vectors, which greatly shortens the combinational path delay and its required hardware resources.

Figure 5 illustrates the segmented inner-product architectures.

It requires parallel SL multiplier and SL adders, and it experiences a combinational path delay, $dcps$, expressed by Equation (14). Its combinational path delay is realized as multiple clock-cycles via a delay-counter that enables writing the partial inner-product to a memory buffer at the end of the computation cycle, at clock-cycle $Kc = \lceil dcps/tm \rceil$.

$$dcps = dm + da \cdot sl, \quad (14)$$

where dm and $da \cdot sl$ are the propagation delay through the multiplier and the set of adders required for the inner-product of length SL vectors, respectively. In order to reduce the propagation delay, the set of adders is organized as a tree-like structure.

TABLE 1: The number of FPGA resources used in different inner-product unit proposals.

| | Qasim [12] | Tiwari [33] | Segmented (proposed) |
|--------------------|-------------|------------------------|-----------------------|
| Multipliers | 1 | N | SL |
| Input ports | 2 | $2N$ | $2p$ |
| $dcps$ | $d_m + d_a$ | $d_m + \log_2(N-1)d_a$ | $d_m + \log_2(SL)d_a$ |
| Computation cycles | N | 1 | $\lceil N/SL \rceil$ |

For length N vectors, the segmented inner-product architecture requires the same input-data read-time as the nonsegmented architecture, expressed in Equation (9). On the other hand, it dictates a computation time of

$$CT = \lceil N/SL \rceil \cdot \lceil dcps/tm \rceil \cdot tm. \quad (15)$$

Equation (15) reveals that the segmented inner-product architecture incurs some computation time overhead compared to the nonsegmented parallel architecture Equation (8). As will be shown in the experimental results, this overhead is very small compared to the advantages of massive decrease in the number of required resources.

The execution time, ET , for completing the inner-product calculation is the sum of the input-data read-time, RT , and the computation time, CT . Assuming that each data element requires a read time of single clock-cycle, the execution time of the length N inner-product, represented as the number of clock-cycles, is

$$ET = N + \lceil N/SL \rceil \cdot \lceil dcps/tm \rceil. \quad (16)$$

The segment length is a design degree of freedom that allows the trade-off between the performance and the resources usage and determines the level of computation parallelism as well as the maximum number of input ports. The number of parallel input ports is another design degree of freedom for more performance gain. Increasing the number of parallel input ports using multiple Block-RAMs for feeding segmented inner-product unit with multiple elements of the two input vectors, A_i and X , simultaneously, decreases the read-time so as to increase the performance gain.

Let p denote the parallel input ports, $p < N$; the execution time of the length N inner-product, represented as the number of clock-cycles, becomes

$$ET = \frac{N}{p} + \left\lceil \frac{N}{SL} \right\rceil \cdot \left\lceil \frac{dcps}{tm} \right\rceil. \quad (17)$$

The proposed segmented architecture turns the inner-product unit to be a parametric module, with the segment length set by the designer to fulfill the required performance while meeting the resources constraint. Table 1 compares hardware architectures requirements of parallel inner-product computations that serve as the kernel operations in matrix multiplication algorithms.

3.5. Segment-Based Matrix-Vector Multiplication Architecture. The segmented inner-product architecture

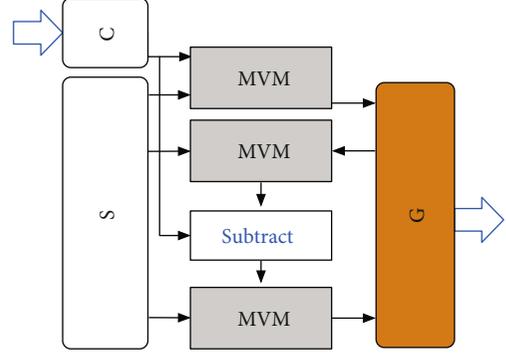


FIGURE 6: iLBP Architecture using Segment-based MVM.

serves as the core unit for resource-sharing-based approach of matrix multiplication; and each matrix-vector multiplication stage of the image reconstruction algorithm is built around this shared parallel segmented inner-product architecture.

Using Equation (13) and letting $Y_s = [y_{1s} y_{2s} \dots y_{Ms}]^T$, the generic matrix-vector multiplication, $Y = AX$, in the segmented form, is written as in Equation (18) and represented in Equation (20)

$$Y = Y_1 + \dots + Y_s + \dots + Y_{Ns}, \quad (18)$$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} a_{11} & \dots & a_{1,SL} & a_{1,SL+1} & \dots & \dots & a_{1,N} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{M1} & \dots & a_{M,SL} & a_{M,SL+1} & \dots & \dots & a_{M,N} \end{bmatrix}, \quad (19)$$

$$\begin{bmatrix} x_1 \\ \vdots \\ x_{SL} \\ x_{SL+1} \\ \vdots \\ x_N \end{bmatrix}. \quad (20)$$

Using Equation (18), MVM requires an execution time of

$$ET = M \cdot \frac{N}{p} + M \cdot \left\lceil \frac{N}{SL} \right\rceil \cdot \left\lceil \frac{dcps}{tm} \right\rceil. \quad (21)$$

In the model-based design at the system-level, the segment length works as an input to the design flow. First, the designer calculates the estimated execution time (using Equation (21)) for small segment length—in order to preserve the HW resources—and checks the achievement of the required performance. Then, the segment length can be increased, and the estimated execution time is recalculated until the performance requirement is met. Inserting the designated segment length in the system-level model, the system design can be flexibly tailored to the designer specifications to fulfill the required performance while meeting the resources constraint. Coupling the allocated segment length with the MBD procedures will greatly minimize the development time and effort and alleviate the designer from remodeling the system in each design cycle.

Based on this segment-based MVM architecture, the iLBP architecture is shown in Figure 6. The first LBP matrix-vector multiplication stage of the iLBP algorithm is organized as to feed each row vector of S^T matrix as well as the C vector to the segmented inner-product unit in $\lceil M/SL \rceil$ computation cycles. In this architecture, LBP MVM requires $N \cdot \lceil M/SL \rceil$ computation cycles. Similarly, the second and third iLBP MVM stages require $M \cdot \lceil N/SL \rceil$ and $N \cdot \lceil M/SL \rceil$ computation cycles, respectively.

4. System Modeling and Implementation

4.1. Hardware Platform. The Altera Cyclone V SoC FPGA platform [8] is used for the ECT-DPU system implementation as a SoC platform. This SoC platform is an evaluation board. It includes the Cyclone V 5CSXFC6D6F31C6 FPGA device integrating a multicore ARM processor subsystem into the FPGA fabric [37], in addition to DDR3 memory and common interface controllers. The dual-core ARM Cortex-A9 MPCore processor operating at a 925 MHz subsystem connected to a rich set of connected peripherals constitutes the hard processing system (HPS) side of the Cyclone V SoC device. The communication between the HPS and the FPGA fabric is achieved through the Standard AXI4 bridge. The HPSFPGA AXI bridges allow masters in the FPGA fabric to communicate with slaves in the HPS logic, and vice versa [37].

4.2. System-Level Modeling and Simulation. MathWorks has introduced a complete model-based design platform based on its MATLAB environment [15]. It covers the whole design flow from modeling and simulation using the MATLAB/Simulink down to the deployment on the FPGA SoC platform. MATLAB's MBD workflow depends on two key technologies, the "HDL Coder" toolbox used to generate synthesizable HDL code and the Embedded Coder toolbox for the embedded C code generation, from both MATLAB code and Simulink model.

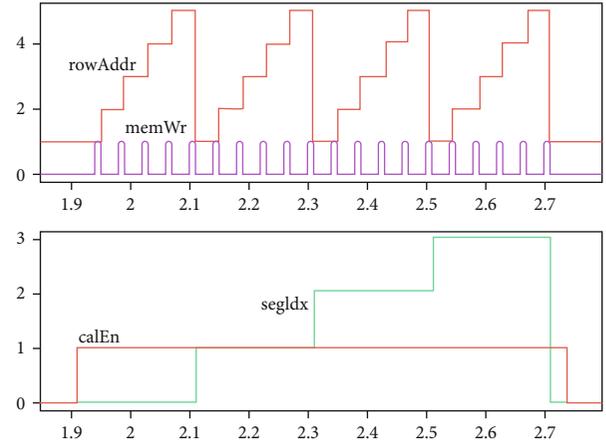


FIGURE 7: Timing behavior of segmented inner-product.

The ECT-DPU system design procedures follow the MBD approach. It is based on the MATLAB MBD flow using the MATLAB HDL Coder and Embedded Coder toolboxes [15]. For a complete process, these toolboxes are linked to the hardware synthesis tools: Altera suite of development tools, Quartus II, and the software compilation tools: the SoC Embedded Design Suite (SoC-EDS) [11].

The ECT-DPU based on the proposed segment-based MVM architecture is modeled using MATLAB Simulink and HDL Coder toolboxes. Its functional behavior is verified at the system-level. The equivalent VHDL code is generated, synthesis and timing analysis are performed via integration with Altera Quartus II, and the design is deployed to the Altera Cyclone V FPGA device. These FPGA design steps are automated using the HDL Workflow Advisor of the HDL Coder.

The image reconstruction subsystem of the ECT-DPU system is coded with HDL-synthesizable MATLAB code, while the main DPU controller is modeled with a C-compatible MATLAB code. Both are modeled as a MATLAB function block inside a Simulink model. In MBD, the system-level simulation of the cycle-accurate model allows functional verification as well as cycle-based performance measurement. The MATLAB Simulation Data Inspector tool is used in this purpose for the ECT-DPU to inspect and verify the behavior of the segmented inner-product core used inside each of the three stages of the iLBP IR-*alg*, as well as the handshaking signals between the DPU-C and IR-C controllers' modules. Figure 7 illustrates the timing behavior at cycle-level of the segmented inner-product core in a single stage of MVM for 5x32-element matrix and 32-element vector example, with the SL parameter set to 8. In addition, HPS-to-FPGA, h2f, and FPGA-to-HPS, f2h, handshaking signals are shown in Figures 8(a) and 8(b).

4.3. Code Generation. Using the HDL Workflow Advisor tool of the HDL Coder toolbox, the HDL-code as an IP-core for the image reconstruction subsystem (IR-*alg* and IR-C modules) is generated. In the same process, the IP interface logic as well as its abstract software interface model to the ARM

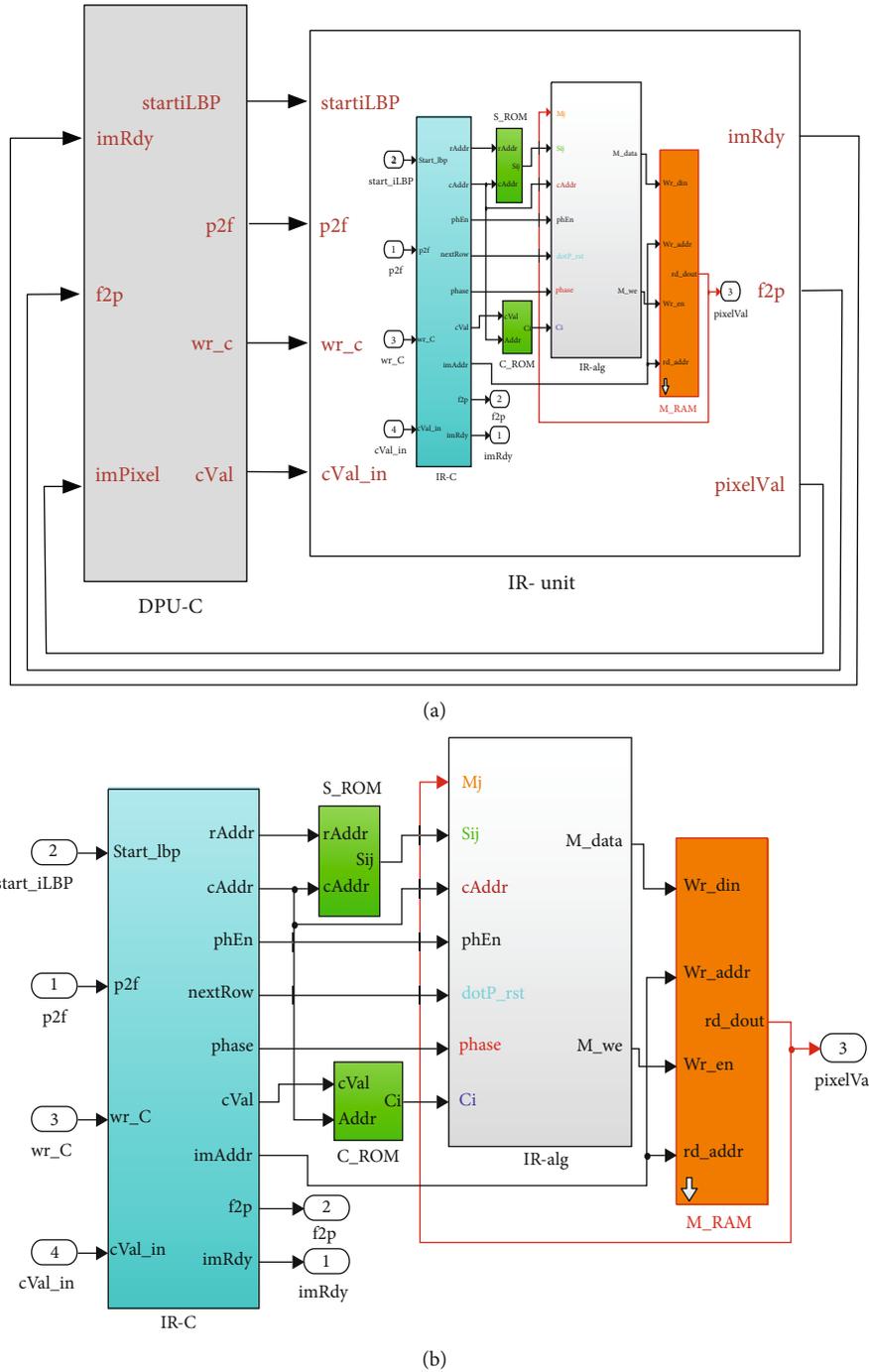


FIGURE 8: ECT-DPU Simulink model.

processor is automatically generated according to the AXI4 interface standard.

On the other hand, the generation of the DPU-C corresponding C code for the ARM processor is handled by the Embedded Coder toolbox so as to be connected to the generated AXI4 interface model. The compiled and bit-stream configuration files of the generated C and HDL code parts, respectively, of the system can be deployed to the FPGA platform directly from within the HDL Workflow Advisor tool. The generated C code for the DPU-C is ready to be integrated

with the rest of the software components of the complete ECT-DPU system. Using the Qsys tools of the Quartus II CAD system, the generated HDL IP-core for the image reconstruction subsystem can be reused within other related ECT-DPU system.

4.4. *FPGA Test and Verification.* The model-based design approach is used for the design and implementation of the ECT-DPU starting from an executable functional system-level model down to the FPGA implementation and testing.

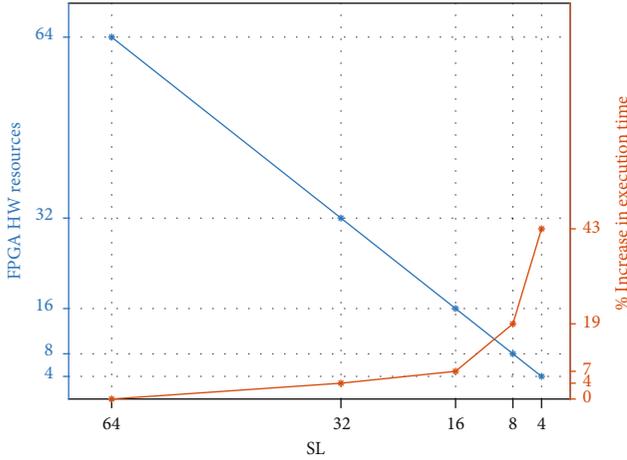


FIGURE 9: Effect of changing the segment length parameter on the hardware resources and the execution time.

TABLE 2: FPGA HW resources.

| SL | 64 | 32 | 16 | 8 | 4 |
|-----------------------------|------|-----|-----|-----|-----|
| Logic utilization (in ALMs) | 1496 | 746 | 382 | 216 | 113 |
| DSP blocks | 64 | 32 | 16 | 8 | 4 |
| Registers | 727 | 375 | 199 | 111 | 67 |

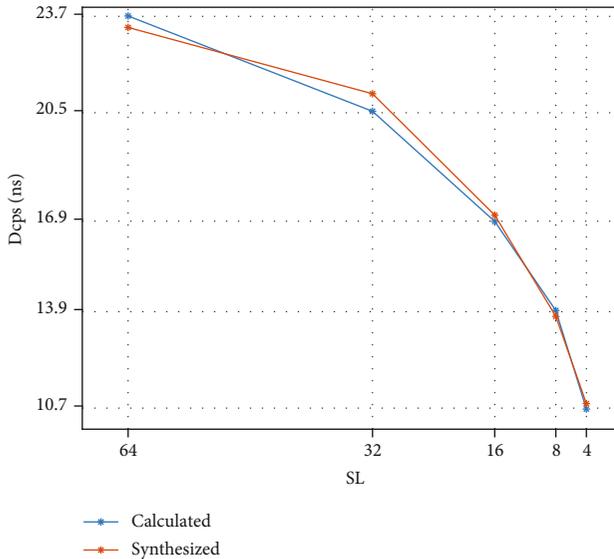


FIGURE 10: Propagation delay of the segmented inner-product.

In each design cycle, the segment length parameter of the segment-based MVM architecture, of the three stages of the iLBP image reconstruction algorithm, is set with one of the test values as an input to the design flow at the system-level. In each design cycle, the ECT-DPU is simulated, tested, and verified, and the code is generated for both the FPGA fabric and the HPS ARM processor.

Using MBD can greatly minimize the development time and reduces the design cycle as well as alleviates the remod-

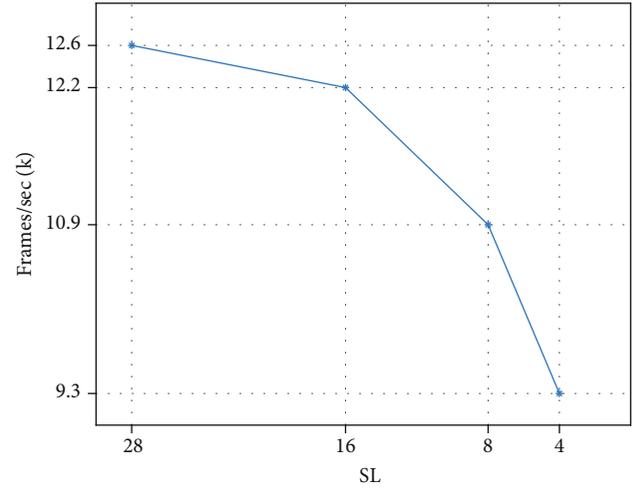


FIGURE 11: LBP frame rate.

eling of the system in each design cycle. The effect of changing the segment length parameter of the segment-based MVM architecture on the required hardware resources and the execution time overhead illustrated with 64-element vectors, as an example is shown in Figure 9. The segment length of its segmented inner-product shared core unit is decreased from the full-length vector of 64 elements to 4 elements. Figure 9 shows the required hardware resources exemplified by the number of multipliers, alongside with the overhead in the execution time as a percentage of the nonsegmented version. The analytical model in Equation (16) is used to generate the calculated data as the first step in the highest-level of MBD scheme. On the other hand, down to the code generation and implementation, the synthesized FPGA hardware resources is recorded in Table 2 for these segment length test values. The Altera TimeQuest [10] analysis tool on the CVSoC FPGA device with an operating frequency of 100 MHz is used to obtain the propagation delay of the combinational path components. The propagation delay of the synthesized segmented inner-product is compared against the calculated analytical data as illustrated in Figure 10 for a 64-element vectors. It shows the close matching of the analytical model in Equation (16) to the synthesized results.

Both of the calculated data and the synthesized FPGA hardware resources illustrate the linear effect of reducing the segment length on minimizing FPGA hardware resources and its impact on the trade-off between the performance and resource usage. This illustrates how the MBD approach can help the designer of the ECT-DPU to experiment with input parameters to the design flow starting at the system-level model down to the FPGA implementation without the need of remodeling the system in each design cycle.

The segmentation effect on the LBP image reconstruction algorithm, Equation (4), is illustrated in Figure 11. The illustrated frame rates are in kilo frames/sec for a 28x256-element sensitivity matrix in the segment-based MVM architecture. It illustrates that the segmentation scheme achieves high resource saving of 43% and 71% (corresponding to segment length of 57% and 29% of the full-length vector) for a small degradation in a frame rate of 3% and 14%, respectively.

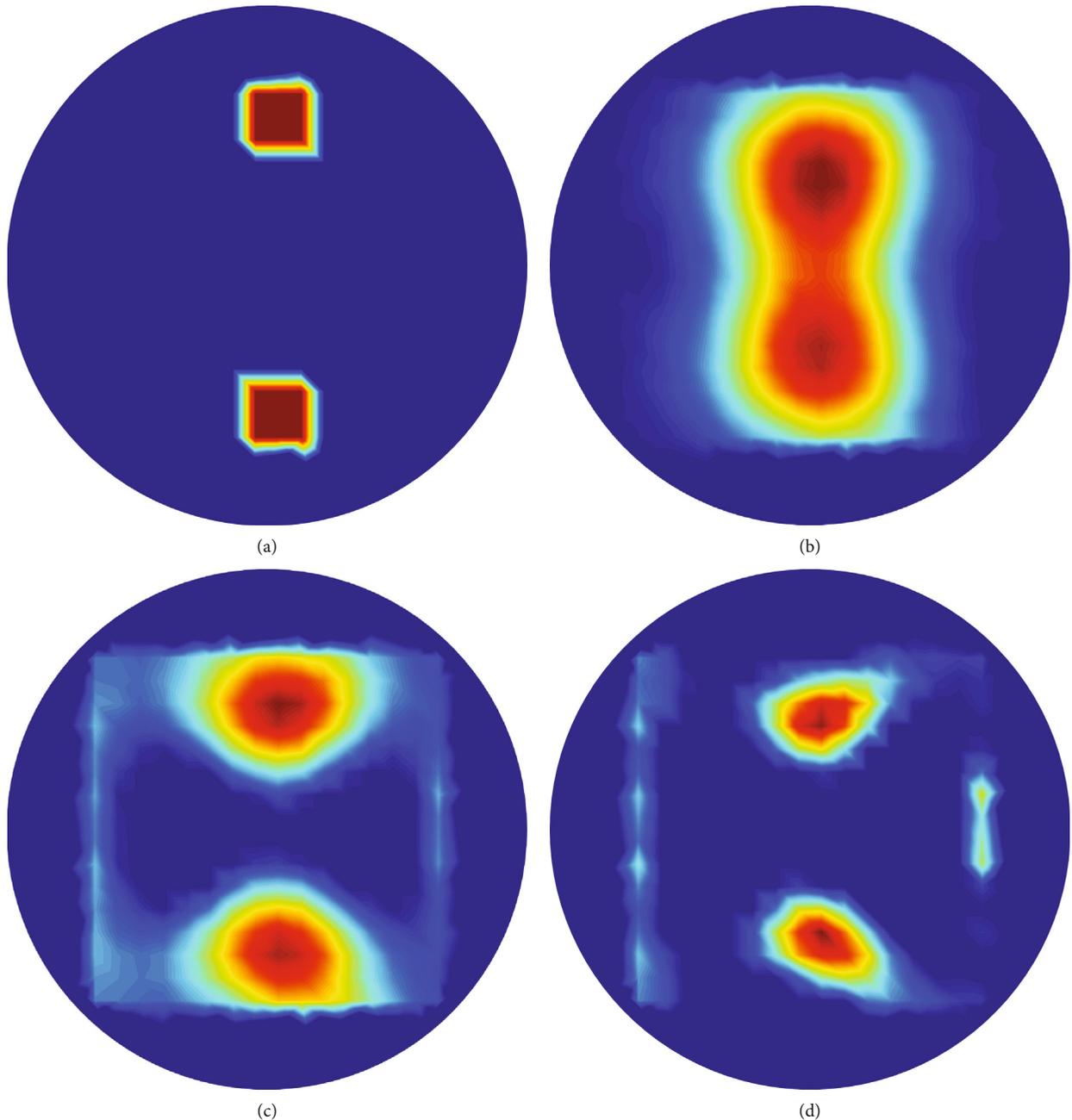


FIGURE 12: Reconstructed tomographic images on FPGA SoC using LBP and iLBP algorithms. (a) Original image. (b) LBP. (c) iLBP 10. (d) iLBP 200.

Pipelined architecture of the three-stage iLBP image reconstruction algorithm, Equation (5), achieves the same frame throughput.

The proposed algorithm is validated by applying synthetic data which collected from an ECT system shown in Figure 1 with $N = 8$ electrodes. These electrodes are uniformly distributed around the 2-D plane of the vessel to be imaged. The voltage is applied on one electrode at a time to work as transmitter while the rest electrodes are receivers. By measuring the produced charges on these electrodes, $M = N(N - 1)/2 = 28$ independent mutual capacitance values were collected. Typically, the quality of the generated images

can be enhanced by increasing the number sensing electrodes around the imaging area. However, this radically increased the complexity of the measuring circuit as well as the cost of the hardware design of the system.

The results shown here represent two different permittivity variations. The mutual capacitances between the electrodes corresponding to each permittivity distribution were estimated using the FEM model [4]. The FEM mesh consists of 720 linear triangular elements. An image area of size 16×16 pixel is selected in the center of the FEM model to reduce the complexity of the computation. Therefore, the sensitivity matrix S of size 28×256 and a measurement electrode

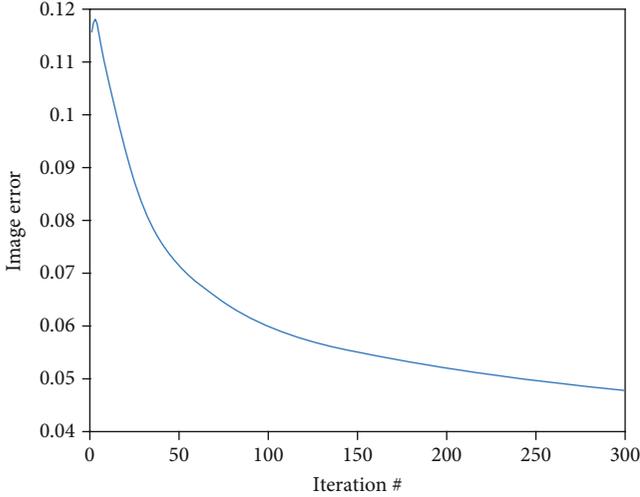


FIGURE 13: Image error for the objects shown in Figure 12.

capacitance C vector of size 28 elements were applied during the experiments. The permittivity value of the inhomogeneous material is 1.8, while the permittivity value of the area when it is empty is 1.0. The sensitivity matrix is also calculated based on Equation (3) from the FEM model. The relaxation parameter $\lambda = 10^{-3}$ was adjusted and selected to give better reconstructed images.

The MBD approach for the ECT-DPU is tested and verified for both LBP and iLBP image reconstruction algorithms implemented on the FPGA. The capacitance measurements and the sensitivity values are stored on the SDRAM. LBP and iLBP image reconstruction algorithms are applied to detect the distribution of the materials inside the imaging area. Two objects separated by a distance are used to test both algorithms as shown in Figure 12. The LBP reconstructed image on FPGA is shown in Figure 12(b), while the iLBP reconstructed image on FPGA for 10 and 200 iterations are depicted in Figures 12(c) and 12(d), respectively. Figure 12 proves the ability of the FPGA implementation of the ECT-DPU in detecting multiple objects from the capacitance measurements of the ECT system. Figures 12(c) and 12(d) verify that the iLBP algorithm is accurately able to detect the size and the location of that object more than the LBP algorithm. In addition, this illustrates the trade-off between the number of iLBP iterations and the reconstructed image quality resulted from the ECT-DPU FPGA implementation.

Errors between the real and the reconstructed images were calculated as Equation (22) [26].

$$\text{Image Error} = \frac{\|\hat{G} - G\|}{\|G\|}, \quad (22)$$

where \hat{G} is the reconstructed image and G is the real image distribution. The quality of the images increases as the relative error decreases.

The error during 300 iterations for the objects shown in Figure 12 is illustrated in Figure 13. The error decreases as the number of the iterations increases.

5. Conclusion

In this paper, a segment-based matrix-vector multiplication architecture is proposed to work as the core unit of the ECT digital processing unit. In addition, the hardware-software (HW/SW) codesign of the ECT digital processing unit is proposed. The design and implementation of the ECT-DPU follow the MBD approach. It is based on the MATLAB MBD flow using the MATLAB HDL Coder and Embedded Coder toolboxes, linked to the Altera suite of development tools, Quartus II and SoC-EDS.

In each design cycle, the segment length parameter of the segment-based MVM architecture, of the three stages of the iLBP image reconstruction algorithm, is set with one of the test values as an input to the design flow at the system-level. In each design cycle, the ECT-DPU is simulated, tested, and verified, and the code is generated for both FPGA fabric as well as the HPS ARM processor. The architecture was evaluated and deployed to the Altera Cyclone SoC FPGA platform. These FPGA design steps are automated using the HDL Workflow Advisor of the HDL Coder.

Design and implementation of the ECT-DPU via MBD has greatly minimized the development time and reduces the design cycle as well as alleviates remodeling the system in each design cycle. Both of the calculated data and the synthesized FPGA hardware resources illustrate the linear effect of reducing the segment length on minimizing FPGA hardware resources and its impact on the trade-off between the performance and resource usage. In LBP algorithm, segmentation scheme has illustrated high resource saving of 43% and 71% for a small degradation in frame rate of 3% and 14%, respectively.

Data Availability

Data is available per request.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was funded by the National Plan for Science, Technology and Innovation (MAARIFAH), King Abdulaziz City for Science and Technology, the Kingdom of Saudi Arabia (grant number 13-ELE469-10).

References

- [1] N. A. A. Rahman, R. A. Rahim, A. M. Nawi et al., "A review on electrical capacitance tomography sensor development," *J. Teknol*, vol. 73, no. 3, pp. 35–41, 2015.
- [2] M. Meribout and S. Teniou, "A pipelined parallel hardware architecture for 2-d real-time electrical capacitance tomography imaging using interframe correlation," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 4, pp. 1320–1328, 2017.

- [3] W. Yang, Y. Li, Z. Wu et al., "Multiphase OW measurement by electrical capacitance tomography," in *2011 IEEE International Conference on Imaging Systems and Techniques*, pp. 108–111, Batu Ferringhi, Malaysia, 2011.
- [4] M. A. Abdelrahman, A. Gupta, and W. A. Deabes, "A feature-based solution to forward problem in electrical capacitance tomography of conductive materials," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 2, pp. 430–441, 2011.
- [5] A. K. Allam and W. A. Deabes, "Electrical capacitance tomography digital processing platform (ECT-DPU)," in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, pp. 4767–4771, Florence, Italy, 2016.
- [6] R. Tessier, K. Pocek, and A. DeHon, "Reconfigurable Computing Architectures," *Proceedings of the IEEE*, vol. 103, no. 3, pp. 332–354, 2015.
- [7] H. Zhou, L. Xu, Z. Cao, X. Liu, and S. Liu, "A complex programmable logic device-based high-precision electrical capacitance tomography system," *Measurement Science and Technology*, vol. 24, no. 7, p. 074006, 2013.
- [8] Altera Corporation, *Cyclone V SoC Development Board Schematic*, Tech. rep., Altera, 2015, https://www.altera.com/content/dam/altera-www/global/en_US/support/boards-kits/C5SOCDEVKITE.pdf.
- [9] G. Xilinx and S. Guide, *Zynq-7000 All programmable SoC technical reference manual (UG585)*, Tech. rep., Xilinx, 2014, <https://www.xilinx.com/support/documentation/userguides/ug585-Zynq-7000-TRM.pdf>.
- [10] Altera, *Quartus Prime Standard Edition Handbook Volume 1: Design and Synthesis*, Tech. rep., Altera, 2015, https://www.altera.com/en_US/pdfs/literature/hb/qts/qts-qps-5v1.pdf.
- [11] Altera Corporation, *Altera SoC Embedded Design Suite (ug-1137)*, Tech. rep., Altera, 2014, https://www.altera.com/en_US/pdfs/literature/ug/ug_soc_ed.pdf.
- [12] S. M. Qasim, A. A. Telba, and A. Y. AlMazroo, "FPGA design and implementation of matrix multiplier architectures for image and signal processing applications," *IJCSNS - Int. J. Comput. Sci. Netw. Secur.*, vol. 10, no. 2, pp. 168–176, 2010.
- [13] F. Cacciatore, R. Sanchez, A. Agenjo et al., "Rapid deployment of design environment for EUCLID AOCs design," in *6th International Conference on Astrodynamics Tools and Techniques*, Darmstadt, Germany, 2016.
- [14] O. Raque and K. Schneider, "Towards the standardization of plug-and-play devices for model-based designs of embedded systems," in *2016 11th IEEE Symposium on Industrial Embedded Systems (SIES)*, pp. 1–4, Krakow, Poland, 2016.
- [15] Mathworks, *Model based design toolbox*, Tech. rep, Mathworks, 2018, <https://www.mathworks.com/solutions/model-based-design.html>.
- [16] X. Cai, M. Zhou, and X. Huang, "Model-based design for software Defined radio on an FPGA," *IEEE Access*, vol. 5, no. 1, pp. 8276–8283, 2017.
- [17] S. B. Junior, V. C. De Oliveira, and G. B. Junior, "Software Defined Radio Implementation of a QPSK Modulator/Demodulator in an Extensive Hardware Platform Based on FPGAs Xilinx ZYNQ," *Journal of Computer Science*, vol. 11, no. 4, pp. 598–611, 2015.
- [18] N. Yang, G. Hua, J. Li-li, and Z. Peng-yu, "Model-based design methodology for digital up and down conversion of software Defined radio," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 11, no. 4, pp. 27–36, 2016.
- [19] K. M. Deliparaschos, K. Michail, S. G. Tzafestas, and A. C. Zolotas, "A model based embedded control hardware/software co-design approach for optimized sensor selection of industrial systems," in *2015 23rd Mediterranean Conference on Control and Automation (MED)*, pp. 889–894, Torremolinos, Spain, 2015.
- [20] F. Memon, F. Jameel, M. Arif, and F. A. Memon, "Model Based FPGA Design of Histogram Equalization," *Sindh University Research Journal-SURJ (Science Series)*, vol. 48, no. 2, pp. 435–440, 2016.
- [21] J. W. Jang, S. Choi, and V. K. K. Prasanna, "Area and time efficient implementations of matrix multiplication on FPGAs," in *2002 IEEE International Conference on Field-Programmable Technology, 2002. (FPT). Proceedings.*, pp. 93–100, Hong Kong, China, 2002.
- [22] R. Guha, "Implementation of Kalman filter and Sonar image processing on FPGA platform," in *2015 International Conference on Industrial Engineering and Operations Management (IEOM)*, pp. 1–7, Dubai, United Arab Emirates, 2015.
- [23] S. R. Huddar, S. R. Rupanagudi, R. Ravi, S. Yadav, and S. Jain, "Novel architecture for inverse mix columns for AES using ancient Vedic Mathematics on FPGA," in *2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 1924–1929, Mysore, India, 2013.
- [24] W. Deabes, M. Abdallah, O. Elkeelany, and M. Abdelrahman, "Reconfigurable wireless stand-alone platform for Electrical Capacitance Tomography," in *2009 IEEE Symposium on Computational Intelligence in Control and Automation*, pp. 112–116, Nashville, TN, USA, 2009.
- [25] A. F. Firdaua and M. Meribout, "A new parallel VLSI architecture for real-time electrical capacitance tomography," *IEEE Transactions on Computers*, vol. 65, no. 1, pp. 30–41, 2015.
- [26] W. Q. Yang and L. Peng, "Image reconstruction algorithms for electrical capacitance tomography," *Measurement Science and Technology*, vol. 14, no. 1, pp. R1–R13, 2003.
- [27] W. A. Deabes and M. A. Abdelrahman, "A nonlinear fuzzy assisted image reconstruction algorithm for electrical capacitance tomography," *ISA Transactions*, vol. 49, no. 1, pp. 10–18, 2010.
- [28] K. Andryc, M. Merchant, and R. Tessier, "FlexGrip: a soft GPGPU for FPGAs," in *2013 International Conference on Field-Programmable Technology (FPT)*, pp. 230–237, Kyoto, Japan, 2013.
- [29] S. Williams, L. Oliker, R. Vuduc, J. Shalf, K. Yelick, and J. Demmel, "Optimization of sparse matrix-vector multiplication on emerging multicore platforms," *Parallel Computing*, vol. 35, no. 3, pp. 178–194, 2009.
- [30] W. Deabes, "Fpga implementation of ect digital system for imaging conductive materials," *Algorithms*, vol. 12, no. 2, p. 28, 2019.
- [31] S. Kestur, J. D. Davis, and E. S. Chung, "Towards a universal FPGA matrix-vector multiplication architecture," in *2012 IEEE 20th International Symposium on Field-Programmable Custom Computing Machines*, pp. 9–16, Toronto, ON, Canada, 2012.
- [32] Sangjin Hong, Kyoung-Su Park, and Jun-Hee Mun, "Design and implementation of a high-speed matrix multiplier based on word-width decomposition," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, no. 4, pp. 380–392, 2006.
- [33] S. Tiwari and N. Meena, "Efficient hardware design for implementation of matrix multiplication by using PPI-SO,"

International Journal of Innovative Research in Computer and Communication Engineering, vol. 1, no. 4, pp. 1020–1024, 2013.

- [34] O. Isaksen, “A review of reconstruction techniques for capacitance tomography,” *Measurement Science and Technology*, vol. 7, no. 3, pp. 325–337, 1996.
- [35] P. R. Schaumont, *A Practical Introduction to Hardware/Software Code-sign*, Springer US, Boston, MA, 2013.
- [36] S. Tiwari, S. Singh, and N. Meena, “FPGA Design and Implementation of matrix multiplication architecture by PPI-MO techniques,” *International Journal of Computers and Applications*, vol. 80, no. 1, pp. 19–22, 2013.
- [37] Altera, *Cyclone V Hard Processor System Technical Reference Manual*, Tech. rep., Altera, 2016, <http://www.altera.com/literature/hb/cyclone-v/cv.5v4.pdf>.