

## FPGA-Based Architecture for a Generalized Parallel 2-D MRI Filtering Algorithm

Sami Hasan, Said Boussakta and Alex Yakovlev

School of Electrical, Electronic and Computer Engineering,  
Newcastle University Newcastle Upon Tyne, NE1 7RU, England, UK

**Abstract:** **Problem statement:** Current Neuroimaging developments, in biological research and diagnostics, demand an edge-defined and noise-free MRI scans. Thus, this study presents a generalized parallel 2-D MRI filtering algorithm with their FPGA-based implementation in a single unified architecture. The parallel 2-D MRI filtering algorithms are Edge, Sobel X, Sobel Y, Sobel X-Y, Blur, Smooth, Sharpen, Gaussian and Beta (HYB). Then, the nine MRI image filtering algorithm, has empirically improved to generate enhanced MRI scans filtering results without significantly affecting the developed performance indices of high throughput and low power consumption at maximum operating frequency. **Approach:** The parallel 2-d MRI filtering algorithms are developed and FPGA implemented using Xilinx System Generator tool within the ISE 12.3 development suite. Two unified architectures are behaviorally developed, depending on the abstraction level of implementation. For performance indices comparison, two Virtex-6 FPGA boards, namely, xc6vlX240T1-1lff1759 and xc6vlX130T1-1lff1156 are behaviorally targeted. **Results:** The improved parallel 2-D filtering algorithms enhanced the filtered MRI scans to be edge-defined and noise free grayscale imaging. The single architecture is efficiently prototyped to achieve: high filtering performance of (11230 frames/second) throughput for 64\*64 MRI grayscale scan, minimum power consumption of 0.86 Watt with a junction temperature of 52°C and a maximum frequency of up to (230 MHz). **Conclusion:** The improved parallel MRI filtering algorithms which are developed as a single unified architecture provide visibility enhancement within the filtered MRI scan to aid the physician in detecting brain diseases, e.g., trauma or intracranial haemorrhage. The high filtering throughput is feasibly nominee the nine parallel MRI filtering algorithms for applications such as real-time MRI potential future applications. Future Work: a set of parallel 3-D fMRI filtering algorithms will be investigated to be developed and fast FPGA prototyped for future research project.

**Key words:** 2-D MIR filtering algorithms, FPGA implementation, parallel algorithms, Xilinx system generator, Virtex-6 FPGA, Trauma, intracranial haemorrhage

## INTRODUCTION

FPGAs are increasingly used in modern parallel Filtering algorithm applications such as medical imaging (Leeser *et al.*, 2005), Mapping DSP Algorithms (Maslennikow and Sergiyenko, 2006), image processing (Kiran, 2008), power consumption in portable image processing (Atabany, 2008), MPEG-4 motion estimation in mobile applications (Gao, 2003), satellite data processing (Nataraj *et al.*, 2009), new Mersenne Number Transform (Nibouche *et al.*, 2009), high speed wavelet-based image compression (Masoudnia *et al.*, 2005) and even the global communication link (Mak *et al.*, 2008). Most of the above FPGA-based solutions are typically programmed with hardware description languages (HDL) inherited from ASIC (Chang, 2005) and microprocessor-based DSP design methodologies (Aziz, 2004; Alshibami, 2001).

On the other hand, parallel multidimensional filtering algorithms (Boussakta, 1999; Wing-Kuen Ling, 2002), to be efficiently implemented, demand high computational performance per Watt at maximum sampling frequency (Hasan *et al.*, 2010). Consequently, this study proposes system-level implementation of parallel reconfigurable architectures for nine different 2-D MRI digital filtering algorithms: Edge, Sobel X, Sobel Y, Sobel X-Y, Blur, Smooth, Sharpen, Gaussian and Beta (HYB).

The 2-D image filtering purpose of the above nine per-processing algorithms is detecting sharp changes in image brightness by significantly reducing the amount of data to be processed, filtering out information that may be regarded as less relevant, while preserving the important structural properties of an image. Thus each of these nine algorithms is one of the fundamental steps in

**Corresponding Author:** Sami Hasan, School of Electrical, Electronic and Computer Engineering, Newcastle University, Newcastle Upon Tyne, NE1 7RU, England, UK

image processing, image analysis, image pattern recognition and computer imaging techniques.

The nine different MRI filtering algorithms are efficiently developed, implemented and, then, improved in a unified architecture using Xilinx system generator tool (Xilinx, 2010) within the ISE 12.3 development suite to target two Virtex-6 FPGA (Virtex-6, 2010) boards, namely, xc6vlX240T1-1lff1759 and xc6vlX130T1-1lff1156.

The unified architecture is an open reconfigurable parallel circuit that can be used for, other than the above mentioned nine algorithms, any parallel 2-D filtering algorithms with convolutional filtering structure.

The study is organized in the following layout of sections: after the introduction, parallel 2-D image filtering algorithms for their functional parallel structure, the nine parallel 2-D MRI algorithms capture for the FPGA-based implementation, discussing results and, then, conclusions before the references.

**Parallel 2-D image filtering algorithms:** Parallel 2-D MRI filtering algorithms are a 5x5 convolution kernel mask based image processing algorithms. Generally, the parallel architecture of these algorithms is constructed of serial to parallel input stage, 2-D convolution filtering vector for processing and a parallel to serial reconstructed output stage, as shown in Fig. 1.

**1 Input 2-D Segmentation MRI Stage:** The serial to parallel input segmentation stage can be achieved by two steps. First step is reshaping. Second step is segmentation and buffering samples.

First step; the 2-D MRI matrix  $x(n_1, n_2)$  of size ( $N \times N$ ) is behaviorally reshaped, within the input stage, from (row  $\times$  column) matrix to be (time stamp  $\times$  MRI samples) Matrix format. The reshaped MRI matrix has a time stamp in the first column and a vector containing the corresponding MRI samples stream in the subsequent column,  $x(t, p)$ , as in (1) Eq. 1:

$$x(n_1, n_2) = x(t, p) \quad (1)$$

Where;  $t = 0, 1 \dots n_1 \times n_2 - 1$  and  $p = 1, 2 \dots n_1 \times n_2$ . Since the System Generator is a time based DSP development tool thus the time stamp variable,  $t$  in (1), is implicitly considered by the parallel MRI filtering algorithm. Hence (1) is simplified to Eq. 2:

$$x(n_1, n_2) = x_{n_1, n_2}(p) = x(p) \quad (2)$$

Second step; the 2-D MRI samples stream, in (2), are equally split to five samples sub-segments, as formulated in Eq. 3:

$$x(p) = [x_j(p)], j=1, 2, \dots, 5 \quad (3)$$

**Parallel 2-D convolution filtering stage:** The parallel 2-D filtering algorithm is processing the MRI pixel streams using convolution filters vector as shown in Fig. 1. Each convolution filter is a 5-tap MAC FIR filter. The filter architecture, as shown in Fig. 2, consists of an image sample stream buffer, filter coefficient memory, comparator, address control unit, MAC unit and capture register.

The image sample stream buffer and the filter coefficient memory store  $N$  MRI stream sub-segments and  $M$  coefficients respectively. The comparator generates the 'reset' pulse and 'enable' pulses for the accumulator and capture register respectively. The pulse is asserted when the address is zero and is delayed to account for pipeline stages. The address control unit provides the necessary address logic for the filter coefficient memory and the image sample stream buffer, in addition to the timing control for the comparator.

The MAC unit is pipelined to sum up an inner Fig. 2. The Convolution Filter algorithm product of a set of  $M$  coefficients by  $N$  respective MRI samples sub-sequence to form an individual result. Each MAC FIR is characterized by its 1-D kernel,  $\beta(m_1)$  of size ( $M$ ), to convolve MRI samples sub-sequences,  $x_j(p/5)$ , of length  $N$ . This 1-D convolution filter produces filtered MRI samples sub-segment,  $y_j(p/5)$ . Thus Eq. 4:

$$y_{n_1}(\frac{p}{5}) = \sum_{m_1=0}^{N-1} \beta(n_1 - m_1) x_{m_1}(\frac{p}{5}) \quad (4)$$

Where,  $n_1 = 0, 1, \dots, N+M-1$

As shown in Fig. 1, five parallel MAC FIR filters, of (4), constitute a 5x5 filter which is characterized by its 2-D convolution kernel,  $\beta(m_1, m_2)$  of size ( $M \times M$ ). This 5x5 filter convolves five MRI samples sub-sequences,  $x_j(p/5)$ , of length  $N \times N$  to produce a 2-D matrix filtered MRI samples sub-segment,  $y_j(p)$ . Then (4) becomes Eq. 5 and 6:

$$y_{n_1, n_2}(p) = \sum_{m_1=0}^{N-1} \sum_{m_2=0}^{N-1} \beta(n_1 - m_1, n_2 - m_2) x_{m_1, m_2}(\frac{p}{5}) \quad (5)$$

where,  $n_1 = n_2 = 0, 1, \dots, N+M-1$ .

**Output 2-D MRI reconstruction stage:** The final output 2-D MRI reconstruction stage is a parallel to serial conversion by summing up, pipelining and reshaping the filtered MRI samples sub-segments stream into the filtered 2-D MRI scan. Since  $x_{m_1, m_2}(p)$  and  $y_{n_1, n_2}(p)$  are to be a 2-D reshaped matrix for the MRI input,  $x(n_1, n_2)$  and a 2-D filtered MRI output,  $y(n_1, n_2)$ , as shown in Fig. 1, within the input stage and the output stage respectively. Thus, (5) can be re-expressed as:

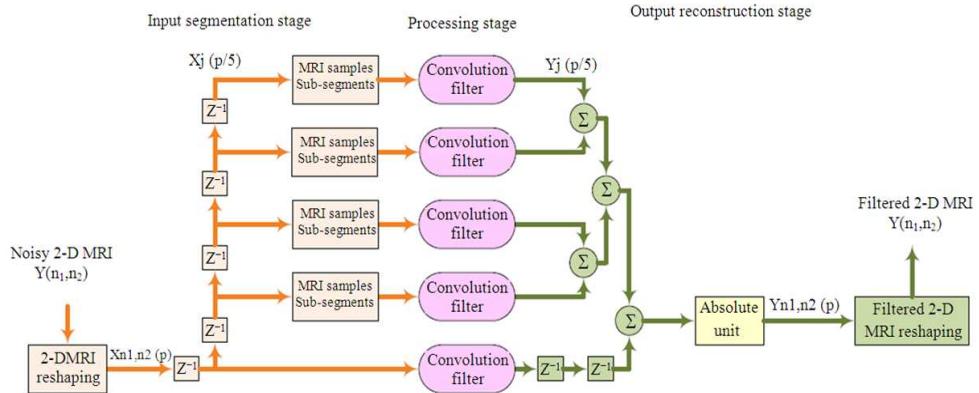


Fig. 1: A generalize parallel 2-D MRI filtering algorithms

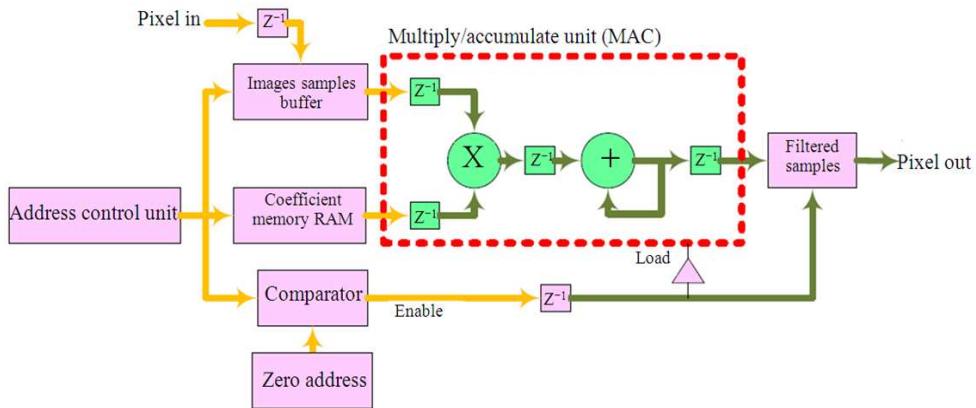


Fig. 2: The Convolution Filter algorithm

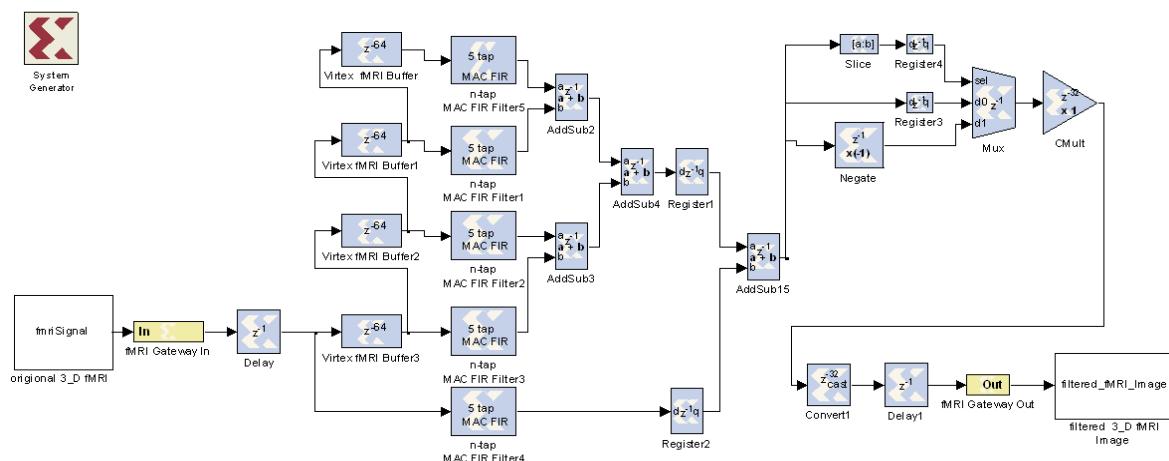


Fig. 3: Architecture 1: as one of the low-level abstracted implementation for the nine parallel 2-D MRI filtering algorithms

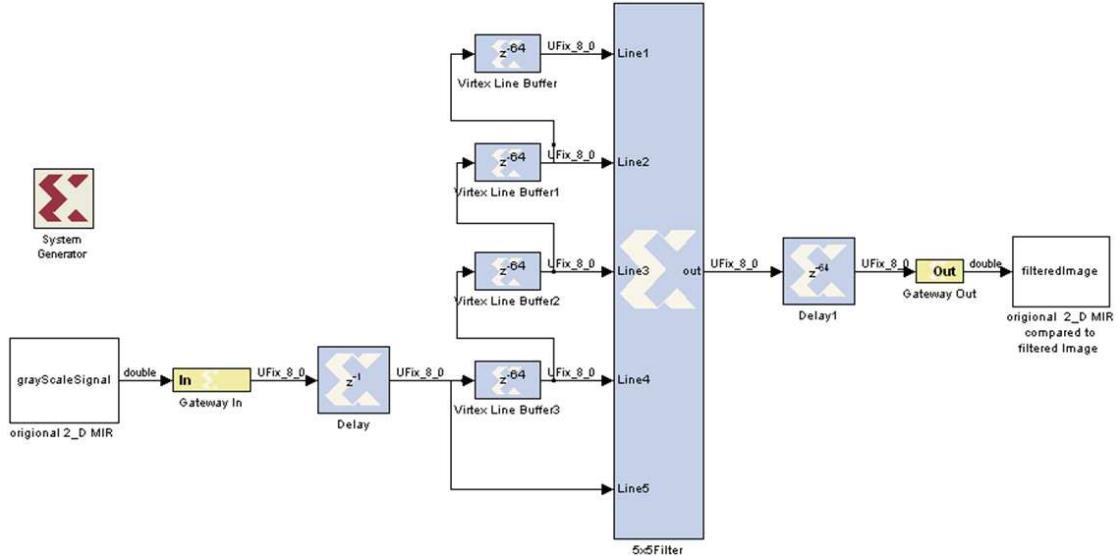


Fig. 4: Architecture 2: as a high-level abstracted implementation for the parallel 2-D MRI filtering algorithms

$$y(n_1, n_2) = \sum_{m_1=0}^{N-1} \sum_{m_2=0}^{N-1} x(m_1, m_2) \beta(n_1 - m_1, n_2 - m_2) \quad (6)$$

Where,  $0 \leq n_1, n_2 < N+M-1$ .

The next challenging goal is efficiently prototyping the nine parallel 2-D filtering algorithms into a single FPGA-base architecture.

**Parallel 2-D MRI algorithms capture:** Xilinx System Generator is utilized to develop an efficient FPGA-based architecture for the nine parallel 2-D MRI filtering algorithms with minimal idle operations. The clock signals and its corresponding enable logic do not appear in the architecture's circuit. These signals are internally generated when the FPGA implementation is behaviourally compiled within Xilinx/Simulink environment.

Consequently, these nine different parallel 2-D MRI image filtering algorithms can be behaviorally captured by more than one performance efficient architecture, depending on the abstraction level of implementation. Two of these circuits are shown in Fig. 3 and 4 as architecture 1 and architecture 2 respectively.

Both architectures consist of three stages; MRI input, processing and output. In the first stage, the magnetic resonance imaging (MRI) pixels are sequentially streamed into four virtex line buffers via a pipelined gateway block. Each line is delayed by 64 samples and the fifth line is a copy of the MRI scan.

The second stage is a parallel five 5-tap MAC FIR filters pipeline-balanced structure, as in the circuit of Fig. 3. Alternatively, the 5x5 convolution operations can be performed via the 5x5 filter block, as in the circuit of Fig. 4. Hence, both processing stages are to filter any noisy 2-D image and as a special case; the 64x64 grayscale MRI scan. Then the computed 5x5 convolution operators are summed up the results by four adder blocks. The absolute value of the FIR filters is computed and the data is narrowed to 8 bits.

## RESULTS AND DISCUSSION

One of the challenging goals of this study is developing an efficient FPGA implementation that provides fast FPGA prototyping for high filtering performance of the nine parallel 2-D MRI filtering algorithms. A time analysis compilation tool is needed to evaluate the area/speed/power consumption performance indices. Thus the Xilinx Timing Analyzer is utilized to generate time statistics, total power analysis and histogram charts of FPGA implementation paths delay. This provides guides to clarify the bottleneck in the implementation and focus on the optimization of the slow paths outliers.

The results presented into three forms: performance index table as in Table 1, grayscale MRI filtered images with their corresponding kernels as in Table 2and Table 3, Logic assets utilization as in Table 4 then Histogram Charts of path delay distribution as in Fig. 5-8.

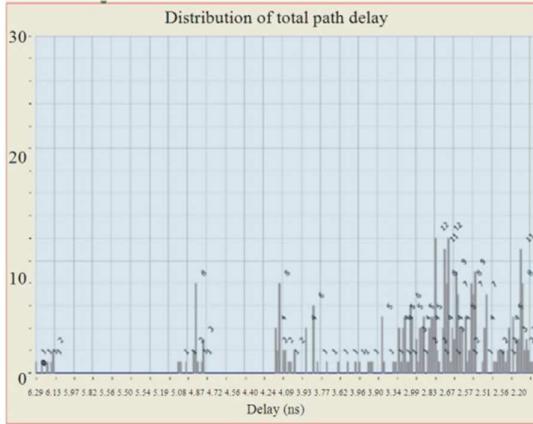


Fig. 5: Chart depicts the total paths delay distribution of the MRI Edge filter captured behaviorally via (X240T) FPGA board

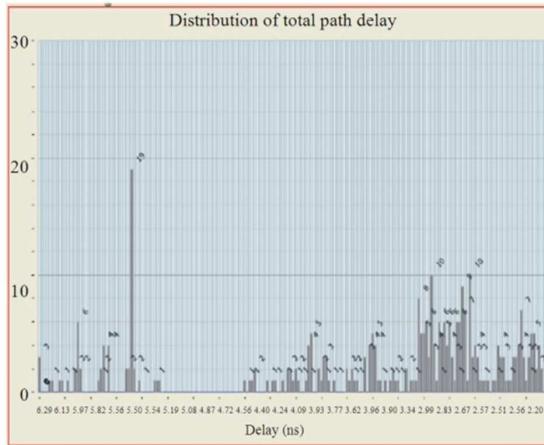


Fig. 6: Histogram Chart depicts the total paths delay distribution of the MRI Edge filter captured behaviorally via (X130T) FPGA board

The performance efficient implementation results can be behaviorally achieved by low power consumption at maximum frequency for the nine parallel 2-D MRI image filtering algorithms. Consequently, comparative results of two Virtex-6 FPGA boards, xc6vlX240Tl-1lff1759 and xc6vlX130Tl-1lff1156 are compiled for the nine 2-D filters by two sets of 5x5 coefficient mask. The first set is the generic 5x5 kernels. And the second set is the improved 5x5 kernels to a new 5x5 Enhancement Orthogonal Kernels.

**Power:** The total power consumption for architecture 2 has two elements: the static power and the dynamic power (Yakovlev, 2011).

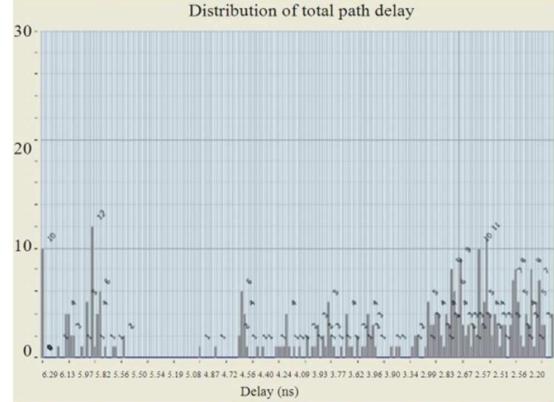


Fig. 7: Histogram Chart depicts the total path delay distribution of the improved Edge filter captured behaviorally via (X240T) FPGA

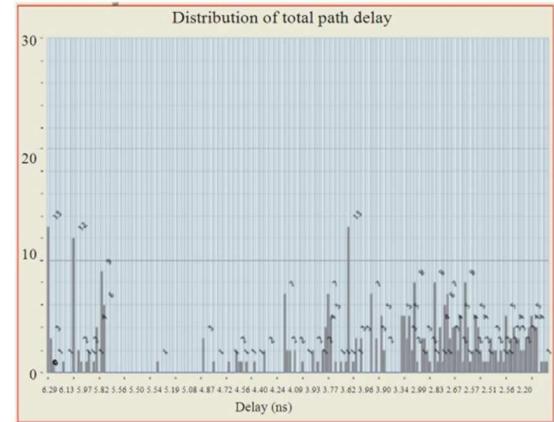


Fig. 8: Histogram Chart depicts the path delays distribution of the improved Edge filter captured behaviorally via (X130T) FPGA

Table 1: Performance indices

| 2-D MRI Filtering Algorithms | Power Consumption (Watt) |       | Maximum Frequency (MHz) |       |
|------------------------------|--------------------------|-------|-------------------------|-------|
|                              | X240T                    | X130T | X240T                   | X130T |
| Edge                         | 1.38                     | 0.86  | 194                     | 230   |
| SobelX                       | 1.38                     | 0.86  | 213                     | 225   |
| SobelY                       | 1.38                     | 0.86  | 214                     | 230   |
| SobelXY                      | 1.38                     | 0.86  | 213                     | 225   |
| Blur                         | 1.38                     | 0.86  | 213                     | 230   |
| Smooth                       | 1.38                     | 0.86  | 211                     | 217   |
| Sharpen                      | 1.38                     | 0.86  | 230                     | 230   |
| Gaussian                     | 1.38                     | 0.86  | 227                     | 230   |
| Beta(HYB)                    | 1.38                     | 0.86  | 211                     | 230   |

Table 1 shows the performance indices of power consumption (Watt) and the corresponding maximum operating frequency (MHz) for the developed nine parallel 2-D MRI filtering algorithms.

Table 2: The generic parallel MRI filtering algorithms

| 2-D MRI<br>filtering algorithms | Generic<br>5x5 Kernel   | Corresponding filtered MRI using |       |
|---------------------------------|---|----------------------------------|-------|
|                                 |   | X240T                            | X130T |
| Edge                            | $\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & -1 & 0 \\ 0 & -1 & 8 & -1 & 0 \\ 0 & -1 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ |                                  |       |
| SobelX                          | $\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & -2 & 0 & 2 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$      |                                  |       |
| SobelY                          | $\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$      |                                  |       |
| SobelXY                         | $\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$        |                                  |       |
| Blur; DF = $(\frac{1}{16})$     | $\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$         |                                  |       |
| Smooth; DF = $(\frac{1}{100})$  | $\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 5 & 5 & 5 & 1 \\ 1 & 5 & 44 & 1 & 1 \\ 1 & 5 & 5 & 5 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$        |                                  |       |
| Sharpen; DF = $(\frac{1}{-16})$ | $\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 2 & 2 & 0 \\ 0 & 2 & -32 & 2 & 0 \\ 0 & 2 & 2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$       |                                  |       |
| Gaussian DF = $(\frac{1}{52})$  | $\begin{bmatrix} 1 & 1 & 2 & 1 & 1 \\ 1 & 2 & 4 & 2 & 1 \\ 2 & 4 & 8 & 4 & 2 \\ 1 & 2 & 4 & 2 & 1 \\ 1 & 1 & 2 & 1 & 1 \end{bmatrix}$         |                                  |       |
| Identity                        | $\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$         |                                  |       |

The performance indices of Table 1 show that the X130T FPGA implementation outperforms X240T FPGA according to its minimum total power consumption (around 0.86 at junction temperature = 52 C°) and maximum frequency (mostly around 230 MHZ). Table 1 is fairly remained unchanged after improving the nine 5x5 kernels.

**Filtering:** The filtered 2-D MRI images of Table 2 and Table 3 are generated from the two 5x5 kernels sets, the generic and the improved, respectively, of the nine parallel algorithms implementation using Virtex-6

X240T and X130T FPGAs. By inspection, the filtered MRI scans of Table 3 are image enhanced compared to those of Table 2 without affecting the developed performance indices of lower power consumption at maximum operating frequency. In both tables, the D.F is stand for Division Factor of the 5x5 kernel.

Furthermore, the generic 5x5 mask-based convolution kernels,  $\beta(m_1, m_2)$ , for the nine filtering algorithms: Edge, Sobel X, Sobel Y, Sobel X-Y, Blur, Smooth, Sharpen, Gaussian and identity are all showing the filtering portability, whether, using X130T FPGA or X240T.

Table 3: The improved parallel filtering algorithms

| 2-D MRI<br>filtering algorithms | Developed<br>5x5 Kernel  | Corresponding filtered MRI |             |
|---------------------------------|--|----------------------------|-------------|
|                                 |  | Using X240T                | Using X130T |
| Edge D.F = $(\frac{1}{-8})$     | $\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & -16 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$  |                            |             |
| SobelX D.F = $(\frac{1}{-8})$   | $\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 1 & 32 & -1 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$                                      |                            |             |
| SobelY D.F = $(\frac{1}{-8})$   | $\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & -1 & -10 \\ 0 & 0 & 32 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$                                    |                            |             |
| SobelXY D.F = $(\frac{1}{-8})$  | $\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & -1 & 32 & 1 & 0 \\ 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$                                      |                            |             |
| Blur; DF = $(\frac{1}{-16})$    | $\begin{bmatrix} -1 & 4 & -14 & -1 \\ -1 & 0 & 0 & -1 \\ 4 & 4 & 4 & 4 \\ -1 & 0 & 0 & -1 \\ -14 & -1 & 4 & -1 \end{bmatrix}$  |                            |             |
| Smooth; DF = $(\frac{1}{100})$  | $\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & -5 & 120 & -5 & 1 \\ 1 & 120 & 480 & 120 & 1 \\ 1 & -5 & 120 & 5 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$                             |                            |             |
| Sharpen; DF = $(\frac{1}{16})$  | $\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & -1 & 0 \\ 0 & 1 & 64 & -1 & 0 \\ 0 & -1 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$                                  |                            |             |
| Gaussian DF = $(\frac{1}{-52})$ | $\begin{bmatrix} 1 & 1 & 2 & 1 & 1 \\ 1 & 2 & -20 & 2 & 1 \\ 2 & 20 & -80 & -20 & 2 \\ 1 & 2 & -20 & 2 & 1 \\ 1 & 1 & 2 & 1 & 1 \end{bmatrix}$                                 |                            |             |
| Beta (HYB)                      | $\begin{bmatrix} 0.2 & 0.4 & 1 & 0.4 & 0.2 \\ 0.4 & 1 & - & 3.3 & 10.4 \\ 1 & 3.3 & -4.4 & 3.3 & 1 \\ 0.41 & -3.3 & -4.4 & 3.3 & 1 \\ 0.2 & 0.4 & 1 & 0.4 & 0.2 \end{bmatrix}$ |                            |             |

Table 4: Typical device utilization summary

| Logic utilization | Used | Available | Utilization (%) |
|-------------------|------|-----------|-----------------|
| FFs               | 578  | 301,440   | 1               |
| LUTs              | 412  | 150,720   | 1               |
| Slices            | 172  | 37,680    | 1               |
| IOBs              | 17   | 720       | 2               |
| TBUFs             | 1    | 32        | 3               |
| DSP48E1s          | 5    | 768       | 1               |

The same observation is applicable for their corresponding improved parallel filtering algorithms.

The ninth improved algorithm is renamed as “Beta (HYB)” which is the authors’ initials.

**Area:** The FPGA-based architecture 2 of Fig. 4 is occupying the proper resources of logic devices as in Table 4. This instantiation is compared to the available Logic assets as a utilization percentage. The efficient implementation hierarchy of Clock trees, Logic, signals, I/O’s and Hard IPs such as DSP blocks subsequently improves the performance indices of power consumption and operating frequency. The

device utilization of architecture 1 is occupying the same logic assets as that of architecture 2 of Fig. 3.

**Speed:** The histogram time charts, in Fig. 5 and 6 depict the slow paths distributions of the generic 2-D MRI Edge filter captured behaviorally via X240T and X130T FPGA board respectively. And, the histogram time charts, in Fig. 7 and 8 depict the slow paths distributions of the improved 2-D MRI Edge filter captured behaviorally via X240T and X130T FPGA board respectively. Each histogram chart is a useful metric to analyze the FPGA implementation. Where are the slowest paths concentrated? How many slow paths are in each bin? How efficient is the implementation to meet timing? Accordingly, the FPGA implementation can be adjusted. Each histogram's slow paths are grouped into regions of roughly formed normal distribution groups. The numbers at the top of the bins show the number of paths in each bin.

Figure 5 shows 308 paths that are roughly forming five groups. These groups are probably from different portions of the system generator architecture, as in Fig. 3, or from different timing clock region constraints. This shows that most of the slow paths are concentrated around (2.81 ns). The slowest path is about (6.15 ns). There are an outlier group of slow paths in the time range (6.13ns-6.30ns) with empty bins to the right of it. That is because the FPGA implementation frequency, from Table 1, is the slowest (194 MHz) for this 2-D MRI Edge filter. However, there are no red/ pink bins or portions that do not meet the timing constraints.

Figure 6 shows a shorter histogram chart of 308 paths that form totally different distributed histogram with roughly only three normally distributed paths groups between (2.2 ns) and (4.36 ns). That is because the FPGA implementation frequency, from Table 1, is the highest (230 MHz) for the same 2-D MRI Edge filter.

The slow paths are concentrated between (2.2ns) and (2.8ns). The slowest path is about (4.2ns). Moreover, the greater number of only one path per bin, distributed throughout the nanosecond domain demonstrate the highly outperformance efficient implementation of (230 MHz) maximum frequency. Consequently, there are no red/pink bins or portions that do not meet the timing constraints.

The histogram charts, in Fig. 7 and 8 are displaying the reflections of the new maximum sampling frequencies over the slow paths concentration for the improved Edge filter FPGA implementation of X240T and X130T respectively.

Figure 7 chart shows a shorted histogram compared to that of Fig. 8, because of the new maximum frequency (229 MHz). This chart depicts 308 paths

grouped roughly into four bell curve regions. Most of the slow paths are concentrated around (2.4 ns). The slowest path is about (4 ns). Consequently, the outlier groups of the slowest paths are shifted to the time range of 3.88ns-4.20ns with empty bins to the right of it. There are no red/ pink bins or portions that do not meet the timing constraints.

Figure 8 histogram is distributed 308 slow paths to roughly form three bell shape distribution between (2 ns) and (4.2 ns). The slowest path is about (4.09 ns). There are less one path bins compared to those of Fig. 7. There are no red/pink bins or portions that do not meet the timing constraints.

**Throughput:** One of the FPGA-based architecture's efficient performance indices is the filtering frame rate, i.e. architecture throughput. Since the architecture is operating at (230 MHz) and each of the five 5-tap MAC FIR filters is clocked 5 times faster than the MRI streams input rate. Therefore, the architecture throughput (frames/second), as a filtering performance, is  $230\text{ MHz} / 5 = 46$  million MRI samples/second. For the  $64 \times 64$  greyscale MRI scan, the throughput is  $46 \times 10^6 / (64 \times 64) = 11230$  frames/second. If the filtered MRI is of  $256 \times 256$  scan then the throughput would be 701 frames/sec and for a  $512 \times 512$  scan it would be 172 frames/sec. Thus the architecture throughput is MRI scan size dependent.

**Performance Comparison:** The nine parallel 2-D MRI filtering algorithms architecture 1 and 2 have efficiently implemented utilizing hard IPs (DSPs) and minimal resources of logic devices. This is to achieve the highly filtered performance of (11230 frames/second) throughput per minimum power consumption of (0.86 Watt at 25 °C via X130T) and up to (1.138 Watt at 75 °C via X240T) at a maximum operating frequency of up to (230 MHZ).

Moreo *et al* (2005) filtered  $256 \times 256$  grayscale image using  $3 \times 3$  convolution filter and  $5 \times 5$  convolution filter to only implement the generic smooth filtering algorithm and the generic sharp filtering algorithm respectively, without mentioning their power consumption. The device selected for the above mentioned existing work is Xilinx Virtex, XCV800 HQ240, speed-6. Table 5 shows the comparative results for area, speed and power.

Moreo *et al.* (2005), the proposed algorithm was prototyped using only the logic devices resources without using any IP cores of DSPs. which produce higher logic utilization percentage and reduces the maximum operating frequency to (69 MHz).

Table 5: Comparative results of area, speed and power

| Logic utilization             | Conv. 3x3 (%) | Conv. 5x5x (%) | Architecture 2(%) |
|-------------------------------|---------------|----------------|-------------------|
| FFs                           | 2.0           | 4.0            | 1.0               |
| LUTs                          | 2.0           | 4.0            | 1.0               |
| Slices                        | 3.0           | 6.0            | 1.0               |
| IOBs                          | 9.0           | 9.0            | 2.0               |
| DSP48E1s                      | NA            | NA             | 1.0               |
| Maximum operating speed (MHz) | 76            | 69             | 230               |
| Power Consumed (Watt)         | NA            | NA             | 0.86              |

## CONCLUSION

This study presented a generalized 2-D MRI filtering algorithm and, then prototyped them in a single FPGA-based architecture using Xilinx System Generator. Two architectures are prototyped, depending on the abstraction level of implementation. This fast FPGA prototyping provides high filtering throughput performance of (11230 frames/second) per minimum total power consumption down to (0.86 Watt) at a maximum sampling frequency of up to (230 MHz).

## REFERENCES

- Alshibami, O., S. Boussakta and M. Aziz, 2001. Fast algorithm for the 2-D new Mersenne number Transform. *Signal Process.*, 81: 1725-1735. DOI: 10.1016/S0165-1684(01)00068-8
- Atabany, W. and P. Degenaar, 2008. Parallelism to reduce power consumption on FPGA Spatiotemporal image processing. Proceedings of the IEEE International Symposium on Circuits and Systems, May 18-21, IEEE Xplore Press, Seattle, pp: 1476-1479. DOI: 10.1109/ISCAS.2008.4541708
- Aziz, M., 2004. Parallel Digital Filtering Algorithms for Multiprocessor DSP systems. A PhD Thesis, University Of Leeds.
- Boussakta, S., 1999. A novel method for parallel image processing applications. *J. Syst. Architecture*, 45: 825-839. DOI: 10.1016/S1383-7621(98)00041-1
- Chang, C., 2005. Design and Applications of a Reconfigurable Computing System for High Performance Digital Signal Processing. Ph.D. Thesis, University of California, Berkeley, pp: 368.
- Leeser, M., S. Coric, E. Miller, H. Yu and M. Trepanier, 2005. Parallel-Beam backprojection: An FPGA implementation optimized for medical imaging. *J. VLSI Signal Process. Syst. Signal, Image, Video Technol.*, 39: 295-311. DOI: 10.1007/s11265-005-4846-5
- Gao, R., D. Xu and J.P. Bentley, 2003. Reconfigurable Hardware Implementation of an Improved Parallel Architecture for MPEG-4 Motion Estimation in Mobile Applications. *IEEE Trans. Consumer Elect.*, 49: 1383-1390. DOI: 10.1109/TCE.2003.1261244
- Hasan, S., A. Yakovlev and S. Boussakta, 2010. Performance efficient FPGA implementation of parallel 2-D MRI image filtering algorithms using Xilinx system generator. Proceedings of the 7th International Symposium on Communication Systems Networks and Digital Signal Processing, Jul. 21-23, IEEE Xplore Press, Newcastle Upon Tyne, pp: 765-769.
- Kiran, M., K.M. War, L.M. Kuan, L.K. Meng and L.W. Kin, 2008. Implementing image processing algorithms using 'Hardware in the loop' approach for Xilinx FPGA. Proceedings of the International Conference on Electronic Design, Dec. 1-3, IEEE Xplore Press, Penang, pp: 1-6. DOI: 10.1109/ICED.2008.4786653
- Mak, T., C. D'Alessandro, P. Sedcole, P.Y.K. Cheung and A. Yakovlev *et al.*, 2008. Implementation of wave-pipelined interconnects in FPGAs. Proceedings of the 2nd IEEE Intern. Symposium on NOCS, April 7-10, IEEE Xplore Press, Newcastle Upon Tyne, pp: 213-214. DOI: 10.1109/NOCS.2008.4492743
- Maslenikow, O. and A. Sergiyenko, 2006. Mapping DSP Algorithms into FPGA. Proceedings of the International Symposium on Parallel Computing in Electrical Engineering, Sept. 13-17, IEEE Xplore Press, Bialystok, pp: 208-213. DOI: 10.1109/PARELEC.2006.51
- Masoudnia, A., H. Sarbazi-Azad and S. Boussakta, 2005. Design and performance of a pixel-level pipelined-parallel architecture for high speed wavelet-based image compression. *Comput. Elect. Eng.*, 31: 572-588. DOI: 10.1016/j.compeleceng.2005.07.005
- Moreo, A.T., P.N. Lorente, F.S Valles, J.S. Muro and C.F. Andres, 2005. Experiences on developing computer vision hardware algorithms using Xilinx system generator. *Microprocessors Microsystems*, 29: 411-419. DOI: 10.1016/j.micpro.2004.11.002
- Nataraj, K.R., S. Ramachandran and B.S. Nagabushan, 2009. Design of architecture for sampling rate converter of demodulator. Proceedings of the 2nd International conf. on Computer and Electrical Engineering, Dec. 28-30, IEEE Xplore Press, Dubai, pp: 427-430. DOI: 10.1109/ICCEE.2009.262
- Nibouche, O., S. Boussakta and M. Darnell, 2009. Pipeline architectures for radix-2 new Mersenne number transform. *IEEE Transactions on Circuits and Systems I: Regular Papers* 56: 1668-1680. DOI: 10.1109/TCSI.2008.2008266

Virtex-6 FPGA Xilinx documentation 2010, from:  
Wing-Kuen Ling, B. and P. Kwong-Shun Tam, 2002.  
Edge detection via fuzzy switch. SPIE's Intern.  
Technical Group Newsletter, 12: 2.  
Xilinx System Generator for DSP user guides, 2010,

Yakovlev, A., 2011. Energy-Modulated Computing.  
Proceedings of the Design, Automation and Test in  
Europe Conference and Exhibition (DATE), March  
14-18, IEEE Xplore Press, Grenoble, pp: 1-6.