RESEARCH ARTICLE OPEN ACCESS

An Efficient Implementation of Median Filter Using Matlab Simulink and Vhdl

Ms. K. R. Katole[1], Mrs. S.P.Balwir[2], Ms. N. W.Umredkar[3]
[1,2,3] Asst.Prof. Department of Electronics Engineering, Dr. Babasaheb Ambedkar College of Engineering & research, wanadongari, Nagpur
kkmanekar8@gmail.com , swatibalwir8@gmail.com

ABSTRACT
This paper presents an efficient image filtering algorithm for median filter and Characterization using Xilinx System Generator (XSG). This architecture offers an alternative through a graphical user interface that combines MATLAB, Simulink and XSG and explores important aspects concerned to hardware implementation. Performance of this architecture implemented on Virtex5 kit

KEYWORDS: *Median Filter, MATLAB Simulink, HDL Coder, FPGA.*

I. INTRODUCTION

Median filter is one of the image filtering technique used for handling of digital images in different areas such as medical, technological applications and many others. This non- linear technique has proven to be a good alternative to linear filtering as it can effectively suppress impulse noise while preserving edge information. This paper focuses on processing an image pixel by pixel and in modification of pixel neighborhoods and the transformation that can be applied to the whole image or only a partial region MATLAB and Simulink for Model-Based Design provide signal, image, and video processing applications with a development platform that spans design, modeling, simulation, code generation, and implementation. A Model-Based Design to target FPGA can design and simulate systems with MATLAB, Simulink and then generate bit-true, cycle-accurate, synthesizable VHDL code using HDL Coder.

This paper presents an architecture of filtering images for noise removal using MATLAB Simulink and The work is organized as follows: in section II the median filter algorithm is described, section III deals with the design of median filter with MATLAB Simulink , in section IV the generated HDL code of filter is simulated in Xilinx and dumped in FPGA, and finally in section V the result and conclusions are presented.

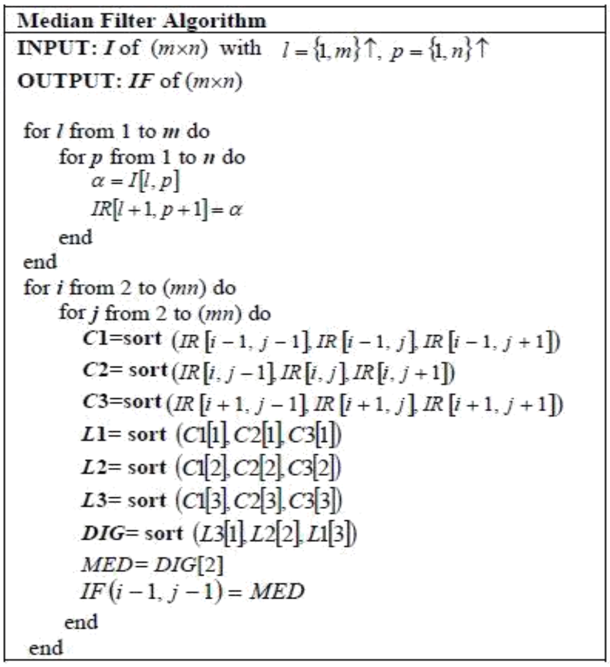
II. MEDIAN FILTER ALGORITHM

Defination:- A Median is defined as the middle element of a group of numbers when the numbers are sorted .The group contains odd number of elements.

The algorithm has been taken form [2].Let us consider a grayscale image *I*, divided in (*m*× *n*) pixels (squares) and affected by impulsive noise. In order to achieve the filtered image *IF* which is obtained though the median filtering operation, the value of each output pixel *IF*(*x*,*y*) must be computed by using iteratively a set of 9 pixels from *I*

These pixels are inside of a 3×3 mask with center in *I*(*x*,*y*). The position of this mask is shifted recursively along to the entire image until the median filtering process is completed. From a purely numerical point of view, the image *I* is treated as a matrix array. In this sense, because of the mask operates over the pixel neighbor then it is required to add elements (zeros) around the matrix *I*, it gives as result the(*m* + 2) × (*n* + 2) matrix *IR*. This procedure is summarized in the algorithm described as follows:

HDL coder.



```
Median Filter Algorithm
INPUT: I of $(m \times n)$ with $l = \{1, m\}\uparrow,\ p = \{1, n\}\uparrow$
OUTPUT: IF of $(m \times n)$

for $l$ from 1 to $m$ do
    for $p$ from 1 to $n$ do
        $\alpha = I[l, p]$
        $IR[l+1, p+1] = \alpha$
    end
end
for $i$ from 2 to $(mn)$ do
    for $j$ from 2 to $(mn)$ do
        C1=sort $(IR[i-1, j-1], IR[i-1, j], IR[i-1, j+1])$
        C2= sort $(IR[i, j-1], IR[i, j], IR[i, j+1])$
        C3=sort $(IR[i+1, j-1], IR[i+1, j], IR[i+1, j+1])$
        L1= sort $(C1[1], C2[1], C3[1])$
        L2= sort $(C1[2], C2[2], C3[2])$
        L3= sort $(C1[3], C2[3], C3[3])$
        DIG= sort $(L3[1], L2[2], L1[3])$
        $MED = DIG[2]$
        $IF(i-1, j-1) = MED$
    end
end
```

III. PROPOSED DESIGN

The steps for design and implementation of median filter is shown in the flow diagram.

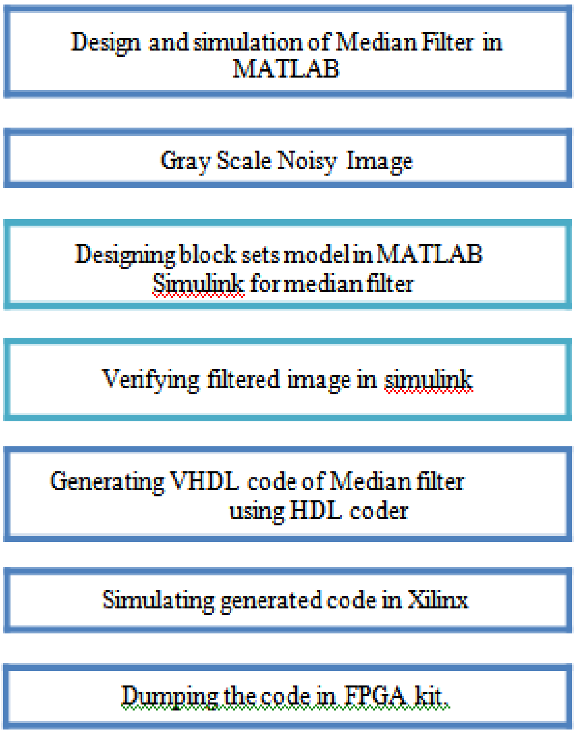

Design and simulation of Median Filter in MATLAB
Gray Scale Noisy Image
Designing block sets model in MATLAB Simulink for median filter
Verifying filtered image in simulink
Generating VHDL code of Median filter using HDL coder
Simulating generated code in Xilinx
Dumping the code in FPGA kit.

Fig. 1: Flow diagram for design and implementation of median filter in FPGA in HDL coder

The median filter is designed in MATLAB and the output image is observed.

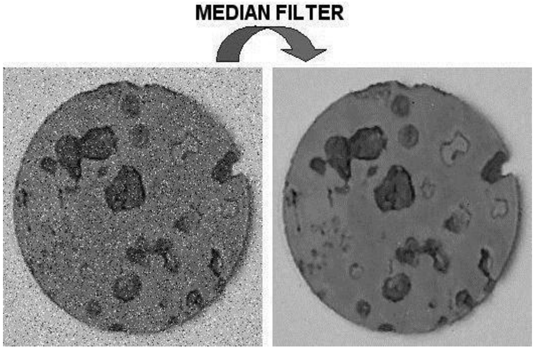


Fig. 2: Median filter results in Matlab

Then using MATLAB Simulink block sets a model for median filter is designed.

Here Input gray scale images are provided as input to the File block. Then this image is partitioned by column. It capture a column of data (9x1) of the image and feeds into the median filter block. Filter buffers the column for first 9 cycles to make a 9x9 region.

In this block min/max/median value computation for a 3x3 region requires 36 comparators (9 stages of 4 comparators each). Each computational element compares the neighboring pixels (simple Mux in h/w that rearranges input pixels). Adaptive Median filter is very resource intensive. To reduces error rate of pixel computation for 3x3 region we used the 1D median filter

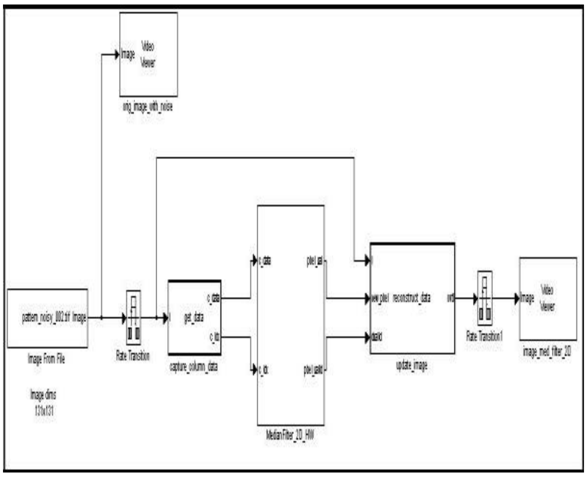


Fig. 3: Simulink model of an image filtering system

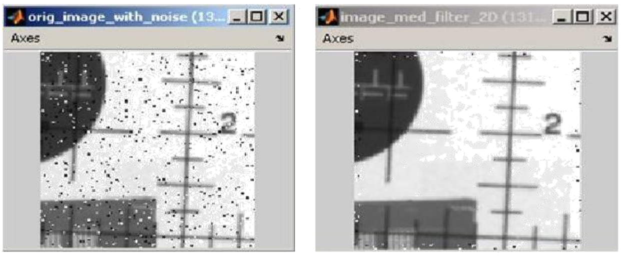


Fig. 4: Input output image in Matlab simulink

IV. Simulation And Implementation

After simulating the model of filter in MATLAB Simulink HDL code is generated using, Setting code generation option. This code is then simulated with Xilinx.

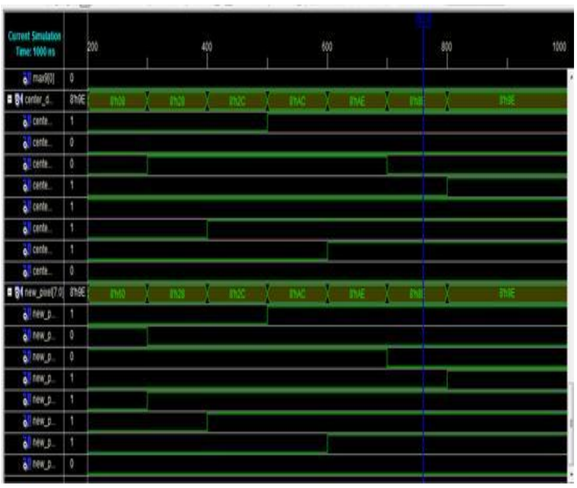


Fig. 5: Simulation in Xilinx simulator

After simulation in Xilinx, the code implemented on Spartan 6.Figure 5 shows the final output where three LEDs indicates min, max and median of the image.

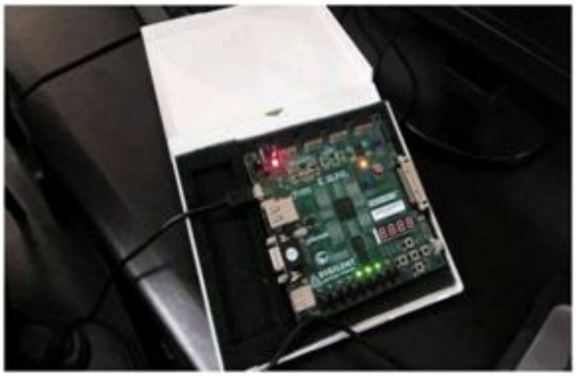


Fig. 6: FPGA implementation on Spartan 6.

V. CONCLUSIONS

The HDL coder tool in MATLAB Simulink is a new application in image processing, and offers a model based design for processing. The filters are designed by blocks here. This tool support software simulation, but most importantly it generates necessary files for implementation in all Xilinx FPGAs, with the parallelism, robust, speed and automatic area minimization. This design has various applications where speed and hardware constraints are important. These features could be extended to real time image processing.

References

[1] FPGA Implementation of Median Filter- RajuI Maheshwari, S. S.Rao,P. g. Poonacha , 10thInifernational Conference on VLSI Design - January 1997

[2] Digital Circuit Architecture for a Median Filter of Grayscale Images Based on Sorting Network, Victor Jimenez-Fernandez, Denisse Martinez-Navarrete, Carlos Ventura-Arizmendi, Zulma Hernandez-Paxtian, Joel Ramirez-Rodriguez, International Journal Of circuits, systems and signal processing, issue 3, volume 5, 2011

[3] Mrs. S. Allin christe,, Mr. M. Vignesh, Dr. A. Kanda swamy "An Efficient Fpga Implementation Of Mri Image Filtering And Tumour Characterization Using Xilinx System Generator" International Journal of VLSI design & Communication Systems (VLSICS) Vol.2, No.4, December 2011