A PROJECT REPORT ON

IMPLEMENTATION OF 7-TAP 1-D MEDIAN FILTER ON FPGA BOARD USING VERILOG HDL

UNDER THE SUPERVISION OF

Dr. KISHOR P. SARWADEKAR

SUBMITTED BY

JATAN GAUR (18092012)

MASTER OF TECHNOLOGY

IN

DIGITAL TECHNIQUES AND INSTRUMENTATION

DEPARTMENT OF ELECTRONICS ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY (BHU)

VARANASI - 221005

APRIL, 2019

Index

1. Index…….1

2. Introduction…….2

2.1 Median Filtering…….……..2

2.2 Real Examples……..3

3. Proposed Work……5

3.1 I/O Specifications……..……..6

3.2 Procedure……..7

4. Verilog Code……8

4.1 Med_Filt( )……....8

4.2 sort( )……...22

4.3 push_bttn( )…….24

4.4 d_ff( )……..24

4.5 Test Bench - Median_Filter_tb( )……..……25

4.6 Constraint File (.xdc)……..50

4.7 MATLAB script……..51

5. Results……52

6. References……...54

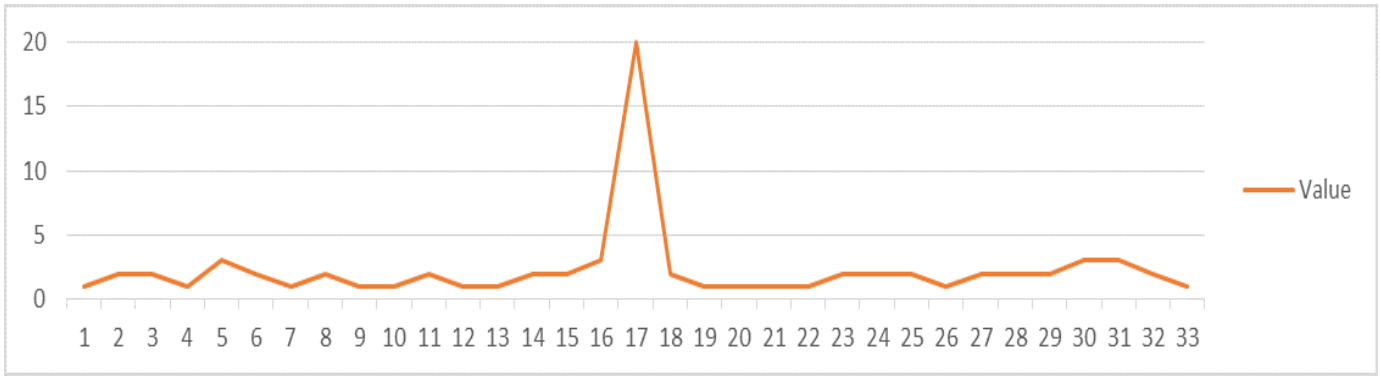
Introduction

Median filtering is a nonlinear process useful in reducing impulsive, or salt and-pepper noise. It is also useful in preserving edges in an image while reducing random noise. Impulsive or salt-and pepper noise can occur due to a random bit error in a communication channel. In a median filter, a window slides along the image, and the median intensity value of the pixels within the window becomes the output intensity of the pixel being processed. Like low pass filtering, median filtering smoothes the image and is thus useful in reducing noise. Unlike low pass filtering, median filtering can preserve discontinuities in a step function and can smooth a few pixels whose values differ significantly from their surroundings without affecting the other pixels.

Median Filter is better compared to Averaging Filter, which is also used in Image processing, in terms of better sensitivity towards edges and curves in images.

A 1-Dimensional array of data is shown below as example of salt and pepper noise, where the ‘Sample #’ indicates the indices of the data, and the Value row shows the value of the signal on that point.

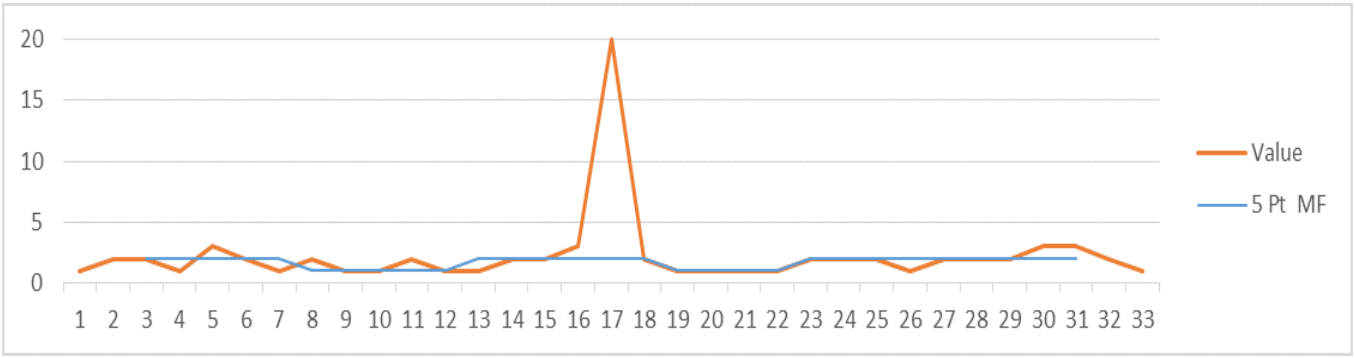
Sample #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
Value	1	2	2	1	3	2	1	2	1	1	2	1	1	2	2	3	20	2	1	1	1	1	2	2	2	1	2	2	2	3	3	2	1



[1]

To remove the salt and pepper noise present at index 17 in the above example, a 5 point median filtering is done and the results are as shown below,

Sample #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
5-point median filter		2	2	2	2	2	1	1	1	1	1	1	2	2	2	2	2	2	1	1	1	1	2	2	2	2	2	2	2	2	2		



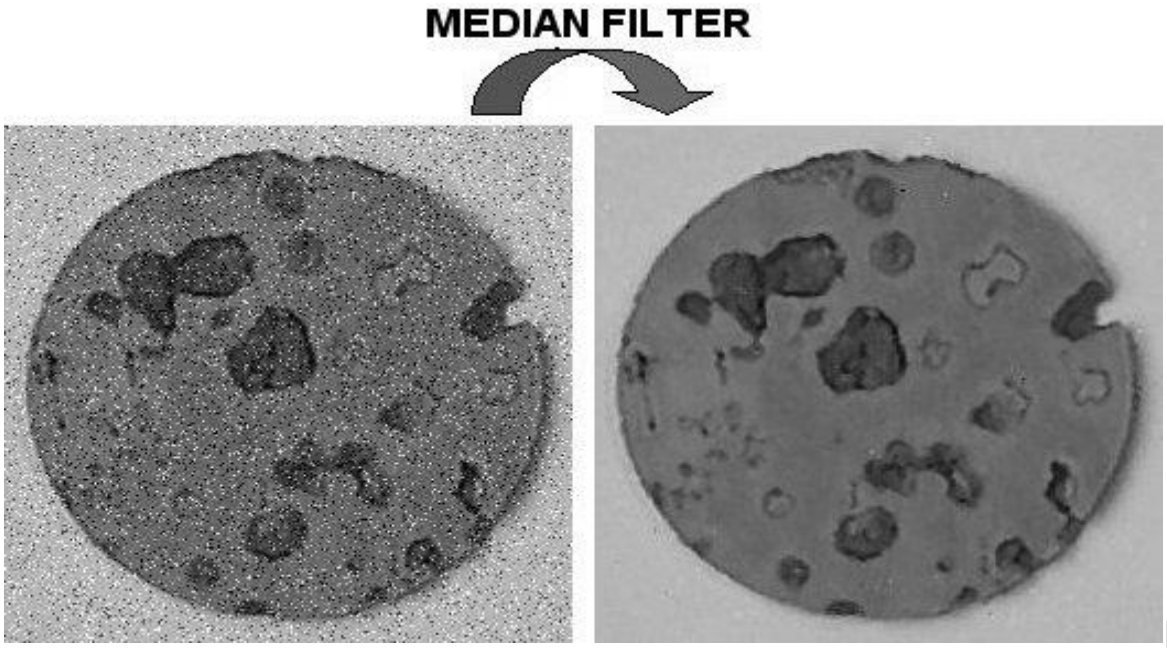
[1]

So the noise present at the index 17 is removed by median filter, by replacing the noisy data with the median of the 5 point sliding window of the median filter.

Examples are given below to show the importance and function of a median filter in an actual noisy image,


[2]

 [3]



[4]

Proposed Work

A 7-tap 1-Dimensional Median Filter is implemented here, which is having an array of 100 data elements, 8 bit each. 1-Dimensional Median filter deals with array of data that is 1-D. This Median Filter has a sliding window of size 7, i.e. it takes 7 values at a time to process, hence the name 7-tap filter.

The proposed median filter takes a group of 7 data elements, sorts them in ascending order. The sorting algorithm used is Bubble Sort. After sorting, the mid value of the sorted array i.e. the median is taken and the mid value of the group of data (7 elements) is replaced by the median. For example, suppose value of memory 'mem' is taken from mem[1] to mem[7], and sorted, and the median value i.e. mem[4] is replaced in the memory with the new median value. Like this, the window keeps incrementing, i.e. from 1 to 7, 2 to 8, 3 to 9 etcetera, up to the final value, i.e. 94 to 100.

Random data is generated for testing purpose through a MATLAB script, and the data is copied to the ROM array every time there is a need to have new data.

This is the mechanism that is being followed to calculate the median, and update the data, and get the filtered data as output. To display output on the FPGA board (BASYS 3 used), two push buttons are taken, the functionality of which is provided in the module named push_bttn(). The Switch De-Bounce logic is used for the noise filtering of push button, such that the push button, when pressed, only generates one pulse, irrespective of the duration of pressing of button. This pulse from the output of push button is then used to drive other logic blocks.

Pressing the first push button displays the raw data on the seven segment display, and pressing the second push button displays the filtered data, one by one each.

Hardware Specification

FPGA Board used for the project is BASYS 3, which has Artix-7 cpg236 (xc7a35tcpg236-1) package. The link for reference manual of the board is provided in the references.

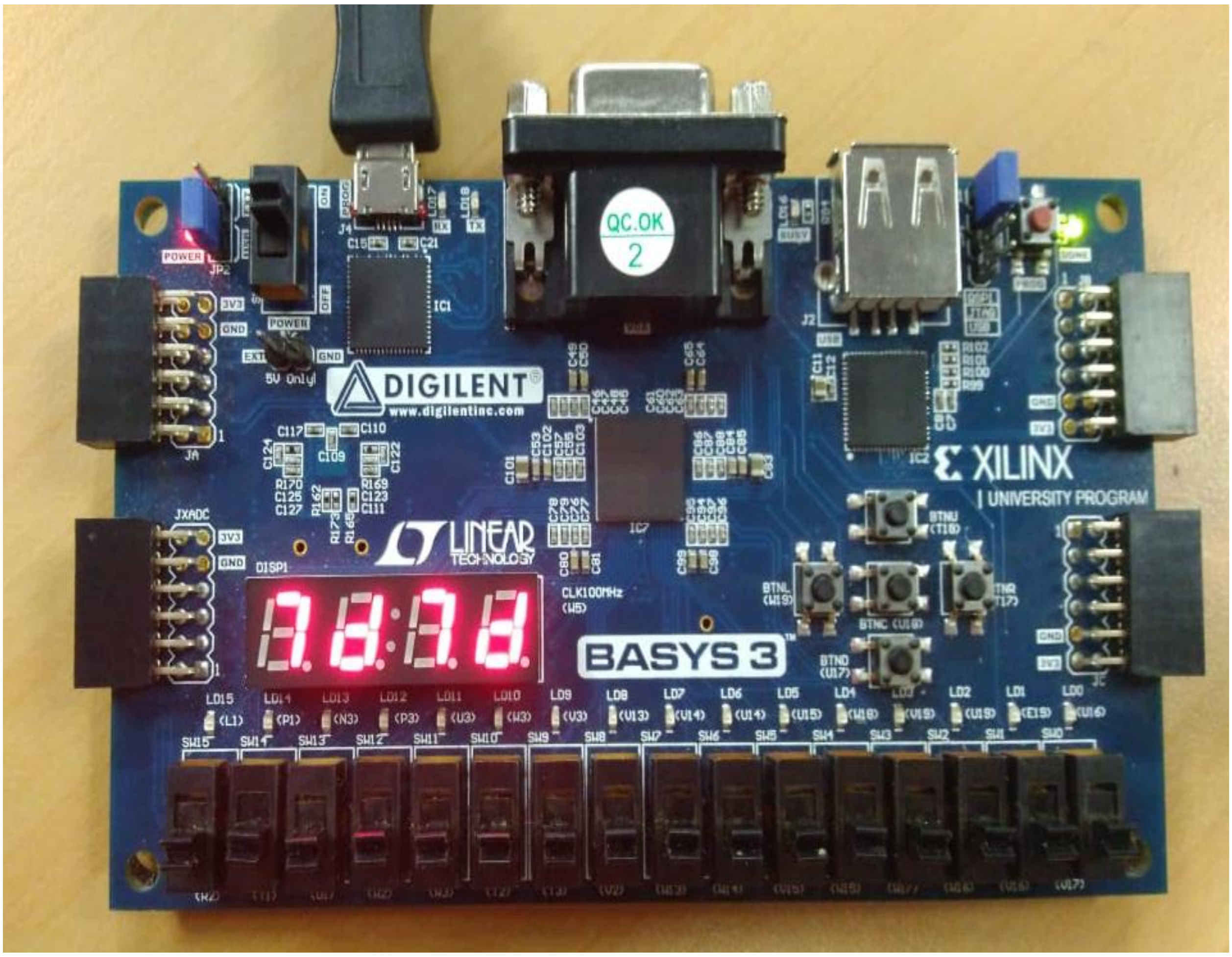
Variables		Function	Pin on BASYS 3
clk		Clock signal	W5
rst		Reset Signal	R2
load		To load the data in the beginning	V17
pb_in1		Input from push button 1	W19
pb_in2		Input from push button 2	T18
anode[3:0]	anode[3]	Pins to select the seven segment anode from the four available anodes (active low)	W4
	anode[2]		V4
	anode[1]		U4
	anode[0]		U2
cathode[6:0]	cathode[6]	Pins to select the cathode values of each seven segment to display the data (whichever pin is logic low, will work- active low)	W7
	cathode[5]		W6
	cathode[4]		U8
	cathode[3]		V8
	cathode[2]		U5
	cathode[1]		V5
	cathode[0]		U7

On the FPGA board, anode and cathode pins are active low, i.e. if anode[0] is to be selected, the value of anode will be 4'b1110. Similarly, if value 'F' is to be displayed, cathode value will be 7'b0111000.

The load signal is used to load the values into memory on the positive edge of the push button 2.

Procedure to get output on board:

1. Reset the board (R2 switch on board).
2. Set load value to 1 (V17 switch on the board).
3. Press the push button 2 once while load is 1, and set the load value to 0.
4. Now press the push button 1 for original data and push button 2 for filtered data, one by one.


QC.OK
2
DIGILENT
www.digilentinc.com
XILINX
UNIVERSITY PROGRAM
LINEAR
TECHNOLOGY
BASYS 3
CLK100MHz
POWER
5V Only!

References

1. https://www.eddyfi.com/ndt/surface-inspection/road-to-median-filter-mastery-part-1/
2. http://www.librow.com/articles/article-1
3. https://en.wikipedia.org/wiki/Median_filter
4. http://tracer.lcc.uma.es/problems/mfp/mfp.html
5. BASYS 3 Reference Manual
   https://reference.digilentinc.com/_media/basys3:basys3_rm.pdf