Area Efficient Implementation of One-Dimensional Median Filter using BEC CSLA

1Tulasiram K., 2Y. Rajasree and 3K. Anitha Sheela

1 Research Scholar, JNTU, Hyderabad, TS, India
2 Professor, Department of ECE, St.Peter’s Engineering College, Hyderabad, TS, India
3 Professor, Department of ECE, JNTU, Hyderabad, TS, India
Corresponding Author E-mail: ramukorrapati@gmail.com

Abstract: This paper presents a circuit implementation and new architecture of one dimensional median filter. Normally, digital adder affects the overall circuit performance. The proposed method low area carry select adder (CSLA) is mostly used in digital circuits and high speed applications. Due to the presence of two Ripple Carry adders (RCA) in the structure, regular square root CSLA absorbs more power and more area. In proposed method instead of using RCA, Binary to Excess-1 Converter (BEC) is used to reduce the area and power.

Keywords: BEC, RCA, CSLA, Excess-1

1. INTRODUCTION

Image enhancement and the noise filtering are the main application in the image processing. In the method of visual interpretation, these two tasks are very essential in image processing [1]. In previous years, linear filter has been used for noise removal and edge preserving. During this process, the data loss is main problem of linear filter. To minimize data loss problem consider the nonlinear filter, it can be performed edge preservation without any data loss. The inadequacy of image sensors causes the noisy images. Impulse noise is mostly affected by memory location hardware, camera sensors and errors during the data transmission. Impulse noise is commonly classified as two types such as random valued shot noise and salt and pepper noise. In random values shot noise, the arbitrary value can be assigned to the noisy pixels. If the salt and pepper noise affected to the image, minimum or maximum values can take by noisy pixels. So it is hard to abolish these kinds of noise using linear filters. To conquer this problem move to median filter [2]. With the help of median filter, without losing high frequency the impulse noise can be smoothened. By using median filter, it is possible to dismiss the impulse noise from the images, smoothen the transient signals and preserves the edge evidence [3].

Median filter can be separated into three methods such as sorting network architectures, array architectures and stack based architectures. In the sorting network architectures, before choose the sample of corresponding rank to perform ranging the samples. Due to the additional amount of compare-swap units, these architectures
produced higher throughput. In array architectures, each element of the window is connected to the rank. The ranks are updated when the window moves to the succeeding position. In the stack based architectures with the aid of majority of elements, hamming comparators, and threshold logic, translate the filtering into the binary domain [4]. Depending on the number of samples, the hardware design divided into two types such as word level architecture and bit level architecture [5]. In the word level sorting, the input samples can be processed word by word in sequentially, and the receiving sample can be injected into the correct rank in two steps. The first step while moving the samples to the left, the previous sample is detached from the window. The arriving sample can be inserted in the right place after comparing the receiving sample with already organised samples, which performs is second step [6]. In the bit level architecture, the samples are collected in parallel and incoming sample bits are sequentially processed [7]. Two clock cycles are needed for performing these two different architectures. These architectures require additional signal transitions in the circuit, large sample width and more dynamic power. To conquer this problem, the proposed model benefits in the new median filter with CSLA. In digital adders, to propagate a carry the speed of count is restricted by time. The implementation of a Rank generator modules 3 bit, 4 bit, 5 bit and 6 bit digital adders are required to generate the multiplexer output [8]. In proposed method instead of using digital adders, we have to use CSLA adder to produce the multiplexer output [9], [10]. Fast arithmetic function can be performed in CSLA, which is one of the quickest adders helps in various data processors. Minimizing the size and improvement in power consumption can be achieved by CSLA.

2. RELATED WORK

J. O. Cadenas et al. [11] proposed a median filter enterprise in parallel counters. In this paper, based on accumulative parallel counters (APC) to usual the positive integer window. The normal digital adder is utilize to diminish the area and power consumption. But not care about time delay.

R. D. Chen et al. [12] presented an area-efficient one dimensional median filter based on the sorting network. The window is sorting in descending order in word level filter. The new sample get into the block. But, the old sample not following the queue that causes the collision in the sample window.

Kamarujjaman et al. [13] proposed an effective attitude to dominance algorithm and its VLSI design for suppression of salt and pepper noise with higher density. This paper also using FPGA implementation, but not care about the area and power dissipation.

Mukherjee, M. et al. [14] presented a low complexity reconfigurable hardware architecture for adaptive median filter. In this paper, only concentrate on mean square error (MSE) and peak signal to noise ratio (PSNR). Here, the power and area is restricted one. So can’t decrease below from its restricted range.

A. Pereverzev et al. [15] developed the architecture of 1-D median filter which permitted to increasing the length of the aperture. The normal digital adder is utilize in the Verilog description. In this paper not possible to reduce the parameter such as power and area for low level.

3. PROPOSED METHOD

The proposed architecture consists of low power CSLA, MUX and logic gates. The efficiency of the architecture can be estimated by using whole power utilization, area utilization and maximum frequency. The working principle of the architecture is described below:

3.1. Filter architecture

This structure consists of three auxiliary modules such as rank calculation (Rank Cal), rank selection (Rank Sel), and median selection (Median Sel) and N identical cells, which is shown in the fig. 1. All the modules are linked to the X input register and the median can be kept in the Y output register. With help of growing edge of a total
Area Efficient Implementation of One-Dimensional Median Filter using BEC CSLA

clock, the register architecture can be synchronized. Each cell block \( c_i \) consists of three registers such as rank register (\( P_i \)), token register (\( T_i \)), and data register (\( R_i \)). The sample cell \( c_i \) stores in the \( R_i \) register, the rank of the sample keeps in \( P_i \) register, and \( T_i \) keeps the permit signal of \( R_i \). The rank starts from 1 for a cell with a least sample value, and ends with \( N \) for a cell with the greatest example value in the \( N \) window size. The example value \( R_i \) of a cell \( c_i \) whose rank \( P_i \) is equal to \((N+1)/2\), where odd number represent as \( N \). In this architecture, based on FIFO method the input model enters into the block. Once the sample is queued, it won’t be de-queued. If the sample keeps the token 1, it helps to resist the queuing of new input model and de-queuing of old sample at the same time. Once the token is utilized it will give to the succeeding clock cycle. Immobile of \( R_i \), our architecture operates low power applications.

At the first stage, the received sample can be kept in the \( c_i \), exit cell mentioned as \( c_N \) and the shadow circle output mentioned as \( T_N \). the rank cell can be updated, whenever the input sample arrives through the window. It pivot on the token, the circuit performs differently.

3.1.1. Circuit behaviour

Initially, the two-stage pipelined filter executes the succeeding operation for every machine cycle \( t_i \) and input sample \( X \): Initially, need to find the innovative rank cell of each cell. Then, \( X \) value insert to the cell that cell holds the token and the token will proceed to the following cell. At the next cycle \( t_{i+1} \), the new values will be compute and updated for all \( T_i, R_i, \) and \( P_i \) registers. The middle value will be evaluated by using second pipelined stage for the input sample enters into the aperture at the preceding cycle \( t_{i-1} \). The updated value at the upcoming cycle \( t_{i+1} \) will be determined in the output register \( Y \). A window contains nine input window and five cells are given in Tab.1. All the registers such as \( P_i, R_i, \) and \( T_i \) for each and every cells \( C_i \) given in table. First input sample stored in first cell \( C_i \), and the last cell is condescend to hold the taken \((T_5=1)\) at \( t_0 \) clock cycle. The two output/input registers \( Y \) and \( X \) along with the sample and rank values (\( P_i \) and \( R_i \)) of every cells are rearrange to be zero.

Figure 1: Low power one dimensional median filter architecture

![Low power one dimensional median filter architecture](image-url)
At cycle $t_1$, the first sample enters into the window. At the time, the token has been passed to $c_1$ from $c_3$ ($T_1=1$ and $T_3=0$). At cycle $t_0$, the $P_3$ value won’t change from zero. Since $c_1$ holds the token, the new value of $R_1$ represented as 12 to cache the input sample at $t_2$ clock cycle. Since, the sample rate of other four cells is lesser than the sample value of 12, which helps to calculate $P_1$ value as 5. To indicate the token move to $c_2$ from $c_1$, the new values of $T_2$ and $T_1$ can be calculated as 1 and 0 respectively. At the following cycle $t_2$, all the $P_i$, $R_i$, and $T_i$ registers values will be updated. At cycle $t_6$, the cell $c_1$ holds the token again ($T_1=1$), when the block fully busy with useable data. At cycle $t_7$, the median output $Y$ will be enumerated as the value of $R_4$ (47) since the value $p_4$ is equivalent to 3, i.e., $(5+1)/2$. All the $P_i$, $R_i$, and $T_i$ registers also refurbish at the cycle, when the $Y$ is updated to be 47 at cycle $t_7$ for the input sample 66. The new median can be calculated in a cell whose rank is equivalent to $(N+1)/2$.

3.2. Rank updating

Table 1

Example illustrating the insertion of nine input samples into a window

<table>
<thead>
<tr>
<th>Cell Registers</th>
<th>Output Reg Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cik</td>
<td>Input Reg X</td>
</tr>
<tr>
<td>-----</td>
<td>-------------</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>12</td>
<td>35</td>
</tr>
<tr>
<td>13</td>
<td>47</td>
</tr>
<tr>
<td>15</td>
<td>66</td>
</tr>
<tr>
<td>16</td>
<td>52</td>
</tr>
<tr>
<td>17</td>
<td>44</td>
</tr>
<tr>
<td>18</td>
<td>38</td>
</tr>
<tr>
<td>19</td>
<td>26</td>
</tr>
</tbody>
</table>

Table 1 shows the insertion of nine input samples into a window. In this case, two major operation can be considered such as cell with token and cell without token. If the sample contains the token, the sample value replaced with the input sample and rank can be recalculated. The window sample will be inserted, which sample don’t occupy in the block cell.

3.2.1. Cell with the Token

For a $c_i$ cell holds the token, by using input, $X$ the $R_i$ value can be replaced and also $P_i$ has to be reformed. By comparing $X$ with the sample values of other $N-1$ cells the new $P_i$ value can be gained that don’t contain the token. The fresh value of $P_i$ will be $K+1$, when the number of cells ($K$) whose sample value is fewer or equivalent to $X$. For example, at clock cycle $t_0$, the novel rank value $P_i$ will be evaluated as $2+1$ at the following cycle $t_1$.

3.2.2. Cell without the Token

In the method of cell without the token, there are five cases will be used such as decremented by 1, incremented by 1, rest of the three methods are unchanged.

Case I – {Decremented by 1}

In this case, check the condition $P_i > P_j$ and $R_i <= X$. If the condition is fulfilled the reference value will be reduced by 1. At next clock cycle, the ref value is fewer or equivalent to $R$ can be declined by 1. i.e., $P_i$ has to be reduced by 1. For example, take a cycle $t_7$ at rank $P_i$ that value will be discount by 1 (from 4 to 3) at the following cycle $t_8$. 
Case II – (Incremented by 1)

In this case, check the condition $P_i < P_j$ and $R_i > X$. If the condition is gratified the current window will be raised by 1. At next clock cycle, the sample value is fewer or identical to $R_i$ will be upturned by 1. i.e., $P_i$ has to be hiked by 1. For example, take a cycle $t_7$ at rank $P_4$ that value will be inflation by 1 (from 2 to 3) at the succeeding clock cycle $t_8$.

Case III – (kept unchanged)

In this case, check the condition of $P_i < P_j$ and $R_i \leq X$. If this condition satisfied the reference value won’t be changed. At next clock cycle, the number of current block is fewer or equal to $R_i$ at the current cycle can be identical i.e., $P_i$ has to be kept unchanged. For example, take a cycle $t_7$ at rank $P_3$ that value unchanged at the following clock cycle $t_8$.

Case IV – (kept unchanged)

In this case, check the condition of $P_i > P_j$ and $R_i > X$. If this condition satisfied the sample value won’t be changed. At next clock cycle, the sample value is less than or equal to $R_i$ at the current cycle will be identical i.e., $P_i$ has to be kept unchanged. For example, take a cycle $t_3$ at rank $P_2$ that value will be unchanged at the succeeding clock cycle $t_4$.

Case V – (kept unchanged)

In this case, if $P_i = P_j$ the sample will be unchanged. When the block is not yet fully busy with valid data, this case will be occurred. Initially, the rank of every cells resets to be zero. Once the window is fully busy, each cells set to be non-zero value and single rank. For example, take a cycle $t_3$ at rank $P_4$ that value will be zero at the following cycle $t_4$.

4. CIRCUIT IMPLEMENTATION

The rank generation module implementation in a cell $c_i$ is shown in fig.2. The $R_i$ and input $X_i$ value can be performed “$\leq$” operation, which gives the output $F_i$. If $R_i$ greater than $X$, it will gives $F_i=0$ else $F_i=1$.

![Figure 2: Implementation of rank gen module](image)
If $F_i = 1$ and $T_i = 0$, the output of AND gate value should be $A_i = 1$. The output of AND gate gives a value $A_i = 0$, when $R_i$ is fewer or equivalent to $X$ and the cell $c_i$ doesn’t hold the token. Rank cal and $A_i$ signal connecting together, which is charity to find the fresh rank cell that holds the token.

There are four sources is given to the input of 4:1 mux that delevers the one resource signal $Q_i$, which is shown in fig.2. By using Ctrl module, two selection line ($S_0$, $S_1$) will be generated, which is apply to controlled the mux and determine the $F_i$, $T_i$, $G_i$and $E_i$ four signals.

The implementation of Ctrl module shown in fig.3. From the Rank cal module output $A$, the new rank will be reform if the cell $c_i$ holds the token. if $T_i = 1$, the $S_0$ and $S_1$ value denoted as 11 else $T_i$ should be 0.

To transferring the $P_i$ rank to output $B$, the rank sel module can be used if $c_i$ comprises the token when $T_i = 1$. Fig. 4 shows the implementation of Rank Sel module using simple AND, OR gates. If the $T_i$ mentioned as 1, this module gives the output $B$. 

Figure 3: Implementation of the Ctrl module

Figure 4: Implementation of Rank Sel module
The main proposal is instead of using normal adder CSLA adder can be used, which is given in fig.2. The major speed limitation in any adder is in the production of carries and many authors have considered the addition problem. The basic idea of the proposed work is using n-bit Binary to Excess-1 Converters (BEC) to improve the speed of addition. This logic can be implemented with Carry Select Adder to Achieve Low Power and Area Efficiency.

![Figure 5: Low area carry select adder](image)

The CSLA is used in many computational systems to reduce the problem of carry propagation delay by independently generating multiple carries and then select a carry to generate the sum. However, the CSLA is not area efficient because it uses multiple pairs of Ripple Carry Adders (RCA) to generate partial sum and carry by considering carry input Cin = 0 and Cin = 1, then the final sum and carry are selected by the multiplexers (mux). The entire work performed by usage of Binary to Excess-1 Converter (BEC) instead of RCA with Cin = 1 in the regular CSLA to achieve lower power consumption. The main advantage of this BEC logic comes from the lesser number of logic gates than the n-bit Ripple Carry Adder (RCA).

<table>
<thead>
<tr>
<th>B[2:0]</th>
<th>X[2:0]</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>001</td>
</tr>
<tr>
<td>001</td>
<td>010</td>
</tr>
<tr>
<td>010</td>
<td>011</td>
</tr>
<tr>
<td>011***</td>
<td>100</td>
</tr>
<tr>
<td>100</td>
<td>101</td>
</tr>
</tbody>
</table>

Table 2
Functional table of 3-bit BEC
This carry select adder with BEC-1 is used in place of increment and decrement operations in rank generation module. For decrementing you need to add 2’s complement value of 1 to the count value.

4. EXPERIMENTAL SETUP

The proposed method simulated in Modelsim SE 10.1c using Verilog code and also the entire work is done by using I7 system with 8 GB RAM. Similarly, determination of area, power, and delay done by using cadence 180nm technology and RTL compiler.

5. RESULTS AND DISCUSSION

Table 2 and Table 3 gives the comparison of the existing and proposed method in terms of power, core area and delay for different 8 bit and 16 bit sample width. To find the energy for Lena, peppers and baboon image in 8-bit and three different audio in 16-bit.

<table>
<thead>
<tr>
<th>Design</th>
<th>Throughput (#median outputs/clock)</th>
<th>Latency (#clock cycles)</th>
<th>Window size</th>
<th>8-bit sample width</th>
<th>EPS(nJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Core area (um²)</td>
<td>Power (nW)</td>
</tr>
<tr>
<td>Existing</td>
<td>1</td>
<td>w</td>
<td>5</td>
<td>39473.4</td>
<td>792366</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td></td>
<td></td>
<td>79599.2</td>
<td>2073216</td>
</tr>
<tr>
<td>Proposed</td>
<td>1</td>
<td>w</td>
<td>5</td>
<td>17989</td>
<td>1325922</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td></td>
<td></td>
<td>34931</td>
<td>2694208</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Design</th>
<th>Throughput (#median outputs/clock)</th>
<th>Latency (#clock cycles)</th>
<th>Window size</th>
<th>16-bit sample width</th>
<th>EPS(nJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Core area (um²)</td>
<td>Power (nW)</td>
</tr>
<tr>
<td>Existing</td>
<td>1</td>
<td>w</td>
<td>5</td>
<td>59722.7</td>
<td>1402155</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td></td>
<td></td>
<td>116579</td>
<td>3426665</td>
</tr>
<tr>
<td>Proposed</td>
<td>1</td>
<td>w</td>
<td>5</td>
<td>28710</td>
<td>2076690</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td></td>
<td></td>
<td>54272</td>
<td>3917452</td>
</tr>
</tbody>
</table>

6. CONCLUSION

In this paper, one dimensional median filter architecture using CSLA method is presented, which benefits to reduce the area, power consumption and delay. BEC, which is utilize to increase the speed of addition operation. The CSLA adder obtained low power consumption and low area, when it operated in BEC instead of RCA. The following results are the main advantages in BEC such as low area, low power, less number of Full Adder (FA) structure, simple and high efficient in VLSI implementation.
Figure 6: Total area value for window 5 proposed

Figure 7: Total delay for window 5 proposed
Figure 8: Total power value for window 5 proposed

Figure 9: RTL schematic for window 5 proposed
REFERENCES


