FPGA implementation of a low power and high speed Hybrid Multiplier for Image Processing Applications

S.P. Valan Arasu[1] S.Baulkani[2]
[1] Associate Professor, ECE Department
Dr. Sivanthi Aditanar College of Engineering
Tiruchendur, Tamil Nadu – 628 215, India
Email: kakkurichi@yahoo.com
[2] Associate Professor, ECE Department
Government College of Engineering
Tirunelveli, Tamil Nadu – 627 007, India
Email: ramabaulkani@yahoo.co.in

Abstract— **Multipliers are one of the major dynamic power consuming elements in most of the processor architectures. Hence, there is an essential need to focus on designing multipliers with low dynamic power consumption and if possible with higher operating speed. In this work, considering Very Large Scale Integrated (VLSI) system design, architectural modifications in the conventional hybrid multiplier architecture has been attempted and this helps in minimizing the switching activities, thereby reducing the dynamic power consumption. This also leads to lesser propagation delay. The architectural modification in the conventional hybrid multiplier is made such that a Modified Booth Multiplier (MBM) and the Wallace tree multiplier are hybridized with Carry Look-ahead Adder (CLA) to form a low dynamic power consuming high speed hybrid multiplier.** The proposed system is synthesized and simulated using Xilinx 14.1, Cadence 6.1.5 and MATLAB 2013a.

Keywords— MBM, Wallace tree multiplier, 3:2 and 4:2 compressors, CLA, Gaussian filter, Mean filter, Wiener filter.

I. Introduction

Multipliers are one of the major power consuming elements with longer latency period so that it affects the speed of the system with the requirement of high power. Therefore, designing them with low-power and high speed is an important concern in VLSI system design.

A. Wallace Tree Multiplier

In a conventional hybrid multiplier, the Wallace tree multiplier accelerates the accumulation of the partial products by using Carry Save Adder structures. A Carry Save Adder which incorporates a 3:2 compressor, adds up three binary numbers and produces a sum and carry [5]. In the Wallace tree method, when two numbers are multiplied the steps usually followed are:

- The partial products are generated.
- The bit product matrix is “reduced” to a two row matrix by using a tree of carry save adders known as Wallace tree.
- The resultant two rows are summed by using a fast carry-propagate adder to generate the multiplied result which is the product.

Thus, the Wallace tree multiplier has lesser delay when compared to the other conventional multipliers.

B. Modified Booth Multiplier

Modified Booth multiplier produces at most N/2 partial products from an N bit operand. The Modified Radix-4 Booth algorithm is most widely used, when the operands are equal to or greater than 16 bits. The resulting encoded partial products can then be summed up by using a suitable method. By avoiding the time consuming carry propagate additions, the performance of a multiplier can be improved.

C.Gaussian Filter

One of the applications of Gaussian filter is the removal of high frequency salt and pepper noise in an image. In a Gaussian filter, the convolution process is carried out using the formula,

$$y(m,n)=\sum_{i=1}^{rows}\sum_{j=1}^{columns}h(i,j)\cdot x(m-i,n-j) \qquad (1)$$

where, x is the input fingerprint image, h is the filter mask, y is the output image, rows is the number of rows in the fingerprint image and columns is the number of columns in the fingerprint image.

$$\frac{1}{256} * \begin{array}{|c|c|c|}\hline 21 & 31 & 21 \\ \hline 31 & 48 & 31 \\ \hline 21 & 31 & 21 \\ \hline\end{array}$$

Fig. 1. Gaussian Kernel Window.

The Gaussian function is given by,

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}} \tag{2}$$

where, σ is the standard deviation and the value is 1. The Gaussian kernel window is given in Fig. 1.

D. Mean Filter

The Mean filter is a sliding-window spatial filter that replaces the center pixel value in the window with the mean of all the pixel value in the window. This filter is generally used for removing the high frequency noise in images. The formula for mean filter is given by,

$$f(x,y) = \frac{1}{mn} \sum_{(s,t)\in S_{xy}} g(s,t) \tag{3}$$

where, g is the input fingerprint image in area defined by S_{xy}. S_{xy} represent the set of coordinates in a sub image window of size m x n, centered at point (x,y) and f is the restored image.

II. EXISTING METHOD

Hybrid multiplier architecture with the modified booth multiplier, Wallace tree multiplier and carry Look-ahead adder have been proposed in [1]. The flow diagram depicting this architecture is shown in Fig. 2.

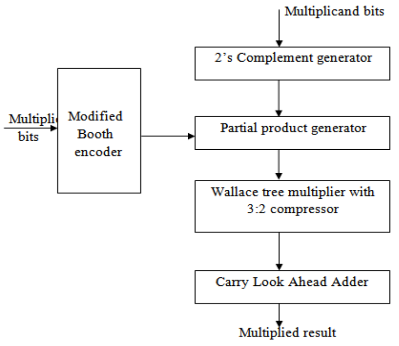


Fig. 2. Flow diagram depicting the architecture of the existing Hybrid Multiplier

This hybrid multiplier is mainly used for reducing the partial products and to enhance the speed of the computation. MBM creates n/2 partial products for 'n' inputs, so that the area is reduced and the speed is increased. Wallace tree is designed by Carry Save Adders (CSA) which is used for fast addition since 3:2 compressors are used. The results of CSA are finally added by Carry Look-ahead Adder (CLA) in order to generate the multiplied result.

TABLE I. ENCODING TABLE

Multiplier bits	Operation
000	Add nothing
001	+ 1 × Multiplicand
010	+ 1 × Multiplicand
011	+ 2 × Multiplicand
100	- 2 × Multiplicand
101	- 1 × Multiplicand
110	- 2 × Multiplicand
111	Subtract nothing

Table I shows the group of three bits formed in the Booth encoder [7]. It is fed to the Modified Booth encoder, and thus it forms the each row of partial products. For 8-bit multiplication, the partial products are assembled in four rows. Thus, the generated partial products have been reduced to half. However, when the input bits are increased, it requires more area for the computation and hence more hardware is utilized.

III. PROPOSED METHOD

A. Proposed Hybrid Multiplier

The proposed multiplier uses two high speed adders such as MBM and Wallace tree multiplier which are hybridized with CLA to perform the final accumulation of the partial products. The multiplication process consists of three steps. They are: 1) generate the partial products; 2) add the generated partial products until the last two rows are remained; 3) compute the final multiplication results by adding the last two rows. The modified Booth algorithm reduces the number of generated partial products by half in the first step.

Fig. 3 shows flow diagram depicting the architecture of the proposed Hybrid multiplier. Here, the multiplier bits are fed to the modified booth encoder, where it performs the encoding operation, where the LSB of the multiplier is appended a '0' bit. Starting from the LSB, each time, three bits are grouped, thus forming a group of overlapping bits and fed according to the Table I, thus forming four rows of partial products [13]. The generated partial products are summed up using the Wallace tree structure which uses the compressor adders. With this, the final product is calculated by using the CLA for getting the final sum and carry.

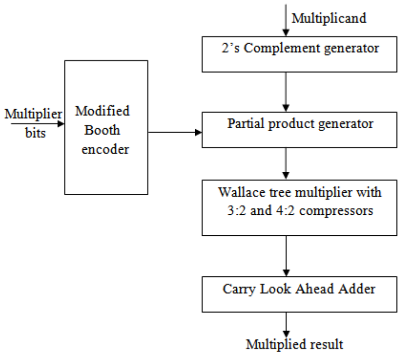


Fig. 3. Flow diagram depicting the architecture of the proposed Hybrid Multiplier

The proposed Wallace tree multiplier has two half adders, two 3:2 compressors and ten 4:2 compressors. The number of partial product addition stages is reduced by using the combination of the 3:2 compressors and 4:2 compressors inside the Wallace tree structure. Compressor is defined as a single bit adder circuit that has more than three inputs as in a full adder and less number of outputs. The 3:2 compressor has 3 inputs and 2 outputs. The output functions of the 3:2 compressor circuits are:

$$Sum = (X1 \wedge X2) * \overline{X3} + \overline{(X1 \wedge X2)} * X3 \tag{4a}$$

$$Carry = (X1 \wedge X2) * X3 + \overline{(X1 \wedge X2)} * X1 \tag{4b}$$

The 4:2 compressor has 4 inputs (x1, x2, x3 and x4) and 2 outputs (Sum & Carry) along with a Carry-in (Cin) and a Carry-out (Cout) as shown in Fig. 4 [3]. The input Cin is the output from the neighboring lower significant compressor. The generated Cout is the output which is fed as input to the next significant stage compressor. The characteristics of the 4:2 compressors are:

- The outputs represent the sum of the five inputs, so it is really a 5 bit adder.

- To avoid carry propagation, the value of Cout depends only on X1, X2, X3 and X4. It is independent of Cin.

- The Cout signal forms the input of the Cin to the 4:2 compressor of the next column.

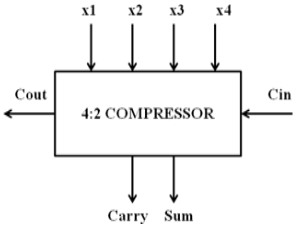


Fig. 4. 4:2 Compressor

The output Cout, is independent of the input Cin which accelerates the carry save summation of the partial products. Output functions of the 4:2 compressor are:

$$Sum = (X1 \wedge X2) * \overline{(X3 \wedge X4)} + \overline{(X1 \wedge X2)} * (X3 \wedge X4) * \overline{Cin} + \overline{(X1 \wedge X2) * \overline{(X3 \wedge X4)} + \overline{(X1 \wedge X2)} * (X3 \wedge X4)} * Cin \tag{5a}$$

$$Cout = (X1 \wedge X2) * X3 + \overline{(X1 \wedge X2)} * X1 \tag{5b}$$

$$Carry = (X1 \wedge X2 \wedge X3 \wedge X4) * Cin + \overline{(X1 \wedge X2 \wedge X3 \wedge X4)} * X4 \tag{5c}$$

With the 4:2 compressor stages, the number of partial product addition stages is reduced. They are usually used to add the partial products in a tree-like fashion to produce two rows of partial products that can be added in the last stage. The proposed Wallace tree architecture is used for the addition of partial products using 3:2 and 4:2 compressors. Then, the CLA is used for getting the sum and carry as the result of the multiplication. Fig. 5 shows a multiplication example of the proposed Hybrid multiplier, where X=106 and Y=105, thus producing the final product of 11130 in binary representation.

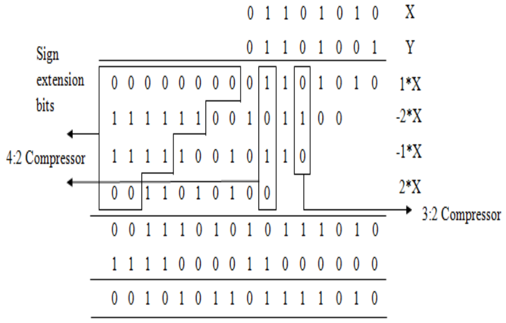


Fig. 5. A Multiplication Example of the proposed Hybrid Multiplier

B. *Application of the proposed hybrid multiplier for high frequency noise removal in fingerprint images*

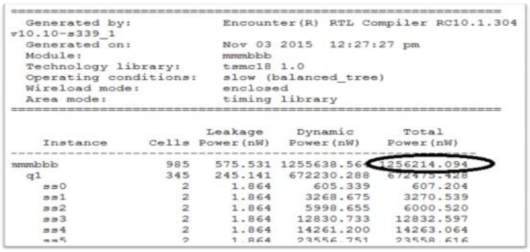
To authenticate a person, the biometric traits such as fingerprint, voice or face recognition are available choices. These biometric samples when processing in a system has the higher possibility of inclusion of the unwanted high frequency noise. Hence, a pre-processing stage is essential in order to remove these noise components. In this work, the proposed hybrid multiplier is utilized in Mean filter, Gaussian filter and Wiener filter structures. Further, a comparative study of the above filter banks with the existing hybrid multiplier and the proposed hybrid multiplier is carried out in order to remove the high frequency noise components in the noisy fingerprint images. The fingerprint images were taken from the database, FVC2004 [14]. The performance parameters of the image processing systems such as mean square error (MSE) and peak signal to noise ratio (PSNR), for the following two cases of input noisy images were analyzed: i) 4% salt and pepper noise added to the base fingerprint image and ii) 2% salt and pepper noise added to the degraded fingerprint image. It is observed that the proposed image processing system employing the proposed hybrid multiplier architecture has low power and high speed when compared to the system employing the existing hybrid multiplier architecture in both the noise analysis cases, thus attaining high performance.

IV. Experimental Results

The power consumption and delay of the existing and proposed hybrid multipliers were compared initially. Then, in order to build the noise free fingerprint images, the proposed hybrid multiplier is incorporated inside the filter banks of Mean, Gaussian and Wiener, for the following two cases of input noisy images: i) 4% salt and pepper noise added to the base fingerprint image and ii) 2% salt and pepper noise added to the degraded fingerprint image. The simulation results were obtained by using Xilinx 14.1, Cadence 6.1.5 and MATLAB 2013a. To have a comparative study about the performance of these image processing systems, the mean square error and peak signal to noise ratio values were analyzed.

A. *Power Consumption Comparison*

The power report of the proposed Hybrid Multiplier obtained using Cadence 6.1.5 is shown in Fig. 6. It shows that the proposed hybrid multiplier has a power consumption of 1.256214 mW. Table II shows a power consumption and delay comparison of this proposed hybrid multiplier with the existing hybrid multiplier, conventional shift-and-add multiplier and the Error tolerant shift-and-add multiplier.



```
================================================================================
  Generated by:            Encounter(R) RTL Compiler RC10.1.304
v10.10-s339_1
  Generated on:            Nov 03 2015  12:27:27 pm
  Module:                  mmmbbb
  Technology library:      tsmc18 1.0
  Operating conditions:    slow (balanced_tree)
  Wireload mode:           enclosed
  Area mode:               timing library
================================================================================

                          Leakage     Dynamic       Total
   Instance       Cells  Power(nW)   Power(nW)    Power(nW)
---------------------------------------------------------------
mmmbbb              985    575.531 1255638.564 1256214.094
  q1                345    245.141  672230.288  672475.428
    ss0               2      1.864     605.339     607.204
    ss1               2      1.864    3268.675    3270.539
    ss2               2      1.864    5998.655    6000.520
    ss3               2      1.864   12830.733   12832.597
    ss4               2      1.864   14261.200   14263.064
    ss5               2      1.864   23556.751   23558.616
```

Fig. 6. Power Report of Proposed Hybrid Multiplier

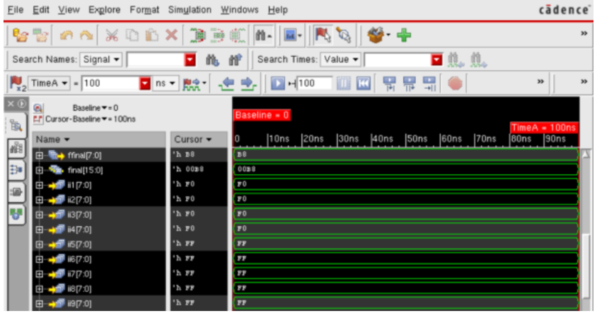
TABLE II. Delay And Power Consumption Comparison Of The Existing 8-Bit Multipliers With That Of The Proposed Hybrid Multiplier

Multiplier Types	Delay (ns)	Power Consumption (mw)
Conventional Shift and Add Multiplier [9]	95.43	295
Error Tolerent Shift and Add Multiplier [9]	49.32	228
Existing Hybrid Multiplier	5.430	1.271083
Proposed Hybrid Multiplier	5.283	1.256214

Table III shows the device utilization summary of the existing and proposed hybrid multipliers in Virtex family XC7VX485TFFG1157-1 kit which gives the utilization of Look Up Table (LUT) and Input/Output (I/O) blocks in the FPGA architecture. The output waveforms of Gaussian and Mean filters by using the proposed hybrid multiplier are shown in Fig. 7 and Fig. 9. The performance analysis of image processing systems employing the existing and proposed hybridized multipliers is depicted in Table IV. It shows that the Gaussian filter and Mean filter implemented with the proposed hybrid multiplier have reduced power consumption when compared with the systems utilizing the existing hybrid multiplier. The power reports of the Gaussian and Mean filters implemented by using the proposed hybrid multiplier are shown in Fig. 8 and Fig. 10.

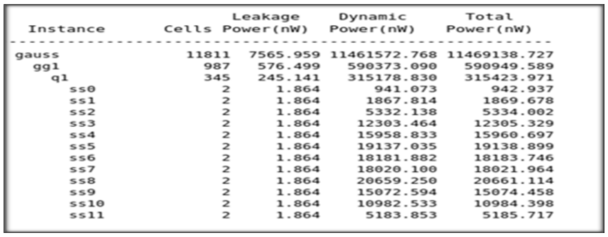
TABLE III. Device Utilization Summary For The 8-Bit Multiplier Utilizing The Existing And The Proposed Hybrid Multiplier Structures

Multiplier Types	LUT (%)	I/O (%)
Hybrid Multiplier	1	7
Proposed Hybrid Multiplier	1	5



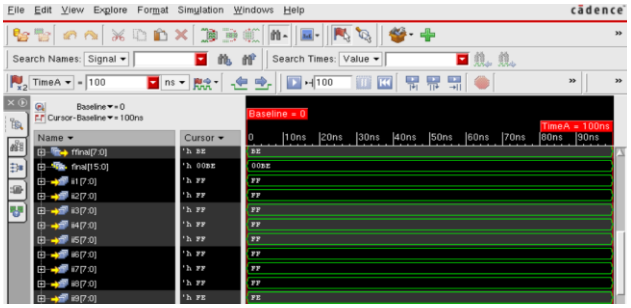
```
ffinal[7:0]   'h B8     B8
final[15:0]   'h 00B8   00B8
ii1[7:0]      'h F0     F0
ii2[7:0]      'h F0     F0
ii3[7:0]      'h F0     F0
ii4[7:0]      'h F0     F0
ii5[7:0]      'h FF     FF
ii6[7:0]      'h FF     FF
ii7[7:0]      'h FF     FF
ii8[7:0]      'h FF     FF
ii9[7:0]      'h FF     FF
```

Fig. 7. Output waveform of Gaussian filter using proposed multiplier



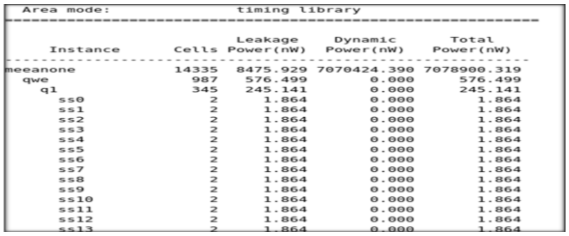
```
                                Leakage      Dynamic          Total
 Instance              Cells   Power(nW)    Power(nW)        Power(nW)
-------------------------------------------------------------------------
gauss                  11811   7565.959  11461572.768  11469138.727
  gg1                    987    576.499    590373.090    590949.589
    q1                   345    245.141    315178.830    315423.971
      ss0                  2      1.864       941.073       942.937
      ss1                  2      1.864      1867.814      1869.678
      ss2                  2      1.864      5332.138      5334.002
      ss3                  2      1.864     12303.464     12305.329
      ss4                  2      1.864     15958.833     15960.697
      ss5                  2      1.864     19137.035     19138.899
      ss6                  2      1.864     18181.882     18183.746
      ss7                  2      1.864     18020.100     18021.964
      ss8                  2      1.864     20659.250     20661.114
      ss9                  2      1.864     15072.594     15074.458
      ss10                 2      1.864     10982.533     10984.398
      ss11                 2      1.864      5183.853      5185.717
```

Fig. 8. Power report of Gaussian filter using proposed multiplier



```
ffinal[7:0]   'h BE     BE
final[15:0]   'h 00BE   00BE
ii1[7:0]      'h FF     FF
ii2[7:0]      'h FF     FF
ii3[7:0]      'h FF     FF
ii4[7:0]      'h FF     FF
ii5[7:0]      'h FF     FF
ii6[7:0]      'h FF     FF
ii7[7:0]      'h FF     FF
ii8[7:0]      'h FF     FF
ii9[7:0]      'h FE     FE
```

Fig. 9. Output waveform of Mean filter using proposed multiplier



```
 Area mode:                timing library
------------------------------------------------------------

                            Leakage    Dynamic      Total
    Instance        Cells Power(nW)  Power(nW)   Power(nW)
-----------------------------------------------------------
meeanone            14335  8475.929 7070424.390 7078900.319
  qwe                 987   576.499       0.000     576.499
    q1                345   245.141       0.000     245.141
      ss0               2     1.864       0.000       1.864
      ss1               2     1.864       0.000       1.864
      ss2               2     1.864       0.000       1.864
      ss3               2     1.864       0.000       1.864
      ss4               2     1.864       0.000       1.864
      ss5               2     1.864       0.000       1.864
      ss6               2     1.864       0.000       1.864
      ss7               2     1.864       0.000       1.864
      ss8               2     1.864       0.000       1.864
      ss9               2     1.864       0.000       1.864
      ss10              2     1.864       0.000       1.864
      ss11              2     1.864       0.000       1.864
      ss12              2     1.864       0.000       1.864
      ss13              2     1.864       0.000       1.864
```

Fig. 10. Power report of Mean filter using proposed multiplier

B. Delay Comparison

Table II shows that the proposed hybrid multiplier has a delay of 5.283 ns which is reduced when compared to the existing hybrid multiplier which has a delay of 5.430 ns. The delay of the image processing system employing the proposed multiplier has reduced when compared with the delay of the system employing the existing multiplier as shown in Table IV. This is true when both these systems incorporate the filter banks such as Gaussian, Mean as well as Wiener filters.

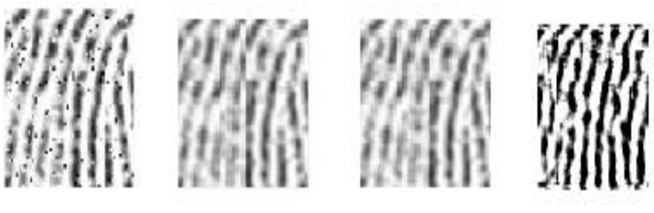
TABLE IV. PERFORMANCE ANALYSIS OF IMAGE PROCESSING SYSTEMS WITH EXISTING AND PROPOSED HYBRID MULTIPLIERS

Image Processing Systems		**Power Consumption (mw)**	**Delay (ns)**
Existing Hybrid Multiplier	Gaussian Filter	15.413516	8.386
	Mean Filter	16.351424	15.5801
Proposed Hybrid Multiplier	Gaussian Filter	11.469138	8.236
	Mean Filter	7.078900	5.166

C. Evaluation of the Performance Parameters for the Image Processing Systems Employing the Proposed Hybrid Multiplier

As the evaluation of the performance parameters like mean square error (MSE) and peak signal to noise ratio (PSNR) for the image processing systems employing the proposed hybrid multiplier, we have considered the following two cases of image denoising. They are: i) Removal of noise in the proposed image processing system with 4% Salt and Pepper noise added to the base fingerprint image and ii) Removal of noise in the proposed image processing system with 2% Salt and Pepper noise added to the degraded fingerprint image. The MSE and PSNR values of the system with mean filtering, Gaussian filtering and Wiener filtering for the 4% Salt and Pepper noise added base fingerprint image are listed in Table V. The MSE and PSNR values of the system with mean filtering, Gaussian filtering and Wiener filtering for the 2% Salt and Pepper noise added degraded fingerprint image are shown in Table VI.

The PSNR for the Wiener filtered image has 17.72% and 22.4% improvements respectively and with Gaussian filtered image has 14.26% and 8.4% improvements respectively, both when compared to the system with Mean filtered image for the above two cases of input noisy images.

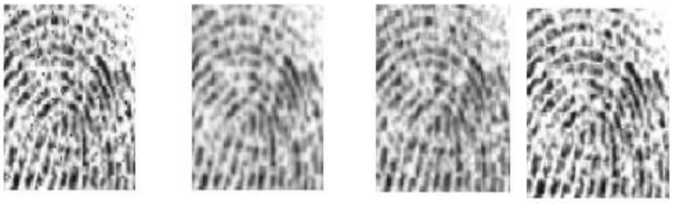


(a) (b) (c) (d)

(a) 4% Salt And Pepper Noise Added To Base Image, (b) Mean Filtered Image, (c) Gaussian Filtered Image (d) Weiner Filtered Image

Fig. 11. Smoothening to remove noise in base fingerprint image in the system employing the proposed hybrid multiplier

The pixel values of the images are calculated by using Verilog code. The numerical values related to the pixels are called by the program written in MATLAB 2013a. Fig.11 shows the fingerprint images after corruption by 4% salt and pepper noise to a standard base image and after the removal of noise by using Mean, Gaussian and Wiener filters. Fig. 12 shows the fingerprint images after corruption by 2% salt and pepper noise to a degraded image and after the removal of noise by using Mean, Gaussian and Wiener filters.



(a) (b) (c) (d)

(a) 2% Salt and pepper noise added to degraded image, (b) Mean filtered image, (c) Gaussian filtered image (d) Weiner filtered image.

Fig. 12. Smoothening to remove noise in degraded fingerprint image in the system using the proposed hybrid multiplier

TABLE V. PERFORMANCE PARAMETERS FOR THE PROPOSED IMAGE PROCESSING SYSTEM WITH 4% SALT AND PEPPER NOISE ADDED TO THE BASE FINGERPRINT IMAGE

Parameters	**Corrupted Image**	**Mean Filtered Image**	**Gaussian Filtered Image**	**Wiener Filtered Image**
MSE	3.0694e+03	94.2969	31.7500	15.427
PSNR(dB)	13.2943	28.4198	33.1474	34.5415

TABLE VI. PERFORMANCE PARAMETERS FOR THE PROPOSED IMAGE PROCESSING SYSTEM WITH 2% SALT AND PEPPER NOISE ADDED TO THE DEGRADED FINGERPRINT IMAGE

Parameters	**Corrupted Image**	**Mean Filtered Image**	**Gaussian Filtered Image**	**Weiner Filtered Image**
MSE	1.3725e+04	1.2827e+04	1.1041e+04	1.0131e+04
PSNR(dB)	6.7898	7.0837	7.7348	9.1289

TABLE VII. DEVICE UTILIZATION SUMMARY FOR THE IMAGE PROCESSING SYSTEMS UTILIZING THE EXISTING AND THE PROPOSED HYBRID MULTIPLIERS

Image Processing System		**LUT (%)**	**I/O (%)**
With Existing Multiplier	Gaussian, Mean and Wiener Filters	1 each	13 each
With Proposed Multiplier	Gaussian, Mean and Wiener Filters	1 each	13 each

Table VII shows the device utilization summary of the image processing systems with the existing and the proposed hybrid multipliers in Virtex family XC7VX485TFFG1157-1 kit which gives the utilization of LUT (Look Up Table) and I/O (Input Output) blocks for the design using Gaussian, mean and Wiener filters.

V. CONCLUSION

A high performance image processing system with the modified hybrid multiplier is designed with reduced delay and lower dynamic power consumption. The design can be used for getting improved results in most of the image processing applications because of its superior performance. Since the partial products are reduced, the speed of computation is increased and also yields lower dynamic power consumption. This proposed Hybrid multiplier structure is utilized in the image processing systems employing Gaussian, Mean and Wiener filters which are used for the removal of salt and pepper noise in the fingerprint images. The image processing system employing the proposed hybrid multiplier using Wiener filter has tremendous improvement in its PSNR value when compared to the system with the Mean filter. Hence, the PSNR improvement for the Wiener filtered image can very well be utilized for enhancing the quality of the images in many real time applications. As the future work, it is proposed to analyze the implementation of this high performance hybrid multiplier in Medical Imaging applications in order to facilitate better diagnosis of diseases. In the proposed architecture, MBM is used to reduce the generated partial products, whereas Wallace tree multiplier is accompanied for fast addition and the CLA is used for final accumulation. The number of partial product addition stages is reduced by using the combination of the 3:2 compressors and 4:2 compressors inside the Wallace tree structure. The simulation results show that this modified hybrid architecture is better in power saving and delay reduction when compared to the existing hybrid multiplier and the conventional shift-and-add and error tolerant shift-and-add types of multipliers. Also, it is found more suitable for adaptation in image processing applications. The simulation results show that it has the improvement in speed by 5% and the reduction in power consumption by 2% when compared to the conventional hybrid multiplier. This proposed hybrid multiplier when implemented in the filter banks for removal of salt and pepper noise in the fingerprint images, the system has tremendous improvement in PSNR (Peak Signal-to-Noise Ratio) values. Also, it has the speed and power consumption advantages. As a comparative study of image denoising, the performance of mean, Gaussian and Wiener filters were analyzed with the proposed hybrid multiplier. The proposed multiplier was implemented in the above referred filter banks for two cases of input noisy images, namely: i) 4% salt and pepper noise added to the base fingerprint image and ii) 2% salt and pepper noise added to the degraded fingerprint image. The PSNR improvement for i) the Gaussian filtered image and for ii) the Wiener filtered image, when compared to the system with Mean filtered image for the above two cases of input noisy images is: i) 14.26% and 8.4% and ii) 17.72% and 22.4% respectively.

REFERENCES

[1] Uvaraj Subramaniam and Srinivasan Alavandar, “Low Power and high Speed computation using Hybridized multiplier”, 2013, IEEE-31661, 4th ICCNT-2013.

[2] Jadhav P A Niranjane V B and Kalyani Ingole, “Design of High Speed Hybridized Multiplier”, International Journal of Research Studies in Science, Engineering and Technology, June 2015, 2(6).

[3] Naveen Kr. Gahlan, Prabhat Shukla and Jasbir Kaur Implementation of Wallace Tree Multiplier Using Compressor, International Journal of Computer Technology and Applications, 2012, 3 (3): 1194-1199.

[4] Sandeep Shrivastava, Jaikaran Singh and Mukesh Tiwari Implementation of Radix-2 Booth Multiplier and Comparison with Radix-4 Encoder Booth Multiplier, International Journal on Emerging Technologies, 2011, 2(1): 14-16.

[5] Sanjeev Kumar and Manoj Kumar, “Low Power High Speed 3-2 Compressor”, International Journal of Electrical, Electronics and Mechanical Controls, May 2013, 2(2).

[6] Satish S Bhairannawar and Rathan R FPGA based efficient Multiplier for Image Processing Applications using Recursive Error Free Mitchell Log Multiplier and KOM Architecture, International Journal of VLSI design & Communication Systems, June 2014, 5(3).

[7] Bodasingi Vijay Bhaskar, Valiveti Ravi Tejesvi and Reddi Surya Prakash Rao, “Implementation of Radix-4 Multiplier with a Parallel MAC unit using MBE Algorithm”, International Journal of Advanced Research in Computer Engineering & Technology, Vol. 1, No. 5, July 2012.

[8] Hoang Q. Dao and Vojin G. Oklobdzija Application of Logical Effort on Delay Analysis of 64-Bit Static Carry-Lookahead Adder, 35th Annual Asilomar Conference on Signals, Systems and Computers Pacific Grove, California, 2001.

[9] Wang J S Kuo C N and Yang T H Low-power fixed-width array multiplier, International Symposium on Low Power Electronics Design, 2004, pp: 307–312.

[10] K.N. Vijeyakumar, V. Sumathy ,Sriram Komanduri and C. Chrisjin Gnana Suji, “Design of Low- Power High-Speed Error Tolerant Shift and Add Multiplier”, Journal of Computer Science, Vol. 12, No.7,pp:1839-1845,2011.

[11] K. Srishylam, Prof. Syed Amjad Ali, M.Praveena, “Implementation of Hybrid CSA, Modified Booth Algorithm and Transient power Minimization techniques in DSP/Multimedia Applications”, International Journal of Engineering Research and Applications, Vol. 2, No. 6, 2012, pp.861-871.

[12] P. A. Jadhav, V.B. Niranjane and Kalyani Ingole, “Design of High Speed Hybridized Multiplier”, International Journal of Research Studies in Science, Engineering and Technology, Vol. 2, No. 6, June 2015, pp. 110-112.

[13] CH.Rama Koti Reddy, K. Satyavathi, R.Raja Kishore, “A New VLSI Architecture to Increase the speed of computation (Modified Booth Algorithm)”, International Journal of Research in Computer and Communication Technology, Vol. 2, No. 9, September 2013.

[14] http://bias.csr.unibo.it/fvc2004/download.asp, FVC2004 Database reference, Third Fingerprint verification competition.