IMAGE AND SIGNAL FILTERING USING FIR FILTER MADE USING APPROXIMATE HYBRID HIGH RADIX ENCODING FOR ENERGY-EFFICIENT INEXACT MULTIPLIERS USING 4:2 COMPRESSORS

G. Sri Vidhya[1], S. Esther Ranjani[2], R.S. Haripriya[3], G. Ashweetha[4], Devareddy Priyanka[5]

[1]*Assistant Professor, Dept. of ECE, Panimalar Engineering College, Poonamalle, Tamilnadu, India*
[2,3,4,5]*UG Students, Dept. Of ECE, Panimalar Engineering College, Poonamalle, Tamilnadu, India*

---***---

Abstract - *Hybrid high base encoding for generating the partial products in signed multiplications that encodes the foremost important bits with the correct **radix-4** Approximate computing forms a design alternative that exploits the intrinsic error resilience of assorted applications and produces energy-efficient circuits with little accuracy loss. During this paper, we propose associate approximate coding and also the least significant bits with an approximate higher base **radix-2^k encoding**. The approximations are performed by rounding the high base values to their nearest power of 2. The projected technique is organized to attain the specified energy–accuracy tradeoffs. Compared with the correct radix-4 multiplier, the projected multipliers accomplish area savings.*

Finite Impulse Response filters are the foremost necessary element in signal processing and communication. Area optimization and speed are the key needs of Finite Impulse Response filters. Finite Impulse Response filter involves multiplications, additions and shifting operations. As the multiplier is the slowest element in the system, it will affect the performance of the FIR filter. In this paper, Finite Impulse Response filter based on modified Booth multiplier is designed and compared with typical filter, during which former reduces both area and delay.

1. INTRODUCTION

Approximate computing has emerged as a possible solution for the design of energy-efficient digital systems. In this paper, the design of inexact multipliers by applying approximations on the partial product generation is done. We propose a novel approximate hybrid high radix encoding with 4:2 compressor techniques[1].In the projected technique, the most significant Bits (MSBs) of the multiplicand are encoded using the radix-4 encoding.

The Least Significant Bits (LSBs) of the multiplicand are encoded using a radix-2^k (with k ≥ 4), whereas the 'k' is Least significant a result. It composes a promising design paradigm targeting energy-efficient systems by extraordinarily decreasing the power consumption of inherently error resilient applications. Approximate computing exploits the innate error tolerance of the assorted applications and deliberately relaxes the correctness of some computations, so as to decrease their power consumption and/or accelerate their execution. Approximations on the partial product generation[1] and approximations on their accumulation are synergistic, and may be applied unitedly so as to attain higher power reduction. In this paper, targeting the design of inexact multipliers by applying approximations on the partial product generation, we propose a novel approximate hybrid high radix encoding. In the planned technique, the most significant bits (MSBs) of the multiplicand are encoded using the radix-4 encoding, whereas the k least significant bits (LSBs) are encoded using a radix-2^k (with k ≥ 4). To simplify the increased complexity induced by the proposed hybrid encoding, the circuit for generating the partial products is approximated by altering accordingly its truth table. Hence, the number of the partial Products decreases considerably and simpler tree architectures are used for their accumulation, reducing the multiplier's energy consumption, area, and critical path delay.

1.1Hybrid high radix encoding

We propose and enable the application of hybrid high base encodings for the generation of energy-efficient approximate multipliers, exceeding the increased hardware complexity of very high radix encodings. The projected technique will be applied to any multiplier architecture and is reconfigurable, enabling the user to pick out the optimum per application energy–error tradeoff.

1.2Analytical error analysis

An analytical error analysis is conducted, showing that the output error of the proposed technique is finite and certain[2]. Such a rigorous error analysis leads to precise and a priori error estimation for any input distribution, without the need of time-consuming simulations.

We show that the proposed technique outperforms related state-of-the-art approximate signed multipliers in terms error values of less energy dissipation for comparable.

The proposed technique is applied into 16×16 bit multipliers operation in addition we implement the 4:2 Compressor to add the product in the multipliers operation. In this compressor techniques should reduce the addition operation[3]. Because of we are using 4:2 compressor, this technique reduce the overall area in the multiplier operation.

1.3PROPOSED SYSTEM

In this paper, we propose approximate high radix encoding technique for Inexact Multiplier using compressor, it is more energy efficient and lesser area occupation comparable with existing. In addition, area observed with the proposed compressor is smaller comparable with normal adder circuit. We propose an approximate hybrid high radix encoding for generating the partial products in signed multiplications that encodes the most significant bits with the accurate radix-4 encoding and least significant bits with an approximate higher radix -2^k encoding. The approximations are performed by rounding high radix value to their nearest power of two. In the proposed technique the multiplicand B is encoded using the approximate high radix encoding, generating B bar, and the approximate multiplication A.B bar is performed. In addition we replace the normal adder circuit into 4:2 Compressor techniques because of area reduction, in the existing once the area will be large because of normal adder circuit occupy larger area in the operation.

1.4 FIR FILTER

Finite Impulse Response (FIR)[6] filters are widely used in Digital Signal Processing (DSP) applications due to their stability and linear-phase property. In today scenario, low power consumption and less area are the most important parameter for the fabrication of DSP systems and high performance systems.

Nowadays, many finite impulse response (FIR) filter designs aimed at either low area or high speed or reduced power consumption are developed [7]. With the increase in area, hardware cost of these FIR filters are increasing.

This leads to design a low area FIR filter with the advantage of moderate speed performance. The implementation of an FIR filter needs three basic building blocks. They are Multiplication [9], Addition and Signal delay.

Multipliers consume the foremost amount of space in a FIR filter design. As the multiplier is the slowest element in the system, it will affect the performance of the FIR filter [8].

Here, Conventional multipliers are replaced by a modified Booth multiplier. Modified booth multiplier performs the computations using lesser number of adders and repetitious steps. As a result of that they cover lesser area as compared to the standard multiplier.

This is a very vital criterion because in the fabrication of chips and high performance system requires components that are as small as possible. Modified Booth is twice as fast as conventional multiplier. It produces only half the number of partial products (PPs) when compared with an ordinary binary multiplication. Modified Booth encoding (MBE) [9] scheme is known as the most effective Booth encoding and decoding scheme.

II.LITERATURE SURVEY:

Approximate computing has received vital attention as a promising strategy to decrease power consumption of inherently error tolerant applications.
In this paper, we tend to target hardware-level approximation by introducing the partial product perforation technique for designing approximate multiplication circuits. Approximate computing is an attractive design methodology to realize low power, high performance (low delay) and reduced circuit complexity by relaxing the requirement of accuracy. In this paper, approximate Booth multipliers are designed based on approximate radix-4 modified Booth encoding (MBE) algorithms and a regular partial product array that employs an approximate Wallace tree.

A multiplier exploitation the radix-4 (or modified Booth) algorithmic rule is incredibly economical because of the benefit of partial product generation, whereas the radix-8 Booth multiplier is slow because of the complexity of generating the odd multiples of the multiplicand. In this paper, this issue is alleviated by the applying of approximate designs.

III. MODULE EXPLANATION:

3.1 APPROXIMATE HYBRID HIGH RADIX MULTIPLIERS

The high radix encodings require complex encoding and partial product generation circuits negating thus the benefits of the partial products reduction [4]. The circuit complexity is presented encoding and the performed approximations for simplifying its proposed hybrid high radix. In the proposed technique, the multiplicand *B* is encoded using the approximate high radix encoding, generating ˜*B*, and the approximate multiplication *A* • ˜*B* is performed [5]. Finally, its adaptation on inexact 16-bit multipliers is described, and a qualitative analysis is conducted, targeting to estimate the potential area gains.

In the proposed hybrid high radix encoding, B is divided in two groups: the MSB part of n-k bits and the LSB part of k bits. The MSB part is encoded using the radix-4 (modified Booth) encoding while the LSB part is encoded with the high radix-2^k encoding. The hybrid high radix encoding is characterized by increased logic complexity, due to the high radix values of y0R2k that are not power of two, and thus, an approximate version is proposed. However, so as to retain high accuracy, the radix-4 encoding of the MSB is performed accurately.

In particular, in the approximate encoding, all the values that are not power of two and the k – 4 smallest powers of two as well, are rounded to the nearest of the 4 largest powers of two or 0, so that the sum of all the values of the approximate Digit is y0^R2k is 0. We choose to keep only the four smallest power of two, so that the radix-2k encoding circuit requires only about area in comparison with the accurate radix-4 encoder.

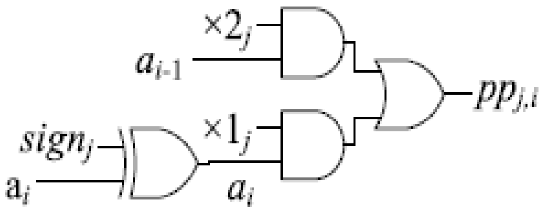


Fig 1: accurate radix-4 encoding

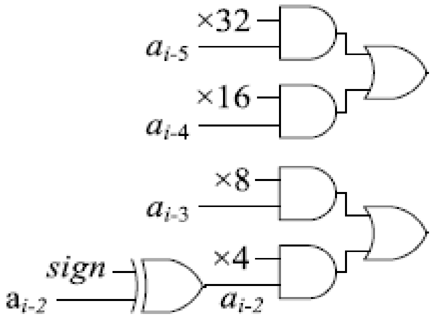


Fig 2: accurate radix -64 encoding

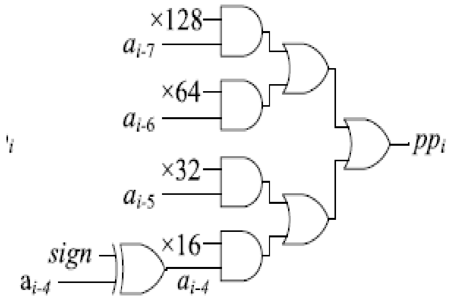


Fig 3: accurate radix-128 encoding

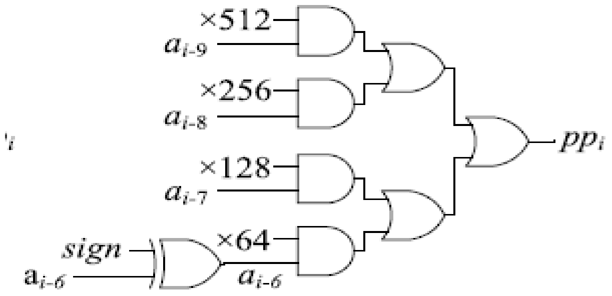
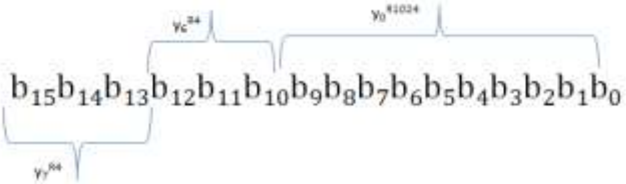


Fig 4: accurate radix-1024 encoding

The effectiveness of the approximate hybrid high radix encoding technique is explored with its application to 16-bit signed numbers, for k = 6, 8, 10, namely the LSBs are encoded using the radix-64, radix-256, and radix-1024 encoding, respectively. We are implementing the radix-1024 in 16-bit multiplier circuit.


y_C^{R4}
y_0^{R1024}
$b_{15}b_{14}b_{13}b_{12}b_{11}b_{10}b_9b_8b_7b_6b_5b_4b_3b_2b_1b_0$
y_Y^{R4}

Example 1: radix-1024 encoding the 16-bit multiplicand **B** grouped as

Similarly, the no powers of two are rounded to ±64, ±128, ±256, or ±512, and the smallest powers of two (±1, ±2, ±4, ±8,±16,±32) are rounded to 0 or ±64. The encoder's inputs are the bits $b9$, $b8$, . . . , $b0$ the approximate radix-1024 digit is $y_{0^\wedge}{}^{R1024}$∈ {0,±64,±128,±256,±512}, and the output signals that define $y_0{}^{\wedge R1024}$ are sign, ×64, ×128, ×256, ×512

3.1.1 Partial Product Generation

In this planned hybrid encoding, the n − k MSBs of multiplicand B are encoded with the accurate radix-4 encoding, whereas the k LSBs are encoded with associate degree approximate radix-2kencoding. The implementation of the partial product accumulation is chosen by the designer. Overall, the multiplication circuit consists of stages of operand hybrid radix encoding, partial product generation, partial product accumulation, and final addition.
The projected approximate multipliers are RAD2k named, showing the chosen approximate high base encoding, e.g., RAD64, RAD256, and RAD1024.The advantage of the approximate hybrid high base multipliers is their simple logic, resulting in fast operation and low power performance. The partial product generator deliver low space, and the reduction of the number of the partial products.

2.1.2 Hardware requirements

Field-Programmable Gate Array (FPGA)

A field-programmable gate array (FPGA) is an integrated circuit designed to be organized by the customer or designer after manufacturing—hence "field-programmable". The FPGA configuration is usually specified using a hardware description language (HDL), almost like that used for an application-specific integrated circuit (ASIC) (circuit diagrams were previously accustomed specify the configuration, as they were for ASICs, however this can be increasingly rare).

FPGAs will be accustomed implement any logical operate that an ASIC might perform. The ability to update the functionality after shipping, partial re-configuration of the portion of the design and the low non-recurring engineering costs relative to an ASIC design (not withstanding the generally higher unit cost), offer advantages for many applications.

3.1.3 SOFTWARE REQUIREMENTS:

- **Verification Tool**
  - ModelSim 6.4c
- **Synthesis Tool**
  - Xilinx ISE 14.5

IV Experimental analysis

SIMULATION RESULTS

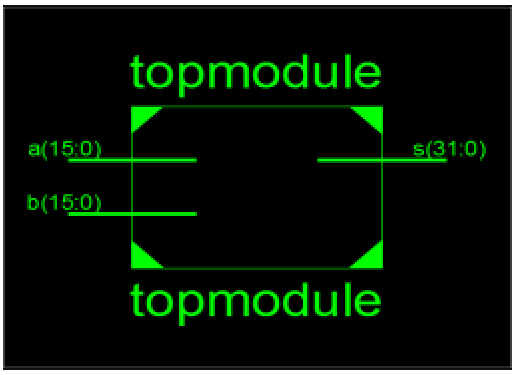
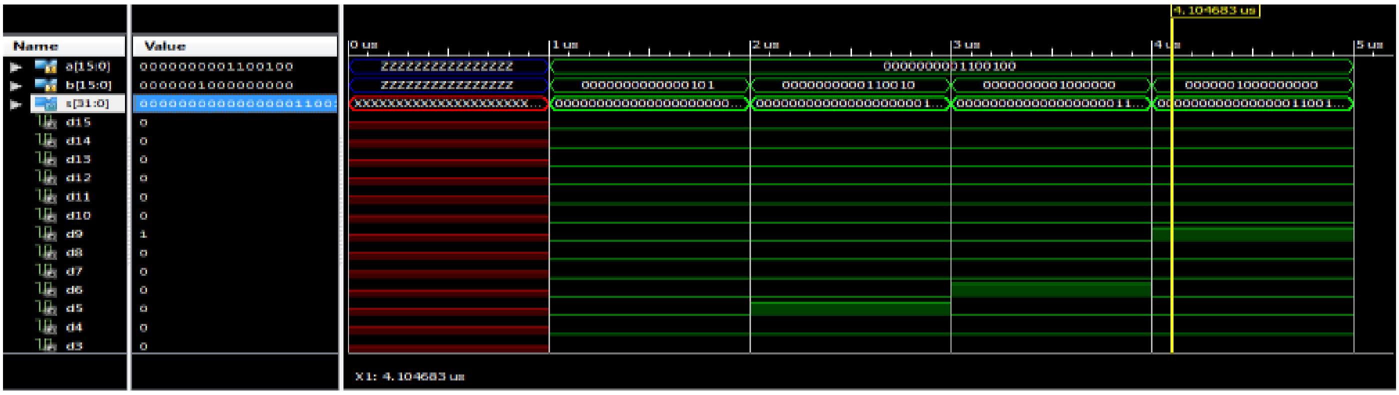
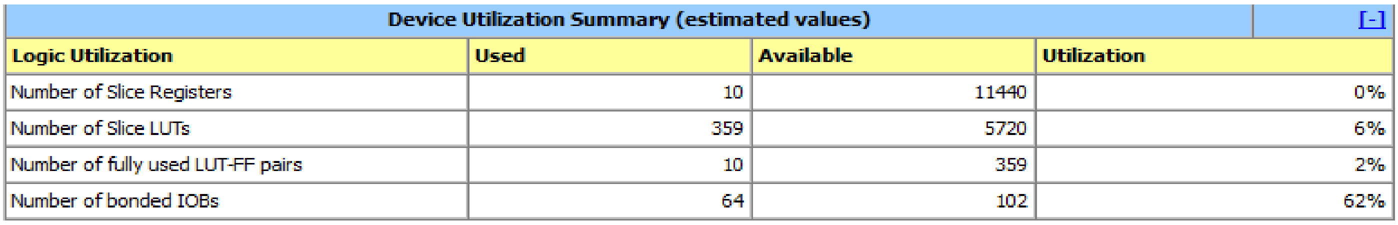


Fig 5: Block diagram of inexact multiplier using compressor



Name	Value
a[15:0]	0000000001100100
b[15:0]	0000001000000000
s[31:0]	000000000000000000001100
d15	0
d14	0
d13	0
d12	0
d11	0
d10	0
d9	1
d8	0
d7	0
d6	0
d5	0
d4	0
d3	0

Fig 6: Output waveform of inexact multipliers using Compressor



Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	10	11440	0%
Number of Slice LUTs	359	5720	6%
Number of fully used LUT-FF pairs	10	359	2%
Number of bonded IOBs	64	102	62%

Fig 7: Design summary of inexact multiplier using Compressor

Fig 6 & 7 shows the simulation output of proposed hybrid high radix encoding for inexact multipliers using 4:2 Compressor technique.

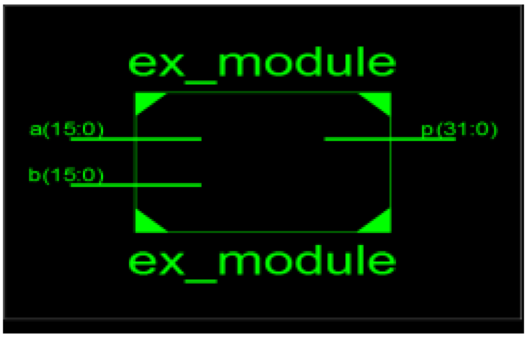


Fig 8: Block diagram of existing multiplier

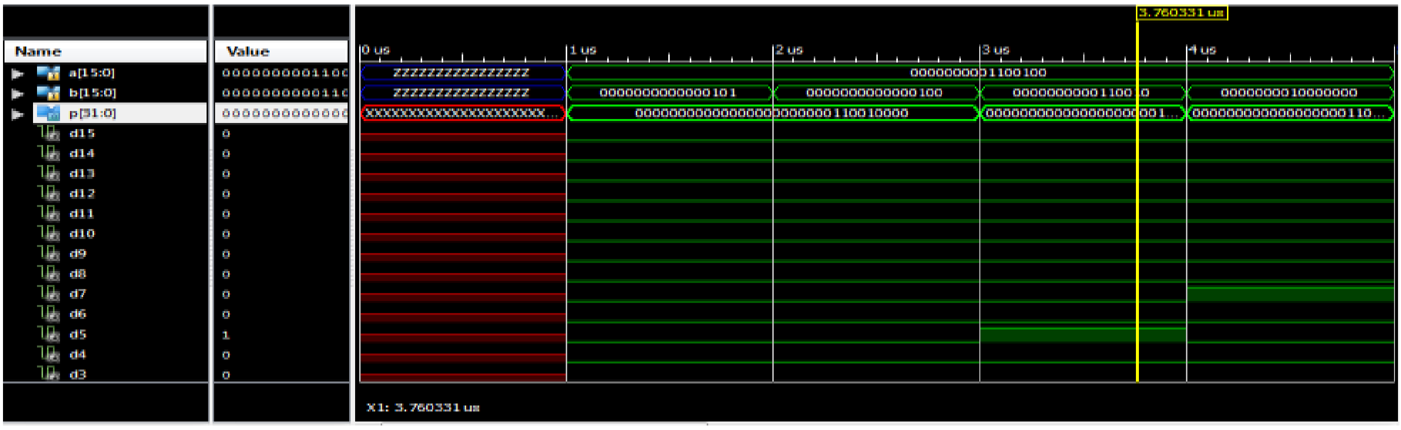
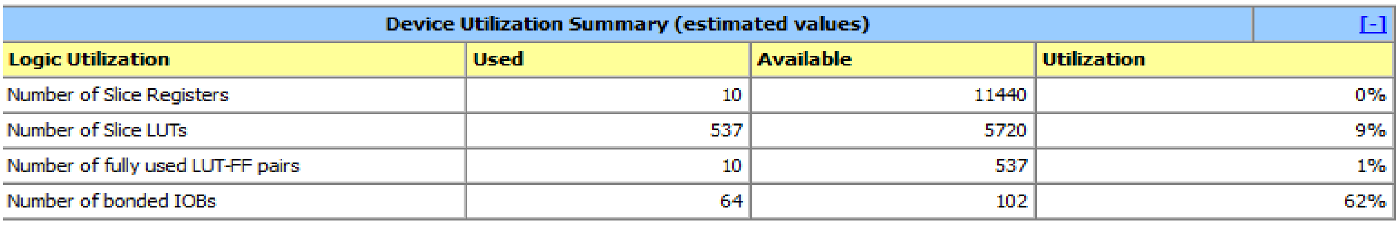


Fig 9: Output waveform of inexact multiplier using adder



Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	10	11440	0%
Number of Slice LUTs	537	5720	9%
Number of fully used LUT-FF pairs	10	537	1%
Number of bonded IOBs	64	102	62%

Fig 10: Design summary of inexact multiplier using adder

Fig 9 & 10 shows the existing hybrid high radix encoding for inexact multipliers.

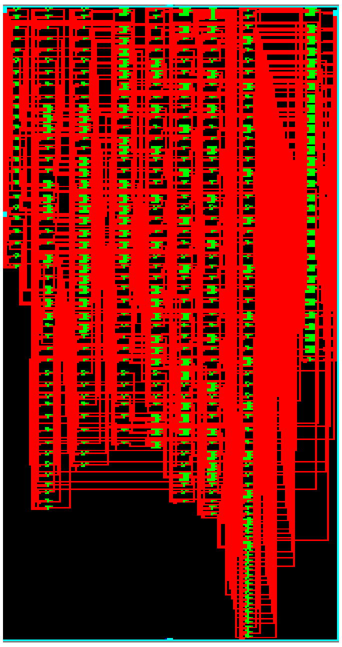


Fig 11: RTL schematic of proposed multipliers

$$y(n) = \sum_{k=0}^{N-1} b_k x(n-k)$$

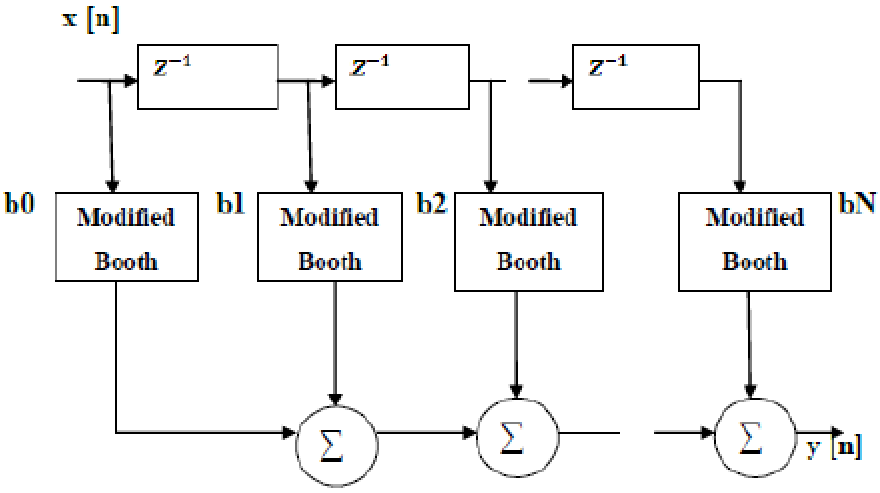


Fig 12: Modified Booth Multiplier Based FIR Filter

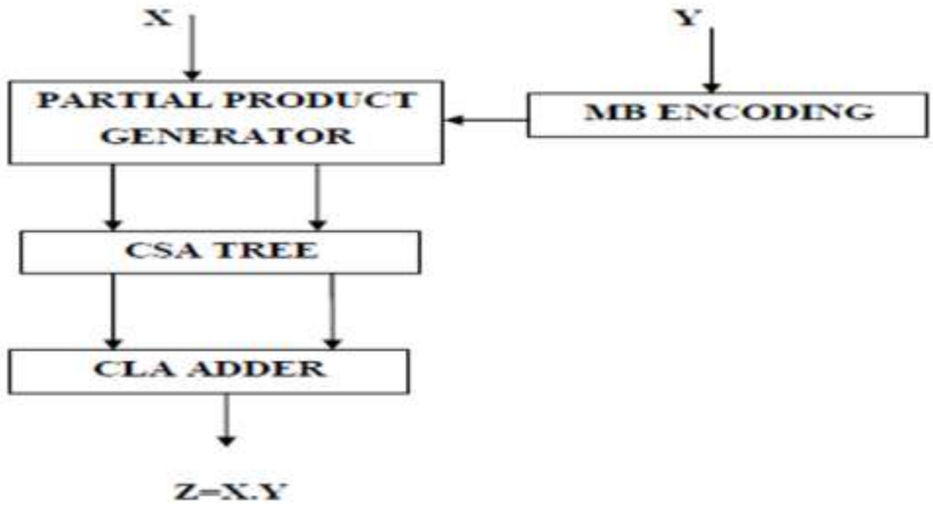
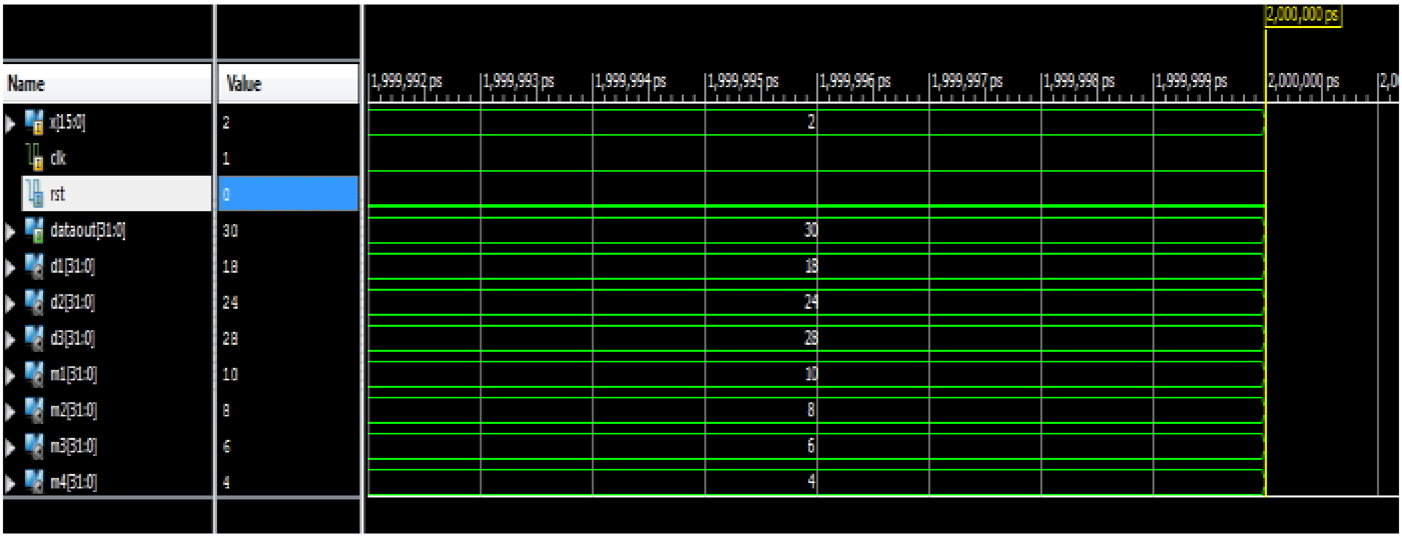


Fig 13:Architecture of the modified Booth multiplier



Name	Value
x[15:0]	2
clk	1
rst	0
dataout[31:0]	30
d1[31:0]	18
d2[31:0]	24
d3[31:0]	28
m1[31:0]	10
m2[31:0]	8
m3[31:0]	6
m4[31:0]	4

Fig 14: Output of FIR Based Booth Multiplication

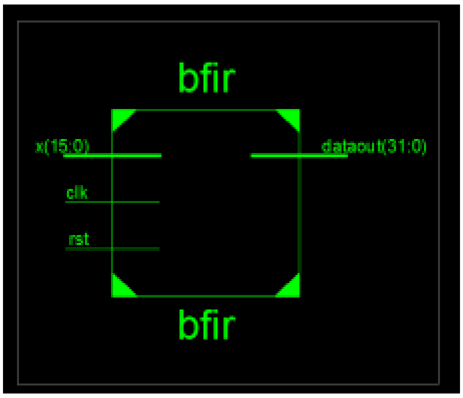


Fig 15: Block diagram of FIR

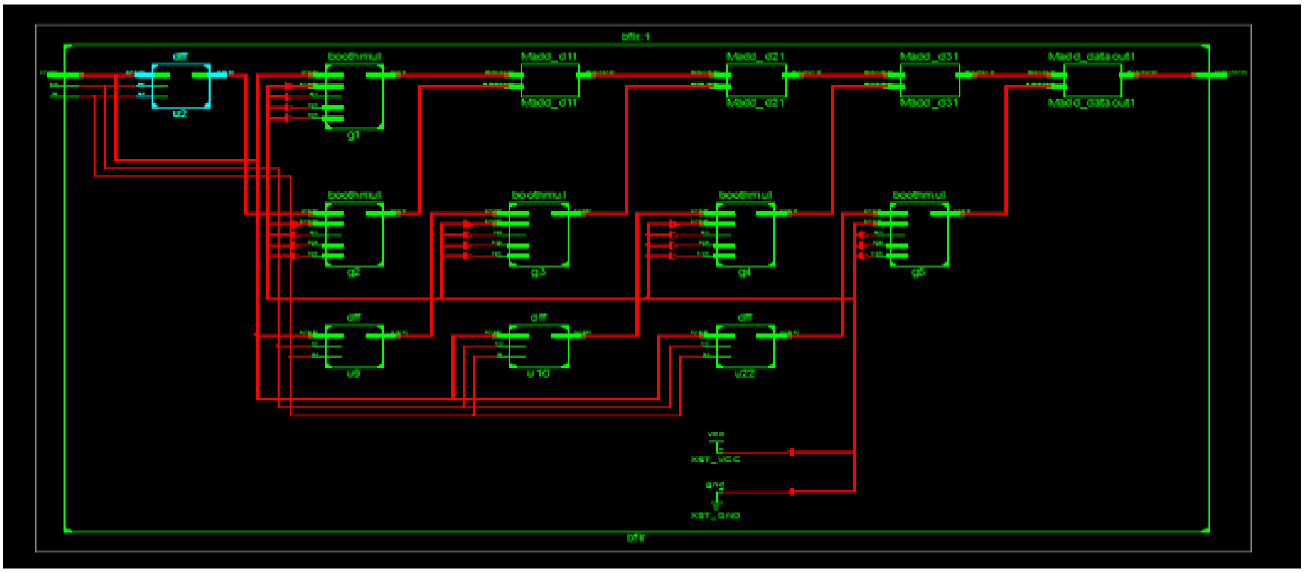


Fig 16: SCHEMATIC VIEW

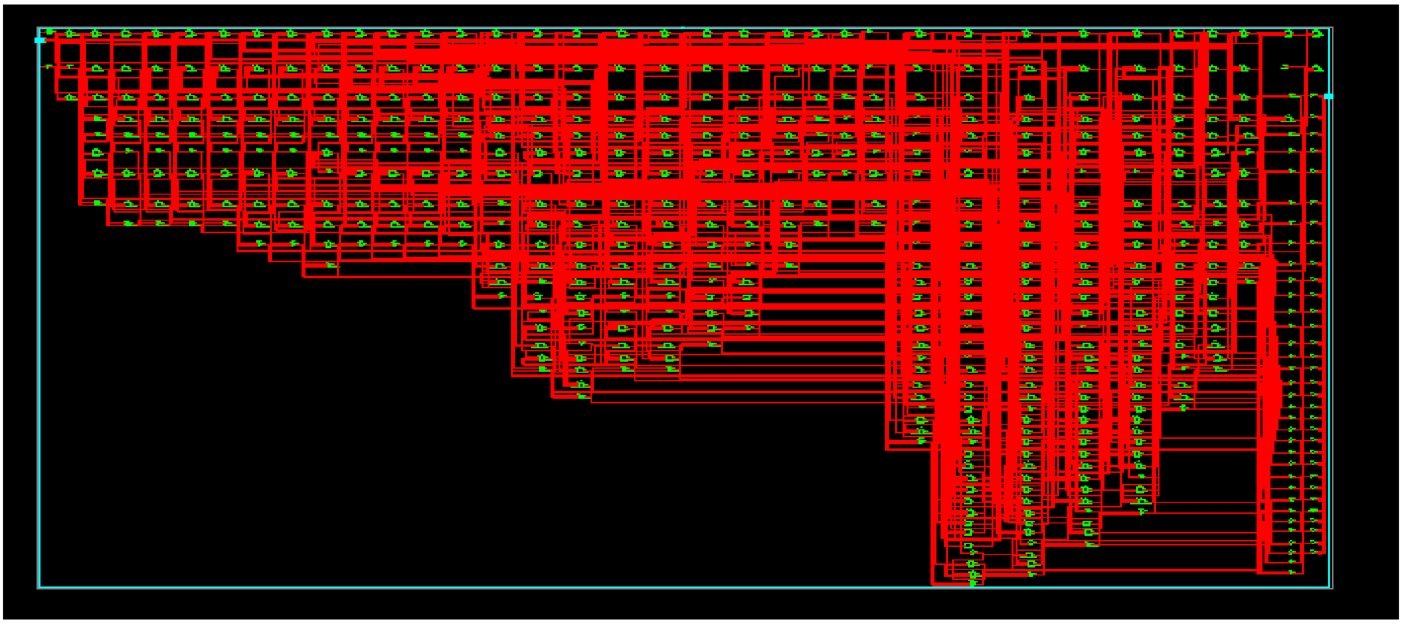
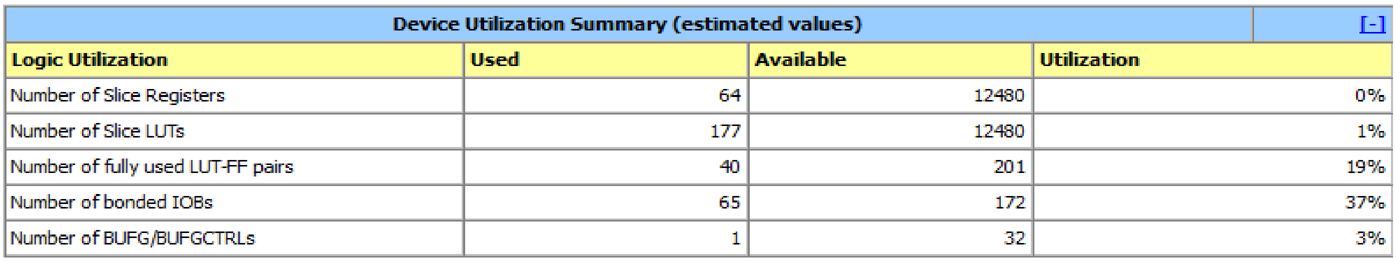


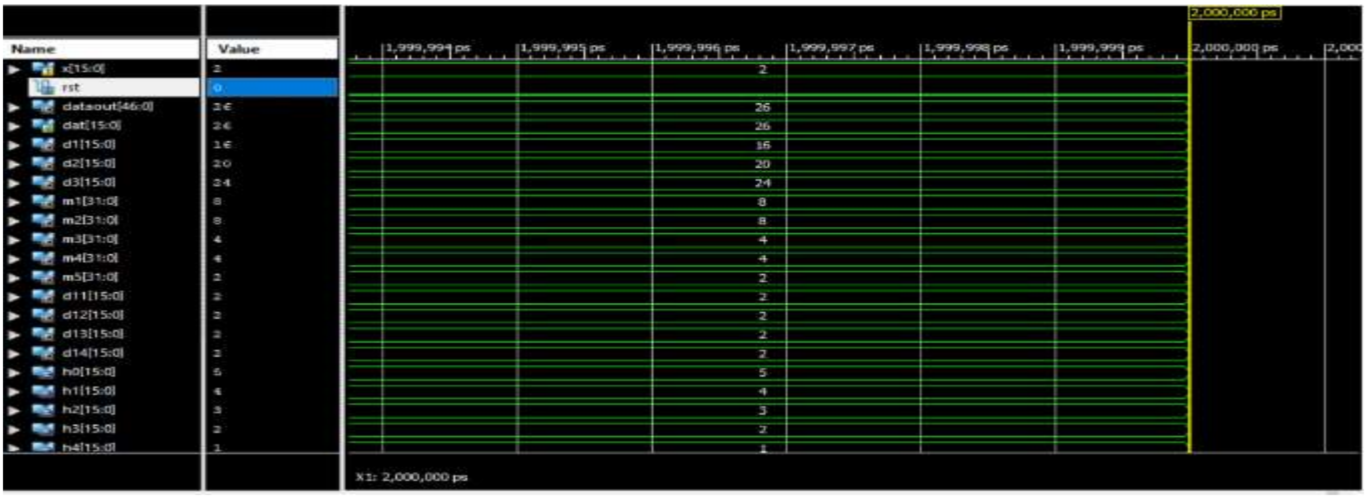
Fig 17: TECHNOLOGY SCHEMATIC



Device Utilization Summary (estimated values) [-]

Logic Utilization	Used	Available	Utilization
Number of Slice Registers	64	12480	0%
Number of Slice LUTs	177	12480	1%
Number of fully used LUT-FF pairs	40	201	19%
Number of bonded IOBs	65	172	37%
Number of BUFG/BUFGCTRLs	1	32	3%

Fig 18: Design summary



```
Name            Value
x[15:0]         2
rst             0
dataout[46:0]   26
dat[15:0]       26
d1[15:0]        16
d2[15:0]        20
d3[15:0]        24
m1[31:0]        8
m2[31:0]        8
m3[31:0]        4
m4[31:0]        4
m5[31:0]        2
d11[15:0]       2
d12[15:0]       2
d13[15:0]       2
d14[15:0]       2
h0[15:0]        5
h1[15:0]        4
h2[15:0]        3
h3[15:0]        2
h4[15:0]        1
```

Fig 19:FIR filter with inexact multiplier output

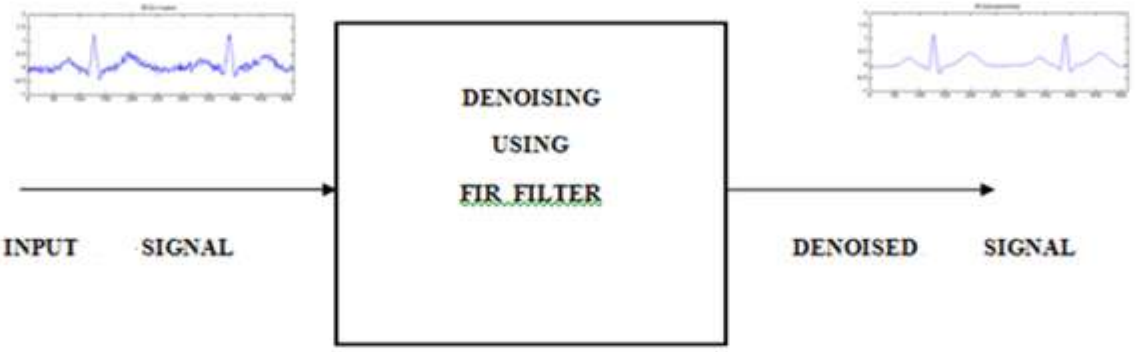


Fig 20: One of the Application of FIR Filter

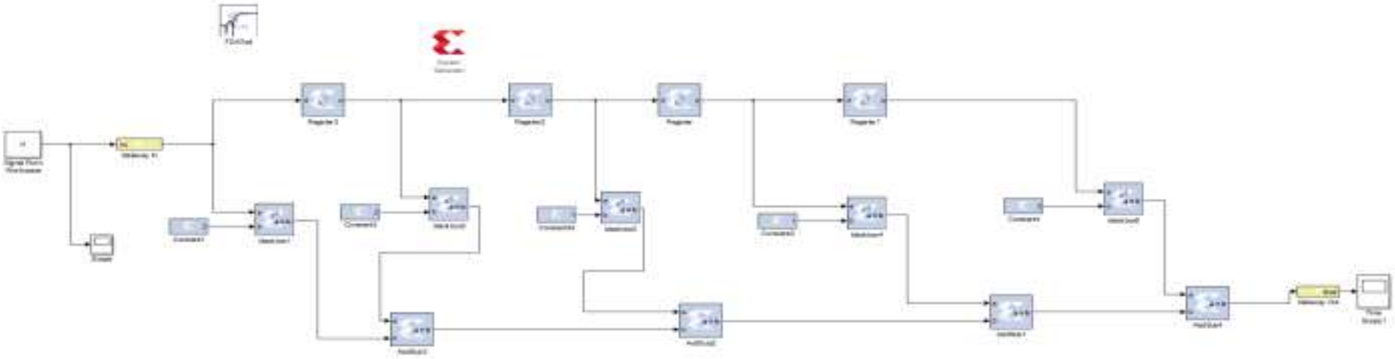


Fig 21: Block diagram

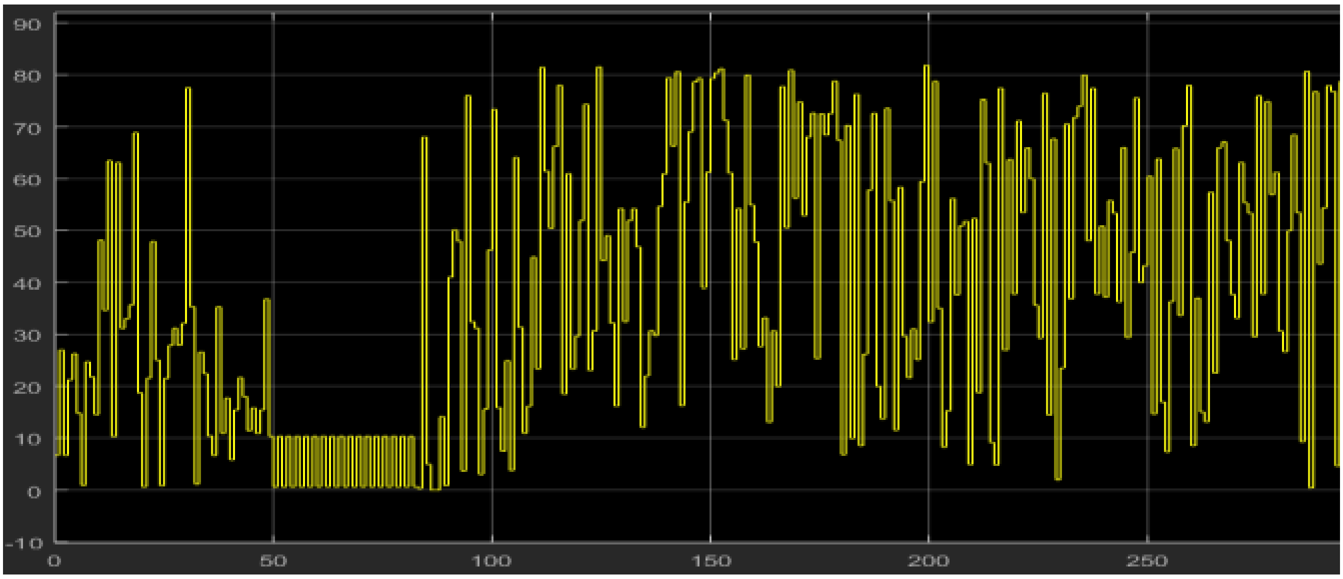


Fig 22: EEG signal input with noise

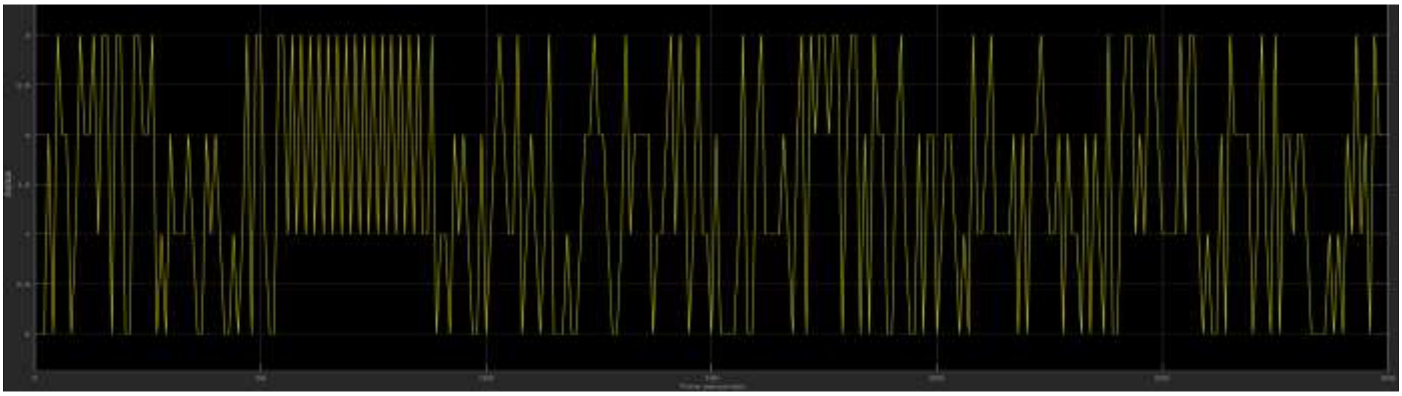


Fig 23: signal EEG output with noise removed

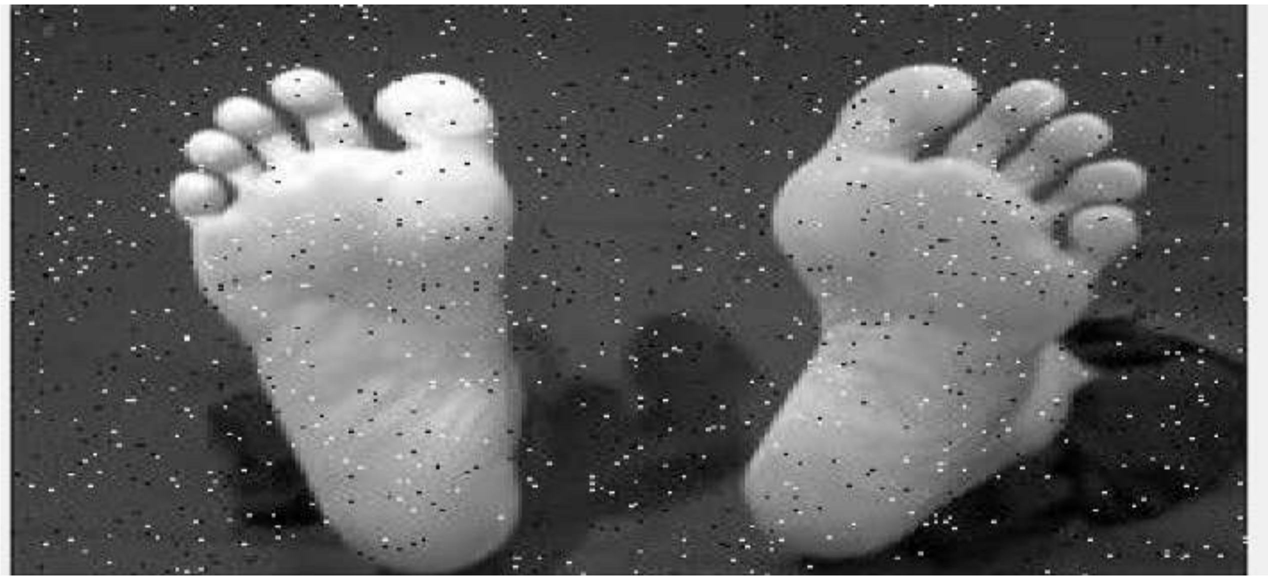


Fig 24: Image with salt and pepper noise as input

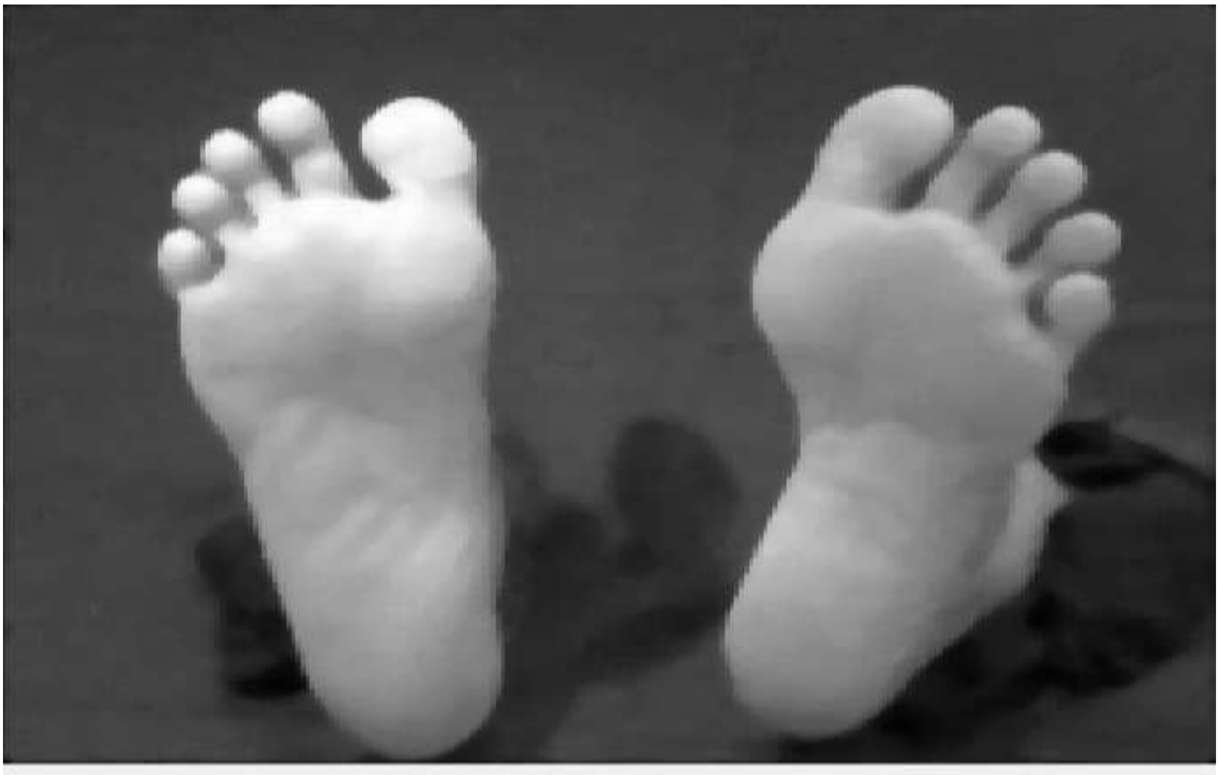


Fig 25: Image output without noise

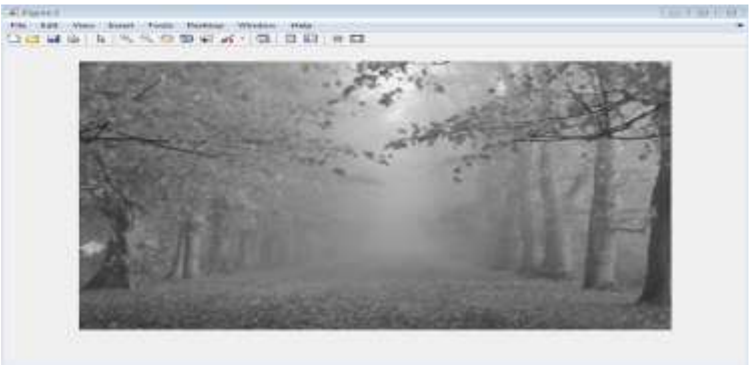


Fig 26: Input image with noise

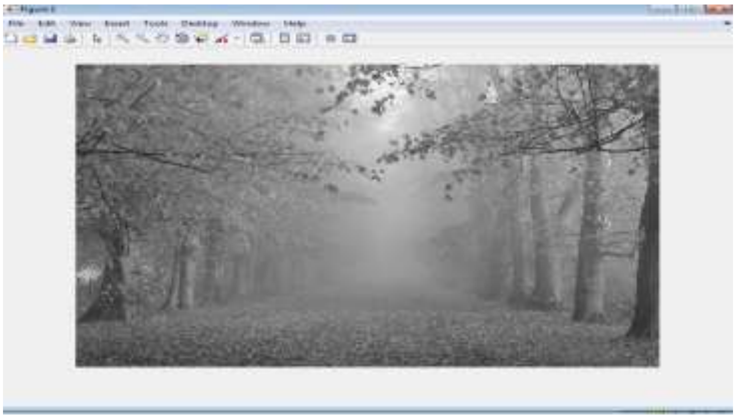


Fig 27: After moving into fir filter, the noise in image is reduced

IV CONCLUSIONS

In this paper, we propose an approximate hybrid high radix encoding for generating the partial products of a signed multiplier. The MSBs of the multiplicand are encoded with the accurate radix-4 encoding, while its k LSBs are encoded-2k with an approximate high radix encoding with k being a configuration parameter that adjusts the tradeoff between accuracy and energy consumption. The error of the proposed technique follows a Gaussian distribution with near zero average[2]. Compared with state-of-the-art inexact multipliers, the proposed ones constitute better approximate design alternative, outperforming them in both energy consumption and accuracy[5].Finally, the proposed technique is scalable, delivering higher energy savings for the same error, as the multiplier's size decreases. With the help of this multiplier we design FIR filter[8]. To this FIR Filter we give distorted EEG signal and an image with salt and pepper as input. And in output we obtain noise free output.

The signal and image processing is achieved through simulink tool in MATLAB.

REFERENCES

1) "Design-Efficient Approximate Multiplication Circuits Through Partial Product Perforation", G. Zervakis, K. Tsoumanis, S. Xydis, D. Soudris, and K. Pekmestzi

2) Design of Approximate Radix-4 Booth Multipliers for Error Tolerant Computing", W. Liu, L. Qian, C. Wang, H. Jiang, J. Han, and F. Lombardi

3) "Design of Low-Error Fixed-Width Modified Booth Multiplier", K.-J. Cho, K.-C. Lee, J.-G. Chung, and K. K. Parhi

4) High-Accuracy Fixed-Width Modified Booth Multipliers for Lossy Applications", J. P. Wang, S. R. Kuang, and S. C. Liang

5) Approximate Radix-8 Booth Multipliers for Low-Power and High-Performance Operation",H. Jiang, J. Han, F. Qiao, and F. Lombardi

6) Performance Analysis of FIR filter using Booth Multiplier D. Jaya Kumar, Dr.E. Logashanmugam..

7) Modified Booth Multiplier Based Low-Cost FIR Filter Design Shelja Jose, Shereena Mytheen.

8) Low power designing of FIR filters -- Sarita Chouhan1, Yogesh Kumar.

9) Design and Implementation of an Efficient Modified Booth Multiplier using VHDL Kavita, Jasbir Kaur.