ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

FPGA Design of a Reduced Complexity Sphere Decoder for Wireless Applications

[I]M.Sheeba, [II]K.Monisha
[I]Final Year, M.E. – VLSI DESIGN, Sethu Institute of Technology, Kariapatti, India
[II]Asst. Professor, ECE Dept., Sethu Institute of Technology, Kariapatti, India

Abstract
Multiple-input and multiple-output (MIMO) technology has been widely applied in wireless communications since it offers significant increases in data throughput a link range without additional bandwidth or increased transmit power. In this project, a MAP Algorithm for Multi input Multi Output (MIMO) Sphere Decoder for wireless applications is proposed. FEC (Forward Error Correction) Algorithm for sphere decoder to reduce its error rate and its VLSI architecture is also proposed for the iterative MIMO receiver. The deeply pipelined architecture employs the optimized hybrid enumeration search for the best child node to estimate efficiently. By adding the counterhypotheses in parallel with other candidates, the proposed iterative MIMO detector improves the detection performance significantly with low detection latency. Compared with previous low-complexity techniques, this work reduces gate counts when compared with existing method and requires only a one-line-buffer memory. This proposed bi-orthogonal based sphere decoder system is designed using Verilog HDL, simulated using Modelsim Software and synthesized using Xilinx Project Navigator.

Keywords
FEC, MIMO, Power, Sphere, Latency and Hybrid.

I. Introduction

Multiple-input and multiple-output (MIMO) technology has been widely applied in wireless communications since it offers significant increases in data throughput and link range without additional bandwidth or increased transmit power. By incorporating MIMO with bit-interleaved coded modulation with iterative detection and decoding (BICM-IDD), the channel capacity can be approached [1] at the cost of much higher complexity and lower throughput compared with non-iterative schemes. Thus, it is very important to develop a high-speed iterative detector to meet the increasing demand for gigabit-per-second wireless systems such as the IEEE 802.11ac wireless local area network (WLAN) and 3GPP LTE Advanced. Due to its practical importance, the very large scale integration (VLSI) design of soft-input soft-output (SISO) detectors has recently received a lot of attention. The first reported implementation of a SISO MIMO detector is based on the minimum mean square error parallel interference cancellation (MMSE-PIC) algorithm [2], but it cannot fully exploit the spatial diversity provided by MIMO. To overcome this limitation, implementations of SISO single tree search (STS) sphere decoding (SD) [3], [4] are presented, which have max-log maximum a posteriori (MAP) performance. However, like other depth-first tree-search algorithms, it suffers from variable throughput and complexity depending on the signal-to-noise ratio (SNR).

More recently, a novel SISO detection algorithm based on trellis search and its VLSI architecture has been proposed in [5], which provides a peak throughput of 1.7 Gbit/s, but it consumes large silicon area and is hard to support high order modulation [e.g., 64 quadrature amplitude modulation (QAM)]. Fixed-complexity SD (FSD) is a breath-first tree-search algorithm previously proposed for hard-output MIMO detection. It is capable of providing near maximum likelihood (ML) detection performance with fixed and low complexity [6]. A highly efficient silicon implementation of FSD is reported in [7], which can achieve a 1.98-Gbit/s detecting throughput with the parallel multistage VLSI architecture. It is very attractive to extend the hard-output base architecture to support iterative MIMO detection.

II. Proposed Maethod

The proposed VLSI architecture of SISO FSD is shown in Fig.1.This architecture consist of processing elements, Candidate generation unit (CGU) and (LCU) log likelihood ratio calculation unit modules.

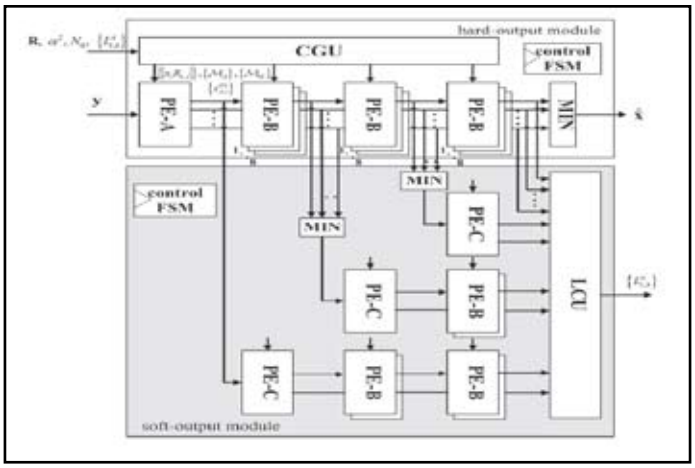

Fig. 1: Proposed VLSI Architecture of SISO FSD

A. Processing element (PE-A) Architecture:

By employing the ORVD, the *MP* computation (i.e., $MP(s_i)$, $MP(s_i+1)$) in two adjacent levels can be conducted in parallel with $R_{i,i+1}$ being zero for $i = 1, 3, \ldots, 2Nt-1$ [7].

As a consequence, the number of processing element (PE) stages is reduced by half compared to those pipelined detectors using traditional real-value decomposition [10]. The architecture supports both hard outputs and soft outputs. The hard-output module generates the original hard-output FSD candidate list L in which the best path with the minimum *MP* is found. The soft-output module generates an expanded list $L+$ by employing the PC-A scheme and calculates the LLRs based on the union of the two lists $L \cup L+$. The PEs in our design is divided into three types: PE-A, PE-B, and PE-C. PE-A is located in the first stage where multiple child nodes are expanded. PE-B performs the single expansion in their remaining three stages. PE-C in the soft-output module adopts the bit-flipping strategy to add

the counterhypotheses to the expanded list $L+$. To identify the partial MAP node among the L, the minimum (MIN) search block at the soft-output module is needed to select the node with the smallest LP. With $\mathbf{L}^{m2}_{Nt-1}$= [7, 7, 5, 5, 3, 3, 1, 1], the number of candidates in L is N^{L}_{cand}= 32. In the hard-output module, we instantiate eight PE-Bs at each stage where eight nodes can be processed simultaneously, and thus, four cycles are needed to complete the processing of all the candidates in L. The candidate generation unit (CGU) is adopted to generate all possible values of $|R_{i,j}s_j|$ which are shared by the $MP(s_i)$ calculations at the same level. Additionally, the $MA(s_i)$ and $M_{si}(s_i)$ of all possible symbols are also precomputed to further enhance the hardware sharing.

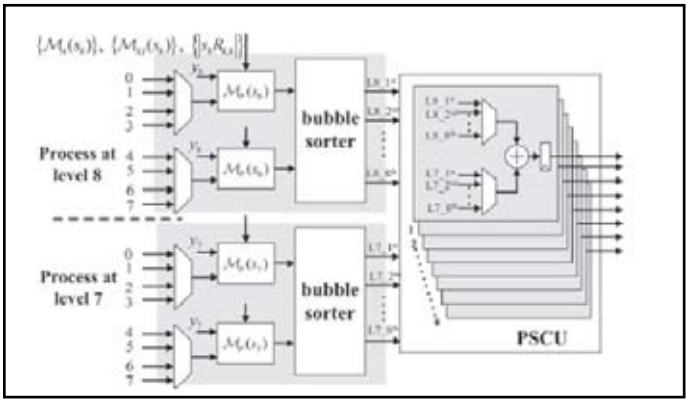

Fig. 2: Architecture of PE-A

Moreover, the best node $s^{(1)}_{A,i}$ with the minimum MA at each level is also identified and buffered in CGU according to the sign of $L^{A}_{i,b}$ which avoids full sorting of the set {MA(s_i)}.The LLR calculation unit (LCU) in the last stage calculates the LLRs of each transmitted bit according to (4) based on the candidate lists L and $L+$.

B. Processing element (PE-B) Architecture:

PE-B is used to implement the single expansion where only the best node estimate is selected and preserved using the proposed OHE.

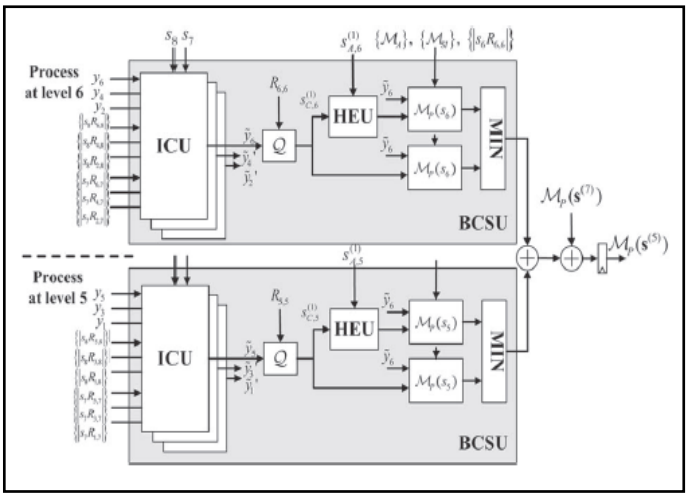

Fig. 3: Architecture of PE-B

As shown in Fig.3, the interference cancellation unit (ICU) in PE-B computes $\tilde{y}$ in (1) to eliminate the inter antenna interference introduced by previously detected symbols. To enumerate the best child node $s^{(1)}_{C,i}$ with the minimum MC, a quantization step Q is required to find the symbol which is next to $\tilde{y}i/Ri,i$. The HE unit (HEU) chooses sCA,i according to step 3) of the OHE method. The MIN block compares the MP of $s^{(1)}_{C,i}$ and sCA,i, and then selects the node with smaller MP.

C. Processing element (PE-C) Architecture:

In PE-C architecture, Best Child node selection unit is employed. This unit is used to select the best child nodes among the set of available child nodes with respect to parent or root node. Fig.4 shows the architecture of PE-C, which implements the PC-A scheme.

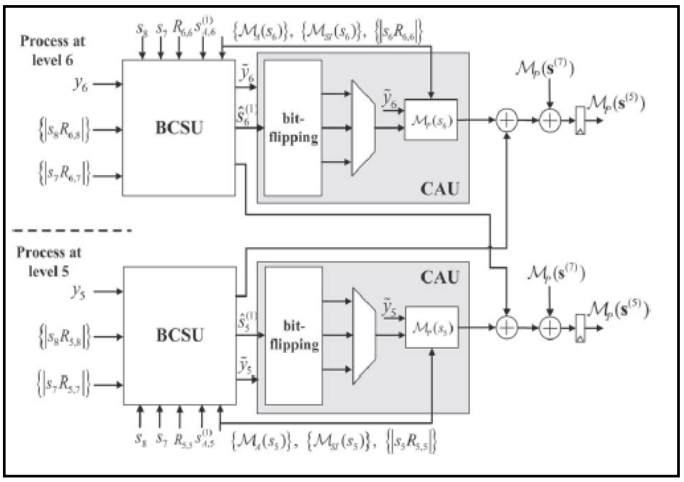

Fig. 4: Architecture of PE-C

The best child estimate selection unit (BCSU) receives the partial MAP node $\hat{s}^{PMAP}_{i+1}$ and finds its best child estimate $\hat{s}^{(1)}_{i}$ by employing the OHE, just the same as it is in PE-B. The candidate adding unit (CAU) uses bit-flipping strategy to add three sibling nodes of $\hat{s}^{(1)}_{i}$, which feed forward to a multiplexer, and only one of them is selected per cycle for MP computation. The serial computation method saves the number of MP computation blocks in CAU by 66.7% and reduces the number of PE-Bs following in the subsequent stages compared to the parallel method, without impacting the throughput of the whole architecture.

D. Candidate Generation Unit (CGU):

A soft-output sphere decoder typically consists of a list generator that finds a set of candidate symbol vector, and a log-likelihood (LLR) generator that calculates the soft-output bit value for the MIMO channel decoder

$$Mp(s_i) = Mc(s_i) + Ma(s_i) - Msi(s_i) \quad (1)$$

where $Msi(s_i) = \alpha^2 |s_i|$ and $\alpha\alpha$ is the regularization parameter.

$$Mc(s_i) = |y_i - \sum_{j=i}^{2Nt} R_{ij} s_i \,|^2 = |\tilde{yi} - R_{ii} s_i|^2 \quad (2)$$

$$Ma(s_i) = -\frac{1}{2} \sum_{b=1}^{Mc/2} x_{i,b} N_0 L^{A}_{i,b} \quad (3)$$

Where $x_{i,b} \in \{+1, -1\}$ denotes the bth bit.

E. LLR Calculation Unit (LCU)

The LLR (log-likelihood ratio) Calculation Unit is used to calculate the likelihood's between the two or more data/models. In LCU unit, MAP (maximum a posteriori) algorithm is proposed to reduce

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

the hardware complexity and FEC (Forward Error Correction) is also proposed to reduce its bit error rate during transmission. The proposed LCU architecture consists of cyclic shift register, XOR matrix, majority gate and control unit. The cyclic shift register will send the inputs to each flipflops and the outputs are fed back in a cyclic manner. In XOR Matrix, the values are taken from the shift register. And these values are XOR one by one. And Parity check sum is performed. That is counting the number of zeros and ones. If number of ones is greater than zeros error correction is performed. Otherwise the bits are under gone cyclic shift. These Processes are done in Majority Gate. And it can be done until all the bits are evaluated.

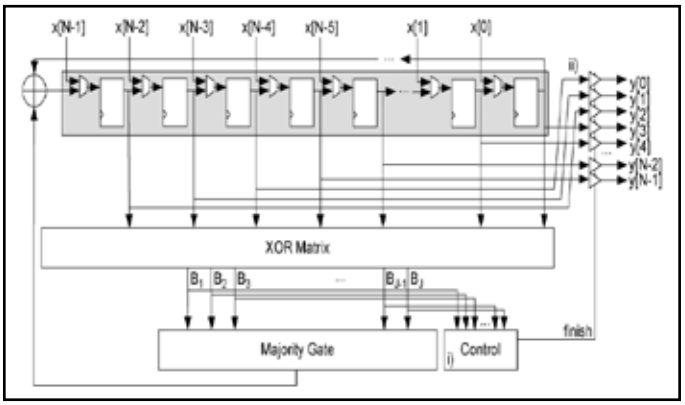

Fig. 5: Proposed Decoder Architecture

The control unit manages the detection process. It uses a counter that counts up to three, which distinguishes the first three iterations of the ML decoding. In these first three iterations, the control unit evaluates the {Bj} by combining them with the OR1 function. This value is fed into a three-stage shift register, which holds the results of the last three cycles.In the third cycle, the OR2 gate evaluates the content of the detection register. When the result is "0," the FSM sends out the finish signal indicating that the processed word is error-free. In the other case, if the result is "1," the ML decoding process runs until the end.

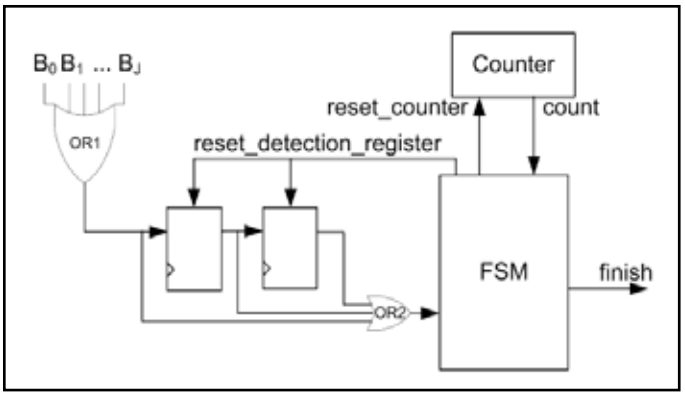

Fig. 6: Architecture of Control Unit

The FEC transmitter is shown in fig 8. The generator receives a data word and generates parity check bits along with the data word. Each of the parity-check bits handles 3 out of 4 bits of the data word using forward error correction code. The forward error correction code for transmitter is given below.

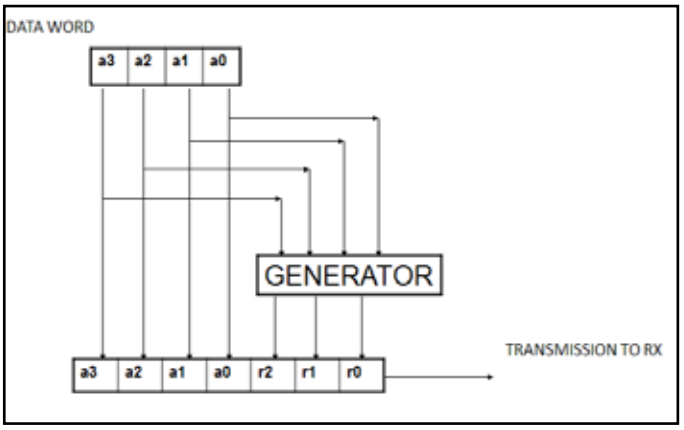

Fig. 7: FEC Transmitter

The forward error correction code is given below:

- r0=a2+a1+a0 modulo-2
- r1=a3+a2+a1 modulo-2
- r2=a1+a0+a3 modulo-2

The FEC receiver is shown in fig 9 that consists of bits from the transmitter and the bits are given to the checker and for each bit the syndrome value will be estimated using the code given below. The correction logic analyzer will check for error according to the calculated syndrome values using the Table I shown below.

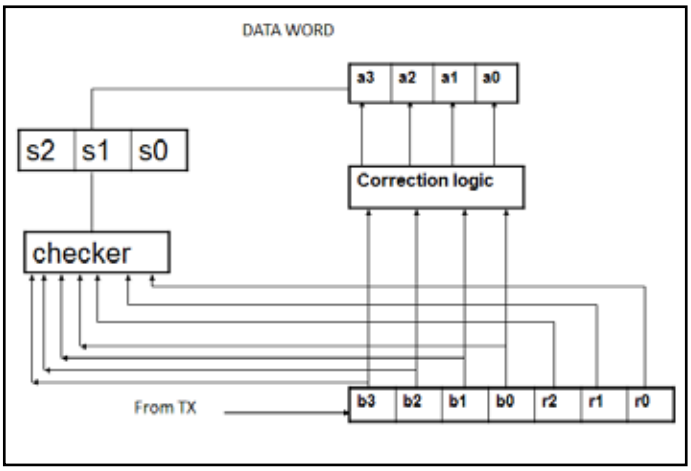

Fig. 8: FEC Receiver

Syndrome estimation:

- $S_0=b_2+b_1+b_0+q_0$ modulo-2
- $S_1=b_3+b_2+b_1+q_1$ modulo-2
- $S_2=b_2+b_1+b_0+q_2$ modulo-2

Table 1 : Logical design made by correction logic analyzer in FEC receiver

Syndrome	000	001	010	011	100	101	110	111
Error	None	q0	q1	b2	q2	b0	b3	b1

MAP Algorithm:

MAP algorithm is used to find the error in multi-bits and it also reduces the hardware complexity.

III. Simulation Results

The proposed sphere decoder architecture is designed using verilog HDL, simulated using modelsim software and synthesized using

Xilinx project navigator. The simulated waveform is illustrated in fig.9 and its RTL schematic view is displayed in fig.10.

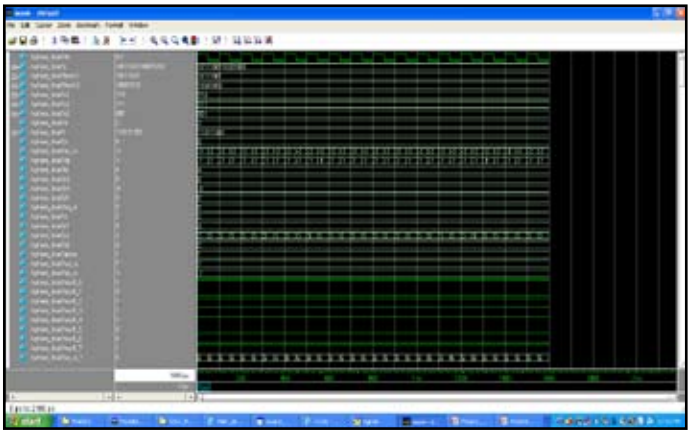

Fig. 9: Waveform Simulation

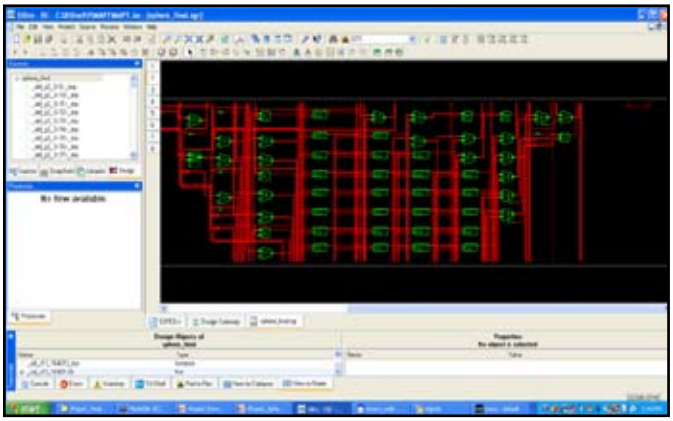

Fig. 10 : RTL schematic view

The Table II shows the power consumption, quiescent current at 1.2v and quiescent current at 2.5v for the proposed architecture.

Table 2 :Performance Analysis

Parameter	Previous	Proposed
Power Consumption	203mw	159mw

IV. Conclusions

This brief presents the algorithm optimization and VLSI implementation of a SISO FSD. Based on the hard-output imbalanced FSD, the proposed SISO FSD algorithm employs the efficient OHE to avoid the exhaustive search of the best child for the soft-input scenario and adopts the simple PCA scheme to improve the quality of the output LLRs. In addition, the compensation of the self-interference caused by channel-matrix regularization is incorporated in the tree search, leading to further performance gain. These proposed techniques can reduce the complexity significantly and provide near max–log–MAP performance. At the architecture level, the proposed multistage architecture using the time-multiplexing hardware sharing fashion further reduces the area cost. Implementation results show that our SISO FSD outperforms other reported iterative MIMO detectors in terms of throughput and area efficiency.

V. Acknowledgement

References

[1] *B. M. Hochwald and S. Brink, "Achieving near-capacity on a multiple antenna channel," IEEE Trans. Commun., vol. 51, no. 3, pp. 389–399,May 2003.*

[2] *C. Studer, S. Fateh, and D. Seethaler, "ASIC implementation of soft input soft-output MIMO detection using MMSE parallel interference cancellation," IEEE J. Solid-State Circuits, vol. 46, no. 7, pp. 1754–1765, Jul. 2011.*

[3] *E. M. Witte, F. Borlenghi, G. Ascheid, R. Leupers, and H. Meyr, "A scalable VLSI architecture for soft-input soft-output single tree-searchsphere decoding," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 57, no. 9, pp. 706–710, Sep.2010.*

[4] *F. Borlenghi, E. M. Witte, G. Ascheid, H. Meyr, and A. Burg, "A 772 Mbit/s 8.81 bit/nJ 90 nm CMOS soft-input soft-output sphere decoder,"in Proc. ASSCC, Jeju, Korea, 2011, pp. 297–300.*

[5] *Y. Sun and J. R. Cavallaro, "Trellis-search based soft-input soft-output MIMO detector: Algorithm and VLSI architecture," IEEE Trans. Signal Process., vol. 60, no. 5, pp. 2617–2627, May 2012.*

[6] *L. G. Barbero and J. S. Thompson, "Fixing the complexity of the sphere decoder for MIMO detection," IEEE Trans. Wireless Commun., vol. 7, no. 6, pp. 2131–2142, Jun. 2008.*

[7] *L. Liu, J. Lofgren, and P. Nilsson, "Area-efficient configurable highthroughput signal detector supporting multiple MIMO modes," IEEETrans. Circuits Syst. I, Reg. Papers, vol. 59, no. 9, pp. 2085–2096, Sep. 2012.*

[8] *L. G. Barbero and J. S. Thompson, "Extending a fixed-complexity sphere decoder to obtain likelihood information for turbo-MIMO systems," IEEE Trans. Veh. Technol., vol. 57, no. 5, pp. 2804–2814, Sep. 2008.*

[9] *C. Studer and H. Bolcskei, "Soft-input soft-output single tree-search sphere decoding," IEEE Trans. Inf. Theory, vol. 56, no. 10, pp. 4827–4842, Oct. 2010.*

Author's Profile

M.SHEEBA received her B.E., degree in Electronics and Communication Engineering from the K.L.N College of Engineering, Sivagangai, India, in 2012. Currently doing M.E - VLSI Design in Sethu Institute of Technology, Virudhunagar, India. Her research interest includes: low Power VLSI and Wireless network.

Ms.K.MONISHA received her B.E., from K.L.N College of Engineering in 2010 and completed her M.E from Sethu Institute of Technology in 2012. Presently working as an Assistant Professor in the Department of ECE at Sethu Institute of Technology, Tamilnadu, India. Her current research areas includes: Image Processing and Signal Processing.