**Review of Information Engineering and Applications**
*2014 Vol. 1, No. 1, 11-23.*
*ISSN(e): 2409-6539*
*ISSN(p): 2412-3676*
*DOI: 10.18488/journal.79/2014.1.1/79.1.11.23*


# VLSI ARCHITECTURE OF MIMO DETECTOR USING FIXED COMPLEXITY SPHERE DECODING

**J.M.Mathana[1] --- P.Rekha[2] --- V.Sai Anitha[3] --- B.Suchitra[4] --- K.B.Bavithra[5]**

[1]*Professor S.A Engineering College Chennai, India*

[2,3,4]*UG student S.A Engineering College Chennai, India*

[5]*Asst. Professor S.A Engineering College Chennai, India*

## ABSTRACT

*Fixed Sphere Decoding is a near optimum tree search detection technique for the spatial multiplexing scheme. The algorithm performs a fixed number of operations to detect the signal independent of the noise level and channel conditions. In this paper, a Soft Input Soft Output Fixed Complexity Sphere Decoding algorithm is proposed for the MIMO receiver using 16 QAM modulation scheme. As the system performance was far from the channel capacity limit, MIMO channel could not support higher spectral efficiencies. Therefore, to obtain power efficiency very close to the Shannon limit, Turbo codes are implemented in MIMO system and provide higher spectral efficiency. The proposed FSD detector is capable of providing a throughput of 1.18 Gbits/s with a critical path delay of 9.603 ns.*

**Keywords:** FSD, MIMO, SISO, QAM, LLR, TURBO codes, APP.

## Contribution/ Originality

The primary contribution of this paper is to design a high efficient soft output fixed complexity sphere detection coupled with a Turbo decoder for MIMO system with low hardware complexity and high throughput. The System can be simulated in Verilog HDL environments and then implemented in a FPGA system to analyze the actual performance.

## 1. INTRODUCTION

Multiple-input multiple output (MIMO) technology uses multiple antennas at the transmitting and receiving ends along with spatial multiplexing and channel coding. It provides reliable, high-speed and bandwidth efficient data transmission. Many wireless communication standards such as IEEE 802.11n and 3GPP-LTE use this technology [1].

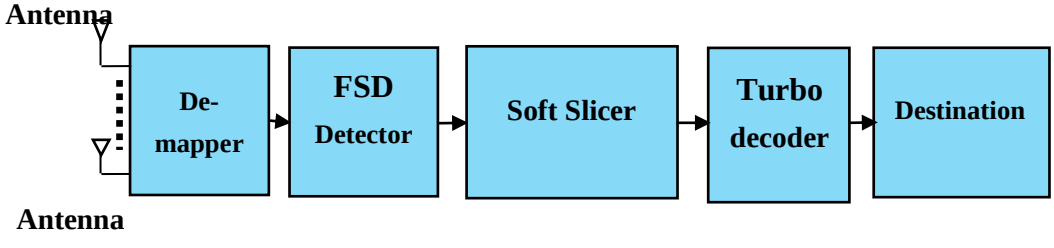
The main challenge in MIMO data detection is computational complexity. Particularly the computational complexity of MIMO detection algorithm [2] depends exponentially with the number of spatial streams used. Thus, research mainly focuses on developing algorithms to reduce the computational complexity of the system. The two solutions to MIMO detection problems [3] are: breadth-first tree-search algorithms such as K-best detection and the depth-first tree-search algorithms such as sphere detection. However, there are some problems in using depth first sphere detection. It uses the lattice structure of the received signal to achieve Maximum Likelihood (ML) performance with reduced complexity than MLD [4]. But the number of visited nodes is large in the low SNR (signal-to-noise) region, while it is small in the high SNR regions. Due to this, it has variable throughput which makes it disadvantageous in systems with strict latency requirements. Also the complexity depends on the channel conditions and the noise level posing a problem if the SD has to be integrated into an actual communication system that requires data to be processed at a constant rate. Therefore, a new detection algorithm that is similar to the sphere decoding but performs a breadth first search called the fixed sphere decoding was proposed [5]. It performs only a fixed number of operations during the detection and achieves quasi ML performance thereby overcoming the two main problems of the SD: its variable complexity and its sequential nature. The algorithm combines ordering of the channel matrix with a search on a small subset of the complete transmit constellation which is independent of the noise level and the channel conditions. It is important to determine the subset of the complete transmit constellation that needs to be searched.

The paper is organized as follows. In section II, a review of MIMO receiver model is discussed. Section III describes the fixed sphere decoding algorithm and the proposed architecture of the same. The soft slicer for the proposed MMO receiver is explained in section IV. Section V describes the architecture of proposed turbo decoder and section VI describes about exception handling. In section 7, simulation and synthesis results are reported and section 8 concludes the paper.

## 2. MIMO RECEIVER MODEL

The MIMO receiver gets the received signal vectors by the multiple receive antennas and decodes the received signal vectors into the original information. A MIMO system consist of $N_t$ transmit antennas and $N_r$ receive antennas [1]. The block diagram of the MIMO receiver is shown in figure 1.

**Figure-1.** Block diagram of the Proposed MIMO receiver



The antenna demapper in figure 1 is used to demap the received signals into single bit stream. A MIMO decoder is configured to obtain a channel matrix and generate a Hermitian transpose of the channel matrix. The products of the Hermitian transpose of the channel matrix and the channel matrix is generated to provide a first product having multiple diagonal elements. A partial matrix inversion of the diagonal elements of the first product is generated to provide a diagonal vector. From the diagonal vector, an antenna layer is selected from the multiple antenna layers and represents the antenna layer selected for a given processing iteration. The selected antenna layer will preferably correspond to that having the lowest inverse channel gain. A partial matrix inversion of the first product along the row corresponding to the selected antenna layer is generated to provide a row vector. A product of the row vector and the Hermitian transpose of the channel matrix is generated to provide an inverse channel gain vector.

In the proposed work, 16 QAM modulation is considered with Mc = 4.Turbo-scheme is used for detection and decoding of the symbols of the MIMO receiver [5]. The soft-information consists of *a posteriori* probability (APP) information expressed in the form of log-likelihood ratios (LLRs). The LLR of a bit is defined as the natural logarithm of the ratio of the probabilities of the bit taking its two possible values and is expressed as,

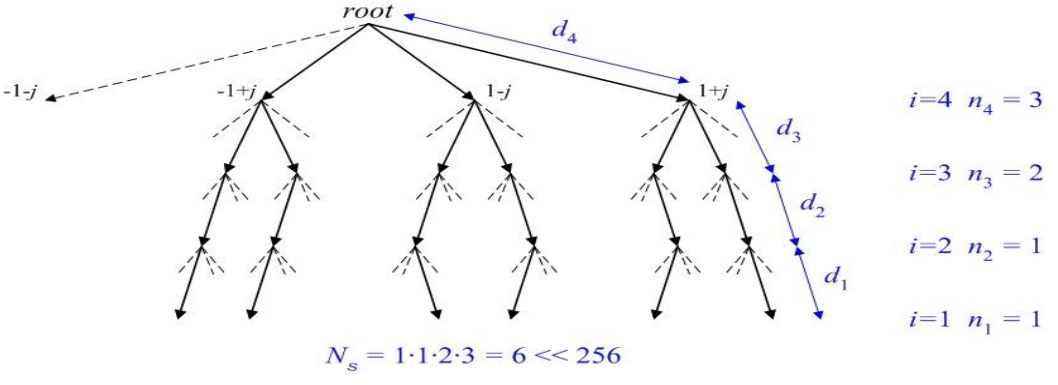
$$L(b_k) = ln\frac{Pr[b_k = +1]}{Pr[b_k = -1]} \qquad (1)$$

where +1 and −1 represent a logical '1' and a logical '0', respectively. The soft-MIMO detector generates APP information about the interleaved bits by taking into account the channel observations and the *a priori* information.

## 3. ARCHITECTURE OF PROPOSED FIXED SPHERE DECODER

One of the important modules in the MIMO receiver is Fixed Sphere Decoder. FSD algorithm is employed to overcome the limitations of sphere decoding [6] [7] which include higher computational complexity and varying throughput depending on the noise signal.

**Figure- 2.** Principle of Fixed complexity sphere decoding



The FSD algorithm is used to detect the transmitted signal with low complexity. FSD is based on the Breadth First Search (BFS) tree technique. In FSD, throughput is increased by performing parallel search. Figure 2 represents the general tree traversal principle of the FSD algorithm.

The detection order of FSD is determined iterative manner [8]. When the algorithm proceeds from level one level to next level, the symbol with the smallest post-detection signal-to-noise ratio (SNR) among those yet to be searched is chosen for detection, if level belongs to the FE stage; otherwise, the one with the largest post-detection SNR is chosen. After the detection order is obtained, reordering of the columns of H is performed to obtain $\hat{H}$ giving y = $\hat{H}$s+n, where QR decomposition of $\hat{H}$ is denoted as $\hat{H}$=Q$\acute{R}$

The steps of the algorithm are as follows, [9]

1. Inputs to FSD detector are the received soft bits (Y) from soft slicer and QAM bits.
2. The detected signal is QR decomposed to get the received signal, Q is a matrix with orthonormal columns, and $\acute{R}$ is a square upper triangular matrix. It is given by,

$$Y = Hs + n \quad (2)$$

where 'Hs' is the channel matrix and 'n' is the white circularly symmetric Gaussian noise vector,

$$H = QR \quad (3)$$

The QR decomposed received signal is,

$$\hat{Y} = Q^H \times Y \quad (4)$$

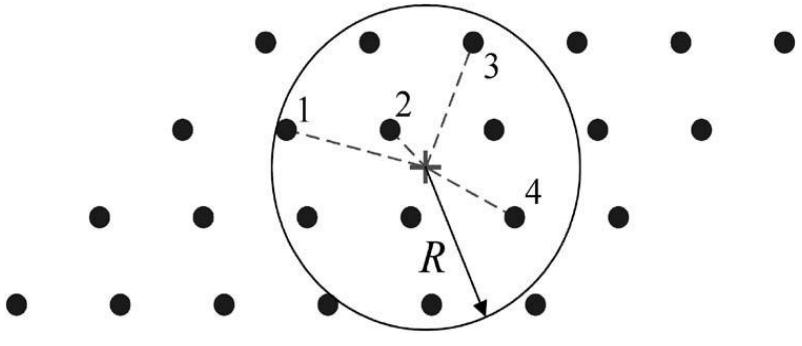
3. The Metric distance is calculated between the QR decomposed received signal and the constellations points as,

$$M_P(si) = M_c(si) + M_A(si) \quad (5)$$

where $M_c(si)$ is the channel based metric increments and $M_A(si)$ is apriori based metric increments.

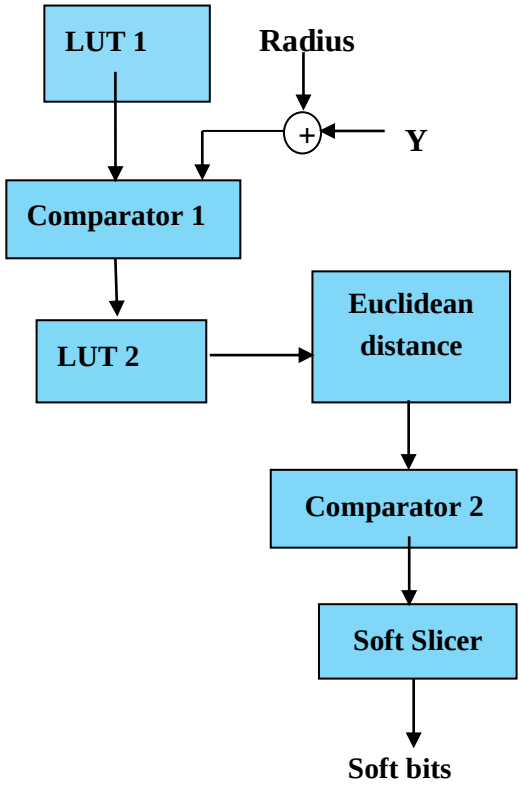
4. The signal with minimum Euclidean distance (minimum metric) is then calculated to find the transmitted signal.

**Figure-3.** Representation of received signal and constellation points in 2-D space.



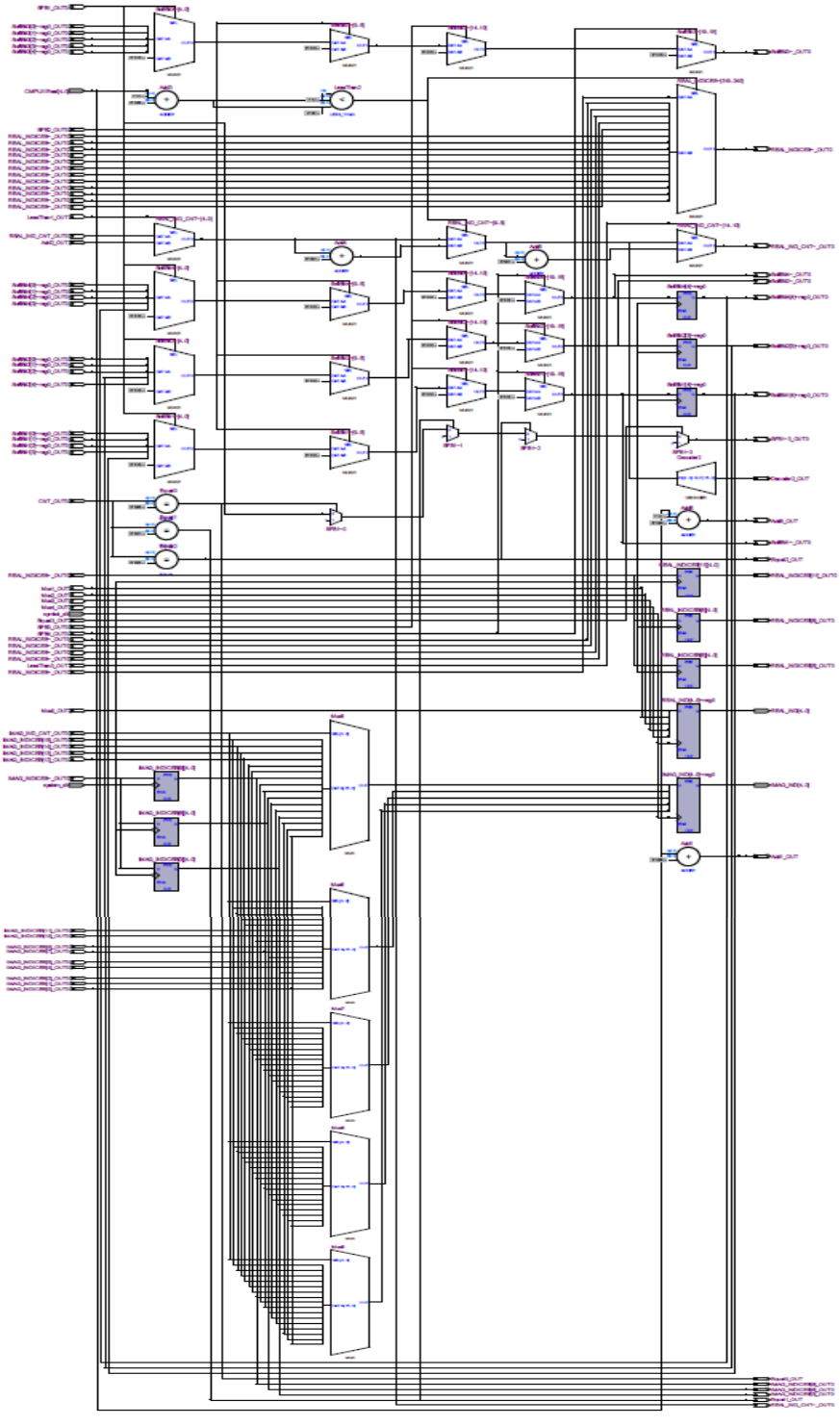
The QAM constellation points are represented as dots in the figure 3 and the received signal with noise function added to it is represented as cross. The Euclidean distance is calculated between the received signal and the constellation points [10]. The flow graph of the fixed complexity sphere decoding is depicted in figure 4.

**Figure-4.** Flow graph of fixed complexity sphere decoding




In figure 4, Look Up Table 1 contains all the 16 constellation points of the QAM modulation technique. The comparator compares the detected signal with the constellation points within the radius 'R'. The constellation points within the given boundary are tabulated in the LUT 2. The Euclidean distance is calculated between the received vector and all the constellation points in the LUT 2 and compared to get the constellation point with minimum euclidean distance. That point is considered as the transmitted signal and it is given as input to the soft slicer to get the soft output bits. The RTL schematic of fixed complexity sphere decoder is depicted in figure 5.

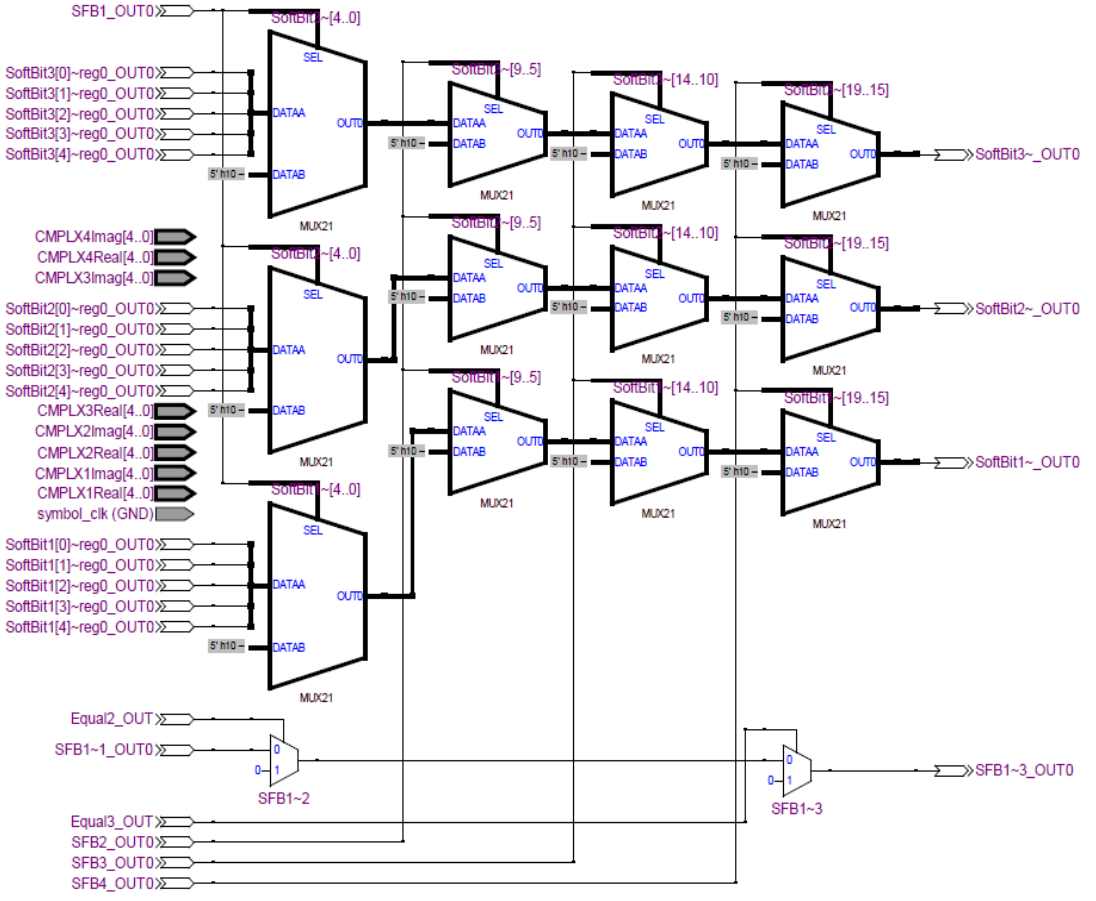
**Figure-5.** RTL Schematic of fixed complexity sphere decoder



## 4. PROPOSED SOFT SLICER

The next block in the proposed receiver is soft slicer. A soft slicer generates quantized data and associated soft data. A decoder with error recovery generates decoded quantized data and a soft sequence and is capable of correcting one bit of the quantized data. Soft bit values are the probability that the given data symbols resides at a particular point in the co-ordinate system [11]. The RTL schematic of the soft slicer is shown in figure 6.

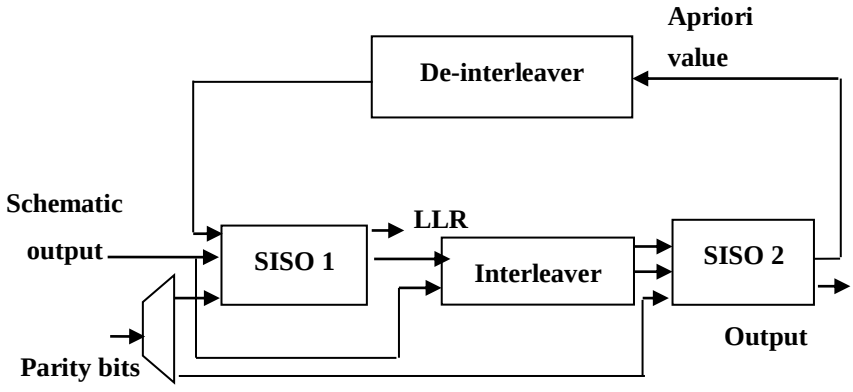
**Figure-6.** RTL Schematic of the Soft Slicer




## 5. ARCHITECTURE OF PROPOSED TURBO DECODER

MIMO systems have been theoretically shown to provide an order of magnitude increase in capacity / spectral efficiency. Since channel coding was not employed in the BLAST system, high wireless SNRs were required to achieve target bit-error rates. Consequently, system performance was far from the channel capacity limit. Turbo decoding techniques employed in MIMO system achieve even higher spectral efficiency. The general block diagram of turbo decoder is shown in figure 7.

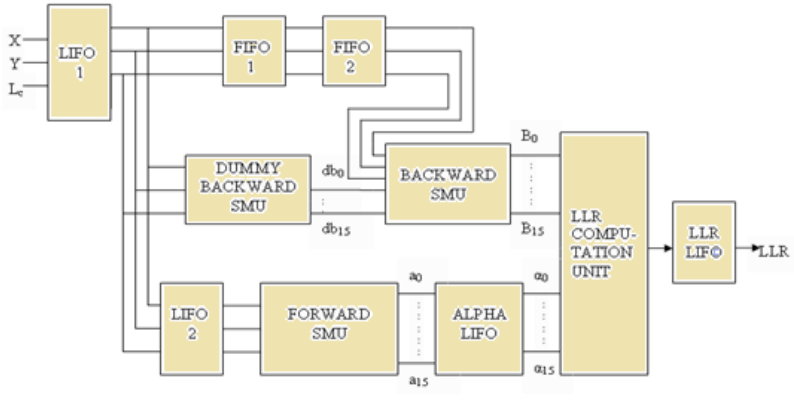
**Figure-7.** General diagram of Turbo decoder




A Turbo decoder consists of two soft-input soft output (SISO) decoders and one interleaver/deinterleaver [12]. Decoding process in a turbo decoder is performed iteratively through the two SISO decoders via the interleaver and the deinterleaver by feeding the extrinsic information of one to the other. LLR values are computed based on the input symbols. The computation of LLR values uses Max-Log MAP algorithm. Further to reduce the memory size, the block of input symbols is sub-divided into several sub-blocks and decoding is done using sliding window method.

As shown in the figure 8, the main components in the SISO decoder are Forward State Metric Unit, Backward State Metric Unit, Dummy Backward State Metric Unit and Likelihood ratio computation unit [13]. Last In First Out (LIFO) and First In First Out (FIFO) are simply memory units used to store the input data symbols and they accompany the Sliding Window (SW) method. Similarly Alpha LIFO and LLR LIFO are used to store the computed state metrics and Likelihood values respectively.

**Figure-8.** Block diagram of SISO decoder




LLR values are represented by the following equation:

$$L_{lr} = \ln \frac{\sum_{S_k} \sum_{s_{k-1}} \gamma_1(S_{k-1}, S_k)\alpha(S_{k-1})\beta(S_k)}{\sum_{S_k} \sum_{s_{k-1}} \gamma_0(S_{k-1}, S_k)\alpha(S_{k-1})\beta(S_k)} = L_1 - L_0 \quad (6)$$

$$L_1 = \ln \sum_{S_k} \sum_{s_{k-1}} \gamma_1(S_{k-1}, S_k)\alpha(S_{k-1})\beta(S_k) \quad (7)$$

$$L_0 = \sum_{S_k} \sum_{s_{k-1}} \gamma_0(S_{k-1}, S_k)\alpha(S_{k-1})\beta(S_k) \quad (8)$$

where γ, α, and β represent the branch, forward, and backward state metric values respectively. The subscript *k* and *S* denote time and state respectively [14]. The LLR value (*Llr*) is calculated by the metric values at all states (*S*) of time *k* and *k-1*. The equation of γ, α, and β can be represented to logarithm form as shown below.

$$ln\gamma(S_{k-1}, S_k) = 1/2\,(L_e\, u_k^s + L_c\, x\, u_k^s + L_c\, y_1\, u_k^p) \quad (9)$$

$$ln\alpha(S_k = ln \sum_{S_{k-1}} exp(\, ln\, \gamma\, (S_{k-1}\, , S_k) + ln\alpha(S_{k-1})) \quad (10)$$

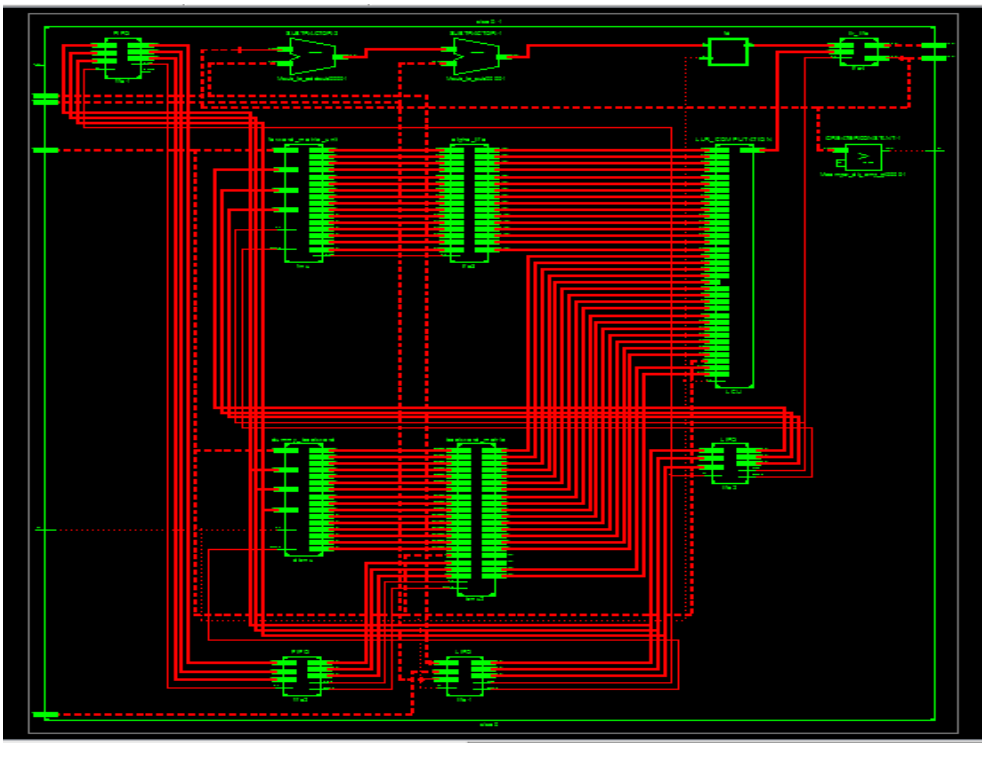
$$\ln \beta\, (S_k) = ln \sum_{S_{k-1}} exp(\, ln\gamma\, (S_k\, , S_{k+1}) + ln\beta(S_{k+1})) \quad (11)$$

The logarithm function makes the MAP algorithm unsuitable for hardware implementation. Therefore, Jacobian logarithm approximation is used and it is given below.

$$\ln(e^x + e^y) = \max(x, y) + \ln(1 + e^{-|x-y|}) \quad (12)$$

This approximation is used to implement the state metric unit (SMU) and LLR computation unit (LCU) in Log MAP and ML-MAP SISO decoder [15]. The second term of the right hand side in the equation is a correction term which can be implemented through a simple look-up table. However, here the SISO decoder is implemented without this correction term.

**Figure-9.** RTL schematic of Turbo decoder



## 6. SIMULATION RESULTS

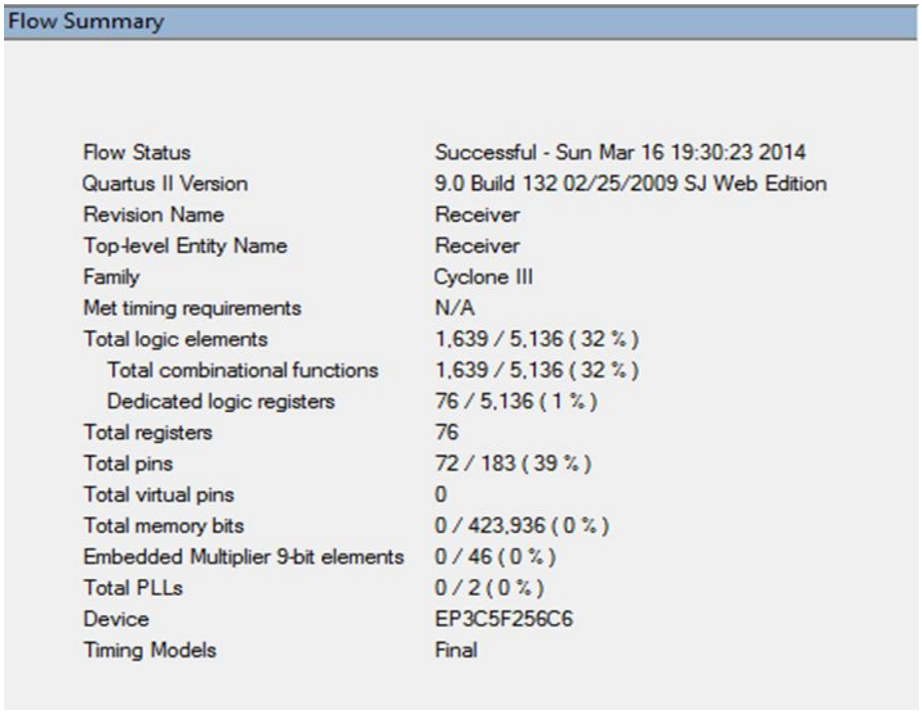
The RTL code for the proposed fixed complexity sphere decoder is written using verilog. The verilog coding is then synthesized and simulated to check the correctness of the design using Altera Quartus II 9.0.

**Table-1.** Resource utilization of fixed complexity sphere decoder

| S no. | Logic utilization | Used | Available | Utilization in % |
|---|---|---|---|---|
| 1 | Total logic elements | 1639 | 5136 | 32 |
| 2 | Total combinational functions | 1639 | 5136 | 32 |
| 3 | Dedicated logic registers | 76 | 5136 | 1 |
| 4 | Total registers | 76 | - | - |
| 5 | Total pins | 72 | 183 | 39 |

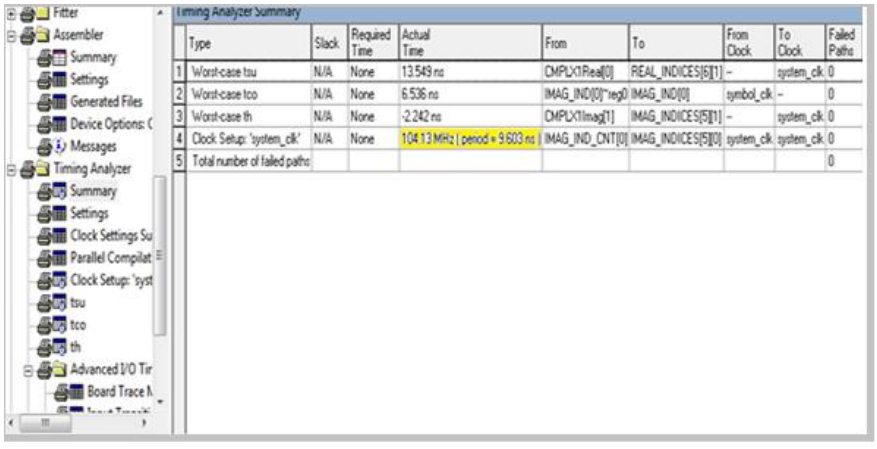
The RTL schematic is shown in figure 5 and its resource utilization summary is depicted in table 1.

**Figure-10.** Flow summary of FSD




The flow summary is provided in figure 10 using Quartus III 9.0.

**Figure-11** Timing Analysis of FSD




The timing analysis of the work is carried out in Altera Quartus II 9.0 and it is shown in figure 11. Source

The throughput of the fixed complexity sphere decoder is calculated using the operating frequency. The throughput is calculated using the formula,

$$\emptyset = \frac{M_C.N_T}{N_{cycle}} f_{clk} \qquad (13)$$

where $M_C$ is number of bits used to represent the QAM constellation points, $N_T$ is the number of transmitting antennas, $f_{clk}$ is the clock frequency and $N_{cycle}$ is the number of clock cycles. The operating frequency is calculated from the timing analysis summary of the FSD as shown in figure 10. For a 16 QAM modulation, the throughput and the critical path delay of the FSD with the operating frequency 104.13 MHz is 1.18 Gbits/s and 9.603 ns respectively.

The performance analysis of the turbo decoder design is performed in Xilinx ISE 9.1i software. The RTL schematic is shown in figure 9 and its resource utilization summary is depicted in table 2.

**Table-2.** Resource utilization of turbo decoder

| Logic Utilization | BMU | ACS | SMU | LMU |
|---|---|---|---|---|
| Number of Slices | 104 | 20 | 329 | 329 |
| Number of 4 input LUTs | 183 | 36 | 582 | 582 |
| Number of bonded IOBs | 55 | 46 | 310 | 310 |
| Number of GCLKs | 1 | 1 | 2 | 2 |

**Table-3.** Path delays of various blocks of turbo decoder

| Name of the module | Obtained critical path delay |
|---|---|
| BMU | 9.030 ns |
| SMU | 9.624 ns |
| LCU | 7.848 ns |

Table 3 represents the comparison of critical path delay for various modules in turbo decoder. From the table it is clear that the State metric unit (SMU) is having the highest critical path delay. The path delay and the operating frequency for the modules FSD and turbo decoder are known and tabulated in table 4.

**Table-4.** Frequency of operation of FSD and Turbo decoder

| Modules | Critical path delay in ns | Operating frequency in MHz |
|---|---|---|
| FSD | 9.603 | 104.13 |
| Turbo decoder | 9.624 | 103.9 |

**Table-5.** Comparison with existing state-of-the-art MIMO receivers

| Work | This work | [8] | [9] |
|---|---|---|---|
| Antennas | 4 x 4 | 4 x 4 | 4 x 4 |
| Modulation | 16- QAM | 16- QAM | 16- QAM |
| Algorithm | FCSD | K-Best | SD |
| Operating frequency | 104.13 MHz | 200MHz | 71MHz |
| Throughput | 1.18 Gbits/s | 106.6 Mbps | 169Mbps |

Table 5 compares the performance with existing state-of-the-art MIMO receivers using FSD. In Guo and Nilsson [8], a MIMO receiver using K-Best algorithm is described and the work in Burg, et al. [9] is implemented a MIMO receiver using SD algorithm.

## 7. CONCLUSION AND FUTURE WORK

The paper has presented a Soft Output Fixed Complexity Sphere Decoding algorithm for the MIMO receiver using 16 QAM modulation scheme. The proposed Fixed Sphere Decoder has a critical path delay of 9.624 ns and its frequency of operation is 104.13 MHz and throughput of FSD is found to be 1.18 Gbits/s. In the design, the speed of the Turbo decoder is decided by the SMU module which has a critical path delay of 9.624 ns and its frequency of operation is 103.9 MHz. In the proposed work, the design of a Fixed Sphere Decoder and turbo decoders for the MIMO system are designed for which throughputs were observed and proper reliability is noticed. This may further be extended by integrating the Turbo decoder and the Fixed Sphere Decoder to achieve higher throughput and better performance and analyze the impact of using FSD detection in the MIMO receiver.

## REFERENCES

[1] X. Chen, H. Guanghui, I. Member, and M. Jun, "VLSI implementation of a high- throughput Iterative fixed-complexity sphere decoder," *IEEE Transactions on Circuits and Systems—II: Express Briefs*, vol. 60, pp. 272-275, May 2013.

[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, pp. 1754–1765, Jul. 2011.

[3] L. G. Barbero and J. S. Thompson, "Fixing the complexity of the sphere decoder for MIMO detection," *IEEE Trans. Wireless Commun*, vol. 7, pp. 2131–2142, Jun. 2008.

[4] 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, pp. 2617–2627, May 2012.

[5] 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, pp. 2804–2814, Sep. 2008.

[6] C. Studer and H. Bolcskei, "Soft-input soft-output single tree-search sphere decoding," *IEEE Trans. Inf. Theory*, vol. 56, pp. 4827– 4842, Oct. 2010.

[7] L. G. Erik, "MIMO detection methods: How they work," *IEEE Signal Processing Magazine*, vol. 3, pp. 91-95, 2009.

[8] Z. Guo and P. Nilsson, "Algorithm and implementation of the k-best sphere decoding for MIMO detection," *IEEE J. Sel. Areas Commun.*, vol. 24, pp. 491–503, March 2006.

[9] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Bolcskei, "VLSI implementation of MIMO detection using the sphere decoding algorithm," *IEEE J. Solid-State Circuits*, vol. 40, p. 1566–1577, July 2005.

[10] W. Xiang, J. S.Thompson., and A. M.Wallace., "An improved sphere decoding scheme for MIMO systems using an adaptive statistical threshold," in *17th European Signal Processing Conference (EUSIPCO 2009)*, 2009, pp. 2668-2672.

[11] B. Bai, X. Ma, and X. Wang, "Novel algorithm for continuous decoding of turbo codes," *IEE PTOC.Commun.*, vol. 146, pp. 271-274, Oct. 1999.

[12] J. M. Dr Mathana and P. Dr Rangarajan, "FPGA implementation of high speed architecture for max log map turbo SISO decoder," *International Journal of Recent Trends in Engineering*, vol. 2, pp. 142-146, November 2009.

[13] 3GPP TS 25.212, "Multiplexing and channel coding (FDD), v. 4.6.0," Sept 2002.

[14] M. Adamantios, "Reliability allocation and optimization for complex systems," presented at the Proceedings Annual Reliability and Maintainability Symposium, Los Angeles, California, USA, 2000.

[15] C. Berrou and A. Glavieux, "Near optimum error correcting coding and encoding: Turbo-code," *In IEEE Transactions on Communications*, vol. 44, pp. 1261-1271, October 1996.