

Performance and Analysis of Low Power Error Resilient Multi Input Multi Output Detectors

S.Sobana

Department of ECE, PSNA college of engineering & Technology, Dindigul, Tamilnadu, India.

S.P.Prabu

M E VLSI Design, PSNA college of engineering & Technology, Dindigul, Tamilnadu, India.

Abstract –Multiple-antenna (MIMO) technology is becoming mature for wireless communications and has been incorporated into wireless broadband standards like LTE and Wi-Fi, the above all detectors facing the power consumption problem. The proposed error resilient K-best MIMO detector system is designed using Euclidian bi-Orthogonal architecture for the 4×4 64-QAM system achieves the better power consumption compared to the conventional K-best MIMO detectors. While using verilog HDL, simulated using modelsim and synthesised using Xilinx project navigator software. The results shows that the number of slices & LUT (look up tables) needed for this error resilient K-best MIMO detector is reduced when compared with the conventional k-best MIMO detectors.

Index terms —Multiple-input multiple-output (MIMO), K-Best, error resilient MIMO detector, very large scale integration (VLSI), and wireless communications.

1. INTRODUCTION

Multiple input multiple output (MIMO) system consists of multiple antennas at the transmitter and receiver ends to improve the power consumption wireless communication in MIMO systems [2]. Orthogonal frequency division multiplexing is efficient in synchronizing the received signal under fading environment and has been used in past times in applications that require a huge data rate. Fast Fourier Transform (FFT)/ inverse FFT (IFFT) processors are proposed for multiple – input multiple – output orthogonal frequency division multiplexing based IEEE 802.11n. Here the processor not only supports the operation of FFT/IFFT but also provides sufficient throughput rates but the drawback is hardware complexity and throughput is less compared with conventional approach [2]. Here the paper mainly focuses on power consumption & throughput of MIMO OFDM system. In requirements the processor supports the operation of FFT/IFFT in 16 points and 64 points can different throughput rates for simultaneous data sequences. The difficulty is number of antenna increases more and more. Hence, the cost and complexity of the system is also increased. IEEE 802.11a established for WLAN standard provides 54 Mbps throughput using SISO OFDM transceiver. The IEEE802.11n provides data rate up to 600 Mbps with transmission speed of 80MHz Latency is incurred by pre-processing channel matrices for MIMO detection. The

VLSI implementation of MIMO OFDM transceiver suitable for low power application [2]. In order to maintain a performance close to maximum-likelihood (ML) detection, we focus on the use of nonlinear breadth-first MIMO detection (also known as K-Best detection)[3]. The major drawback is the rapid degradation in throughput as K- value increases[4]. For example, for a 16-QAM system, the reported throughput is 376 Mb/s for $K = 5$ and degrades to 80 Mb/s for $K = 10$. The VLSI architecture proposed in [5] reduces the power consumption considerably due to its sort-free architecture; however, both the throughput and area are not significantly improved. The largest potential for complexity reduction of highest-performance VLSI circuits for signal processing is in the joint optimization of both the algorithms and the register transfer level architecture with the circuit level tradeoffs in mind [6]. For instance, for 16-QAM, is chosen to be 5 while for 64-QAM, $K = 10$ meaning that the constellation quadruples but the K value only doubles, thus the sub-linear increase. It also has fixed critical path delay independent of the constellation order, value, and the number of antennas. Moreover, it efficiently expands a very small fraction of all possible children in the K-best algorithm and can be applied to infinite lattices. Finally it provides the exact K-best solution, i.e., the solution that implements the original K-best algorithm with all needed expansions [7].

This paper presents a error resilient K-best MIMO detector and its VLSI implementation that can achieve high throughput with low complexity. This paper is organized as follows. A review of MIMO detection is presented in section 2. Section 3 describes the proposed error resilient K-best MIMO detector architecture. The proposed VLSI architecture is described in section 4. The simulation results are presented in section 5. Finally this paper is concluded in section 6.

2. INTRODUCTION TO QAM

We assume that the receiver has acquired knowledge of the Channel (e.g., through a preceding training phase). Algorithms to separate the parallel data streams corresponding to the transmit antennas [8] are as follows,

- Linear detection methods invert the channel matrix using a zero-forcing (ZF) or minimum mean

squared error (MMSE) criterion. The received vectors are then multiplied by the channel inverse, possibly followed by slicing. The drawback is, in general, a rather poor bit-error-rate (BER) performance.[8]

- Ordered successive interference cancellation decoders such as the vertical Bell Laboratories layered space time (VBLAST) algorithm show slightly better performance, but suffer from error propagation and are still suboptimal.[8]
- QAM Modulation
- QAM applications

2.1. QAM Modulation

Quadrature amplitude theory states that both amplitude and phase change within a QAM signal. The basic way in which a QAM signal can be generated is to generate two signals that are 90° out of phase with each other and then sum them. This will generate a signal that is the sum of both waves, which has certain amplitude resulting from the sum of both signals and a phase which again is dependent upon the sum of the signals.

If the amplitude of one of the signals is adjusted then this affects both the phase and amplitude of the overall signal, the phase tending towards that of the signal with the higher amplitude content.

As there are two RF signals that can be modulated, these are referred to as the I - In-phase and Q - Quadrature signals.

The I and Q signals can be represented by the equations below:

$$I = A \cos(\Psi) \quad \text{and} \quad Q = A \sin(\Psi)$$

It can be seen that the I and Q components are represented as cosine and sine. This is because the two signals are 90° out of phase with one another.

Using the two equations it is possible to express the signal as:

$$\cos(\alpha + \beta) = \cos(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta)$$

Using the expression $A \cos(2\pi ft + \Psi)$ for the carrier signal:

$$A \cos(2\pi ft + \Psi) = I \cos(2\pi ft) - Q \sin(2\pi ft)$$

Where f is the carrier frequency.

This expression shows the resulting waveform is a periodic signal for which the phase can be adjusted by changing the amplitude either or both I and Q. This can also result in an amplitude change as well. Accordingly it is possible to digitally modulate a carrier signal by adjusting the amplitude of the two mixed signals. QAM, Quadrature amplitude modulation is widely used in many digital data radio communications and data communications applications. A variety of forms of QAM are available and some of the more common forms include 16 QAM, 32 QAM, 64 QAM, 128 QAM, and 256 QAM. Here the figures

refer to the number of points on the constellation, i.e. the number of distinct states that can exist. The various flavours of QAM may be used when data-rates beyond those offered by 8-PSK are required by a radio communications system. This is because QAM achieves a greater distance between adjacent points in the I-Q plane by distributing the points more evenly. And in this way the points on the constellation are more distinct and data errors are reduced. While it is possible to transmit more bits per symbol, if the energy of the constellation is to remain the same, the points on the constellation must be closer together and the transmission becomes more susceptible to noise. This results in a higher bit error rate than for the lower order QAM variants. In this way there is a balance between obtaining the higher data rates and maintaining an acceptable bit error rate for any radio communications system

2.2. QAM applications

QAM is in many radio communications and data delivery applications. However some specific variants of QAM are used in some specific applications and standards. For domestic broadcast applications for example, 64 QAM and 256 QAM are often used in digital cable television and cable modem applications. In the UK, 16 QAM and 64 QAM are currently used for digital terrestrial television using DVB - Digital Video Broadcasting. In the US, 64 QAM are the mandated modulation schemes for digital cable as standardised by the SCTE in the standard ANSI/SCTE 07 2000. In addition to this, variants of QAM are also used for many wireless and cellular technology applications. The advantage of using QAM is that it is a higher order form of modulation and as a result it is able to carry more bits of information per symbol. By selecting a higher order format of QAM, the data rate of a link can be increased.

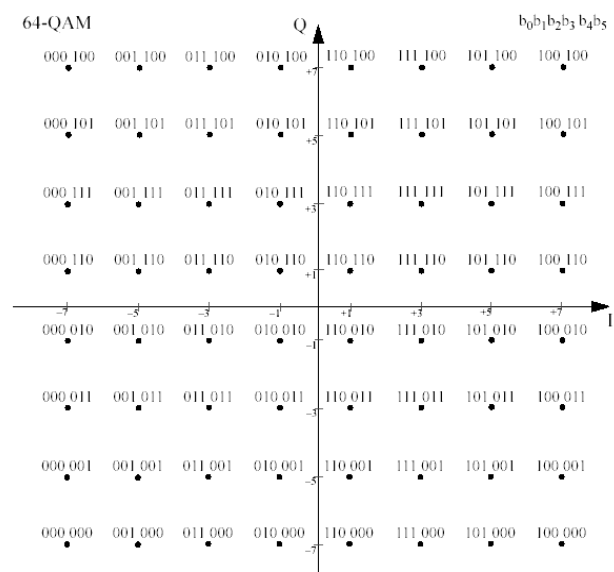


Figure 1 constellation diagram for 64-QAM
The constellation diagram in fig.1 shows the different positions for the states.

3. ERROR-RESILIENT DETECTION ALGORITHM

(Phase I work)

An error-resilient MIMO detection algorithm that can support a 4×4 MIMO system with 16-QAM modulation was designed and presented in [1]. The block diagram of the above mentioned MIMO system is shown in fig 2 then fig 3. Shows the tree structure with erroneous children nodes using 16-QAM. Both hard output and soft output scheme share the main part of the architecture with minor differences. Several complexity-reduction techniques have also been discussed therein. However, from the VLSI architectural point of view, the approach proposed in [11] could still result in degraded throughput and high complexity as compared to the conventional K-Best architectures. In this work, we present a low-complexity algorithm that is more suitable for VLSI architecture design and can achieve close-to optimal PER performance in the presence of joint channel noise and hardware error.

3.1. Two-Way Sorting

One of the main reasons that the approach presented in [1] contains high complexity lies in the fact that it employs an exhaustive enumeration for the child nodes. In other words, while expanding the tree nodes, it considers all the candidates resulting from either the error-free scenario or all possible combinations of error terms. This results in an excessively expanded tree dimension and significantly increased computation overhead. Although a complexity-reduction technique based on the concept of “learning” was utilized in [1], the resulting complexity can still be further decreased.



Figure. 2. A simplified MIMO system.

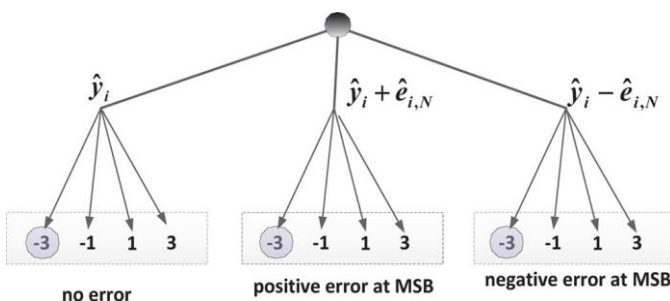


Figure 3 shows the tree structure with erroneous children nodes using 16-QAM as an example.

It is obvious that the optimal error value that could result into the minimum distance is the one closest to (W_i) . Hence, the two distances should be compared to find the minimum one as given by [1]

$$d_i = \min \begin{cases} |W_i| - \log c_0, & \text{error-free} \\ ||W_i| - \hat{e}_i| - \log c_1, & \text{erroneous} \end{cases}$$

4. VLSI ARCHITECTURE OF THE DETECTOR

4.1. History computation unit

The history computation unit shown in fig(5) (bi) circuit calculates the portion of the branch metric that is related to the tree nodes on the history path, that is, to compute the part of ‘bi’ expressed by [1]

$$b_i = \hat{y}_i - \sum_{j=i+1}^{2N_T} R_{ij} s_j$$

4.2. Processing Unit Architecture

Since this result will be identical for the same parent path, only one such unit is required for processing single survivor node. This part of the implementation utilizes a number of multipliers and adders, and occupies a significant amount of area and power consumption. In addition, for each survivor node, the PE is responsible for computing the complete path metrics and identifying the valid children for the symbol candidate from either error-free term or one of the erroneous terms. In other words, the PE will compute W_i and generate D_i represented in the. In order to maintain timing efficiency, in this architecture, multiple PEs are instantiated and running in parallel in which each one of them is operating on one symbol candidate. A 64-QAM system, there are eight PE blocks performing the operations for symbols -7, -5, -3, -1, 1, 3, 5, and 7 respectively. Once these eight valid symbol candidates are delivered, they will be sent to a sort and merge unit, for a pre-sorting. The sorted results will then be saved to the registers. As mentioned previously in Section V-A, the Sorting Unit (SU) employed to identify the K best survivors is based on the WPE structure. The major characteristic of this scheme is to enumerate the child nodes according to their path metrics. Thus, the sort and merge block within the PU will arrange the child nodes for each parent node based on the path metrics, such that the WPE-based SU can extend to the next winner candidate more efficiently. In short, the combination of the comparator circuit within each PE and the sort and merge circuit constructs the first step of the *two-way* sorting mentioned in Section III-A. The PU will iterate K times for processing all the K survivors and thus for each level of the tree, the PU will take up to K clock cycles. Finally, the detailed circuit for the binary search module is illustrated in Fig (6.c) The design is presented as an example of four error terms. However, the design is scalable to accommodate more error terms. First, the metric is compared with the thresholds to identify the selection lines and which are then utilized with a tree of multiplexers to choose the optimum error term.

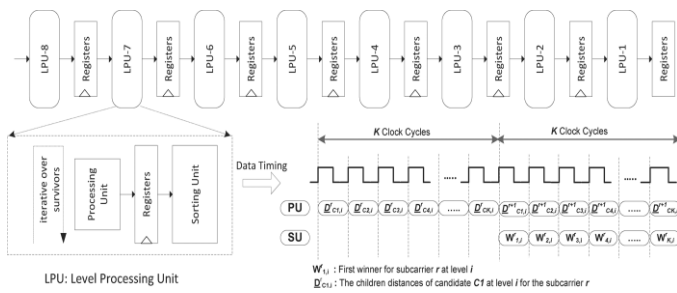


Figure 4 High-Level Overview Of The Proposed Architecture And Data

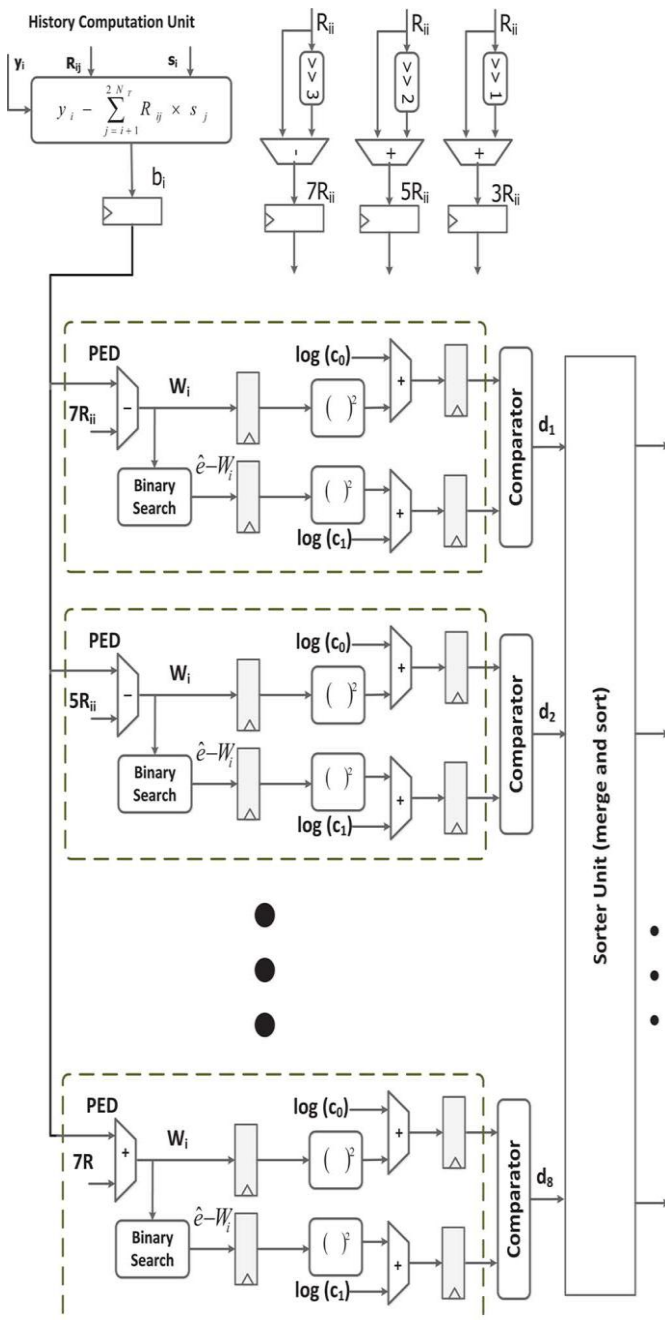


Figure 5 Architecture of the processing unit.

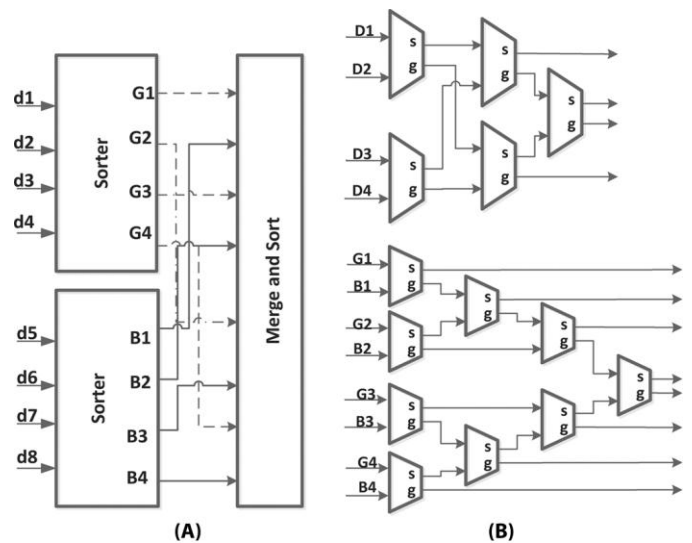


Figure 6 (a) Architectural overview of sort and merge unit within the PU, (b) detailed structure using cascaded 2-input comparator units with smaller and greater outputs, and (c) detailed architecture of binary search circuit.

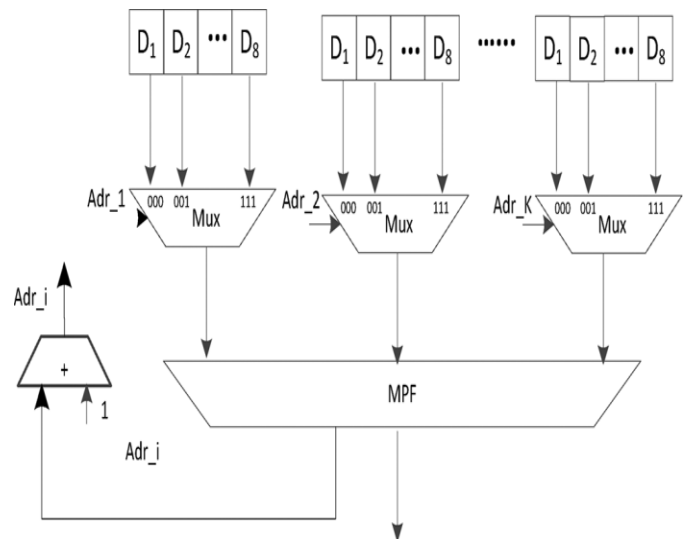


Figure 7 WPE based sorting unit (SU)

4.3. Sorter Unit

The SU shown in fig (7) represents the second step of the two-way sorting algorithm. The architecture of the WPE-based Sorting in which the children distances of each of the K-parents are stored in ascending order. Initially, the multiplexer select line is pointing to the child with the minimum K-parents are stored in ascending order. Initially, the multiplexer select line is pointing to the child with the minimum distance. The Minimum Path Finder (MPF) is a tree of comparators which finds the winner of K candidate distances. The select line of the winner is incremented to choose its sibling for the next search cycle. The sorting unit will take K clock cycles to find the best K children

5. EUCLIDIAN BI-ORTHOGONAL ARCHITECTURE
(Phase-II Work)

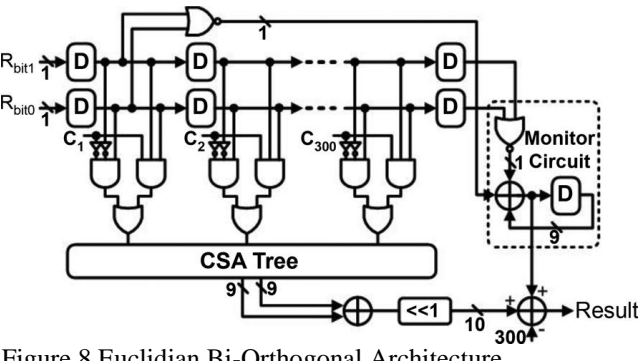


Figure 8 Euclidian Bi-Orthogonal Architecture

The carry-select adder generally consists of two ripple carry adders and a multiplexer. Adding two n-bit numbers with a carry-select adder is done with two adders (therefore two ripple carry adders) in order to perform the calculation twice, one time with the assumption of the carry being zero and the other assuming one. After the two results are calculated, the correct sum, as well as the correct carry, is then selected with the multiplexer once the correct carry is known. The number of bits in each carry select block can be uniform, or variable. In the uniform case, the optimal delay occurs for a block size. When variable, the block size should have a delay, from addition inputs A and B to the carry out, equal to that of the multiplexer chain leading into it, so that the carry out is calculated just in time. The delay is derived from uniform sizing, where the ideal number of full-adder elements per block is equal to the square root of the number of bits being added, since that will yield an equal number of MUX delays. The basic building block of a carry-select adder where the block size is four. Two 4-bit ripple carry adders are multiplexed together, where the resulting carry and sum bits are selected by the carry-in. Since one ripple carry adder assumes a carry-in of 0, and the other assumes a carry-in of 1, selecting which adder had the correct assumption via the actual carry-in yields the desired result. The figure 5 consist 8 processing elements here the

Euclidian bi-orthogonal architecture replaced to the 8 processing elements in the proposed work. The performance analysis of the proposed work results like power consumption, latency both less then compare to the phase-I work .here we got better results compare to the phase-I work

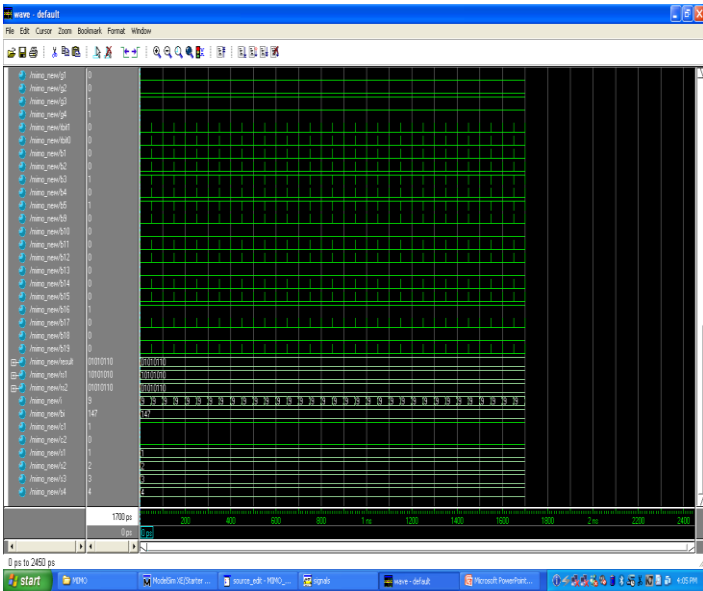


Figure 9 Proposed Work Simulation Result

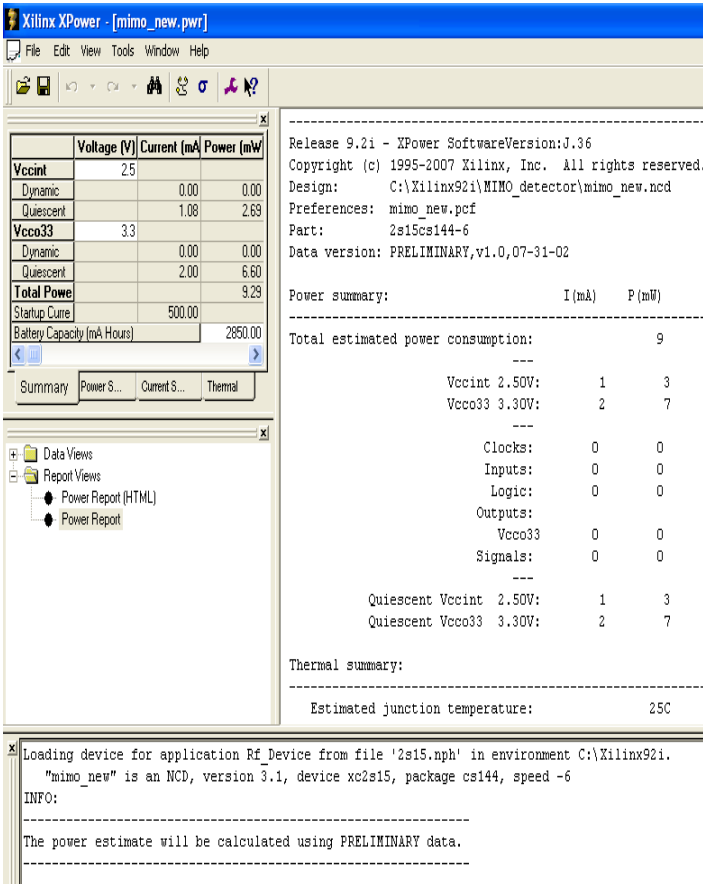


Figure 10 Power Consumption Report

Performance parameter	Phase-I Experimental Result	Phase-II Experimental Result
Power consumption	27 mW	9 mW
slices	430	10
LUT's	725	16
Latency	48.132ns	10.681ns

Table 1 Comparison Result Table

All the comparison results shown by the above table

6. CONCLUSION

In this paper, Based on the error-resilient MIMO detector. The results show that the detector power is reduced when compared with the existing systems. The proposed architecture was synthesized, placed and routed. Layout based performance and complexity benchmarks were presented. In 95 nm (transistor gate size) CMOS, it achieves a throughput of and the power consumption.

REFERENCES

- [1] "Algorithms and Architectures of Energy-Efficient Error- Resilient MIMO Detectors for Memory-Dominated Wireless Communication Systems" IEEE transactions on circuits and systems—i: regular papers, vol. 61, no. 7, July 2014
- [2] Performance Analysis of Low power Low-cost Signal detection of MIMO- OFDM using NSD International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-2, Issue-5, April 2013
- [3] L. Liu, F. Ye, X. Ma, T. Zhang, and J. Ren, "A 1.1-Gb/s 115-pJ/bit configurable MIMO detector using 0.13-um CMOS technology," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 57, no. 9, pp. 701–705, Sep. 2010.
- [4] M.Wenk, M. Zellweger, A. Burg,N. Felber, and W. Fichtner, "K-Best MIMO detection VLSI architectures achieving up to 424 Mbps," in Proc. IEEE ISCAS, May 2006, pp. 1151–1154.
- [5] S.Mondal, A. M. Eltawil, C.-A. Shen, and K. N. Salama, "Design and implementation of a sort-free K-Best sphere decoder," IEEE Trans (VLSI)Syst.[Online].Available:<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5313829>.
- [6] "A Radius Adaptive K-Best Decoder with Early Termination: Algorithm and VLSI Architecture" IEEE TRANSACTIONS ON circuits and systems—i: regular papers, vol. 57, no. 9, september 2010
- [7] M. Shabany and P. G. Gulak, "A 675 Mbps, 4×4 64-QAM K-best MIMO detector in 0.13 CMOS," IEEE Trans. Very Large Scale Integr.(VLSI) Syst., vol. 20, no. 1, pp. 135–147, Jan. 2012.
- [8] VLSI Implementation of MIMO Detection Using the Sphere Decoding Algorithm Andreas Burg, Member, IEEE, Moritz Borgmann, Student Member, IEEE, Markus Wenk, Martin Zellweger, Wolfgang Fichtner, Fellow, IEEE, and Helmut Bölcskei, Senior Member, IEEE
- [9] J. Ketonen, M. Juntti, and J. R. Cavallaro, "Performance—Complexity comparison of receivers for a LTE MIMO-OFDM system," IEEE Trans. Signal Process., vol. 58, no. 6, pp. 3360, 3372, June 2010.
- [10] A. M. A. Hussien, M. S. Khairy, A. Khajeh, A. M. Eltawil, and F. J.Kurdahi, "A class of low power error compensation iterative decoders," in Proc. 2011 IEEE Global Telecommun. Conf. (GLOBECOM 2011), Dec. 5–9, 2011, pp. 1, 6.
- [11] R. A. Abdallah and N. R. Shanbhag, "Error-resilient low-power Viterbi decoder architectures" IEEE Trans. Signal Process., vol. 57, no. 12, pp. 4906–4917, 2009.
- [12] C. Gimmmler-Dumont, M. May, and N. When,"Cross-layer error resilience and its application to wireless communication systems," J. Low Power Electron. (JOLPE), vol. 9, no. 1, Apr. 2013.
- [13] S.P.Prabu,S.Sobana (2014),"Performance andAnalysis of Low Power Error-Resilient MIMO Detectors"International Journal of Advanced and Innovative Research (IJAIR) IJAIR Volume 3 Issue 11, SI/No/34 (November Issue)