FPGA Implementation of High Throughput MIMO Detectors Based on Path Preserving Trellis Search Algorithm

S. Sheshana Priya[1], K. Senthil kumar[2]
[1]PG Student, [2]Assistant professor
Department of electronics and communication engineering
DMI college of engineering, Chennai-602013, India

***Abstract*— In this paper , novel path preserving trellis search algorithm (PPTS) and its high speed VLSI architecture (FPGA) for soft-output multiple-input-multiple-output (MIMO) detection. In MIMO receiver design as the computational complexity increases exponentially with the number of antennas. So in order to overcome this complexity PPTS algorithms are introduced. The PPTS algorithm is a hardware-friendly data-parallel algorithm because the search operations are evenly distributed among multiple trellis node for parallel processing. The PPTS detector is guaranteed to have soft information for every possible symbol transmitted on every antenna so that log-likehood ratio (LLR) for each transmitted data bit can be more accurately formed. Simulating result shows that our proposed scheme PPTS algorithm can achieve low search complexity and high throughput mechanism.**

***Keywords*— FPGA, MIMO, soft output MIMO detectors , shortest path algorithms.**

I.INTRODUCTION

Multiple-input-multiple-output (MIMO) system have potential to increase spectral efficiency by transmitting independent data streams on multiple antennas. MIMO technologies have been adopted in many new wireless standards such as Wi -MAX and WLAN. Soft-output MIMO detection poses significant challenges to the MIMO receiver design as the computational complexity increases exponentially with the number of antennas. However, the optimal soft-decision detector, the maximum a posterior detector, will consume enormous computing power and require tremendous computational resources which make it infeasible to be used in a practical MIMO receiver. So efficient algorithms used to reduce the MIMO detection complexity. The MIMO detection problem is usually tackled based on tree-search algorithms. The tree-search algorithms can be often categorized into the depth first search algorithm and the breadth-first search algorithm. The sphere detection algorithm is a depth-first tree-search algorithm to find the closest lattice point. To provide soft information for outer channel de-coders, a modified version of the sphere detection algorithm, or soft sphere detection algorithm, is introduced. There are many implementations of sphere detector sphere detector suffers from non-deterministic complexity and variable-time throughput. The sequential nature of the depth-first tree-search process significantly limits the throughput of the sphere detector especially when the SNR is low. The k-Best algorithm is a fixed-complexity algorithm based on the breadth-first tree-search algorithm . But this algorithm tends to have a high sorting complexity to find and retain the best candidates, which limits the throughput of the detector especially when k is large. In tree-search algorithms is that the counter- hypotheses for certain bits are missing due to tree pruning. As a consequence of missing counter-hypotheses, the magnitude of the LLRs for certain bits cannot be determined, which will lead to performance degradation. To avoid the missing counter-hypothesis problem and to reduce the search complexity, we investigate high performance MIMO detection algorithms and high-speed VLSI architectures.

II.SYSTEM MODEL

We consider a spatial-multiplexing MIMO system with N_t transmit antennas and N_r receive antennas ($N_r > N_t$) . The MIMO transmission can be modeled as

$$Y = Hs + n \qquad (1)$$

Where H is a $N_r * N_t$ complex matrix and is assumed to be known perfectly at the receiver, S is a Nt * 1 transmit symbol vector S $[S_0\ S_1 \ldots.. S_{Nt-1}]^T$, is a received vector Y= $[Y_0\ Y_1 \ldots. Y_{Nr-1}]^T$, and is a vector of independent zero-mean complex Gaussian noise entries with variance per real component. A real bit-level vector $X_{K\,=}$ [$X_{K,0}$ $X_{K,1}$ …$X_{K,}$ M_{C-1}]^T is mapped to the complex symbol S_K , where the bth bit of X_k is denoted as $X_{k,b}$ and is the number of bits per constellation point. Throughout this paper, the complex symbol and its associated bit vector X_K will be used interchangeably.

The optimal MAP detector is to compute the log-likelihood ratio (LLR) value for the a posteriori probability (APP) of each transmitted bit. Assuming there is no a priori information for the transmitted bit, the LLR APP of each bit $X_{k,b}$ can be computed

$$LLR(xk,b) = \ln \frac{\sum_{s:x_{k,b}=+1} \exp(-1/2\sigma^2 \|y - H.s\|^2)}{\sum_{s:x_{k,b}=-1} \exp(-1/2\sigma^2 \|y - H.s\|^2)} \qquad (2)$$

With max-log approximation,

Note that to form LLR for bit $X_{k,b}$, both the ML hypothesis and the counter-hypothesis of bit $X_{k,b}$ are required. Otherwise, the magnitude of the LLR will be undetermined. If a (sorted) QR decomposition of the channel matrix according to H=QR is used, where Q and R refer to an N_r * N_t unitary matrix and N_t * N_r an upper triangular matrix is changed to

$$LLR(x_{k,b}) = 1/2\sigma^2(\min_{s:x_{k,b=-1}} d(s) - \min_{s:\, x_{k,b=+1}} d(s)) \tag{3}$$

Where the Euclidean distance d(s),

$$\text{d(s)} = \|y = R.s\|^2 = \sum_{k=0}^{N=1} |(y)k - (Rs)k)|^2 \tag{4}$$

III.PATH-PRESERVING TRELLIS-SEARCH ALGORITHM

Computing the bit LLR in (3) requires searching for a ML hypothesis and a counter-hypothesis of this bit over a large search space. To reduce the search complexity and avoid the missing counter-hypothesis problem, we introduce a path-preserving trellis-search (PPTS) algorithm for soft MIMO detection.

A. Unconstrained Trellis Model

The Euclidean distance in (4) can be computed recursively. To visualize the recursion, we create a graph model, which resembles an unconstrained “trellis”. As an example, Fig. 1 shows the trellis graph for the 4× 4 4-QAM system. In this graph, nodes are ordered into N_t vertical slices or stages, where stage k corresponds to symbol S_k transmitted by antenna . The trellis starts with a root node and ends with a dummy sink node. The stages are labeled in descending order. In each stage, there are Q = 2^{M_c} different nodes, where each node maps to a constellation point that belongs to a known alphabet. Thus, each transmitted symbol vector is a path from root to sink.

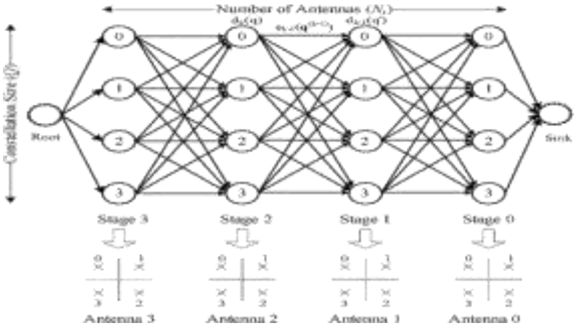

Fig 3.1 Trellis graph for the 4×4 4-QAM system

B. Multiple Shortest Paths Problem

We transform the soft MIMO detection problem into a multiple shortest paths problem. In the trellis graph, each trellis node $\langle k, q\rangle$maps to a complex symbol s_k such that each path from root to sink maps to a symbol vector S. A path weight is a measurement of the soft probability for nodes (symbols) on this path. To make a soft decision for every transmitted bit, finding one shortest path is not enough. We want to find multiple paths which cover every node in the trellis graph. The multiple shortest paths problem is defined as follows. For each ⟨k, q⟩ node in the trellis graph, find a shortest path from root to sink that must visit this node ⟨k, q⟩. The corresponding shortest path weight is related to the symbol probability . If we can find such a conditional shortest path for each node in the trellis, we will then have one soft information value for every possible symbol transmitted on every antenna. As a result, we will have sufficient soft information values to avoid the missing counter-hypothesis problem. Thus, the LLR for every data bit can be formed accurately based on these soft information values.

1V. VLSI ARCHITECTURE

In this section, we describe VLSI architectures for the proposed PPTS detector. We introduce a fully-parallel “systolic” architecture to achieve the maximum throughput performance, and a “folded” architecture to reduce area for lower throughput applications. For the sake of clarity, we describe a PPTS detector architecture with for the 4 × 4 16-QAM system. It should be noted that the architecture described can be easily scaled for other values of and other MIMO configurations.

A. Fully-Parallel Systolic Architecture

The fully-parallel “systolic” architecture for N_t = 4 antenna system. This architecture is fully pipelined so that it can process one MIMO symbol in every clock cycle. In this architecture, the main processing elements include three path reduction units (PRUs), three path extension units (PEUs), four path selection units (PSUs), and four LLR calculation (LLRC) units. The detailed structures of these processing elements will be described in the following subsections.

The three PRUs and one PSU are employed to implement the path reduction algorithm. The main diagonal of the systolic array is related to the path reduction data flow. The PRU implements one main iteration loop of Algorithm by employing path reduction processors to simultaneously process Q nodes in a certain stage implements PSU_0 the final selection step of Algorithm by using Q search units. The data flow for the path reduction is as follows. First, receives R, Y , and the precomputed $|y|$, and it computes all the path candidates β_2^m (i,j) in parallel, which are fed to the next PRU, i.e.,PRU_2. Then,PRU_2 computes β_1^m(i,j) , which are fed to PRU_1 , and so forth. Finally,PSU_0 receives $\beta_0^m(i,j)$ from PRU_1 and computes, which are fed to compute . The three PEUs and three PSUs are employed to implement the path extension algorithm. Each row (but the last) of the systolic array is related to the path extension data flow . The PEU implements one main iteration loop of Algorithm by employing path extension processors to simultaneously extend nodes in a certain stage. The PSU is used to implement the final selection step of Algorithm. The data flow for the path extension is as follows. To detect antenna, k ≥ 1, k-1 number of the PEUs and 1 PSU are used. Let = k-1 . First, $PEU_{k,t}$ receives $\beta_{k-1}^m(i,j)$ from PRU_K and it computes $Q^{(m)}(k,i,t-1,j)$, which are fed to $PEU_{k,t-1}$. Next, PEU $_{k,t-1}$ computes $Q^{(m)}$ (k,i,t-2,j) , which are fed to $PEU_{k,t-2}$, and so forth. Finally, PSU_k receives $Q^{(m)}(k,i,0,j)$ from PEU $_{k,1}$ and computes $\gamma^{(0)}(k,i,0)$, which are fed to $LLRC_k$.

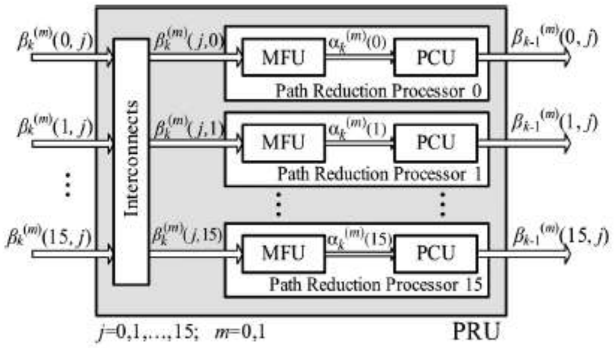

fig 4.1 structure of PRU

B. PRU

The structure of the PRU is shown in Fig 4.1 The PRU is used to implement the path reduction algorithm . The PRU employs Q = 16 path reduction processors to process all the nodes in a certain stage in parallel. Each path reduction processor contains one minimum (min) finder unit (MFU) and one path calculation unit (PCU), where the MFU is used to select the best paths $\alpha_k^{(i,j)}$from the incoming path candidates $\beta_k^{(m)}$ (i,j) the PCU is used to compute the new extended path candidates $\beta_{k-1}^{(m)}$(i,j).

1)MFU: The MFU is used to select the best M = 2 paths From QM = 32 path candidates. Fig. 10 shows the block diagram for the MFU unit which finds the minimum value Z_0 and the second minimum value Z_1 from its 32 data inputs (I_0 toI_{31}). The MFU is comprised of 16 CMP (comparison) units, 15 variable size C-S (compare and select) units, and one MIN unit. The CMP unit compares two data inputs and , and outputs the smaller one (or the “winner”): , and the larger one (or the “loser”):L= max(A,B) , and the sign:S=(A-B).

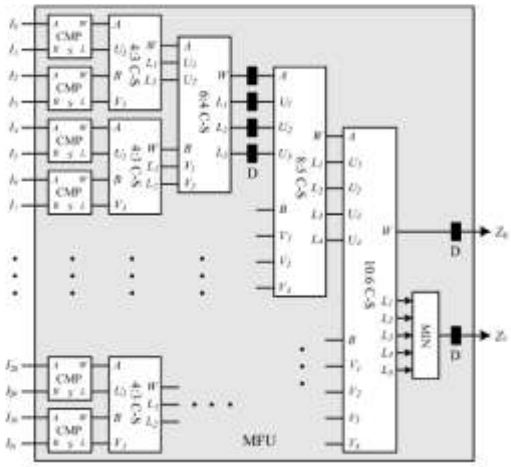

fig 4.2 block diagram for the MFU which use 16 comparator units

Outputs W of the C-S unit are candidates for the second smallest data among all the P inputs. he MFU takes data inputs and feeds them to 16 CMP units, where each CMP unit generates the winner and the loser of its two data inputs. The connection of the computational blocks in the MFU resembles a tree-like structure. Every two CMP units are connected to one 4:3 C-S unit, where the outputs of the 4:3C-S unit are the winner of its four data inputs, and two candidates for the second winner. Every two 4:3 C-S units are connected to one 6:4 C-S unit, where the outputs of the 6:4 C-S unit are the smallest data among its 6 data inputs ,and three candidates for the second smallest data .Similarly, every two 6:4 C-S units are connected to one 8:5 C-S unit, and two 8:5 C-S units are connected to a final 10:6 C-S unit. Finally, output of the 10:6 C-S unit is the smallest data among the 32 data . Outputs of the 10:6 C-S unit are the five candidates for the second smallest data among the 32 data inputs. A MIN unit is used to generate the second smallest data .

*2)PEU:*The PEU implements the path extension algorithm . The PEU has a very similar architecture to the PRU. . Each path extension processor contains one MFU and one PCU, where the MFU is used to select the best M paths $\gamma^{(m)}$ (k,i,t) from QM path candidates $\theta^{(m)}$(k,i,t,j) and the PCU is used to calculate the QM new extended path candidates $\theta^{(m)}$(k,I,t-1,j.)

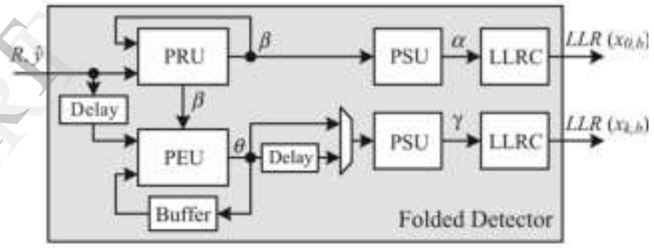

Fig 4.3, folded architecture for the PPTS detector

3)Throughput Performance of the Systolic Architecture

The proposed systolic MIMO detector architecture can provide very high throughput performance. This architecture is fully pipelined so that it can process one MIMO symbol in every clock cycle. Generally, if we let the clock frequency be MHz, then the throughput (Mbps) for a N_t ×N_r Q–QAM system can be expressed as

$$throughput_{systolic} = N_t . log_2 Q . fclk \quad (5)$$

As an example, assuming a system clock of 400 MHz, the systolic architecture can provide a throughput of 6.4 Gbps for a 4 ×4 16-QAM system.

V. SIMULATION RESULT

The proposed systolic PPTS detector with can achieve a 6.4 Gbps throughput when running at 400 MHz. As can be seen from the table, the proposed detectors achieve very high data throughput while still maintaining a low area requirement. Compared to the K-Best detector from the PPTS detectors M=2 with have a better area efficiency based on the area metric . The PPTS detector with achieves a balanced tradeoff between hardware complexity and error performance (0.3 dB loss). The PPTS detector with achieves a close-to-optimal error performance with a reasonable hardware cost. Therefore, the proposed detector is a viable solution for the Gbps MIMO detection problem as it achieves both high throughput performance and good error performance.

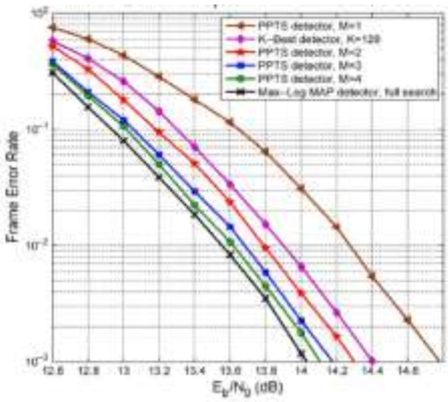

fig 5.1 Error performance of a coded 4×4 4QAM MIMO using PPTS detection algorithm.

VLSI Implementation Result,

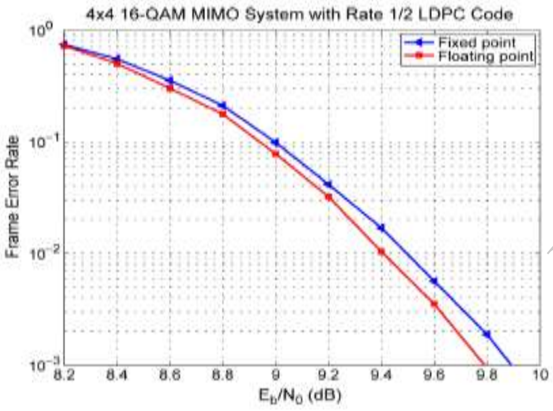

fig 5.2 simulation result for the proposed PPTS detector

VI. CONCLUSION

We propose a novel soft-output MIMO detector architecture based on the PPTS algorithm. The PPTS algorithm is a search efficient algorithm based on a path-preserving trellis search approach. We introduce a path reduction and a path extension algorithm to reduce the search complexity while still maintaining sufficient soft information values to form the LLRs. We avoid the missing counter-hypothesis problem by keeping multiple paths during the trellis search process. Simulation results show that the PPTS algorithm can achieve very good error performance with a small M value. The proposed MIMO detector has great potential for application in the next generation Gbps wireless systems by achieving very high throughput and good error performance.

VII. REFERENCES

1. G. J. Foschini, *"Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas,"* *Bell Labs Techn. J.*, vol. 1, no. 2, pp. 41–59, 1996.
2. G. Fettweis, T. Hentschel, and E. Zimmermann, *"WIGWAM—Awireless gigabit system with advanced multimedia support,"* in *Proc. VDE Kongress*, 2004, pp. 18–20.
3. U. Fincke and M. Pohst, *"Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," Math. Comput*, vol. 44, no. 170, pp. 463–471, Apr. 1985.
4. E. Viterbo and J. Boutros, *"A universal lattice code decoder for fading channels," IEEE Trans. Inf. Theory*, vol. 45, no. 5, pp. 1639–1642, Jul. 1999.
5. M. O. Damen, H. E. Gamal, and G. Caire, *"On maximum-likelihood detection and the search for the closest lattice point," IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2389–2402, Oct. 2003.