A New MIMO Detector Architecture Based on A Forward-Backward Trellis Algorithm

Yang Sun and Joseph R. Cavallaro
Department of Electrical and Computer Engineering
Rice University, Houston, TX 77005
Email: {ysun, cavallar}@rice.edu

***Abstract*— In this paper, a recursive Forward-Backward (F-B) trellis algorithm is proposed for soft-output MIMO detection. Instead of using the traditional tree topology, we represent the search space of the MIMO signals with a fully connected trellis and a Forward-Backward recursion is applied to compute the *a posteriori* probability (APP) for each coded data bit. The proposed detector has the following advantages: a) it keeps a fixed throughput and has a regular datapath structure which makes it amenable to VLSI implementation, and b) it attempts to maximize the *a posteriori* probability by tracing both forward and backward on the trellis and it always ensures that at least one candidate exists for every possible transmitted bit $x_k \in \{-1, +1\}$. Compared with the soft K-best detector, the proposed detector significantly reduces the complexity because sorting is not required, while still maintaining good performance. A maximum throughput of 533Mbps is achievable at a cost of 576K gates for 4×4 16-QAM system.**

I. Introduction

The depth first Sphere Decoder (SD) [1][2] and the breadth first K-best [3][4] algorithms have been proposed by researchers to achieve maximum *a posteriori* (MAP) decoding for coded MIMO systems. The depth first SD algorithm has non-deterministic complexity and variable throughput which makes it sensitive to the channel conditions. The performance of the depth first SD with a small list size suffers degradations due to the inaccurate and especially the infinite log likelihood ratio (LLR). On the other hand, the K-best algorithm has an advantage in hardware implementations since it has fixed complexity, throughput and latency. However, when K is large, the complexity of the K-best algorithm dramatically increases because a large number of paths have to be extended and sorted.

In this paper, a new efficient Forward-Backward (F-B) trellis searching algorithm and its VLSI architecture is introduced for high throughput soft-output MIMO detection. It is based on a suboptimal double-direction trellis traversal algorithm. This algorithm always ensures that a full Euclidean distance will be found for every possible transmitted bit, therefore it avoids the LLR clipping issues that both depth-first SD and K-best detectors have. The soft K-best algorithm usually does not perform backward tree traversal which limits its performance due to the inaccurate LLR generation. In our approach, we add a new feature by traveling both forward and backward in the trellis to generate a more accurate LLR for each coded bit. This low-latency detector offers a good solution for high throughput MIMO detection.

II. System Model

We consider a coded MIMO system with M transmit antennas and N receive antennas. The MIMO transmission can be modeled as:

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{n}. \tag{1}$$

where $\mathbf{H}$ is an $N \times M$ complex matrix, $\mathbf{s} = [s_0, ..., s_{M-1}]^T$ is a transmitted vector, $\mathbf{y}$ is a $N \times 1$ received vector and $\mathbf{n}$ is a complex Gaussian noise vector with variance σ^2. The symbol vector $\mathbf{s}$ is obtained using the mapping function $s_m = \text{map}(\mathbf{x}), m = 0, ..., M-1$ where $\mathbf{x}$ is an $M_c \times 1$ vector of data bits, and M_c is the number of bits per constellation symbol.

The soft-output detector is to compute the APP L-value of the bit $x_k, k = 0, ..., M \cdot M_c - 1$ as

$$L_D(x_k|\mathbf{y}) = \ln \frac{P[x_k = +1|\mathbf{y}]}{P[x_k = -1|\mathbf{y}]} = L_A(x_k) + L_E(x_k), \tag{2}$$

where L_A and L_E denote the *a priori* L-value and extrinsic L-value, respectively. Using Max-Log approximation, $L_D(x_k|\mathbf{y})$ can be simplified to [4][5]

$$L_D(x_k|\mathbf{y}) \approx \min_{\mathbf{x} \in \mathbb{X}_{k,-1}} \Lambda(\mathbf{s}, \mathbf{y}, \mathbf{L}_A) - \min_{\mathbf{x} \in \mathbb{X}_{k,+1}} \Lambda(\mathbf{s}, \mathbf{y}, \mathbf{L}_A), \tag{3}$$

where $\mathbb{X}_{k,\pm 1} = \{\mathbf{x} | x_k = \pm 1\}$, and

$$\Lambda(\mathbf{s}, \mathbf{y}, \mathbf{L}_A) = \frac{1}{\sigma^2} \|\mathbf{y} - \mathbf{H} \cdot \mathbf{s}\|^2 - \frac{1}{2} \mathbf{x}^T \cdot \mathbf{L}_A. \tag{4}$$

Using QR decomposition according to $\mathbf{H} = \mathbf{Q}\mathbf{R}$, where $\mathbf{Q}$ and $\mathbf{R}$ refer to an $N \times M$ unitary matrix and an $M \times M$ upper triangular matrix, respectively, we can write (4) as

$$\Lambda(\mathbf{s}, \mathbf{y}, \mathbf{L}_A) = \frac{1}{\sigma^2} \|\hat{\mathbf{y}} - \mathbf{R} \cdot \mathbf{s}\|^2 - \frac{1}{2} \mathbf{x}^T \cdot \mathbf{L}_A + C. \tag{5}$$

where $\hat{\mathbf{y}} = \mathbf{Q}^H \mathbf{y}$, and C is a constant that does not affect the minimizations in (3).

Solving (3) requires exhaustive search for each bit x_k. In order to reduce the complexity, conventional SD or K-best detectors can be used to generate a list of $\mathcal{L}$ candidates which have the smallest Euclidean distance to approximate (3).

978-1-4244-2941-7/08/$25.00 ©2008 IEEE

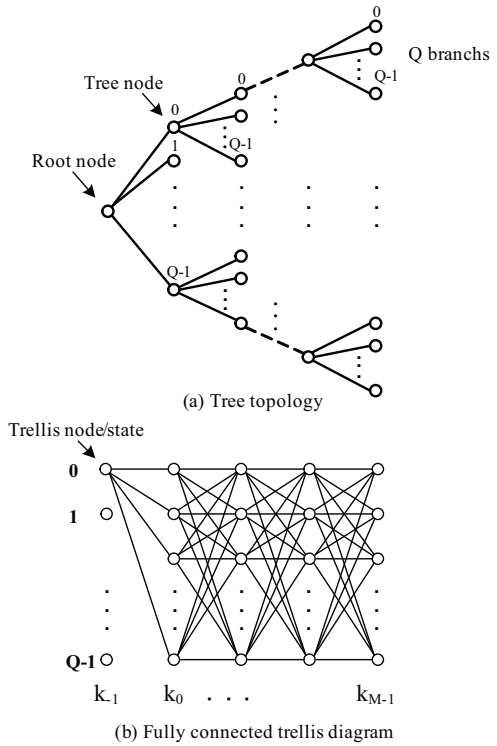

Fig. 1. Tree topology and its equivalent trellis diagram

III. Proposed F-B MIMO Detection Algorithm

We represent the search-tree with its equivalent trellis diagram as shown in Fig. 1 and apply the recursive Forward-Backward algorithm to solve (3). Fig. 1(b) shows a Q ($Q = 2^{M_c}$) state fully connected trellis, numbered from 0 to $Q-1$. There are M steps in the trellis diagram numbered from 0 to $M-1$, note that k_{-1} corresponds to the tree root node in Fig. 1(a). The trellis is fully connected in that each state/node has Q input paths and Q output paths. A path in the trellis can be represented by a state sequence $\{q_0, q_0, .., q_{M-1}\}$, which indicates the trellis path starting at state q_0, passing through every state q_k at time k, and terminating at state q_{M-1}.

To describe the baseline F-B algorithm, for now, we assume there is no *a priori* L-value L_A for the detector. Hence solving (3) is equivalent to find the minimum Euclidean distance

$$\Omega = \|\hat{\mathbf{y}} - \mathbf{R}\cdot\mathbf{s}\|^2 \tag{6}$$

for each coded bit x_k. The F-B algorithm is described as follows:

A) Forward Recursion:

Let $\alpha_k(q)$ be the state metric, which represents the partial Euclidean distance, for state q ($q = 0, 1, ..., Q-1$) at step k ($k = 0, 1, ..., M-1$). Let $\gamma(q', q)$ denote the branch metric from state q' to q. Let the history of the best forward path for state q at step k be stored in an array $\phi_k^q(j)$, where j is the index of the array $0 \le j \le M-1$. The Forward recursion, which searches from antenna $M-1$ to antenna 0, is summarized as follows:

1. Input R, $\hat{\mathbf{y}}$ and Initialization $k = -1$:

$$\begin{aligned}\alpha_k(i) &= \begin{cases} 0, i = 0 \\ \infty\ (1 \le i \le Q-1)\end{cases} \\ \phi_k^i(0) &= \emptyset,\ 0 \le i \le Q-1\end{aligned} \tag{7}$$

2. Forward recursion $k = 0, 1, ..., M-1$:
For each state q ($0 \le q \le Q-1$)

$$\alpha_k(q) = \min_{0\le q'\le Q-1}\{\alpha_{k-1}(q') + \gamma_k(q', q)\}, \tag{8}$$

where the branch metric

$$\gamma(q', q) = \left|\hat{y}_{M-1-k} - \sum_{j=M-1-k}^{M-1} R_{M-1-k,j}\cdot s_j\right|^2. \tag{9}$$

The complex transmitted symbol s_j in (9), is formed by using the constellation mapping function:

$$s_j = \begin{cases} \text{map}(q), & j = M-1-k \\ \text{map}(\phi_{k-1}^{q'}(M-1-j)) & j > M-1-k\end{cases}. \tag{10}$$

After the minimum $\alpha_k(q)$ for each state q is found, the forward history path array ϕ_k^q is updated by

$$\begin{aligned}\tilde{q} &= \underset{0\le q'\le Q-1}{\text{argmin}}\{\alpha_{k-1}(q') + \gamma_k(q', q)\} \\ \phi_k^q(i) &= \begin{cases} q, & i = k, \\ \phi_{k-1}^{\tilde{q}}(i), & 0 \le i \le k-1\end{cases}.\end{aligned} \tag{11}$$

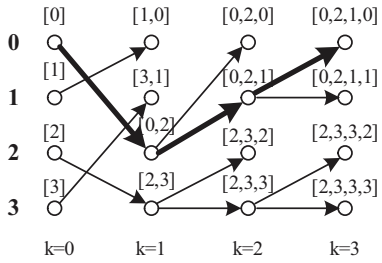

Fig. 2. Example of a 4-state trellis after forward recursion.

A forwarding recursion example for a 4×4 QPSK system is illustrated in Fig. 2, where the forward history path array ϕ_k^q is shown for each state node. For simplicity, only the survivor path for each state is shown. This algorithm can be used to find the ML path for the trellis, which is highlighted with a bold line. However, our goal is to find the minimum Euclidean distance Ω for every coded bit. Except for the first antenna (k=3), not every state node has a fully extended path in Fig. 2. This is because of the greedy path selection algorithm: only the best path will be retained for each node. In order to find a minimum full Euclidean distance for every state node in each antenna, a backward recursion is performed after the forward recursion.

B) Backward Recursion:

Similarly, let $\beta_k(q)$ be the backward state metric. The backward recursion, which searches from antenna 0 to antenna $M-1$, is summarized as follows:

1. Input R, $\hat{\mathbf{y}}$, array ϕ_k^q and Initialization $k = M-1$:

$$\beta_k(i) \quad = \quad 0, \ 0 \le i \le Q-1 \tag{12}$$

2. Backward recursion $k = M-2, ..., 0$:
For each state q $(0 \le q \le Q-1)$

$$\beta_k(q) \quad = \quad \min_{0 \le q' \le Q-1} \left\{ \beta_{k+1}(q') + \tau_k(q', q) \right\}, \tag{13}$$

where the backward branch metric

$$\tau(q', q) \quad = \quad \left| \hat{y}_{M-2-k} - \sum_{j=M-2-k}^{M-1} R_{M-2-k,j} \cdot s_j \right|^2 . \tag{14}$$

The symbol s_j in (14) is formed by using the forward path array ϕ_k^q and the incoming state q'

$$s_j = \begin{cases} \text{map}(q'), & j = M-2-k, \\ \text{map}(\phi_k^q(M-1-j)), & j > M-2-k. \end{cases} \tag{15}$$

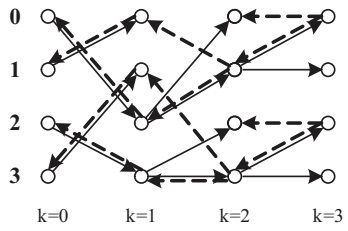

Fig. 3. Example of a 4-state trellis after the backward recursion.

Fig. 3 shows the trellis diagram after the backward recursion. The dotted and the solid lines denote the survivor pathes for the backward recursion and the forward recursion, respectively. The backward β metric is an approximation for the partial Euclidean distance (PED). Combining forward α metric and backward β metric will give us a good approximation of the minimum Euclidean distance for each trellis node. If we examine the trellis diagram after the backward recursion, every state node now has a fully extended minimum path metric Ω that can be used to generate the APP L-value. For pipelined implementation, the APP L-value can be generated in parallel with the backward recursion as shown in Fig. 6.

B.2) LLR Generation $k = M-1, ..., 0$:

$$\Omega_k(q) \quad = \quad \alpha_k(q) + \beta_k(q), \ 0 \le q \le Q-1$$

$$L_D(\mathbf{x}_{M-1-k,j}) \quad = \quad \frac{1}{\sigma^2} \left(\min_{q^{<j>}=-1} \Omega(q) - \min_{q^{<j>}=+1} \Omega(q) \right)$$

where $\Omega_k(q)$ is an approximation of the minimum Euclidean distance by combining the forward and backward metric for state q at step k, and $L_D(\mathbf{x}_{M-1-k,j})$ is the APP L-value for the j-th bit of the transmitted data vector $\mathbf{x_{M-1-k}}$, and $q^{<j>} = \pm 1$ is a sub set of $\{q\}$ $(q = 0, 1, ...Q-1)$ with its j-th bit equal to ± 1 $(0 \le j \le \log_2(Q) - 1)$.

IV. Simulation Results

To evaluate the detection performance, we consider 4×4 16-QAM and 64-QAM MIMO systems (the channel matrices are assumed to have independent Rayleigh fading distribution). In the simulation, the soft-output of the detector is fed to a length 2304, rate 1/2 LDPC decoder [6], which performs up to 15 iterations. Fig. 4 and Fig. 5 show the bit error rate performance for the proposed F-B detector and the soft K-best complex detector with different K values. For the 4×4 16-QAM system, our F-B detector outperforms the K-best detector for K=16 and 32, and achieves similar performance compared with K=64. The same trend has been observed for the 4×4 64-QAM system where our F-B detector outperforms the soft K-best detector with K=32, 48 and 64.

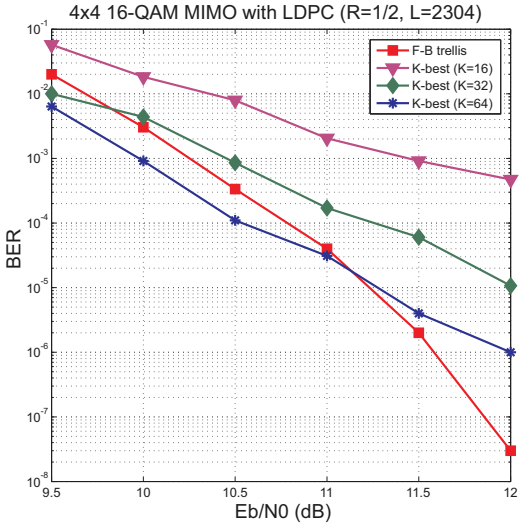

Fig. 4. Simulation results for 4×4 16-QAM MIMO system.

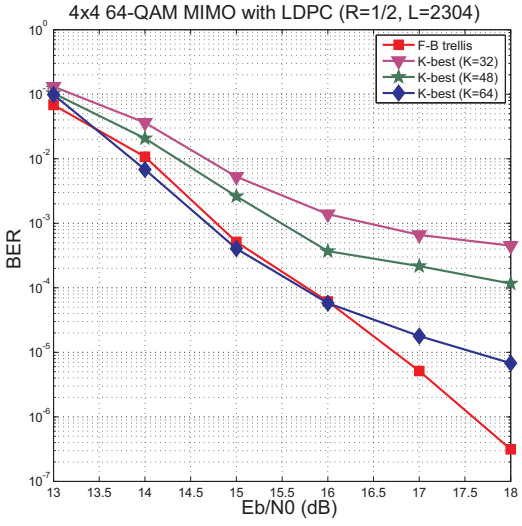

Fig. 5. Simulation results for 4×4 64-QAM MIMO system.

V. Architecture Design

Fig. 6 shows the proposed architecture based on the F-B algorithm. The proposed architecture is very suitable for VLSI implementation because it has a regular data flow, fixed complexity , and fixed throughput and latency. Compared with the soft K-best detector, our architecture has less complexity since no sorting is required. Only finding the minimum value is required for the proposed architecture. Therefore, the critical path of the F-B detector would be shorter than the K-best detector. The latency is also reduced.

A tile chart is used to represent the detector data flow, which is shown in Fig. 7. The X-axis represents the MIMO symbol sequences and the Y-axis represents the decoding time. In Fig. 7, the forward recursion (**F**) is followed by the backward recursion (**B**), and the **LLRs** are generated in parallel with the backward recursion.

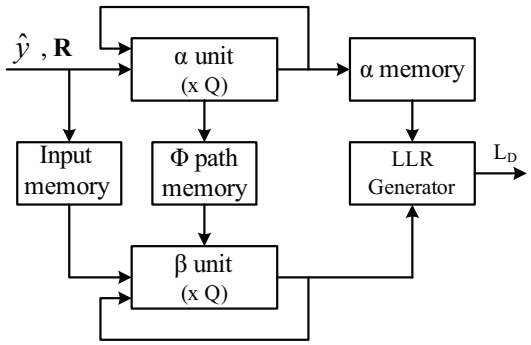

Fig. 6. F-B detector architecture.

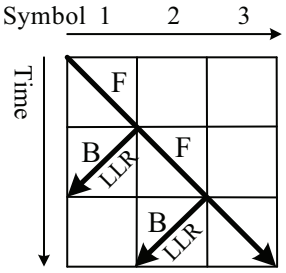

Fig. 7. Detection data flow tile chart.

A. Partial Euclidean Distance Computation Unit

The main computation units of this detector are the α and β units. Since α and β units have very similar structures, we use node processing unit (NPU) hereinafter to represent them. NPU is responsible for calculating the partial Euclidean distance (PED) and finding the minimum PEDs among all the candidates. Because of the upper triangular property of the $\mathbf{R}$ matrix, the PED can be computed in a recursive way which is shown below:

$$\begin{aligned} d_i &= d_{i+1} + ||t_{i+1} + R_{i,i}S_i||^2 \\ t_{i+1} &= \sum_{j=i+1}^{M-1} R_{i,j}S_j - \hat{y}_i, \end{aligned} \tag{16}$$

where d_M is initialized to be 0, and d_0 is the full Euclidean distance.

Since the trellis structure is fully connected, each state node needs to compute Q PEDs. Fig. 8 shows the architecture of the partial Euclidean distance computation unit (PEU). For simplicity, we assume a QPSK modulation scheme with M transmit and receiving antennas. In Fig. 8, SADD stands for shift and add which implements $R_{i,j}S_j$, where Cx ($x = 0, 1, ..Q-1$) are the constant constellation points. For the QPSK scheme, $Q = 4$, new partial Euclidean distances (NPEDs) are computed and are sent to the compare and select unit for further processing.

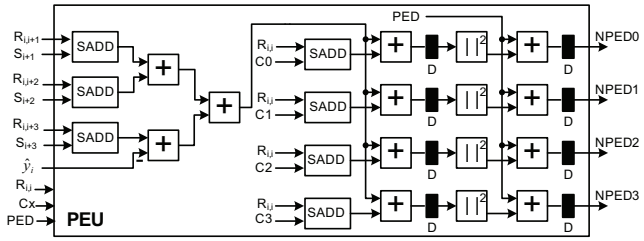

Fig. 8. PEU architecture for QPSK system.

B. Compare and Select Unit

The compare and select (CSU) unit, which is shown in Fig. 9, is used to find the minimum PED from Q input NPEDs.

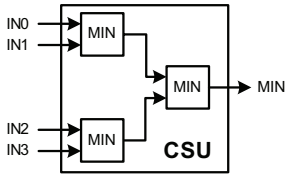

Fig. 9. Compare and select architecture for QPSK system.

C. Node Processing Unit

In each step of the trellis algorithm, Q state nodes are working independently and therefore can be processed in parallel. By instantiating $Q = 4$ PEUs and CSUs, the top level node processing unit (NPU) for QPSK systems is shown in Fig. 10. This is an iterative hardware architecture which implements (16). And the latency for one iteration is 3 cycles.

D. Architecture for Higher order Modulation Systems

We have shown the hardware architecture for QPSK systems, now we will extend it for higher order modulation schemes such as 16-QAM. The PEU-E and CSU-E in Fig. 11 are extensions of the PEU and CSU by replicating the hardware four times to support 16-QAM system. The PEU-E unit is used for computing 16 branches metrics. And the CSU-E unit is used for selecting the minimum PED from 16 candidates.

Based on the PEU-E and CSU-E units, the node processing unit (NPU) for a 16-QAM system has a very similar architecture as the QPSK system. As shown in Fig. 12, 16 PEU-Es

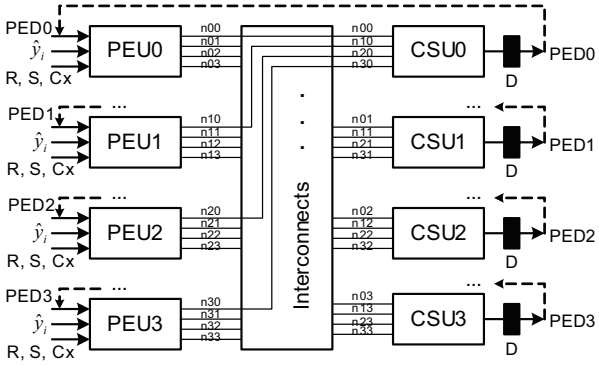

Fig. 10. NPU architecture for QPSK system

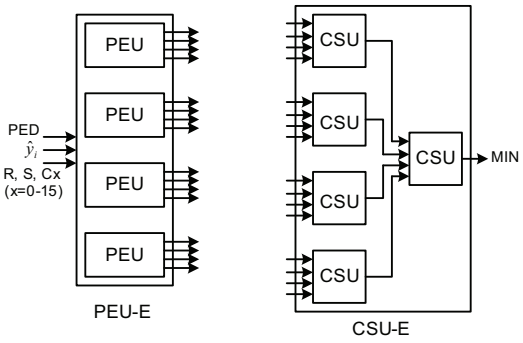

Fig. 11. PEU and CSU architecture for 16-QAM system

and CSU-Es are instantiated so that 16 nodes can be processed in parallel. The latency for each iteration remains to be 3 cycles. Therefore, the throughput for a 16-QAM system will be increased to 2 times that of the QPSK system.

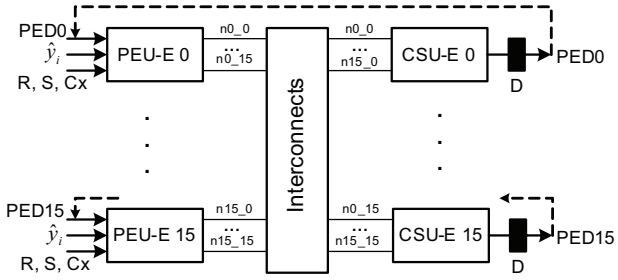

Fig. 12. NPU architecture for 16-QAM system

E. Hardware Complexity and Throughput Analysis

Table I shows the hardware complexity, detection throughput, and latency analysis for 4×4 QPSK and 16-QAM systems. The gate count estimation is based on a TSMC 65nm standard cell CMOS library. The highest clock frequency that the detector can achieve is about 400MHz. The decoding latency for a 4×4 system is $3 \times M = 12$ cycles.

Table II compares the detection throughput and hardware complexity of the proposed F-B solution versus two hardware implementations from the literature: depth-first soft sphere detector with 256 search operations (fclk=122.88MHz) from [1], and soft K-best detector (fclk=200MHz) from [4]. In [4], a real QR decomposition is used with a small K=5. Based on the simulation results in Fig. 4, our solution has a better BER performance than [4] and can achieve a faster throughput because we limit the number of sorting operations which is very expensive in the hardware implementation. On the other hand, at a cost of more hardware resources, the depth-first detector in [1] has a better BER performance than our solution. However [1] has a limited throughput because of the large number of sequential searching operations and the most undesired feature of [1] is its variable throughput at different SNR levels. Our architecture provides a good solution in between the depth-first detector and the K-best detector.

TABLE I
COMPLEXITY AND THROUGHPUT/LATENCY ANALYSIS

	Gate count	Max throughput	Latency
4×4 QPSK	36 K	266 Mbps	12 Cycles
4×4 16-QAM	576 K	533 Mbps	12 Cycles

TABLE II
COMPARISON OF SOFT 4×4 16-QAM MIMO DETECTORS

	[1]	[4]	This work
Throughput	38.8 Mbps	106 Mbps	533 Mbps
Gate count	1100 K	97 K	576 K

VI. CONCLUSION

We propose a new MIMO detector architecture based on the Forward-Backward recursion algorithm. This scheme can achieve very high throughput and can be easily parallelized. Both throughput and latency is deterministic, hence it is very suitable for hardware implementation.

VII. ACKNOWLEDGEMENT

This work was supported in part by Nokia and by NSF under grants CCF-0541363, CNS-0551692, and CNS-0619767.

REFERENCES

[1] D. Garrett, L. Davis, S. ten Brink, B. Hochwald, and G. Knagge, "Silicon Complexity for Maximum Likelihood MIMO Detection Using Spherical Decoding," *IEEE J. Solid-State Circuit*, vol. 39, pp. 1544–1552, Sep 2004.

[2] P. Radosavljevic and J. R. Cavallaro, "Soft Sphere Detection with Bounded Search for High-Throughput MIMO Receivers," in *IEEE Asilomar Conf. on Signals, Syst. and Computers*, Oct 2006, pp. 1175–1179.

[3] K. Wong, C. Tsui, R. Cheng, and W. Mow, "A VLSI architecture of a K-best lattice decoding algorithm for MIMO channels," in *IEEE Int. Symp. on Circuits and Syst.*, vol. 3, May 2002, pp. 273–276.

[4] Z. Guo and P. Nilsson, "Algorithm and implementation of the K-best sphere decoding for MIMO detection," *IEEE J. Selected Areas in Commun.*, vol. 24, pp. 491–503, Mar 2006.

[5] B. Hochwald and S. Brink, "Achieving Near-Capacity on a Multiple-Antenna Channel," *IEEE Tran. Commun.*, vol. 51, pp. 389–399, Mar 2003.

[6] Y. Sun, M. Karkooti, and J. R. Cavallaro, "VLSI Decoder Architecture for High Throughput, Variable Block-size and Multi-rate LDPC Codes," in *IEEE Int. Symp. on Circuits and Systems*, May 2007, pp. 2104–2107.