Low Power Soft-Output Signal Detector Design for Wireless MIMO Communication Systems

Sizhong Chen and Tong Zhang
Electrical, Computer, and Systems Engineering Department
Rensselaer Polytechnic Institute, Troy, NY 12180
sizhong@alum.rpi.edu, tzhang@ecse.rpi.edu

ABSTRACT

Energy-efficient realization of soft-output signal detection is of great importance in emerging high-speed multiple-input multiple-output (MIMO) wireless communication systems. This paper presents three algorithm-level complexity-reduction techniques for soft-output detector design to achieve significant energy savings. To demonstrate their effectiveness, we designed a soft-output detector for 4×4 MIMO with 64-QAM using 65nm CMOS technology. While achieving near-optimum detection performance, the detector can support over 100Mbps throughput with only 0.24mm^2 silicon area and 11mw power, leading to a ×10 improvement over the state of the art.

Categories and Subject Descriptors

B.4.1 [**Input/Output and Data Communications**]: Data Communications Devices

General Terms

Algorithms, Design

Keywords

MIMO, detection, spatial multiplexing, low power, VLSI

1. INTRODUCTION

Multiple-input multiple-output (MIMO) wireless communication technology has attracted a lot of attentions in the past decade and evidently will play an essential role in future high speed wireless communication systems such as the fourth-generation (4G) mobile radio systems, fixed/mobile broadband wireless systems (WiMAX), and wireless local area networks (IEEE 802.11n) [5]. MIMO can be used to enhance the data transmission rate by spatial multiplexing or improve the transmission reliability by space-time coding. This paper concerns the MIMO system with spatial multiplexing which transmits independent data streams from each of the multiple transmit antennas within the same frequency band at the same time. Therefore, throughout this paper, *MIMO* always refers to the MIMO systems with spatial multiplexing. As the cost of the increased transmission rate, the computational complexity and hence the power consumption of MIMO signal detection tend to grow dramatically with the number of transmit antennas and the modulation constellation size. This is particularly true for soft-output MIMO detection that not only provides the binary estimation of each transmitted bit but also the reliability measurement of the binary estimation. Meanwhile, soft-output MIMO signal detection is almost indispensable in practice since most real-life wireless communication systems use error correcting codes (ECC) that demand soft input for decoding, such as convolutional codes, Turbo codes, and low-density parity-check (LDPC) codes. Therefore, effective design solutions for reduced-complexity low-power soft-output MIMO detector are highly desirable.

One family of reduced-complexity detectors is linear detectors based on the principles of zero-forcing (ZF) or minimum mean-square error (MMSE). Although they can greatly reduce the computational complexity, they suffer from significant performance degradation. The successive interference cancellation (SIC) detectors such as the VBLAST architecture [9] are prone to decision error propagation and can only provide modestly better performance. To achieve the performance closer or even equivalent to the optimum detection, researchers have developed several nonlinear detectors that realize hard- or soft- output detection through *non-exhaustive tree search* based on a set of additive metrics, where the goal of hard-output detection is to find one tree leaf with the best metric and the goal of soft-output detection is to find a list of tree leaves to calculate the reliability information.

This paper focuses on the tree-search soft-output MIMO detection because of its near-optimum detection performance. Since the recently approved IEEE 802.11n draft and the on-going 3GPP Long Term Evolution (LTE) project will use 4×4 MIMO transmission (i.e., 4 transmit antennas and 4 receive antennas) with 64-QAM modulation as the most complex MIMO configuration, we are primarily interested in the design and VLSI implementation of tree-search soft-output MIMO detector for 4×4 MIMO with 64-QAM. The authors of [4] developed a tree-search soft-output MIMO detector design solution based on several algorithm-level innovations and presented the corresponding ASIC design results for 4×4 MIMO with 64-QAM. To the best of our knowledge, it is the only tree-search soft-output MIMO detector design reported in the open literature that is capable of supporting our interested 4×4 MIMO with 64-QAM. However, its computational complexity and power consumption (as presented in the next section) are still very high and may be intolerable in many practical applications. In fact, even the previously reported tree-search detectors for 4×4 MIMO with only 16-QAM [6, 7] tend to have very high computational complexity and may be subject to high power consumption.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
ISLPED'07, August 27–29, 2007, Portland, Oregon, USA.
Copyright 2007 ACM 978-1-59593-709-4/07/0008 ...$5.00.

Therefore, the design of low-power tree-search soft-output MIMO detector still remain a big challenge and demand further innovations at algorithm/architecture levels.

As an attempt to tackle this challenge, this paper presents three algorithm-level techniques to largely reduce the computational complexity of conventional tree-search soft-output detection schemes. The proposed techniques include (i) bidirectional partial tree search, (ii) hybrid two-step detection, and (iii) ECC feedback aided detection bypassing. Computer simulations under the framework of MIMO-OFDM wireless communication systems show that these complexity-reduction techniques only incur very small detection performance degradation. To demonstrate their effectiveness on power saving, we designed an ASIC detector optimized for 4×4 MIMO with 64-QAM. We use Synopsys tools for simulation, synthesis, and timing/power analysis with 65nm CMOS standard cell and SRAM libraries. Post-synthesis results show that this detector can achieve above 100 Mbps throughput with the silicon area of 0.24mm^2 and power consumption of only 11mW, which represents a more than one order of magnitude improvement over the state of the art [4] in the open literature.

2. BACKGROUND

2.1 System Model

In MIMO system, on the transmitter side, one $qN_t \times 1$ binary vector $\mathbf{x}$ is mapped to an $N_t \times 1$ symbol vector $\mathbf{s}$ at once, where N_t represents the number of transmit antennas. Each transmitted symbol is taken from a W-QAM constellation with $W = 2^q$. The transmission of each vector $\mathbf{s}$ over flat-fading MIMO channels can be modeled as $\mathbf{y} = \mathbf{H} \cdot \mathbf{s} + \mathbf{n}$, where N_r represents the number of receive antennas, $\mathbf{y}$ is an $N_r \times 1$ signal vector received by a MIMO detector, $\mathbf{H}$ is an $N_r \times N_t$ channel matrix, and $\mathbf{n}$ is a noise vector whose entries are independent complex Gaussian random variables with mean zero and variance $N_0/2$.

2.2 Soft-Output MIMO Signal Detection

The task of a soft-output detector is to compute the log-likelihood ratio (LLR) value of each bit, defined as $L(x_i|\mathbf{y}) = \ln \frac{P(x_i=+1|\mathbf{y})}{P(x_i=-1|\mathbf{y})}$, where x_i denotes the i-th bit of the binary vector $\mathbf{x}$. Through standard simplification [2, 8], $L(x_i|\mathbf{y})$ can be approximated as:

$$\begin{aligned} &L(x_i|\mathbf{y}) \approx \max_{x\in\mathbb{X}_{i,+1}} \{\Lambda(\mathbf{x},\mathbf{y}\} - \max_{x\in\mathbb{X}_{i,-1}} \{\Lambda(\mathbf{x},\mathbf{y})\}, \\ &\text{where } \Lambda(\mathbf{x},\mathbf{y}) = -\frac{1}{N_0}\|\mathbf{y} - \mathbf{H}\cdot\mathbf{s}\|^2. \end{aligned} \tag{1}$$

Using standard matrix decompositions such as Cholesky or QR decomposition, we can obtain $\mathbf{H}^*\mathbf{H} = \mathbf{L}^*\mathbf{L}$, where $\mathbf{L} = (l_{i,j})$ is a lower triangular matrix and $(\cdot)^*$ denotes the complex conjugate transpose. Let $\hat{\mathbf{s}} = (\mathbf{H}^*\mathbf{H})^{-1}\mathbf{H}^*\mathbf{y}$, we have

$$\begin{aligned} \|\mathbf{y} - \mathbf{H}\cdot\mathbf{s}\|^2 &= (\mathbf{s}-\hat{\mathbf{s}})^*\mathbf{L}^*\mathbf{L}(\mathbf{s}-\hat{\mathbf{s}}) \\ &\quad + \mathbf{y}^*(\mathbf{I} - \mathbf{H}(\mathbf{H}^*\mathbf{H})^{-1}\mathbf{H}^*)\mathbf{y}. \end{aligned} \tag{2}$$

Since the second term in (2) is independent of $\mathbf{s}$ and the matrix $\mathbf{L}$ is lower triangular, we can rewrite $\Lambda(\mathbf{x},\mathbf{y})$ in (1) as

$$\Lambda(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{N_t} \Big(-\frac{1}{N_0}\Big|\sum_{j=1}^{i} l_{i,j}(s_j - \hat{s}_j)\Big|^2\Big) = \sum_{i=1}^{N_t} \Lambda_i^s. \tag{3}$$

Hence, we obtain *additive metrics* with the metric increment Λ_i^s that only depends on s_j for $j \le i$. Thus, soft-output MIMO detection can be formulated as an N_t-depth W-ary tree search problem, where the i-th depth of this tree corresponds to the i-th transmitted symbol (or i-th transmit antenna) and each tree node has W child nodes corresponding to the W possible QAM points. Since the term $-\frac{1}{N_0}$ in (3) can be omitted in the tree search, we define the metric increment Λ_i as $|\sum_{j=1}^{i} l_{i,j}(s_j - \hat{s}_j)|^2$ and the metric of a depth-n path as $\Gamma^{(n)} = \sum_{i=1}^{n} \Lambda_i$. Each tree leaf represents a distinct N_t-symbol vector that may be possibly transmitted from the transmitter.

Tree-search soft-output detector finds a set of tree leaves with good metrics (i.e., high probabilities of occurrence), based on which the L-values can be evaluated according to (1). Optimum soft-output detector exhaustively examine all the tree leaves, which nevertheless will result in prohibitive computational complexity. Computational complexity can be significantly reduced by performing non-exhaustive tree search at the cost of certain performance degradation. The essence of non-exhaustive tree search can be briefly described as follows: We depth-by-depth extend/examine the paths from root towards leaves, and at each depth we prune those *bad* paths with relatively low probabilities of occurrence from further examination. Hence, all the tree nodes under those pruned paths will not be examined, which will correspondingly reduce the computational complexity. At each depth, the tree paths that are not pruned are called survivor paths at present depth. Such non-exhaustive tree search can be realized in either depth-first or breath-first manner. This work is interested in soft-output detection based on breadth-first non-exhaustive tree search.

2.3 Breadth-First Tree Search *K*-Best Detector

Broadly speaking, breadth-first tree search extends all the survivor paths at each tree depth at once, purge some paths according to certain criterion, and then continue on to the next tree depth. The *M*-algorithm [1] is the most well-known breadth-first non-exhaustive tree search algorithm. Directly combining the principal of *M*-algorithm with the sphere constraint from the sphere decoding algorithm [8], we have the so-called *K*-best MIMO detectors [7, 10]. A *K*-best detector performs the following operations at each depth:

1. *Path Extension*: Extend each survivor path from the previous depth with the W modulation points, i.e., calculate $\Gamma^{(i)} = \Gamma^{(i-1)} + \Lambda_i$ for each modulation point.

2. *Radius Check*: Delete the extended paths whose metrics are larger than a fixed threshold r^2, that is equivalent to the radius constraint in sphere decoding algorithm.

3. *Path Sort*: Let *R* denote the number of the remaining extended paths. If $R > K$, then sort these *R* paths in ascending order based on the path metric and select the first *K* paths as survivors depth, otherwise all the *R* extended paths are survivors.

After reaching the tree leaves at depth N_t, the detector keeps all the survivors as a list of candidates, based on which the L-values are calculated according to (1). A straightforward implementation of *K*-best detectors tends to suffer from two critical drawbacks that prevent it from achieving high throughput with reasonable silicon area and power consumption, particularly for high order modulation such as 64-QAM. This is briefly explained as follows:

i) The detector *explicitly* examines the extension of each survivor with all the modulation points. Due to the complex computation involved in each path extension, it will incur a large computational complexity overhead. To solve this problem, PSK enumeration technique [3, 4, 8] can be used. Its basic idea is to use the radius constraint to determine an admissible region for the extension of each survivor path at each tree depth, and only examines the modulation points inside the admissible region for path extension. However, since the shape of the admissible region is a circle in PSK enumeration technique, it is not trivial to determine its boundary in hardware implementation. Moreover, since the modulation points are enumerated in a zigzag fashion, the data flow control circuitry becomes quite complex. These may result in non-negligible silicon area cost and energy overhead.

ii) The *search-the-best-K-paths* operation is typically realized by strict sorting (e.g., bubble sorting) in hardware implementation, which will result in significant silicon area cost and energy overhead. To tackle this issue, the authors of [4] proposed to replace the strict sorting with so-called *approximate sorting*. The basic idea of approximate sorting can be described as *sorting with a coarse granularity*: Given the fixed radius constraint r^2, we divide the entire range of the path metric (i.e., [0, r^2]) into a certain number of adjacent regions, and each region is associated with a lower-bound threshold and an upper-bound threshold. Using simple parallel comparison with the lower/upper-bound thresholds, we can arrange all the extended paths into groups corresponding to those path metric regions. Within each group, the paths are not sorted at all. The detectors using such technique is referred to as relaxed *K*-best detectors.

2.4 Results of Prior Work

Table 1 summarizes the hardware design results of tree-search soft-output MIMO detectors reported in the open literature. The soft-output List Sphere Decoder (LSD) [6] performs depth-first tree search and the parameter l denotes the size of the tree leaves list used for calculating the soft output. The *K*-best detector [7] and relaxed *K*-best detector [4] perform breadth-first tree search and the parameter *K* denotes the number of survivors kept at each tree depth. Notice that only the relaxed *K*-best detector can support 64-QAM, while the other two support 16-QAM.

Table 1: Comparison of soft-output MIMO detectors.

Algorithm	LSD (l=256) [6]	*K*-best (*K*=5) [7]	Relaxed *K*-best (*K*=64) [4]
Antennas	4×4		
Modulation	16-QAM	16-QAM	64-QAM
ECC	length-18432 rate-1/2 Turbo	length-18432 rate-1/2 Turbo	length-2304 rate-1/2 LDPC
Technology	0.18μm	0.13μm	0.13μm
Core Area (mm^2)	10	0.56	21.4
Gate Count	N/A	97K	2.5M
Throughput (Mbps)	38.4	106.6	77.1 @17.7dB SNR
Power (mw)	N/A	N/A	847

As suggested in Table 1, breadth-first tree search may be more suitable for high-speed soft-output MIMO detection compared with depth-first tree search. This motivates us to focus on the breadth-first tree-search detector in this work. The value of *K* heavily affects the complexity vs. performance tradeoff in breadth-first detector. The larger the *K* is, the better the detection performance will be but higher implementation complexity and energy consumption will be incurred. Moreover, a bigger modulation constellation size (e.g., 64-QAM vs. 16-QAM) demands a (much) larger value of *K*. Appropriate choice of *K* is also affected by the ECC being used. If a stronger ECC code (e.g., the very long length-18432 code vs. length-2304 code) is used, we can choose a smaller *K* given the same system error rate performance requirement. Unfortunately, in practical wireless communication systems, relatively short (and hence weaker) ECC codes with the codeword length of few thousands are typically used. Therefore, a relatively large value of *K* (i.e., 64) was chosen in [4] where a short ECC is used and the modulation size is 64-QAM. Nevertheless, as listed in Table 1, the silicon area and power consumption of the 64-QAM detector [4] are very high and may be intolerable to real-life wireless communication systems.

3. DEVELOPED COMPLEXITY/POWER REDUCTION TECHNIQUES

This section presents our proposed three algorithm-level techniques that can largely reduce the computational complexity and hence the power consumption of breadth-first tree-search soft-output detection.

3.1 Bidirectional Partial Tree Search

From Section 2.2, we know that each metric increment Λ_i at depth-i can be calculated as $|P_c - P_s|^2$, where $P_c = G_i - \sum_{j=1}^{i-1} l_{i,j} s_j$, $G_i = \sum_{j=1}^{i} l_{i,j} \hat{s}_j$ and $P_s = l_{i,i} s_i$. This suggests that the computational overhead for each path extension will increase as we search deeper towards the tree leaves (i.e., as i increases). We propose a method called bidirectional partial tree search to reduce the path extension computational complexity by limiting the tree search depth. As illustrated in Fig. 1, the basic idea is to perform partial tree search in both forward and backward directions. In the forward direction, we perform the breadth-first tree search over the first and second symbols, while keeping *k*=*K*/2 survivor paths at each tree depth. For the third and fourth symbols, each survivor path is simply extended to only one modulation point that is closest to $\frac{P_c}{l_{i,i}}$ at the i-th depth for $i = 3, 4$.

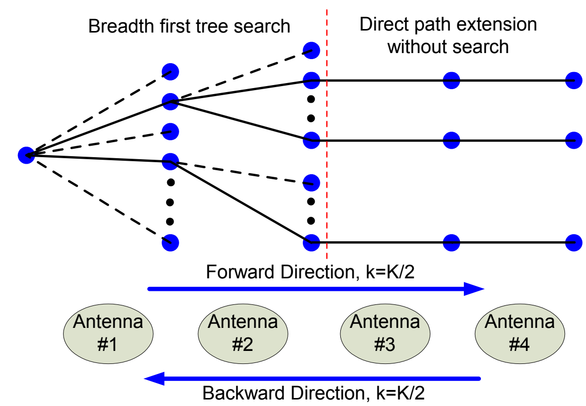

Figure 1: Bidirectional partial tree search diagram.

The same operation is performed in the backward direction by treating the fourth antenna as the root and first antenna as leaf. Notice that since the symbol vector is detected in the reverse order along the backward direction, the channel matrix is the up-down flipped version of the original channel matrix $\mathbf{H}$ and the channel matrix decomposition result used in (3) should be re-computed. After both forward and backward partial tree search are finished, we obtain a list of *K* paths based on which we can calculate the soft output according to (1). Using such bidirectional partial tree search approach, the computational complexity can reduce to only about 35% of a conventional K-best detector. Although the channel matrix decomposition has to be done twice, the cost can be marginal since the channel is typically considered to be constant during the transmission of several consecutive packets.

To evaluate the signal detection performance of the proposed bidirectional partial tree search approach, we performed computer simulations using the following system configurations that will also be used for all the other simulations throughout the paper: We consider LDPC-coded MIMO-OFDM system with 64-point FFT for 4×4 MIMO transmission with 64-QAM. Out of the 64 subcarriers, 48 are data carriers while the rest are used for pilots and virtual carriers, as defined in the IEEE 802.11a standard. Each subcarrier MIMO channel remains constant during transmission of one packet and is subject to flat fading, i.e., all the entries in the MIMO channel matrix are independent random Gaussian variables. Each packet is protected by a length-2304, rate-1/2 LDPC code, where the LDPC code decoder performs up to 15 decoding iterations. There is no iteration between MIMO detection and LDPC decoding. For the definition of the MIMO channel SNR, we follow the one proposed in [8]: Let R denote the channel code rate ($R = 1$ for uncoded systems), SNR is defined as:

$$\frac{E_b}{N_0}\Big|_{dB} = \frac{E_s}{N_0}\Big|_{dB} + 10\log_{10}\frac{N_r}{R \cdot N_t \cdot q},$$

where E_s denotes the average symbol energy of the QAM constellation. As pointed out earlier, the radius r^2 is calculated as $2\alpha N_r \sigma^2$ [8], where $\alpha = 6$ is a predefined constant parameter and σ is the noise standard deviation. For the purpose of comparison, we also carried out simulations using depth-first sphere decoding algorithm and conventional K-best detection scheme. The sphere decoder *exhaustively* examines all the paths that satisfy the sphere radius constraint for the calculation of the soft output. Therefore, it can achieve better performance than the LSD scheme used in [6] that does not exhaustively examines all the paths satisfying the sphere radius constraint. All the detectors use the same radius constraint. As shown in Fig. 2, the K-best and bidirectional detectors have almost the same performance. Both of them have only very small degradation compared to sphere decoder when K is large (e.g., 64). At very small value of K, the bidirectional K-best detector is even slightly better than the conventional K-best detector.

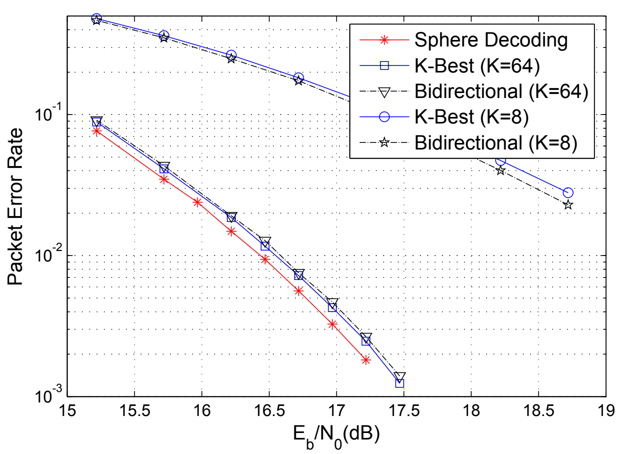

Figure 2: Simulated PER performance for 4×4 MIMO 64-QAM.

3.2 Hybrid Two-Step Detection

Practical wireless communication systems (such as IEEE 802.11n and 3GPP LTE) typically target on achieving the packet error rate (PER) of $10^{-2} \sim 10^{-3}$ at low SNR. Due to the significant impact of the value of K on the detection performance as shown in Fig. 2, we may need to use a large value of K (e.g., 64), which nevertheless tends to incur very high computational complexity. We note that, as shown in Fig. 2, although a small K (e.g., 8) will result in a big SNR loss at the target detection PER, it still can achieve fair PERs (close to 10^{-1}) in the low SNR range. This motivates us to propose a hybrid two-step technique to reduce the average detection computational complexity as described in the following. As shown in Fig. 3, in the first step, a simpler detector with K=8 is used. If the ECC decoder fails to decode the packet correctly, all the vectors in the packet will be re-detected with K=64 in the second step. Therefore, majority of the packets can be successfully processed with the simpler detector and only about 10% of packets need to be further processed by the much more costly detector with K=64. It should be pointed out that, based on our computer simulations, the iterative detection/decoding scheme (i.e., feedback the soft-output of the ECC decoder in the first step to the detection in the second step) fails to improve the overall system performance when the first-step detection is performed with very small K (e.g., 8). Therefore, we do not consider the use of iterative detection/decoding in this work.

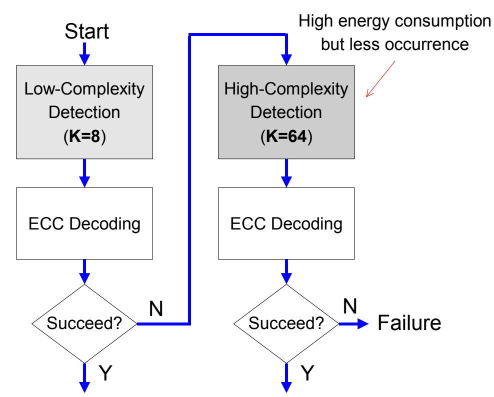

Figure 3: The flow diagram of the hybrid two-step detection.

To further simplify the search for the admissible region during the path extension, we propose to use rectangular shaped admissible region instead of the circular shaped admissible region in PSK enumeration technique. In the case of K=8, locating the admissible points is very easy, especially when the bidirectional partial tree search is used. Since each survivor only extend with k=K/2=4 modulation points, we only need to find the 4 modulation points around $\frac{P_c}{l_{i,i}}$ at the i-th depth as shown in Fig. 4. For K=64, a rectangular shaped region is formed by extending the inner 4-point square with the radius constraint as shown in Fig. 4. Although a few more points may be included in the rectangular shaped region than the circular shaped region, it is much easier to define the boundary and identify the admissible points. In the corresponding VLSI architecture design, the data flow will become very regular and the control circuitry can be significantly simplified. Moreover, we propose to realize path purge differently for K=8 and K=64: For K=8, we simply use the bubble sorter to select the best ones among all the extended paths, while for K=64, we propose to use the memory based-approximate sorter [4], as described in 2.3, in order to reduce the silicon area cost and power consumption.

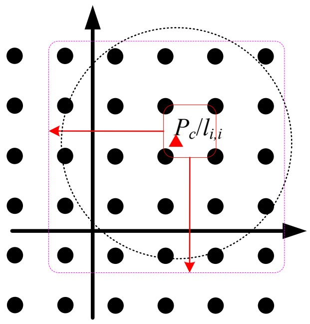

Figure 4: Rectangular shaped admissible region.

3.3 ECC Feedback Aided Detection Bypassing

In a straightforward realization of the above hybrid two-step detection, when the ECC decoder fails to decode a packet for the first time, the detector will repeat the detection with large K on *all* the vectors in the packet. However, our simulations show that it is typically not necessary to re-detect all the vectors since some vectors may already have good enough soft detection output after the first detection step. We further observed that, if the ECC decoder can generate soft output (e.g., the decoding of Turbo code and LDPC code), it is possible to use the corresponding ECC decoding soft output to distinguish the *good* vectors from *bad* vectors with reasonably good confidence and perform the second-step high-complexity detection only for those *bad* vectors. Through such detection bypassing on *good* vectors, further saving of computational complexity and power consumption can be achieved.

In this work, we propose the following procedure to identify those *bad* vectors. We claim a vector to be a *bad* vector if it contains more than α bad bits, where α is called vector classification threshold. The *bad* bits are identified as follows:

1. Let L_D denote the *a posteriori* soft-output of the ECC decoder for each bit. If the absolute value of L_D is less than a pre-specified threshold β_0, the corresponding bit is declared to be a *bad* bit.

2. Let L_{E1} denote the *extrinsic* soft-output of the detector for each bit during the first-step detection, which is also the *a prior* soft-input L_A of the ECC decoder. Let L_{E2} denote the *extrinsic* soft-output of the ECC decoder. We define a parameter γ as follows: When L_A and L_{E2} have the same sign, which means the detector and ECC decoder agree with each other on the bit decision, γ is 0. If L_A and L_{E2} have different signs, γ equals to $|L_A - L_{E2}|$. If γ is greater than a pre-specified threshold β_1, the corresponding bit is declared to be a *bad* bit.

To determine the appropriate value of α, β_0, and β_1, we typically rely on extensive computer simulations. For our interested 4×4 MIMO with 64-QAM under LDPC-coded MIMO-OFDM transmission environment as described in the above, Fig. 5 show the simulated probability distribution of $|L_D|$ (i.e., the absolute value of L_D) and γ, based on which we decide to choose $\alpha = 3$, $\beta_0 = 3$, and $\beta_1 = 10$. Our simulations show that, under such setup, about 50% of the vectors can be by-passed in the second-step detection.

Fig. 6 shows the simulation results to demonstrate the detection performance while using the above proposed three design techniques and their comparison with the sphere-constrained exhaustive-search sphere decoding and conventional K-best detection. The direct combination of these three techniques only incur about 0.5dB performance degradation. If we further use the memory-based approximate sorting to reduce the second-step detection complexity, another 0.1dB loss will be incurred. This leads to a total 0.6dB degradation compared with the sphere decoding.

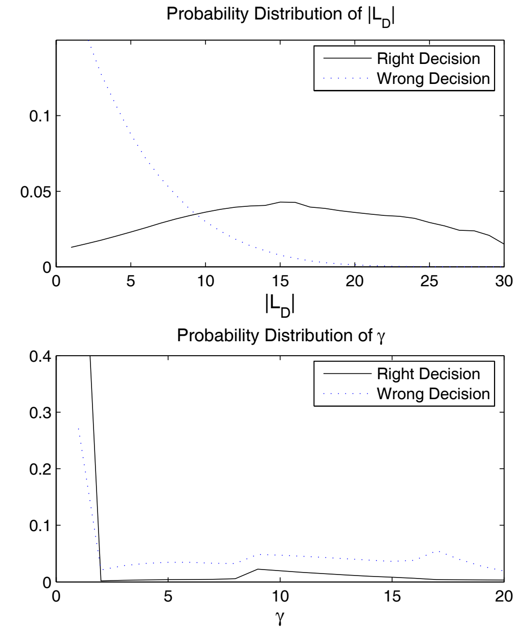

Figure 5: The simulated distributions of $|L_D|$ and γ.

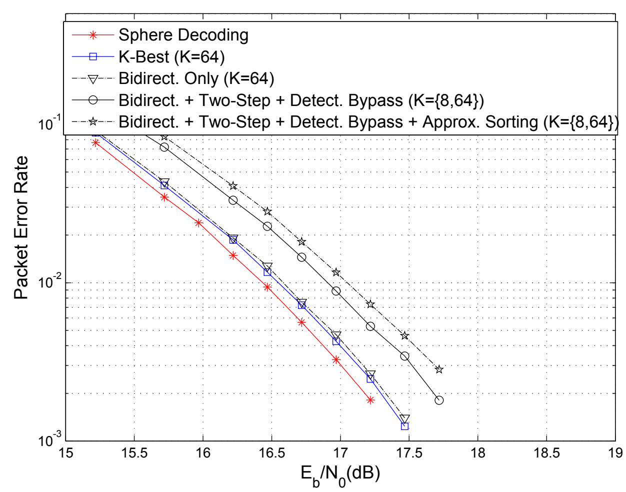

Figure 6: Simulated PER performance for 4×4 MIMO 64-QAM.

4. DETECTOR ASIC DESIGN

To evaluate the effectiveness of the above proposed three design techniques (i.e., bidirectional detection, hybrid two-step detection, and ECC feedback aided detection bypassing), we designed a soft-output detector optimized for 4×4 MIMO with 64-QAM. Fig. 7 shows the principal structure of the detector core. The survivor extension block extends the survivors from the previous depth using the rectangular shaped admissible region as described in Section

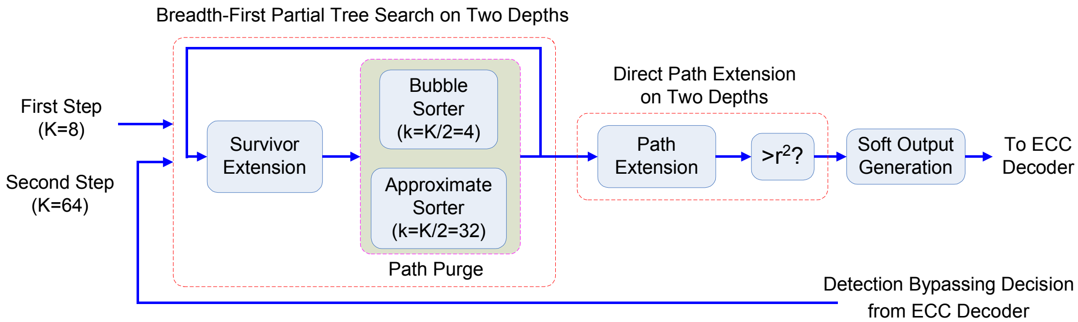

Figure 7: Principal structure of the detector core that implements the above proposed three design techniques.

3.2. For the path purge, we use a bubble sorter to select the best 4 paths when K=8, and use the memory-based approximate sorter to select 32 paths when K=64. In the realization of memory-based approximate sorter, each memory can store 128 paths and is divided into 16 segments. Readers are referred to [4] for a detailed description of its implementation. The same hardware is used for both forward detection and backward detection. The ECC feedback aided detection bypassing is realized using the rules and parameters described in Section 3.3. The detector is designed using TSMC 65nm CMOS standard cell and SRAM libraries. Synopsys tools are used for the simulation, synthesis, and timing/power analysis. The design metrics are summarized in Table 2.

Table 2: Detector AISC Design Metrics.

V_{DD}	f_{clk}	Area	Gate Count	Power
0.9 V	200 MHz	0.24 mm^2	174K	11 mW

Because of the use of the sphere constraint, the number of survivors may vary during the tree search. Therefore, the run-time detection throughput will vary and the average throughput is highly dependent on the SNR. At high SNR, the throughput will be higher because: 1) The sphere constraint will become tighter as SNR increase; 2) At high SNR, it is more likely the detector only need to perform the first-step detection; 3) At high SNR, more vectors will be declared as *good* vectors and will be bypassed during the second-step detection. Table 3 shows the estimated average detection throughput at four different SNRs for 4×4 MIMO 64-QAM. The above results show that, compared with the design solution reported in [4] for 4×4 MIMO with 64-QAM, our proposed design can achieve almost the same detection performance and much higher throughput, while reducing the silicon area and power consumption by more than one order of magnitude.

Table 3: Average throughput of the detector.

E_b/N_0 (dB)	17.7	17.2	16.7	16.2
Throughput (Mbps)	114.8	85.0	60.3	41.7

5. CONCLUSIONS

This paper presents a new energy-efficient soft-output breadth-first tree-search detector design solution for wireless MIMO communication systems. Three algorithm-level techniques are developed to significantly reduce the computational complexity and power consumption while maintaining the near-optimum detection performance. Using 65nm CMOS standard cell and SRAM libraries, we designed a soft-output detector that can support 4×4 MIMO with 64-QAM. To achieve a throughput over 100Mbps, the detector only occupies 0.24mm^2 silicon area (equivalent to 174K logic gates) and consumes 11mw, which represents a more than one order of magnitude improvement over the state of the art reported in the open literature.

6. REFERENCES

[1] J. B. Anderson and S. Mohan. Sequential coding algorithms: A survey and cost analysis. *IEEE Transactions on Communications*, 32:169–176, Feb. 1984.

[2] S. Baro, J. Hagenauer, and M. Witzke. Iterative detection of MIMO transmission using a list-sequential (LISS) detector. In *Proc. of IEEE International Conference on Communications*, pages 2653–2657, May 2003.

[3] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Bölcskei. VLSI implementation of MIMO detection using the sphere decoding algorithm. *Journal of Solid-State Circuits*, 40:1566 – 1577, July 2005.

[4] S. Chen, F. Sun, and T. Zhang. Nonlinear Soft-Output Signal Detector Design and Implementation for MIMO Communication Systems with High Spectral Efficiency. In *IEEE Custom Integrated Circuits Conference (CICC)*, September 2006.

[5] M. S. et al. (ed.). Special Issues on MIMO Systems and Applications (I/II). *IEEE Journal on Selected Areas in Communications*, 21, April/June 2003.

[6] D. Garrett, L. Davis, S. ten Brink, B. Hochwald, and G. Knagge. Silicon complexity for maximum likelihood MIMO detection using spherical decoding. *IEEE Journal of Solid-State Circuits*, 39:1544–1552, Sept. 2004.

[7] Z. Guo and P. Nilsson. Algorithm and implementation of the k-best sphere decoding for mimo detection. *IEEE Journal on Selected Areas in Communication*, 24:491–503, March 2006.

[8] B. M. Hochwald and S. ten Brink. Achieving near-capacity on a multiple-antenna channel. *IEEE Transactions on Communications*, 51:389–399, March 2003.

[9] P. W. Wolniansky, G. J. Foschini, G. D. Golden, and R. A. Valenzuela. V-BLAST: An architecture for realizing very high data rates over the rich-scattering wireless channel. In *Proc. of URSI ISSSE*, pages 295–300, 1998.

[10] K.-W. Wong, C.-Y. Tsui, R. S. Cheng, and W.-H. Mow. A VLSI architecture of a K-best lattice decoding algorithm for MIMO channels. In *IEEE International Symposium on Circuits and Systems*, pages III–273–III–276, May 2002.