Low-Power On-the-Fly Reconfigurable Iterative MIMO Detection and LDPC Decoding Design

Ehsan Rohani\textsuperscript{a}, Jingwei Xu\textsuperscript{b}, Gwan Choi\textsuperscript{c} and Mi Lu\textsuperscript{d}

Department of Electrical and Computer Engineering
Texas A&M University, Texas, TX, 77843. USA
\textsuperscript{a}ehsanrohani@tamu.edu, \textsuperscript{b}xujw07@tamu.edu, \textsuperscript{c}gchoi@tamu.edu, \textsuperscript{d}mlu@tamu.edu

Keywords: MIMO, Iterative, LDPC, Reconfigurable, Low-power.

Abstract. Manufacturing and operation of wireless systems require a practical solution for achieving low-power and high-performance when using advance communication apparatus such as that using multiple-input and multiple-output (MIMO). Often algorithm solutions achieve very high performance but over only in a narrow range of operating parameters. This paper presents a hardware design of MIMO detection that allows real-time switching between various algorithms and detection effort to achieve high performance over the wide-range of signal to noise ratio (SNR) found in realistic operating conditions. We illustrate a design with over 80% reduction in detection power that satisfies the required quality of service (QoS) in SNRs (Eb/No) as low as 8.7 dB.

Introduction

MIMO systems are known for their ability to provide higher data rate in a given channel. Many new standards use MIMO as their link between transmitter and receiver. Standards based on their primary usage have required QoS. IEEE standards 802.16e and 802.11 a/g/n specified $10^{-6}$ bit error rate (BER) and 0.1 packet error rate (PER) as the required QoS. In this content, reaching performances beyond the required minimum error rate seems not only unnecessary but also wasteful [1]. The main challenge in use of MIMO systems is the complexity and computation power necessary for detection of received signal. Suboptimal detectors that avoid exhaustive search have been shown near-ML performance with less complexity. These detectors have been extended further to provide soft input-soft output (SISO) detectors that are suitable for iterative decoding[2].

With the vast coverage range of new standards, mobility of the subscriber station, and different working environment of the communication systems the change in quality of the received signal is high. Also; with everyday decreasing of the silicon cost, it is reasonable to instantiate multiple detectors on a chip and select the proper one based on the system demands and channel conditions. This paper presents a system that enables switching between different detectors in conjunction with iterative detection and decoding (outer-iteration). The hardware and power costs and benefits of the system is also provided and optimized to accommodate the low power requirements of the system.

In the rest of this paper, we first introduce the MIMO detection problem. Next we discuss the basics of iterative detection. Later we explain hardware of the detectors. We present the LDPC decoder of our choice and the details of our case study after that. Then we describe the global architecture of the design and discuss the results. Finally we conclude the paper.

MIMO Detection

In this section we will show how MIMO detection problem will lead to a lattice tree search. If the transmitted signal is $s$, $H$ is the channel matrix (its elements are complex random numbers with average of zero and variance of one), and $e$ is a vector of additive complex symmetric Gaussian noise then the received signal is $y=Hs+e$. We assume that the elements of $H$ are constants during the transmission of a block of data (block fading). The entries of $s$ are chosen from a set of complex constellation of $\Omega$. For ML detection of $s$ we can write the problem as, \textit{Problem: Min} $||y-Hs||^2$.

Sphere decoding successively reduces the search space by evaluating only those constellations that
fit inside a sphere around the received signal, \( \text{Problem: } \min ||y - Hs||^2: d(s) < r^2 \). Using QR decomposition (QRD) we will change the problem to a tree search problem \((H = QR)\).

\[
\hat{y} \triangleq Q^Ty \Rightarrow \text{Problem: } d(s) = \min ||\hat{y} - Rs||^2: d(s) < r^2. \tag{1}
\]

Knowing that \( R \) is a right sided matrix, the search tree can be seen. The tree has a root and the root has size \((\Omega)\) branches. Then each branch is divided to size \((\Omega)\) branches and this will continue equal to the number of transmit antennas. The goal of the tree search is to find branches at the last layer which have the smallest distance. Next, we discuss how soft LLR values can be calculated.

**SISO for Outer Loop**

For iterative detection and decoding of the MIMO system, it is necessary to implement a SISO detector and decoder. LDPC soft decoder was implemented more than 12 years ago. For detectors, Hochwald et al. in [2] described a method to efficiently calculate the approximate LLRs. Using Eq. 2 makes it possible to implement a SISO detector (LLR update unit).

\[
L_k(x_k|Y) \approx \frac{1}{2} \max_{x \in X_k+1} \left\{ \frac{1}{\sigma^2} \| y - H.s \|^2 + X_{[k]}^T L_{A,[k]} \right\} - \frac{1}{2} \max_{x \in X_k-1} \left\{ \frac{1}{\sigma^2} \| y - H.s \|^2 + X_{[k]}^T L_{A,[k]} \right\}. \tag{2}
\]

Where \( X_{[k]} \) (-1 or 1) and \( L_{A,[k]} \) are the candidates and LLR values with the elimination of \( k \)th bit of candidate. The LLR values at the output of LDPC decoder can be feed back to LLR update unit. Calculating LLRs from a limited set of candidates creates the over confidence LLR problem which is the result of absence of hypothesis. As explained in [3], LLR clipping can mitigate this problem.

**Detectors**

We consider three different detectors; minimum mean-squared error based parallel interference cancellation (MMSE-PIC), layered orthogonal lattice detector (LORD), and depth first search list sphere decoding (DFS-LSD). We choose these detectors due to their fundamental differences. In this section we describe the details of the detectors and then we focus on the performance results.

**MMSE-PIC**. The first detector is MMSE-PIC. We used the architecture in [4]. This detector has less complexity in comparison to ML detectors and consumes less power. But this detector uses the maximum hardware among our detectors when it adopted to perform three outer-iterations (777 KGE and 75 mW). Although using MMSE-PIC we achieved 0.3 dB gain in performance (compared to DFS-LSD) we decide that implementing this detector is not optimum.

**LORD** The detail architecture and discussion about a complex domain LORD is proposed in our previous works [5]. By permuting the columns of the \( H \) matrix in QRD, we can precisely control the QAM symbol that will be processed first in tree search. LORD algorithm can compute the LLRs of the bits corresponding at the top level. To get LLRs in the below levels, it recomputes the QRD with columns-permuted \( H \) matrix. The cost of this architecture is 12.4 mW and 69 KGE.

**DFS-LSD** This detector searches the lattice tree and builds a list of the strongest candidates for each transmission then using Eq. 2 generates and updates LLRs. The task of generating (updating) the LLRs will be done with a block called LLR calculation (update) unit. This detector dose not search the entire tree and it only searches for candidates within a distance of received signal (sphere).

There are two properties in architecture of this detector which reduces the power consumption. First, for iterations after first iteration there is no need to search the tree again and LLRs will be updated using pre calculated distances and feedback LLRs. As the result, although LSD tree search unit uses more power for performing one iteration compared to MMSE-PIC, in second iteration and iterations after that it will be more power efficient. Also, the LLR unit in first iteration (LLR calculation unit) has less complexity compared to later iterations (LLR update unit). This is due to the fact that in first iteration there is no feedback LLR exist so Eq. 2 can be rewritten as Eq. 3.

\[
L_k(x_k|Y) \approx \frac{1}{2} \max_{x \in X_k+1} \left\{ \frac{1}{\sigma^2} \| y - H.s \|^2 \right\} - \frac{1}{2} \max_{x \in X_k-1} \left\{ \frac{1}{\sigma^2} \| y - H.s \|^2 \right\}. \tag{3}
\]
Using LSD we obtain almost 0.8 dB gain in second and 1.1 dB in third iteration compare to its first. The cost of this detector is 36.2 mW, 48.8 mW, and 61.1 mW for first to third iteration and 240.6 KGE. The architecture of two major units of the LSD-DFS detector is explained next.

**Tree Search Unit** We used the architecture presented in [6]. In order for presented architecture to support list detector the tree search unit needs the following provisions. First, the algorithm keeps the K best distances at the last level of the three. Second, as a new node with less distance is reached, the search radius is not necessarily decreased unless all the candidates of the list have shorter distances. Also, we do not initially limit the search radius, and \( r \) in Eq. 1 will be infinity.

One of the critical parameters of LSD is its list size. Studies show that the performance of this kind of detector is highly related to the list size [3]. In [3] the LLR clipping has been used to reduce the list size effectively. The maximum list size which is beneficiary for a 4x4, 16 QAM MIMO architecture is 32 with clipping of eight. The architecture of LSD is presented in Fig. 1(a).

- **LLR Update Unit** This unit consists of 16 adders and computes Eq. 2 using the input distances and LLRs. The update unit can update the value of Eq.2 once in each clock cycle. It takes one of the candidates at each clock and computes the LLR value, then the new LLR will be compared to the maximum of previous LLRs and the total maximum of entries will reach the last level. The update unit keeps track of two numbers; one for those that the \( k \)th bit of candidate is ‘1’ (Lambda-ML) and the other for ‘0’ (Lambda-ML-bar). After that LLR update unit applies the same process to all the candidates, the LLR value will be calculated from Lambda-ML and Lambda-ML-bar. The architecture of basic block of LLR update unit is presented in Fig. 1(b). We minimized the hardware cost of the architecture. The blocks at the first layer of the circuit are basically XORs which calculate the one’s compliment of input LLRs. For the calculation of two’s compliments the sign bit of each LLR is assigned to the carry input of an adder. Saturation blocks prevent the overflow and the registers are inserted to increase the throughput. Targeting the throughput of 480 Mbps we used eight basic blocks working in parallel. The latency of the circuit is 28 clock cycles.

**LDPC Decoder**

For the iterative decoder system presented in this paper, we implemented the LDPC decoder based on our previous work [7]. The decoder accommodates irregular decoding with throughput of 480 Mbps and uses 19.7 mW. LDPC decoder is allowed to iterate up to maximum of 25 iterations. With three outer-iterations, the LDPC decoder will take (maximum latency) 1467 clock cycles or 4.52 \( \mu s \). The number of LDPC iterations is highly sensitive to channel SNR. On the average, the number of LDPC iterations is 23.4 for first and second outer-iterations at SNR (Eb/No) of 7.5 dB, and it can be as low as 3.57 iterations for third and fourth outer-iterations at SNR (Eb/No) of 10 dB.
Case Study

This paper case study is based on IEEE 802.11n standard modulation and coding scheme (MCS) index 27. This MCS uses 16 QAM with 4x4 antenna arrangement. The rate of LDPC coding is ½ and the decoded data rate is 240 Mbps. The PER shall be less than 10%, for a physical layer service data unit of 4096 octets. In Fig. 2 the simulation of all the possible regimes are demonstrated. We see that applying more than three outer-iterations is not cost-effective. Also; the second iterations of MMSE-PIC and LSD are providing almost the same performance. The final observation is that MMSE-PIC first iteration is less efficient compared to LORD. Simulations in Fig. 2 show that using three iterations of MMSE-PIC will provide us 0.3 dB performance gain. As using all other regimes of MMSE-PIC is not beneficiary we decide not to include this detector in the final receiver.

![Fig. 2: The simulation of the detectors.](image)

Architecture of the Design

In this section a brief overview of the system is presented. Top level architecture of the system is demonstrated in Fig. 3. In the architecture each block is powered with separate power supply to enable power gating ability. The architecture also uses clock gating which enables us to save power in boundary conditions. In case that the operating SNR of the system is in boundary range control unit will turn on the adjacent regime detector to pass initialization phase and disables its clock signal. The control unit changes the multiplexer to proper position at the beginning of the next received symbol and turns the other detector off. This technique makes sure that we do not lose any data during the transition period. In Fig 3 we used three LDPC and LLR update unit in our design with throughput of 480 Mbps instead of using one decoder with 3x throughput. Using this topology we are able to simply turn the unused decoders off and save the power without dealing with DVFS.

![Fig. 3: High level block diagram of the system.](image)

Results

Table 1 shows the boundary values for regime changes. These margins are the lowest SNRs (Eb/No) for each beneficiary regime that the QoS satisfies the requirement of the standard. Table 2 shows the hardware results and properties of our detectors. The LSD detector uses 61.4 mW
and 48.8 mW for performing three and for two outer-iterations. As the result; if the SNR (Eb/No) is more than 9.1 dB we are able to save 20% in detection power and still guarantee the required QoS. This detector will use 36.2 mW for performing one outer-iteration which will result in further 26% reduction in power. In case that the SNR (Eb/No) is higher than 10.25 dB the system is able to switch to LORD and at this point it will only use 12.6 mW. This will result in 66% power saving compared to previous regime. Using LORD we are able to reduce the detector power consumption by 80% and for SNRs (Eb/No) greater than 8.7 dB we will satisfy the QoS requirement.

<table>
<thead>
<tr>
<th>SNR Range (dB)</th>
<th>8.7-9.1</th>
<th>9.1-9.8</th>
<th>9.8-10.25</th>
<th>10.25-+∞</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detector regime</td>
<td>DFS-LSD 3rd iteration</td>
<td>DFS-LSD 2nd iteration</td>
<td>DFS-LSD 1st iteration</td>
<td>LORD</td>
</tr>
</tbody>
</table>

Table 1: Margin SNRs (Eb/No) of different regimes.

Conclusion

In this paper we considered three different types of MIMO detectors on a special case of IEEE 802.11n standard and we used two of the studied detectors in our subsequent iterative detector. Base on the properties of these three detectors we achieve different power, performance and complexity characteristics. The system gives us ability to save up to 80% in power consumption of detection subsystem and 73% of power usage of detection and decoding without compromising the required performance of the system with the cost of increased chip area.

<table>
<thead>
<tr>
<th>Reference</th>
<th>LORD [5]</th>
<th>DFS-LSD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antenna Conf.</td>
<td>4x4</td>
<td></td>
</tr>
<tr>
<td>Modulation</td>
<td>16 QAM</td>
<td></td>
</tr>
<tr>
<td>Detector</td>
<td>LORD</td>
<td>DFS-LSD</td>
</tr>
<tr>
<td>Performance</td>
<td>Near-ML</td>
<td>SISO iterative (1.6 dB performance gain compared to LORD)</td>
</tr>
<tr>
<td>Technology (µm)</td>
<td>0.045</td>
<td>0.045</td>
</tr>
<tr>
<td>Area (GE)</td>
<td>69K</td>
<td>209K</td>
</tr>
<tr>
<td>Throughput (Mbps)</td>
<td>480</td>
<td>480@ 8.5 dB &amp; 580@ 20 dB</td>
</tr>
<tr>
<td>Power (mW)</td>
<td>12.4</td>
<td>35.3</td>
</tr>
<tr>
<td>Latency (ns)</td>
<td>38</td>
<td>264</td>
</tr>
</tbody>
</table>

Table 2: Implementation results of different Detectors.

References


