Algorithm and hardware design of a 2D sorter-based $K$-best MIMO decoder

Thi Hong Tran$^{1*}$, Yuhei Nagao$^2$ and Hiroshi Ochi$^1$

Abstract

In the field of multiple input multiple output (MIMO) decoder, $K$-best has been well investigated because it guarantees an SNR-independent fixed-throughput with a performance close to the optimal maximum likelihood detection (MLD). However, the complexity of its expansion and sorting tasks is significantly affected by the constellation size $W$. In this paper, we propose an algorithm and hardware design of a 2D sorter-based $K$-best MIMO decoder whose complexity is negligibly affected by $W$. The main novelties of the algorithm are the following: (1) Direct expansion and parent node grouping ideas are proposed for reducing the expansion task’s complexity. (2) Two-dimensional (2D) sorter is proposed for simplifying the sorting task. The hardware design of the decoder supports up to 256-QAM modulation, which aims to apply into 4 $\times$ 4 MIMO 802.11n and 11ac systems. The paper shows that the proposed decoder outperforms the Bell Labs layered space-time (BLAST) minimum mean square error (MMSE) and lattice-reduction aided (LRA) MMSE, and is close to the full $K$-best in terms of bit error rate (BER) performance. The hardware design of the decoder is synthesized in application specific integrated circuit (ASIC) and compared with the previous works. As a result, it achieves the highest throughput (up to 2.7 Gbps), consumes the least power (56 mW), obtains the best hardware efficiency (15.2 Mbps/Kgate), and has the shortest latency (0.07 $\mu$s).

Keywords: Maximum likelihood detection (MLD); $K$-best; MIMO decoder; IEEE 802.11n/ac; 256-QAM

1 Introduction

Multiple input multiple output (MIMO) technology has shown a great promise for the future wireless communication because of its high spectral efficiency. For example, it has been applied in many wireless communication standards such as IEEE 802.16 e/m and IEEE 802.11 n/ac [1]. As an important part of the MIMO system, the MIMO decoder has been well investigated recently. Several types, such as maximum likelihood detection (MLD), linear minimum mean square error (LMMSE), Bell Labs layered space-time MMSE (BLAST MMSE), and lattice-reduction aided MMSE (LRA MMSE), have been proposed. Among these, it is well known that the MLD is the optimal approach in terms of bit error rate (BER) performance. However, its complexity increases exponentially with the number of constellation points of the modulation and with the number of spatial streams [2]. Several researches on suboptimal MLD algorithms, especially on the full $K$-best, have been done instead. If a MIMO system sends data via $N$ spatial streams, the full $K$-best will process through $N$ stages. In each stage, it firstly computes the Euclidean distance from the received information to all of the constellation nodes (i.e., expansion task) and then sorts the obtained results (i.e., sorting task) to select $K$ best nodes. If we denote $W$ as the number of constellation nodes, complexity of the expansion and sorting tasks increases proportionally to $W$ and $W^2$, respectively.

To reduce the $K$-best’s complexity, several researches were carried out and published already. These researches can be classified into two methods named as complex domain and real domain. The former one processes through $N$ stages as the full $K$-best does. However, new ideas are proposed to reduce the complexity in trade-off with an acceptable performance degradation. Some typical proposals on this method are a fixed sphere decoder algorithm - FSD in [3], a step reduced $K$-best sphere decoder algorithm in [4], and a zigzag on-demand expansion scheme in [5]. On the other hand, the real domain method separates the in-phase (IP) and quadrature-phase (QP) components of a complex data into two independent...
real data and processes these data in real domain. Thus, the complexity of each stage is reduced, while the number of stages is increased from \(N\) (in complex domain) to \(2N\) (in real domain). The well-known researches on this method are [6-9]. Studying these proposals, we recognize that the expansion and sorting tasks are still too complex for practical implementation if a large value of \(K\) and high-order modulation types such as 256-QAM are needed.

In this paper, we propose an algorithm and hardware design of a low complexity 2D sorter-based \(K\)-best MIMO decoder. The proposal bases on the complex domain method. The contributions of this paper is briefly described as follows:

- In terms of algorithm, we propose direct expansion and parent node grouping methods to reduce the expansion’s complexity, and two dimensional (2D) sorter to simplify the sorting task. The direct expansion specifies the best candidates directly without searching all the constellation nodes. Consequently, complexity of the algorithm is negligibly affected by constellation size. The Euclidean distance computation becomes simpler, and the divider is eliminated. The parent node grouping helps to reduce the number of search candidates within an acceptable amount without trade-off of the BER performance. The 2D sorter does the matrix-based sorting. It has low complexity, is suitable for hardware resource sharing, and provides approximate result.

- In terms of hardware architecture, a prototype of the algorithm which aims to support 4 \(\times\) 4 MIMO 802.11n/ac systems is developed. We utilize some techniques such as resource sharing and GAIN-MUX-based multiplier to further reduce the complexity.

The rest of this paper is organized as follows: Section 2 shows the preliminary information such as notations, channel model, and full \(K\)-best algorithm. Section 3 describes our algorithm. Section 4 focuses on hardware design. Sections 5 and 6 compare the proposed one with the previous works in terms of BER performance and application specific integrated circuit (ASIC) results, respectively. We conclude the paper in Section 7.

2 Background

2.1 Notations

We shall use bold lowercase letters for vectors and bold capital letters for matrices. Furthermore, \(\|\cdot\|\) denotes the \(L-2\) norm distance or Euclidean distance, \((\cdot)^H\) denotes the Hermitian transpose of a matrix, and \((\cdot)^I\) and \((\cdot)^Q\) respectively denote the in-phase (IP) and quadrature-phase (QP) parts of a signal.

2.2 Channel model and full \(K\)-best algorithm

This paper considers a MIMO system with spatial multiplexing signaling (i.e., the signal transmitted from individual antennas are independent of each other). Let \(N\) and \(M\) represent the number of transmit and receive antennas, respectively, with \(M \geq N\). Assume that the transmit symbol is taken from a quadrature amplitude modulation (QAM) which has \(W\) constellation nodes.

\[
y = Hx + n
\]  

The transmission of each vector \(x\) over flat-fading MIMO channels can be modeled as (1), in which \(y\) is the \(M \times 1\) received signal vector, \(H\) is the \(M \times N\) channel matrix, \(x\) is the \(N \times 1\) transmit symbol vector, and \(n\) is the \(M \times 1\) independent identically distributed (i.i.d.) Gaussian white noise vector. Channel \(H\) is decomposed into two matrices \(Q\) and \(R\): \(H = QR\), in which \(Q\) is an \(M \times M\) unitary matrix and \(R\) is an \(M \times N\) upper triangular matrix. In case \(M > N\), the last \(M-N\) rows of \(R\) are zero, and the size of the \(R\) matrix thus becomes \(N \times N\). For simplicity, in this paper we assume that \(M = N\).

\[
\hat{x} = \arg\min_{x \in \Omega^N} \|z - Rx\|^2 = \arg\min_{x \in \Omega^N} \sum_{i=1}^{N} |z_i - \sum_{j=1}^{N} r_{ij}x_j|^2
\]  

(2)

\[
PED_n = \sum_{i=N}^{n+1} |z_i - \sum_{j=1}^{N} r_{ij}x_j|^2 + |z_n - \sum_{j=n}^{M} r_{nj}x_j|^2
\]  

(3)

The full \(K\)-best finds the transmitted symbol \(x\) by solving (2). In this equation, \(\Omega^N\) denotes \(W^N\) possible sets of the transmitted symbol vector \(x\), and \(z = Q^H y\). Equation 2 is computed through \(N\) stages in the order from \(N\) to 1, one after another. The \(n\)th stage \((n = N, \ldots, 1)\) computes the \(n\)th partial Euclidean distance \((PED_n)\) in (3) by adding \(PED_{n+1}\) (i.e., results of the \((n+1)\)th stage) with \(D_n\) (i.e., calculated in the \(n\)th stage).

Two main tasks - expansion and sorting - will be done in stage \(n\) \((n = N, \ldots, 1)\) (refer to [5] for details).

- **Expansion** task firstly computes \(K \times W\) values of \(D_n\) and \(x_n\) (i.e., child nodes) from \(K\) parent nodes selected from stage \(n + 1\). It then calculates \(PED_n = PED_{n+1} + D_n\).

- **Sorting** task sorts \(K \times W\) values of \(PED_n\) to find the \(K\) smallest values of \(PED_n\) and the corresponding \(\{x_N, \ldots, x_1\}\). The selected data will become the parent nodes of the next stage (i.e., stage \(n - 1\)).

The processing of the two first stages (i.e., \(N\) and \(N - 1\)) is illustrated in Figure 1. Notice that \(K = 1\) if \(n = N\).

At the final stage (i.e., stage 1), the sorting is not performed. All \(K \times W\) values of \(PED_1\) are used for the final
decision, whether hard or soft decision. The hard decision method finds the value of \(\{x_N, \ldots, x_1\}\) that is equivalent to the smallest value of PED and decides this value as the decoded data, while the soft decision method calculates the log likelihood ratio (LLR) of all information bits.

3 The proposed algorithm

Firstly, we use sorted QR decompose (SQRD) pre-processing: \(H = SQR\) instead of the conventional QRD: \(H = QR\) to improve the BER performance. In [10], the authors have shown that a low complex SQRD can be designed by using the modified Gram-Schmidt algorithm with pipelining and resource sharing.

The main process of our algorithm is done through \(N\) stages as the full \(K\)-best does. The following ideas are proposed to reduce the complexity.

3.1 Direct expansion

Firstly, \(D_n\) in (3) is rewritten into (4) and (5) as follows.

\[
D_n = |z_n - \sum_{j=n+1}^{N} r_n x_j - r_{nm} x_m|^2 = |f_n - r_{nm} x_m|^2 \quad (4)
\]

\[
= \left(\frac{f_n^l - r_{nm} x_n^l}{D_{I_n}}\right)^2 + \left(\frac{f_n^Q - r_{nm} x_n^Q}{D_{Q_n}}\right)^2 \quad (5)
\]

In the first quarter of the constellation (in which IP and QP parts are both non-negative), we divide the IP space into \(\sqrt{W} - 1\) subdomains such as \([0, r_{nm}), [r_{nm}, 2r_{nm}), \ldots, [\sqrt{W} - 2, \infty)\). Each subdomain is associated with a set of \(ceil(\sqrt{L})\) best values of \(x_n^l\). For example, if the modulation is 16-QAM and \(L = 9\), the IP space is divided into \([0, r_{nm}), [r_{nm}, 2r_{nm}), \ldots, [2r_{nm}, \infty)\) subdomains. The corresponding three best values of \(x_n^l\) are \((1, -1, 3), (1, 3, -1),\) and \((3, 1, -1)\), respectively (refer to Figure 2a). With QP space and \(x_n^Q\), we do similarly.

The \(L\) best child nodes per parent node in stage \(n\) \((n = N, \ldots, 1)\) are directly specified as follows:

**Step 1.** Calculate \(f_n\) that is defined in (4).

**Step 2.** Determine the IP subdomain that \(f_n^l\) belongs to by comparing \(f_n^l\) with values such as \(r_{nm}, 2r_{nm}, \ldots, (\sqrt{W} - 2) r_{nm}\). From that, the \(ceil(\sqrt{L})\) best values of \(x_n^l\) will be known. If \(f_n^l < 0\), the signs of \(x_n^l\) are reversed. Then, we calculate the corresponding \(D_{I_n}\) in (5). The \(x_n^Q\) and \(D_{Q_n}\) are found similarly (refer to Figure 2a).

**Step 3.** From \(ceil(\sqrt{L})\) best values of \(x_n^l, D_{I_n}\), and \(x_n^Q, D_{Q_n}\), we compute \(L\) best values of \(x_n^l, D_{I_n}\), and \(D_{Q_n}\), which are already in ascending order. The combination of the sum \(D_n = D_{I_n} + D_{Q_n}\) is arranged so that the sum \(i_n^l + q_n^Q\) increases. Consequently, the results of \(D_n\) are approximately in ascending order without sorting (refer to Figure 2b).

To expand \(L\) best child nodes from a parent node, the previous works such as [5] firstly finds the center node by rounding the result \(x_c = f_n/r_{nm}\). It then seeks for \(L\) nearest nodes to the center node. The divider is thus required. By comparing as step 2, the proposed algorithm can eliminate the divider \(f_n/r_{nm}\). Furthermore, by using (5), \(L\) values of \(D_n\) are obtained from \(ceil(\sqrt{L})\) values of \(D_{I_n}\) and \(D_{Q_n}\). The complexity of computing Euclidean distance \(D_n\) is reduced.
3.2 Parent node grouping

It is important to know how much should the number of child nodes per parent node (L) be. If L is too large, BER performance is improved. However, the decoder’s complexity is also increased. If L is too small, the BER performance may be too small to fulfill the system requirement.

Notice that once the L best child nodes are directly specified as mentioned in Section 3.1, if \( L > K \), there is no probability that one of the last \( L-K \) child nodes of any parent will become the final selection. Thus, selecting \( L \leq K \) is a way to reduce the complexity without trade-off of the performance.

In another aspect, assume that \( k \) and \( c \) are the index number of the \( K \) parent nodes (\( \text{PED}_{n+1} \)) and of the \( L \) child nodes (\( D_n \)) per parent node in stage \( n \), respectively. Because values of \( \text{PED}_{n+1} \) are already sorted in stage \( n+1 \), the parent node that has high index \( k \) will have a large value of \( \text{PED}_{n+1} \). Thus, its child nodes are expected to have low probability to be selected as one of the \( K \) smallest (best) nodes for the next stage. To prove this analysis, we did the simulation and computed the probability (in \%) in which a child node might become one of the \( K \) best nodes. The result is shown in Figure 3. From this figure, it can be seen that the larger the index \( k \) is, the smaller the number of child nodes may be selected.

Based on that fact, we propose a parent node grouping method as follows: The \( K \) parent nodes are divided into \( G \) groups. Each group has \( A = K/G \) parent nodes. Note that \( K \) and \( G \) should be selected so that \( K \) is divisible to \( G \) (i.e., \( \text{mod}(K, G) = 0 \)). Group 1 contains the best parent nodes, while group \( G \) contains the worst parent nodes. Each parent node of the \( g \)-th \( (g = 1, 2, \ldots, G) \) group is expanded by \( L_g \) child nodes so that \( L_G < \cdots < L_1 \leq K \).

3.3 Two-dimensional sorter

Sorting is the major bottleneck of the \( K \)-best decoder because of its high complexity. Theoretically, the sorting of \( n \) elements requires \((n^2 - n)/2 \) comparators.

In this subsection, we propose a two-dimensional (2D) sorter which has low complexity, is suitable for hardware resource sharing, and produces approximate result. The 2D sorter for sorting \( C = \sum_{g=1}^{G} A L_g \) child nodes is described as follows: we put the \( C \) child nodes into an \( A \times B \) matrix, in which \( B = \sum_{g=1}^{G} L_g \). The \( j \)-th row of the matrix contains all the child nodes of the \( j \)-th parent of all groups. The illustration in the case \( G = 3 \) is shown in Figure 4. The matrix operates through two processes called as row sorting and column sorting, one after the other, as follows:

- **Row sorting.** The \( B \) elements in a row are sorted. The smallest value is located in the left of the row. This sorting is repeated for all rows.

- **Column sorting.** The \( A \) elements in a column are sorted. The smallest value is located in the top of the column. This sorting is repeated for all columns.

After completing the row and column sorting, the \( K \) top-left elements of the sorted matrix are expected to be the best (smallest) values and are selected. A simulation is needed in advance to correctly determine the position of the best candidates.

To verify the correctness of the 2D sorter, we did the simulation and measured the probability (in \%) in which an element of the sorted matrix might become one of the actual \( K = 7, K = 14 \), and \( K = 21 \) best nodes. The results are shown in Figure 5. From these results, positions of the 1st to the 7th (yellow color), 8th to the 14th (green color), and 15th to the 21st (blue color) best nodes are one by one.
Figure 3 Probability (in %) that a child node may be selected as one of \( K \) best nodes. The simulation parameters are 4 \( \times \) 4 IEEE 802.11ac simulator, 256-QAM modulation, 148,000 data samples, \( K = 21 \), and \( L = 9 \). The data which have probability \( \geq 1\% \) are marked by yellow color.

one determined. The figure also shows that the obtained results (in %) are slightly affected by channel type. However, the influence is too small so that the position of the best nodes is not affected by channel type.

The 2D sorter is suitable for hardware resource sharing because all the rows (columns) do the same task. A circuit which sorts \( B \) elements of the 1st row in the 1st cycle can be reused to sort the 2nd, ..., \( A \)th rows in the 2nd, ..., \( A \)th cycles.

Figure 4 Illustration of a 2D sorter. The parameters are \( G = 3 \), \( K = 21 \), and \( A = 7 \).

4 Hardware design

4.1 Overview architecture

To determine the effectiveness of the proposed algorithm practically, we develop a 4 \( \times \) 4 2D sorter-based \( K \)-best MIMO decoder for 802.11n and 11.ac systems. The decoder supports five modulation types such as BPSK, QPSK, 16-QAM, 64-QAM, and 256-QAM. After completing exhaustive simulation and considering the trade-off between BER performance and complexity, the decoder is configured as follows:

- At all stages, we select \( K = 21 \), \( G = 3 \), \( A = K/G = 7 \), \( L_1 = 4 \), \( L_2 = 3 \), \( L_3 = 1 \), \( B = 8 \), and \( C = 56 \). In the case of 16-QAM, QPSK, and BPSK, which has \( W < K \), the numbers of parent nodes of stages 3, 2, and 1 (denoted by \( K_3 \), \( K_2 \), and \( K_1 \), respectively) are selected as follows: with 16-QAM mode, \( K_3 = 14 \) and \( K_2 = K_1 = 21 \); with QPSK mode, \( K_3 = 4 \), \( K_2 = 14 \), and \( K_1 = 21 \); and with BPSK mode, \( K_3 = 2 \), \( K_2 = 4 \), and \( K_1 = 8 \).
- Stage 4 does not use the sorter, while stages 3 and 2 use the proposed 2D sorter with the matrix size of 7 \( \times \) 8.
- Soft decision is used to achieve high BER performance.
This configuration is illustrated in Figure 6a. The decoder needs multipliers to compute many data, such as \( r_{44}x_4 \), \( r_{33}x_3 \), and \( r_{22}x_2 \), while possible values of \( x_4 \), \( x_3 \), and \( x_2 \) are the same. Thus, one ‘GAIN’ block can be shared among them. The ‘Multiplier-Less’ block implements this ‘GAIN’ block.

4.2 Hardware implementation

To achieve low complexity, in addition to utilize the proposed algorithm, the following implementation points are worth to be noticed.

4.2.1 GAIN-MUX-based multiplier

From (6) to (12), it can be seen that the decoder requires a large number of multipliers to compute \( r_{ij}x_j \) (\( i = 3, 2, 1; j \geq i \)). For example, \( 2\sqrt{K} \) multipliers are needed to compute \( r_{44}x_4 \) and \( r_{44}x_4^Q \) in stage 4 (see (6)), and the multiplier costs large hardware resource.

To compute \( r_{ij}x_j \) (\( i = 3, 2, 1; j \geq i \)), instead of using the multiplier, we implement GAIN and multiplexer (MUX) as shown in Figure 7. This figure illustrates the case of multiplying \( r_{ij} \) by \( m \) best values of \( x_j \). The left figure shows the conventional method which uses \( m \) different multipliers. The right figure is our proposed GAIN-MUX-based multipliers. The input data \( r_{ij} \) firstly goes into the ‘GAIN’ block that amplifies \( r_{ij} \) by the modulation gain \( D \) and then by the values of the constellation points such as \( 1, 3, 5, \ldots, 15 \). Notice that all the possible values of \( x_j \) are \( \{D, 3D, \ldots, 15D\} \). The outputs of ‘GAIN’ blocks are then inputted to \( m \) MUX blocks. Each MUX is controlled by a select signal of \( x_j \) (i.e., denoted by \( sel_{x_j}^{(m)} \)). If values of \( x_j^{(m)} \) are \( \{D, 3D, \ldots, 15D\} \), values of \( sel_{x_j}^{(m)} \) will be \( \{0, 1, \ldots, 7\} \). Consequently, both the outputs of MUX blocks are equivalent to the outputs of multipliers in the left figure.

Meanwhile, hardware cost for MUX is much smaller than that for the multiplier.

The decoder needs multipliers to compute many data, such as \( r_{44}x_4 \), \( r_{33}x_3 \), and \( r_{22}x_2 \), while possible values of \( x_4 \), \( x_3 \), and \( x_2 \) are the same. Thus, one ‘GAIN’ block can be shared among them. The ‘Multiplier-Less’ block implements this ‘GAIN’ block.

4.2.2 Resource sharing

This technique is implemented in STAGE 4, STAGE 3, STAGE 2, and STAGE 1 blocks.

The STAGE 4 block computes \( K \) best values of \( PED_4 \) and \( x_4 \) in (6). Based on the direct expansion method, it finds \( cell(\sqrt{2T}) = 5 \) best values of \( DL_4 \) and \( DQ_4 \) and then adds these values together. Because the processes of finding \( DL_4 \) and \( DQ_4 \) are similar to each other, they share the same circuit. Figure 8a shows the block diagram inside STAGE 4, in which, ‘BLOCK A’ is shared to find best values of \( DL_4, x_4 \) and \( DQ_4, x_4^Q \) in two clock cycles. In other words, the sharing factor of this block is 2. The design of BLOCK A is shown in Figure 8b, in which the...
Figure 6 Hardware design. (a) Decoder’s configuration and (b) the corresponding overview hardware architecture.

'SIGN ABS' block determines the sign and absolute value of $|z_4^I|$ (and $|z_4^Q|$). The 'CONS-LOCAT' block specifies the subdomain in the constellation that $|z_4^I|$ (and $|z_4^Q|$) belongs to. Based on information of the CONS-LOCAT block, the 'DI/DQ CAL' block computes the best values of $D_{I4}$ and $D_{Q4}$, while the 'XDE-CODE' block finds the best values of $x_{I4}^*$ and $x_{Q4}^*$.

The STAGE 3 block computes the best values of PED$_3$ and the corresponding $\{x_4, x_3\}$ in (8). The block diagram is shown in Figure 8c, in which 'B1,' 'B2,' and 'B3' respectively

Figure 7 Conventional multiplier versus GAIN-MUX-based multiplier. This figure illustrates the case of multiplying $r_{ij}$ with $m$ best values of $x_j$. 

perform the direct expansion for the 1st → 7th (i.e., group 1), 8th → 14th (i.e., group 2), and 15th → 21st (i.e., group 3) parent nodes. Because all parent nodes in the same group process similarly, they can share the same circuit. Consequently, the B1 block is designed to find $L_1$ best child nodes of one parent node only. It is then reused in seven clock cycles to complete the direct expansion for seven parent nodes of group 1. The sharing factor is 7. Similarly, the B2 and B3 blocks are shared by seven times. Each B1, B2, and B3 block has the following components: 'CAL $f_3$' computes $f_3$ in (7), 'BLOCK A*' computes $D_{I3}$ and $D_{Q3}$, and 'SUM' computes $PED_3$ from $PED_4$, $DI_3$, and $DQ_3$ (see (8)).

After each clock cycle, B1, B2, and B3 output the best child nodes of one parent node in all groups. In other words, all elements of one row in the sort matrix (see Section 3.3) are obtained per cycle. Block ‘2D-SORT” thus requires only a one-row-sorting circuit. This circuit is then shared to sort all seven rows in seven clock cycles. The sharing factor is 7. The hardware design of the 2D-SORT block is shown in Figure 9. The ‘ROW-SORT’ block sorts eight outputs of B1, B2, and B3 per clock cycle. Only four best data are obtained. In the ‘COL-SORT”, the ‘1to7’ collects the best values from ROW-SORT in seven cycles and sorts them. The ‘1to6’ block collects the 2nd best values from ROW-SORT in seven cycles, sorts them, and obtains six best data, so on. The designs of ROW-SORT and ‘1to3’ of COL-SORT are shown in Figure 9b,c, respectively. It can be seen that the 2D-SORT needs only 36 comparators to sort 56 child nodes, which is significantly reduced as compared to $(56^2 - 56)/2 = 1,540$ comparators if using the full sort.

The architectures of STAGE 2 and STAGE 1 are similar to STAGE 3. The sharing factor of these blocks is 7. However, the 2D-SORT block is not implemented in STAGE 1. Instead, the results of B1, B2, and B3 are directly passed to the LLR block.

5. BER performance comparison
The 802.11ac simulator with the following options were used in our simulation: $4 \times 4$ MIMO and transfer packet number of 5,000. Total transfer data was $2.5 \times 10^6$ bytes. Bandwidth was 80 MHz. Channel type was D. Forward error correction (FEC) type was binary block code (BCC).

5.1 QRD versus SQRD
Figure 10 shows that using SQRD pre-processing helps to improve BER performance by 0.6 dB, 0.8 dB (at BER = $10^{-3}$), and 1 dB (at BER = $10^{-2}$) for cases of $K = 21$, $K = 10$, and $K = 6$, respectively, as compared to the case of using QRD.

5.2 Parent node grouping
Figure 11 shows that the BER performance is insignificantly degraded when ‘L1-L2-L3’ is decreased from 256-256-256 (full $K$-best) to 9-9-9, 9-6-3, 4-4-4, and 4-3-1. Numerically, the performance degradation of $L_1-L_2-L_3 = 4-3-2$ is about 0.15 dB as compared to the full $K$-best. However, when continuing to reduce the number of child nodes per parent node to $L_1-L_2-L_3 = 1-1-1$, the performance degradation is about 1.2 dB, which is considerable, as compared to the full $K$-best.

5.3 2D sorter
Figure 12 shows that the BER performance of $S4$ NoSort - $S32$ 2D Sort is insignificantly degraded as compared to the
Figure 9 The design of '2D-SORT' block. (a) 2D-SORT, (b) ROW-SORT, and (c) 1to3.

Figure 10 BER of 802.11ac system: QRD versus SQRD. 256-QAM was used in all cases.
Figure 11 BER of 802.11ac system: parent node grouping. $G = 3$, $K = 21$, 256-QAM, and SQRD were used in all cases.

Figure 12 BER of 802.11ac system: 2D Sorter. $K = 21$, $G = 3$, $L_1 = 4$, $L_2 = 3$, $L_3 = 1$, 256-QAM, and SQRD were used in all cases. The terms 'S4 FullSort', 'S4 NoSort', 'S32 FullSort', and 'S32 2D Sort' denote that a full sorter is used in stage 4, no sorter is used in stage 4, full sorters are used in stages 3 and 2, and 2D sorters are used in stages 3 and 2, respectively.
case of S4 FullSort - S32 FullSort. The amount of degradation is about 0.08 dB (at BER = $10^{-3}$). In other words, (1) by applying the direct expansion method, the sorter can be eliminated in stage 4, and (2) the 2D sorter is an acceptable approximation of the full sorter. It can be used in trade-off with about 0.08-dB BER performance.

5.4 The proposed decoder

Figure 13 shows the BER of 4 × 4 MIMO 802.11ac system when applying BLAST MMSE, LRA-MMSE, full K-best (soft decision), and the proposed decoder (soft and hard decisions).

From this figure, it can be seen that for all modulation types (16-QAM, 64-QAM, and 256-QAM), the proposed decoder with soft decision (green line) outperforms the BLAST MMSE (blue line) and LRA MMSE (black line), and is close to the full K-best with soft decision (red line). Numerically, at the observation point of BER = $10^{-3}$, the proposed decoder (with soft decision) is better than BLAST MMSE by 6.7, 3.7, and 2.3 dB, respectively. It is better than LRA MMSE by 1, 0.5, and 0.02 dB, respectively. As compared to the full K-best, the BER performance degradation of the proposed one is about 0.2 dB for all cases. In addition, using soft decision can improve the performance of the proposed decoder by about 2 dB as compared to the hard decision (green line versus pink line).

From this figure, we also see that the BER performance’s gap from the proposed decoder (soft decision) and the full K-best to the LRA MMSE and the BLAST MMSE decreases when the modulation types increase from 16-QAM to 64-QAM and to 256-QAM. That is because the modulation size increases while the K value is fixed to 21. Consequently, the BER performance of the proposed decoder and of the full K-best is expected to be worse as the modulation size increases.

Notice that in cases of BPSK and QPSK, the proposed decoder searches all of the constellation nodes; it thus achieves the same BER as the optimal MLD does.

6 Complexity comparison

Due to the application of the direct expansion method, the number of search candidates (or visited nodes) of the proposed decoder is no longer affected by the constellation size. It is affected by $K$, $L_g$ ($g = 1, \ldots, G$), and $N$ only.

Numerically, we compare the complexity of the proposed algorithm with the previous works in terms of total number of visited nodes (shorted as ‘total nodes’) in Table 1. All the compared algorithms are configured to be 4 × 4 MIMO decoder ($N = 4$). The data of [3] and [4] are obtained from their papers. Data of [5] is calculated by ourselves after understanding the algorithm. In the best of our knowledge, this algorithm needs to visit $(\sqrt{W} + K + 1) + 2K(RSE_{\text{num}} + CSE_{\text{num}} + 1) + K$.
Table 1 Total visited nodes of $4 \times 4$ K-best-based MIMO decoders

| Algorithm | [3], 2008 | [4], 2012 | [5], 2013 | Proposed |
|-----------|-----------|-----------|-----------|----------|
| Modulation | 256-QAM  | 256-QAM  | 64-QAM    | 256-QAM  |
| K value   | 26        | 26        | 10        | 21       |
| Total nodes | 1024     | 1004      | 189       | 387      |

nodes, in which $RSE_{\text{num}} = 4$ and $RSE_{\text{num}} = 3$ are reported to be optimal for the case of $N = 4, K = 10$, and $W = 64$ (64-QAM).

This table shows that

- As compared to [3] and [4], the total nodes of the proposed algorithm reduces about 8.5 times, while the gap of the $K$ value is about 1.24 times.
- The total nodes of the proposed algorithm is about half of that of [5], while both have the same $K = 21$ and the proposed one supports higher modulation than [5] (256-QAM versus 64-QAM). In case [5] supports $K = 10$ and the proposed supports $K = 21$, they have the same total nodes.

The comparison in Table 1, however, just reflects the algorithm's complexity in terms of total nodes. The complexity on computing the Euclidean distance of each visited node and on sorting the nodes cannot be seen.

To compare the decoder with the previous ones thoroughly, we designed and synthesized our decoder in ASIC. The synthesis tool was the Design Vision of Synopsys. The CMOS SAED 90 nm technology and saed90nm_min library were used. The applied voltage was 1.32 V.

The ASIC synthesis results are shown and compared in Table 2. All the designs are $4 \times 4$ K-best-based MIMO decoders. From this table, the contribution of the proposed decoder can be seen as follows:

**High throughput.** The proposed decoder achieves the highest throughput among all designs. Comparing with the most recent work in [5], the proposed decoder’s throughput is two times higher.

**Low power consumption.** Among all the designs, the proposed design consumes the least power, which is about 56 mW.

**Small area.** Although supporting higher modulation (i.e., 256-QAM) and larger $K$ (i.e., $K = 21$) than the most recent work in [5], the proposed decoder occupies less hardware area. It needs 180 K gates, which is almost half of [5]. Remember that the proposed decoder and [5] have the same number of visited nodes (see Table 1). This is the evidence for the effectiveness of the 2D sorter and computation method of the direct expansion.

**High normalized hardware efficiency (NHE).** The proposed design obtains the highest NHE. It is 15.2 Mbps/Kgate, which is better than [8,11,12], and [5] by 50.7, 29.2, 8.5, and 3.6 times, respectively.

**Short latency.** The proposed design has the shortest latency. It is 0.07 μs.

7 Conclusions

In this paper, we have proposed an algorithm and hardware design of a 2D sorter-based $K$-best MIMO decoder that supports up to 256-QAM. By utilizing the ideas such as direct expansion, parent node grouping, and 2D sorter, the algorithm has been proven to be less complex than the previous works, and its complexity is negligibly affected by the constellation size. A prototype hardware architecture of the algorithm has been developed to support $4 \times 4$ MIMO 802.11n and 11ac systems. Some techniques

Table 2 ASIC synthesis results of $4 \times 4$ K-best-based MIMO decoders

| Design | [7], 2006 | [6], 2006 | [8], 2010 | [9], 2012 | [10], 2007 | [11], 2012 | [12], 2010 | [5], 2013 | Proposed |
|--------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|----------|
| Modulation | 16-QAM | 16-QAM | 64-QAM | 64-QAM | 64-QAM | 4(64)/QAM | 64-QAM | (2-256)/QAM |
| K value | 5 | 5 | 5-64 | 10 | 64 | N/A | 10 | 21 |
| Method | Real | Real | Real | Complex | Complex | Complex | Complex | Complex |
| Process | 0.35 μm | 0.25 μm | 65 nm | 0.13 μm | 0.13 μm | 0.13 μm | 0.13 μm | 90 nm |
| Hard/soft decision | N/A | N/A | Hard | Hard | Soft | Soft | Soft | Hard |
| $f_{\text{max}}$ (MHz) | 100 | 132 | 158 | 282 | 270 | 198 | 417 | 590 |
| Throughput (Mbps) | 54 | 424 | 732-100 | 675 | 100 | 285-431 | 1,000 | 2,700 |
| Power (mW) | 210a | 1,178a | 529 – 72a | 975a | 140a | 411 – 623a | 1,444a | 2,700a |
| Area (Kgate) | 91 | 114 | 1,760 | 114 | 280 | 350 | 340 | 180 |
| Power (mW) | 626 | N/A | 165 | 135 | 94 | 57-74 | 1,700 | 56 |
| Latency (μs) | 2.33 | 10.3 | 0.3-0.04 | 8.5 | 0.52 | 1.18-1.79 | 4.26 | 15.2 |

aNormalized throughput from $S$ technology to 90 nm = (throughput at $S$) × $\frac{90}{S}$. bNormalized hardware efficiency (NHE) = $\frac{\text{Normalized throughput (Mbps)}}{\text{Area (Kgate)}}$.
such as resource sharing, and MUX-GAIN-based multiplier have been implemented to further reduce the complexity.

The paper has shown that the proposed decoder outperforms the BLAST MMSE and LRA MMSE, and is close to the full \( K \) best in terms of BER performance. The hardware design of the decoder achieves the highest throughput (2.7 Gbps), consumes the least power (56 mW), obtains the best hardware efficiency (15.2 Mbps/Kgate), and has the shortest latency (0.07 \( \mu \)s). This research is, thus, expected to be utilized not only in 802.11n/ac but also in other MIMO systems.

Our future work is to upgrade the designed decoder so that it supports from 1 \( \times \) 1 to 8 \( \times \) 8 MIMO cases.

Competing interests
The authors declare that they have no competing interests.

Acknowledgements
The authors would like to thank Assoc. Prof. Masayuki Kurosaki and Ms. Reina Hongyo for their help on tool licenses and hardware design. This work was partly supported by Japan Ministry of Education KAKENHI (22360156) and by VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with Synopsys.

Author details
1 Kyushu Institute of Technology, 680-4 Kawazu Iizuka, Fukuoka 820-8502, Japan. 2 Radrix Co., Ltd., 680-4 Kawazu Iizuka, Fukuoka 820-8502, Japan.

Received: 10 February 2014 Accepted: 28 May 2014
Published: 12 June 2014

References
1. TH Tran, Y Nagao, M Kurosaki, B Sai, H Ochi, ASIC implement of 600 Mbps IEEE 802.11n 4 \( \times \) 4 MIMO wireless LAN system, in The 14th IEEE Int. Conf. on Advan. Commu. Tech. (ICACT). Pyeongchang Korea, 19-22 Feb. 2012, pp. 360–363
2. L Azzam, E Ayanoglu, Reduction of ML decoding complexity for MIMO sphere decoding, QOSTBC, and OSTBC, in Information Theory and Application Workshop. San Diego, CA, USA, 27 Jan - 1 Feb 2008, pp. 18–25
3. LG Barbero, JS Thompson, Fixing the complexity of the sphere decoder for MIMO detection. IEEE Trans. Wireless Commun. 7, 2131–2142 (2008)
4. X Mao, Y Cheng, L Ma, H Xiang, Step reduced K-best sphere decoding, in Vehicular Technology Conference (VTC Fall). Quebec, Canada, 3–6 Sept 2012, pp. 1–4
5. M Mahdavi, M Shabany, Novel MIMO detection algorithm for high-order constellations in the complex domain. IEEE Trans. VLSI Syst. 21, 834–847 (2013)
6. M Wenk, M Zellweger, A Burg, N Felber, W Fichtner, K-best MIMO detection VLSI architectures achieving up to 424 Mbps, in Proc. Int. Symp. Circuits and Systems (ISCAS 2006). Kos Island, Greece, 21-24 May 2006, pp. 1151–1154
7. Z Guo, P Nilsson, Algorithm and implementation of the K-best sphere decoder for MIMO detection. IEEE Trans. Sel. Areas Commun. 24, 491–503 (2006)
8. S Mondal, A Eltawil, CA Shen, KN Salama, Design and implementation of a sort free K-best sphere decoder. IEEE Trans. VLSI Syst. 18, 1497–1501 (2010)
9. M Shabany, PG Gulak, A 675 Mb/s, 4 \( \times \) 4 64-QAM K-best MIMO detector in 0.13 um CMOS. IEEE Trans. VLSI Syst. 20, 135–147 (2012)
10. Y Miyaoka, Y Nagao, M Kurosaki, H Ochi, RTL design of high-speed QR decomposition for MIMO decoder. IEICE Trans. Fund. Elec., Commu. Comp. Sci. 1991–1997 (A E95)
11. S Chen, T Zhang, Y Xin, Relaxed K-best MIMO signal detector design and VLSI implementation. IEEE Trans. VLSI Syst. 15, 328–337 (2007)
12. C Liao, T Wang, T Chiueh, A 74.8 mW soft-output detector IC for 8 \( \times \) 8 spatial-multiplexing MIMO communications. IEEE Trans. Solid State Circuits. 45, 411–421 (2010)