IMPLEMENTATION OF A K-BEST BASED MIMO-OFDM DETECTOR ALGORITHM

Johanna Kerttula, Markus Myllylä, Markku Juntti

Centre for Wireless Communications
P.O. Box 4500, FIN-90014 University of Oulu, Finland
{johanna.kerttula, markus.myllyla, markku.juntti}@ee.oulu.fi

ABSTRACT

The combination of multiple-input multiple-output (MIMO) and orthogonal frequency-division multiplexing (OFDM) is a promising solution for high-data-rate transmission. An architecture of a K-best based list sphere detector (LSD) algorithm for a MIMO-OFDM system is introduced in this paper. The architecture was designed for a 2×2 *antenna system with quadrature phase shift keying (QPSK) and 16-quadrature amplitude modulation (QAM). The implementation of the architecture was synthesized for a field programmable gate array (FPGA). The feasibility of the implementation for wireless local area network (WLAN) and third generation (3G) long term evolution (LTE) is considered.*

1. INTRODUCTION

The need for higher data rates is growing. Orthogonal frequency-division multiplexing (OFDM) [1] is a popular technique for wireless high-data-rate transmission because it enables efficient use of the available bandwidth and a simple implementation. Multiple-input multiple-output (MIMO) [2] techniques offer an increase in capacity or diversity by bringing an extra dimension to the system. The combination of MIMO and OFDM is a promising broadband wireless access scheme [3]. OFDM is included in wireless local area network (WLAN) [4] and third generation (3G) long term evolution (LTE) [5] standards.

Spatial multiplexing (SM) can be used to transmit independent data streams using multiple antennas [2]. The maximum *a posteriori* (MAP) detection is optimal for systems where channel coding is applied. However, its computational complexity limits its use in most practical systems. Suboptimal minimum mean square error (MMSE) and zero forcing (ZF) criteria based detectors can be used, but they perform poorly in bad channel conditions.

Sphere detectors calculate a maximum likelihood (ML) solution with a reduced computational complexity [6]. List sphere detectors (LSD) can be used to approximate the MAP detector and to provide soft outputs for the decoder [7]. The K-best sphere detection algorithm [8] guarantees a fixed throughput and complexity. Parallel and pipelined implementations can also be applied.

In this paper, an architecture of the K-best LSD is presented. The architecture was designed for a complex valued 2×2 antenna system with operating modes for quadrature phase shift keying (QPSK) and 16-quadrature amplitude modulation (QAM). The use of different list sizes is possible. The implementation was synthesized for a field programmable gate array (FPGA) and the suitability of the implementation for WLAN and 3G LTE is evaluated. The word lengths used in the implementation were determined with simulations using the 3G LTE parameters and realistic channel models. The used list sizes were determined in [9]. The implementation was verified with hardware co-simulation.

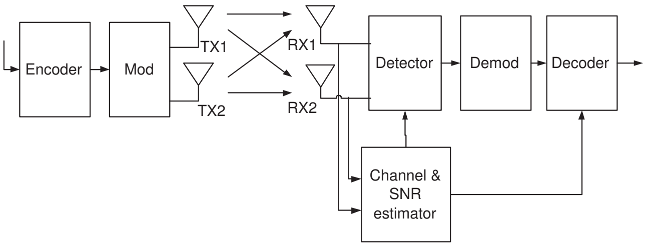

Figure 1: MIMO receiver structure.

The paper is organized as follows. The system model is presented in Section 2. The K-best LSD algorithm is introduced in Section 3. The architecture is presented in Section 4 and the implementation results in Section 5. Conclusions are presented in Section 6.

2. SYSTEM MODEL

An OFDM based MIMO transmission system with N transmit (TX) and M receive (RX) antennas, where $N \leq M$, is considered in this paper. Spatial multiplexing with vertical encoding [10] is applied. A structure of a MIMO system with two RX and two TX antennas is illustrated in Figure 1. The received signal can be described with the equation

$$\mathbf{y}_p = \mathbf{H}_p \mathbf{x}_p + \eta_p, \quad p = 1, 2, \ldots, P, \tag{1}$$

where P is the number of subcarriers, $\mathbf{x}_p \in \mathbb{C}^{N \times 1}$ is the transmitted signal, $\eta_p \in \mathbb{C}^{M \times 1}$ is a vector containing identically distributed complex Gaussian noise and $\mathbf{H}_p \in \mathbb{C}^{M \times N}$ is the channel matrix containing complex Gaussian fading coefficients. The entries of $\mathbf{x}_p$ are from a complex quadrature amplitude modulation (QAM) constellation Ω and $|\Omega| = 2^Q$, where Q is the number of bits per symbol. The set of possible transmitted symbol vectors is $|\Omega|^N = 2^{QN}$.

The maximum likelihood (ML) detection method minimizes the average error probability and it is the optimal method for finding the closest lattice point [6]. The ML detector calculates Euclidean distances (EDs) between the received signal vector $\mathbf{y}$ and lattice points $\mathbf{Hx}$, and returns the vector $\mathbf{x}$ with the smallest distance, i.e., it minimizes

$$\hat{\mathbf{x}}_{\mathrm{ML}} = \arg \min_{x \in \Omega^N} ||\mathbf{y} - \mathbf{Hx}||^2. \tag{2}$$

The subindices are omitted in (2) and in the sequel for notational simplicity.

The sphere detector (SD) algorithms solve the ML solution with a reduced number of considered candidate symbol vectors. They take into account only the lattice points that are inside a sphere of a given radius. The considered lattice points are inside a hyper-sphere $S(\mathbf{y},\sqrt{C_0})$, where C_0 is the squared radius of the sphere and $\mathbf{y}$ is the center of the sphere [6]. The condition that the lattice point lies inside the sphere can be written as

$$||\mathbf{y}-\mathbf{Hx}||^2 \quad \leq \quad C_0. \tag{3}$$

As the channel matrix $\mathbf{H}$ in (3) is decomposed by QR decomposition (QRD) as $\mathbf{H}=\mathbf{QR}$, the equation (3) can be written as

$$||\mathbf{y}'-\mathbf{Rx}||^2 \quad \leq \quad C_0^{'}, \tag{4}$$

where $C_0^{'}=C_0-||(\mathbf{Q}')^{\mathrm{H}}\mathbf{y}||^2$, $\mathbf{y}'=\mathbf{Q}^{\mathrm{H}}\mathbf{y}$, $\mathbf{R}\in\mathbb{C}^{N\times N}$ is an upper triangular matrix with positive diagonal elements, $\mathbf{Q}\in\mathbb{C}^{M\times N}$ is a matrix with orthogonal columns and $\mathbf{Q}'\in\mathbb{C}^{M\times(M-N)}$ is a matrix with orthogonal columns. The vector $\mathbf{x}$ can be solved from (4) using back-substitution due to the upper-triangular form of matrix $\mathbf{R}$. The values of $\mathbf{x}$ are solved level by level. First, the set of admissible values of the last component $\mathbf{x}_N$ are calculated and the final values calculated are for component $\mathbf{x}_1$. The squared partial Euclidean distance (PED) of $\mathbf{x}_i^N$, i.e., the distance between the partial candidate symbol vector and the partial received vector, can be calculated as

$$d(\mathbf{x}_i^N)=\sum_{j=i}^{N}\left|y_j^{'}-\sum_{l=j}^{N}r_{j,l}x_l\right|^2 \quad \leq \quad C_0^{'}, \tag{5}$$

where $i=N\ldots,1$ and $\mathbf{x}_i^N$ denotes the last $N-i+1$ components of vector $\mathbf{x}$ [6].

3. K-BEST LSD ALGORITHM

The sphere detector algorithm can be divided into depth-first and breadth-first groups based on their search strategy. The depth-first algorithms process one candidate symbol vector at a time. The breadth-first algorithms process all the partial candidate symbol vectors on each level before moving to the next level. The K-best algorithm [8] is a breadth-first search based algorithm, and keeps the K nodes which have the smallest accumulated Euclidean distances at each level. If the PED is greater than the squared sphere radius C_0, the corresponding node will not be expanded.

A list sphere detector (LSD) [7] is a variant of the sphere detector. It provides a list of candidates $\mathscr{L}$ and their Euclidean distances as an output. An approximation of the bit *a posteriori* probabilities are calculated from the output. The channel decoder then gets the log-likelihood ratios (LLR) from the list sphere detector.

The K-best LSD is a modification of the K-best algorithm [8] and it outputs a list of candidate vectors and the corresponding Euclidean distances. The K-best LSD algorithm is an interesting choice for implementation because it guarantees a fixed throughput and complexity. It can, therefore, be implemented in a pipelined and parallel fashion.

The size N_{cand} of the output list $\mathscr{L}$ has an impact on the performance of the sphere detector. With a small N_{cand}, the complexity is lower and the detection process faster, but the performance can be worse than with a full list.

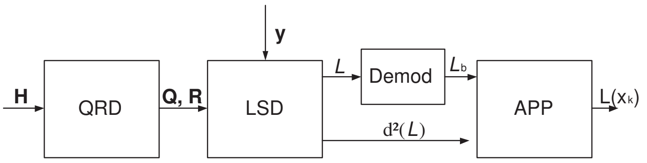

Figure 2: List sphere detector structure.

A high level architecture of a list sphere detector is displayed in Figure 2. The LSD consists of a QR decomposition block, a LSD algorithm block, a demodulation block and an *a posteriori* probability (APP) computation block.

In an OFDM system, each subcarrier has to be detected separately. The QR decomposition has to be done for each subcarrier in an OFDM system and it has to be repeated every time the channel realization changes. The LSD algorithm calculates outputs for each subcarrier and the received signal vector $\mathbf{y}$. The *a posteriori* probability (APP) block calculates log-likelihood ratios

$$L(x_k)=\ln\frac{\Pr(+1|\mathbf{y})}{\Pr(-1|\mathbf{y})} \tag{6}$$

of the transmitted bit k.

4. ARCHITECTURE

The top level structure of the list sphere detector is shown in Figure 3. The input signals to the detector are the received signal vector $\mathbf{y}$, matrices $\mathbf{Q}$ and $\mathbf{R}$ from the QR decomposition, the list size K, a reset signal and a mode signal. The mode signal indicates the modulation used. The sphere detector can operate with QPSK or 16-QAM. The radius of the sphere was set to infinity and, thus, every possible partial candidate symbol vector is included in the calculations and no enumeration method is used.

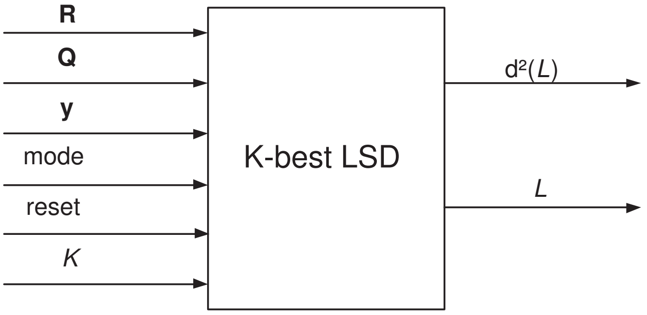

Figure 3: The top level structure of the detector.

The high level architecture of the LSD can be seen in Figure 4. The architecture was divided into separate units and the processing can therefore be pipelined. Pipelining increases the throughput of the sphere detector. In the matrix multiplication unit, the inputs $\mathbf{y}$, $\mathbf{Q}$ and $\mathbf{R}$ are buffered, sliced and the received vector $\mathbf{y}$ is multiplied with matrix $\mathbf{Q}^{\mathrm{H}}$. Each input matrix and vector is divided into real and imaginary elements. PED1 unit calculates the partial Euclidean distances with $d(\mathbf{x}_2^2)=||y_2^{'}-r_{2,2}x_2||^2$, where x_2 is the

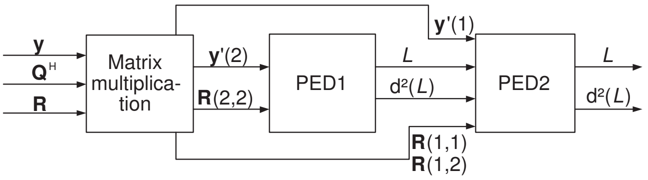

Figure 4: Main blocks of the detector.

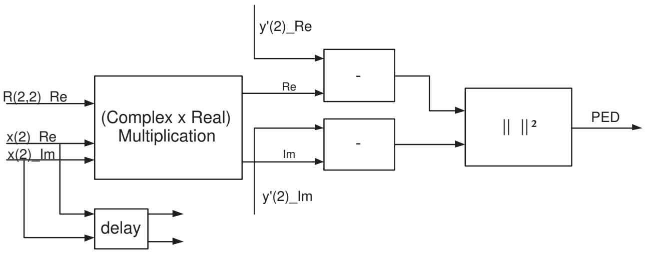

Figure 5: First PED calculation.

set of possible partial transmitted symbol vectors. The block outputs a list of candidate symbols $\mathscr{L}$ and a list of PEDs $d^2(\mathscr{L})$. The lists are then sorted according to the PEDs. The PED2 unit calculates the final Euclidean distances. The PEDs are calculated with $d(\mathbf{x}_1^2) = ||y_1^{'} - (r_{1,1}x_1 + r_{1,2}x_2)||^2$ and the PED from the previous unit is added to the result corresponding to x_2. The output lists are sorted and K candidates with the lowest EDs are kept.

The input signals are buffered to shift registers in the matrix multiplication unit in Figure 4. The buffering is done according to the WLAN parameters [4], i.e., 52 signals are buffered at a time. The next OFDM symbol is buffered after the previous symbol has been processed. After slicing and quantization, the multiplication of $\mathbf{y}$ with matrix $\mathbf{Q}^{\mathrm{H}}$ is performed. Since a Hermitian transpose of the matrix $\mathbf{Q}$ is needed, the imaginary parts of the elements of $\mathbf{Q}$ are negated. The transpose is performed by directing the signals accordingly.

In PED1 unit from Figure 4, the PEDs are calculated and the resulting candidate list is sorted in an ascending order according to the PEDs. The sorting is not necessary in the PED1 unit if all the PEDs are passed to the next level. The architecture of the PED calculation is illustrated in Figure 5. The real and imaginary parts of the partial candidate are transformed to unsigned integers and concatenated to a 16 bit integer. The candidate and the corresponding PED are output simultaneously. The sorting after the first PED block is performed with a modification of the bubble sorting algorithm, which is also known as sorting by exchange [11]. The bubble sorting algorithm is easy to implement. It has $O(n^2)$ worst case complexity, where n is the number of elements to be sorted [12]. Since the timing of the first sorting is not critical, the bubble sorter can be used. The sorter consists of 16 consecutive bubble units.

An architecture of the second PED calculation is displayed in Figure 6. The first complex multiplication block multiplies $r_{1,1}$ with candidate x_1 and the second complex multiplication is performed with $r_{1,2}$ and x_2. The sum of the multiplication results is subtracted from y_1'. The final PED is a result of squaring and adding the real and imaginary parts. The PED is added to the PED from the previous block and the values of x_1 and x_2 are transformed to unsigned integers and concatenated into a 32 bit number. There are registers which are used for holding the candidates and PEDs from the previous level until they have been processed.

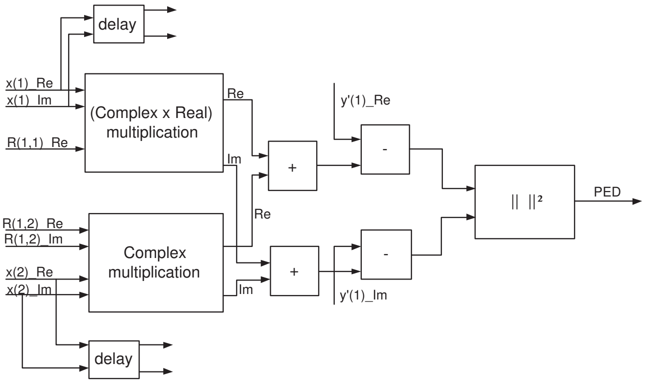

Figure 6: Second PED calculation.

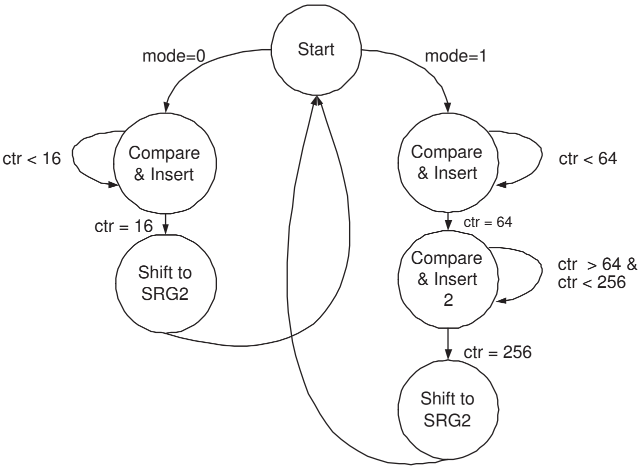

Figure 7: Functionality of the insertion sorter.

Sorting in the PED2 block from Figure 4 is performed with an insertion sorting algorithm. The algorithm is a modification of the parallel insertion sorting algorithm presented in [13]. The sorter consists of four shift registers with lengths of 64 registers since the maximum list size is 64. The data is input to the sorter in a serial form. Each input ED is compared to the EDs in the first shift register and inserted to a corresponding register. After all EDs are sorted, the 64 smallest values remain in the shift register in an ascending order. The data is then shifted to the second shift register, where it is output serially. After the sorter, the whole list of candidates can be passed to the output or only the K best values. The functionality of the insertion sorter is illustrated in Figure 7. The insertion sorter has a worst case complexity of $O(K \cdot n)$, where K is the maximum list size and n is the number of elements to be sorted. The EDs in the shift register are always sorted and only 64 registers are needed to store the best EDs [13].

The calculation of the PEDs can be performed in parallel to improve the speed of the detector and to decrease the latency. Since the calculation of the final list of candidates at the second PED is the most time consuming part of the de-

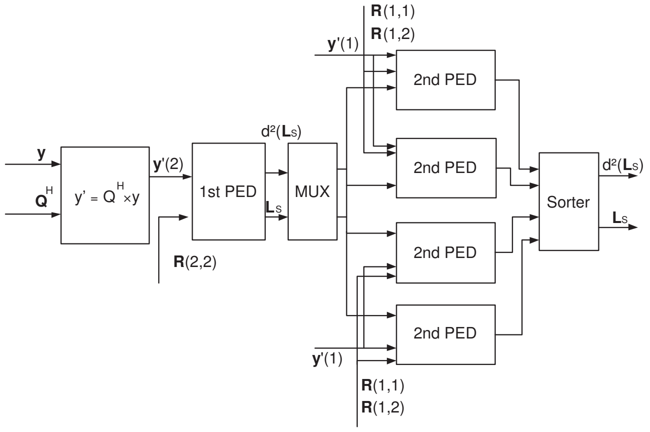

Figure 8: A parallel architecture of the 2×2 antenna detector.

tector, the parallelization of the second PED block is reasonable. A parallel architecture can be seen in Figure 8. The first PEDs are calculated serially and their calculation is pipelined to output one PED on every clock cycle. The second PED calculations are divided into four separate calculation blocks which operate in parallel. The partial candidate list $\mathscr{L}$ and the PED list $d^2(\mathscr{L})$ are multiplexed. There are four candidates after the first sorter in the QPSK mode, and each is divided into separate PED calculation block. Each PED calculation block then calculates four PEDs. In 16-QAM mode, each PED calculation block calculates 64 PEDs.

The final sorting has to be implemented differently from the serial sphere detector to get the advantage of the parallel PED blocks. In the serial detector, the PEDs are input to the sorter in a serial form. The PEDs are input to the sorter partially in parallel with the architecture in Figure 8. There are naturally also other alternatives for parallelization. For example, the first PED could be divided into two calculation blocks. The second PED calculation block would then have to be divided into 8 blocks to achieve equal processing times for both PEDs.

5. IMPLEMENTATION RESULTS

The word lengths used in the implementation were determined with computer simulations using the 3G LTE parameters [5] as in Table 1. The 3G LTE and WLAN parameters are listed in Table 1. The word lengths of the input and output signals are presented in Table 2.

Table 1: 3G LTE OFDM parameter set candidate and WLAN parameters

Parameter	3G LTE	WLAN
Number of OFDM symbols:	7	16
Symbol duration:	71.36 μs	4 μs
Cyclic prefix duration:	4.69 μs	0.8 μs
Useful symbol duration:	66.67 μs	3.2 μs
Channel bandwidth:	5 MHz	16 MHz
Subcarrier spacing:	15 kHz	312.5 kHz
Number of subcarriers per OFDM symbol:	300	52

Table 2: Word lengths used in the implementation

Signal	Word length
$\mathbf{R}$	14 bits
$\mathbf{Q}$	9 bits
$\mathbf{y}$	15 bits
$d^2(\mathscr{L})$	20 bits
$\mathscr{L}$	8 bits

The architecture of the K-best LSD algorithm was implemented using the Xilinx System Generator and synthesized to a Xilinx Virtex-IIv6000 FPGA. The resources used by each main block are displayed in Table 3. The synthesis results are unconstrained. The resources are specified in slices, 18-bit × 18-bit embedded multipliers and block random access memory (BRAM). The insertion sorter is the most complex part of the detector.

Table 3: Synthesis results

Block	Slices	Emb. mult.	BRAM	Max. Clock Freq.
Matrix mult.	332	12	0	115.8 MHz
PED1	309	10	4	70.14 MHz
Bubble sorter	1544	0	0	164.6 MHz
PED2	554	13	4	73.43 MHz
Insertion sorter	16446	0	0	75.5 MHz
Total	20560	35	18	59 MHz

The latencies of the sphere detector are presented in Table 4. The insertion sorter has the highest latency, which is 18 clock cycles in QPSK mode and 258 clock cycles in 16-QAM mode. The sorter has to get the whole list of candidate symbol vectors before it can output the first ED and candidate. However, it can take a new candidate on every clock cycle, except when the shifting to the second shift registers occurs. The bubble sorter has the second highest latency. The latency is 8 clock cycles in QPSK mode and 18 clock cycles in 16-QAM mode. The bubble sorter cannot take in the next list of candidate symbol vectors until the previous list has come out and the registers cleared. The bubble sorter could be removed from the system, since full lists of candidate symbol vectors are passed to PED2. However, it would only have an impact on the overall latency and complexity, and there would not be an increase in throughput.

Table 4: Latencies of the main blocks

Block	Modulation	Latency in clock cycles
Matrix mult.	QPSK and 16-QAM	2
PED1	QPSK and 16-QAM	5
PED2	QPSK and 16-QAM	6
Bubble sorter	QPSK	8
Bubble sorter	16-QAM	18
Insertion sorter	QPSK	18
Insertion sorter	16-QAM	258
Total	QPSK	61
Total	16-QAM	373

The sphere detector would have 71.36 μs to process 300 subcarriers according to the 3G LTE parameters. There is approximately 3 times more time to process each subcarrier with the 3G LTE parameters than with the WLAN parameters. Some parallelism would have to be introduced for the given time frames. The parallelism needed to process the subcarriers in the time frames given by the standards is presented in Table 5. With QPSK and 3G LTE parameters, the second PED calculation would have to be divided in to two PED calculation blocks. It would then take 40.7 μs to process 300 symbol vectors with a 59 MHz clock frequency. With WLAN and QPSK, the second PED calculation would have to be divided in to four PED calculation blocks, leaving 3.5 μs to process 52 symbol vectors.

Table 5: Parallelism with 3G LTE and WLAN parameters

Standard	Modul.	Time for subcarrier	Parallelism needed
WLAN	QPSK	76.9 ns / 4 clock cycles	4 PED2
3G LTE	QPSK	238 ns / 14 clock cycles	2 PED2
WLAN	16-QAM	76.9 ns / 4 clock cycles	4 PED1, 64 PED2
3G LTE	16-QAM	238 ns / 14 clock cycles	2 PED1, 32 PED2

Table 6 shows the amount of complexity needed to achieve the same throughput as in the computer simulations. In the simulations, the processing of a symbol vector was assumed to be done in the same time frame with all modulations. Parallelism is therefore required to achieve the same throughput as in the simulations. Table 6 shows only the complexity of the PED calculation blocks. In the 4×4 antenna system, there would also be more sorters compared to the 2×2 system. To achieve twice the throughput in 16-QAM compared to QPSK, it would require at least 5 times the complexity.

Table 6: The complexity required to achieve a target throughput

Modulation	Antennas.	SNR	Throughput	PED complexity
QPSK	2×2	6 dB	8 Mbps	918 slices
16-QAM	2×2	14 dB	16 Mbps	5391 slices
64-QAM	2×2	22 dB	24 Mbps	107424 slices
QPSK	4×4	8 dB	16 Mbps	4524 slices
16-QAM	4×4	18 dB	32 Mbps	63087 slices
64-QAM	4×4	26 dB	50 Mbps	670624 slices

There are some published implementations of the K-best algorithm in the literature [8, 14]. The comparison of the implementation carried out in this work on the implementations in [8] and [14] is difficult because they are implemented for a 4×4 antenna system with 16-QAM and have considerably smaller list sizes. Also an enumeration method is used, which reduces the amount of partial candidates. The implementations are targeted on application-specific integrated circuits (ASIC) which allows the placing of the components more freely, and higher clock frequencies. Therefore our implementation cannot reach the same decoding throughput as that in [14]. The throughput of the hard-output sphere detector in [8] can be reached with QPSK with an ASIC implementation. The implementation in [8] uses a higher clock frequency and each signal vector contains 16 bits. The implemented sphere detector uses a 59 MHz clock frequency, each signal vector contains 4 bits and it calculates 10 times less EDs with QPSK than the implementation in [8]. It can be seen that the throughput of a detector correlates with the maximum number of calculated PEDs.

6. CONCLUSIONS

The architecture of a 2×2 antenna system sphere detector for QPSK and 16-QAM and implementation results were presented. The architecture is pipelined which increases the throughput of the detector. Parallel calculation of the PEDs would also increase the throughput and decrease the latency. Sorting is the most complex part of the sphere detector. The complexity of the sorter grows with the maximum list size. The suitability of the sphere detector for WLAN and 3G LTE systems was discussed. The detector needs parallelism with both standards. A sphere detector designed for a system using the 3G LTE parameters would require two times less parallelism than a detector in a WLAN system and therefore would be less complex.

REFERENCES

[1] H. Yang, “A road to future broadband future wireless access: MIMO-OFDM-based air interface,” *IEEE Communications Magazine*, vol. 43, no. 1, pp. 53–60, January 2005.

[2] D. Gesbert, M. Shafi, D. Shiu, P. J. Smith, and A. Naguib, “From theory to practice: An overview of MIMO space-time coded wireless systems,” *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 3, pp. 281–302, April 2003.

[3] H. Bölcskei and E. Zurich, “MIMO-OFDM wireless systems: basics, perspectives, and challenges,” *IEEE Wireless Communications*, vol. 13, no. 4, pp. 31–37, August 2006.

[4] ANSI/IEEE Standard 802.11,1999 Edition (R2003), “Information technology - telecommunications and information exchange between systems - local and metropolitan area networks - specific requirements part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications,” 2003.

[5] 3rd Generation Partnership Project (3GPP); Technical Specification Group Radio Access Network, “Physical layer aspects for evolved UTRA (TR 25.814 version 1.5.0 (release 7)),” Tech. Rep., 3rd Generation Partnership Project (3GPP), 2006.

[6] M. O. Damen, H. El Gamal, and G. Caire, “On maximum–likelihood detection and the search for the closest lattice point,” *IEEE Transactions on Information Theory*, vol. 49, no. 10, pp. 2389–2402, October 2003.

[7] B. Hochwald and S. ten Brink, “Achieving near-capacity on a multiple-antenna channel,” *IEEE Transactions on Communications*, vol. 51, no. 3, pp. 389–399, March 2003.

[8] K. Wong, C. Tsui, R.-K. Cheng, and W. Mow, “A VLSI Architecture of a K-best Lattice Decoding Algorithm for MIMO Channels,” in *Proc. IEEE Int. Symp. Circuits and Systems*, Helsinki, Finland, June 2002, vol. 3, pp. 273–276.

[9] M. Myllylä, P. Silvola, M. Juntti, and J. Cavallaro, “Comparison of two novel list sphere detector algorithms for MIMO-OFDM systems,” in *Proc. IEEE Int. Symp. Pers., Indoor, Mobile Radio Commun.*, Helsinki, Finland, September 2006.

[10] 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 *International Symposium on Signals, Systems, and Electronics (ISSSE)*, Pisa, Italy, September 1998, pp. 295–300.

[11] E. Friend, “Sorting on electronic computer systems,” *Journal of the ACM*, vol. 3, no. 3, pp. 134 – 168, July 1956.

[12] D. Knuth, *The Art of Computer Programming, Volume 3*, Addison-Wesley, Reading, Massachusetts, 1973.

[13] P.A. Bengough and S.J. Simmons, “Sorting-based VLSI architectures for the M-algorithm and T-algorithm trellis decoders,” *IEEE Transactions on Communications*, vol. 43, no. 234, pp. 514 – 522, February 1995.

[14] Z. Guo and P. Nilsson, “Algorithm and implementation of the K-best sphere decoding for MIMO detection,” *IEEE Journal on Select Areas in Communications*, vol. 24, no. 3, pp. 491–503, March 2006.