

Implementation of a Reduced-Lattice MIMO Detector for OFDM Systems

Josep Soler-Garrido, Henning Vetter, Magnus Sandell, David Milford and Andy Lillie[†]

Toshiba Research Europe Ltd.
Telecommunications Research Laboratory
32 Queen Square, Bristol BS1 4ND
United Kingdom
Email: Josep.Soler@toshiba-trel.com

Abstract—This paper presents a novel VLSI implementation of a MIMO detector for OFDM systems. The proposed architecture is able to perform both linear MMSE and reduced lattice-aided MIMO detection, making it possible to adjust the balance between performance and power consumption. In order to facilitate real-time detection in reduced lattice mode of operation, a novel fixed-complexity version of the LLL lattice reduction algorithm has been developed, allowing for strict practical timing requirements, such as those specified for new generation IEEE 802.11n wireless LAN systems, to be met. An implementation of the MIMO detector for a system employing up to 4 transmit and receive antennas is described and its complexity and performance are evaluated.

I. INTRODUCTION

In recent times, MIMO systems have become a popular alternative to their single antenna counterparts thanks to the increase in throughput they can achieve with the use of spatial multiplexing techniques. However, this increase in performance comes at the price of added complexity at the receiver, making optimal detection of MIMO signals impractical at the data rates required by modern wireless systems. In order to circumvent this problem, conventional receivers resort to linear detection techniques such as minimum mean squared error (MMSE), which incur a substantial performance degradation if the MIMO channel is poorly conditioned.

Recent research has shown that lattice-reduction-aided detection is able to offer near-optimal performance with a complexity close to that of linear techniques [1]. Lattice reduction offers an alternative way of describing a MIMO channel which is less sensitive to noise enhancement and may be computed using *e.g.* the Lenstra-Lenstra-Lovász algorithm (LLL) [2]. One of the main challenges for hardware implementation of LLL is its variable execution time. For this reason, a modified LLL algorithm [3] offering fixed complexity while maintaining high performance is considered for our design.

Previous work on the implementation of lattice reduction techniques, such as [4], has focused on the original LLL algorithm in a standalone mode. In this paper, we take it one step forward and consider a VLSI implementation of a full lattice reduction-aided MIMO detector for orthogonal

frequency-division multiplexing (OFDM) systems, *i.e.* MIMO detection is carried out individually for the multiple subcarriers within an OFDM symbol. Moreover, the presented architecture allows for the lattice reduction engine to be disabled and linear MMSE detection to be performed when high energy efficiency or low latency are required.

The paper is organized as follows: the OFDM system model and lattice-reduction-aided detection are explained in Section II. The VLSI architecture for the implemented detector is described in Section III. Application to a wireless LAN IEEE 802.11n receiver is discussed in Section IV. Finally, conclusions are drawn in Section V.

II. SYSTEM MODEL AND MIMO DETECTION

A frequency domain narrowband model of a MIMO OFDM system is considered

$$\mathbf{y}(f) = \mathbf{H}(f)\mathbf{x}(f) + \mathbf{v}(f) \quad (1)$$

where $\mathbf{y}(f) \in \mathbb{C}^{N \times 1}$ is the received signal vector, $\mathbf{H}(f) \in \mathbb{C}^{N \times M}$ the MIMO channel, $\mathbf{x}(f) \in \mathbb{C}^{M \times 1}$ the transmitted signal and $\mathbf{v}(f) \in \mathbb{C}^{N \times 1}$ an AWGN vector of noise variance σ_v^2 , with M and N being the number of transmit and receive antennas respectively, and f the subcarrier index. For brevity, the index f is omitted in the sequel.

A quasi-static block fading channel is assumed, where \mathbf{H} is constant over a packet, possibly comprising multiple OFDM symbols.

Linear MMSE detection of the transmitted symbols is formulated using an extended system model, defining [1]

$$\mathbf{H}_e = \begin{pmatrix} \mathbf{H} \\ \sigma_v \mathbf{I}_M \end{pmatrix} \quad \mathbf{y}_e = \begin{pmatrix} \mathbf{y} \\ \mathbf{0}_M \end{pmatrix} \quad (2)$$

where for simplicity we assumed the normalizations $E\{|x_m|^2\} = E\{|h_{n,m}|^2\} = 1$. \mathbf{I}_M is the $M \times M$ identity matrix and $\mathbf{0}_M$ a $M \times 1$ zero vector. The output of the MMSE filter is

$$\hat{\mathbf{x}}_{\text{MMSE}} = (\mathbf{H}^H \mathbf{H} + \sigma_v^2 \mathbf{I}_M)^{-1} \mathbf{H}^H \mathbf{y} = \mathbf{H}_e^+ \mathbf{y}_e \quad (3)$$

where $\mathbf{H}_e^+ = (\mathbf{H}_e^H \mathbf{H}_e)^{-1} \mathbf{H}_e^H$ is the pseudoinverse of \mathbf{H}_e . An arguably more implementation-friendly formulation employs a QR decomposition $\mathbf{H}_e = \mathbf{Q}\mathbf{R}$, with $\mathbf{Q} \in \mathbb{C}^{(M+N) \times M}$ being

[†]A. Lillie was with Toshiba Research Europe Ltd. and is now with Nextwave Wireless, Greenways Business Park Unit 7, Bellinger Close, Chippenham, Wiltshire SN15 1BN, UK.

a unitary matrix and $\mathbf{R} \in \mathbb{C}^{M \times M}$ being an upper triangular matrix, leading to the expression

$$\hat{\mathbf{x}}_{\text{MMSE}} = \mathbf{R}^{-1} \mathbf{Q}^H \mathbf{y}_e. \quad (4)$$

As aforementioned, when \mathbf{H}_e is poorly conditioned, the performance of this estimator can be very poor due to noise enhancement. Transformation of the detection model into a reduced lattice representation can mitigate this problem. Given a unimodular transformation matrix \mathbf{T} , *i.e.* with integer entries and absolute value of determinant 1, lattice-reduction-aided detection can be expressed as

$$\mathbf{y} = \mathbf{H} \mathbf{T} \mathbf{T}^{-1} \mathbf{x} + \mathbf{v} = \tilde{\mathbf{H}} \tilde{\mathbf{x}} + \mathbf{v} \quad (5)$$

with $\tilde{\mathbf{H}} = \mathbf{H} \mathbf{T}$ being the reduced lattice basis and $\tilde{\mathbf{x}} = \mathbf{T}^{-1} \mathbf{x}$. The MMSE filter applied in the reduced lattice $\tilde{\mathbf{H}}_e = \mathbf{H}_e \mathbf{T}$ is then

$$\hat{\tilde{\mathbf{x}}} = \frac{1}{\alpha} \tilde{\mathbf{H}}_e^+ \mathbf{y}_e. \quad (6)$$

The transformation \mathbf{T} can be found by means of a lattice reduction algorithm such as LLL or, in our case, a fixed complexity modification of it [3]. A modulation format dependent scaling factor $1/\alpha$ is used to force the estimation $\hat{\tilde{\mathbf{x}}}$ in the reduced lattice to an integer grid. This facilitates a quantization operation which is required prior to transformation back to the original lattice. Again, a more suitable expression using a decomposition $\tilde{\mathbf{H}}_e = \tilde{\mathbf{Q}} \tilde{\mathbf{R}}$ is

$$\hat{\tilde{\mathbf{x}}} = \frac{1}{\alpha} \tilde{\mathbf{R}}^{-1} \tilde{\mathbf{Q}}^H \mathbf{y}_e. \quad (7)$$

Therefore, the final reduced-lattice detector estimate is given by

$$\hat{\mathbf{x}}_{\text{RL}} = \alpha \mathbf{T} \mathcal{Q} \left\{ \hat{\tilde{\mathbf{x}}} \right\} \quad (8)$$

where $\mathcal{Q}\{\cdot\}$ quantizes toward the nearest valid complex integer vector, and \mathbf{T} transforms the quantized estimate back to the original domain. In the case that one or more elements of $\hat{\mathbf{x}}_{\text{RL}}$ fall outside the set of valid constellation points, they are mapped to the nearest ones.

III. VLSI ARCHITECTURE

A top level block diagram describing the proposed detector architecture is given in Figure 1. As aforementioned, it is assumed that the detector is used in a quasi-static fading environment, *i.e.* the channel \mathbf{H} remains constant throughout a packet. This means that QR decomposition and lattice reduction operations, which are performed on the channel matrix, need only be performed once for each received packet, regardless of the number of OFDM symbols it comprises. This leads to a division of the detector into two main blocks, a Pre-Processing Engine (PPE) and a Data-Processing Engine (DPE).

The PPE performs factorization and lattice reduction operations. In MMSE mode, a single QR decomposition of \mathbf{H}_e is performed, and the resulting matrices \mathbf{R} and \mathbf{Q} are stored. In reduced-lattice mode, matrix \mathbf{R} is sent to the lattice reduction engine which computes the transformation matrix \mathbf{T} .

Moreover, the lattice reduction engine is also able to obtain $\tilde{\mathbf{H}}_e = \mathbf{H}_e \mathbf{T}$ which is fed back to the QR decomposition engine. A second QR decomposition, this time on $\tilde{\mathbf{H}}_e$ results in matrices $\tilde{\mathbf{R}}$ and $\tilde{\mathbf{Q}}$ which are finally stored in memory for later use.

The task of the DPE is to use the previously stored values to compute equation (4) or (8) depending on the mode of operation, which respectively obtain MMSE and reduced-lattice MMSE estimates. This has to be performed for every OFDM symbol received.

The rest of this section presents details for the blocks of Figure 1 considering a $N = M = 4$ MIMO detector implementation.

A. QR Decomposition and Data Rotation engines

The QR decomposition engine is based on a systolic array architecture [5] where factorization of the matrix is performed using successive Givens rotations on the channel values. The array is constructed with two different building blocks, both of which are based upon CORDIC processors, namely boundary cells and internal cells.

Boundary (B) cells contain CORDIC processors in *vectoring* mode, performing the rotations necessary to introduce zeros in the desired positions of the \mathbf{R} matrix. These cells produce a set of controls \mathbf{C} which represent the rotations performed. Internal cells (I), on the other hand, employ CORDIC processors in *rotational* mode which rotate the remaining channel values as prescribed by \mathbf{C} .

This architecture is able to perform at the same time both QR decomposition and the data rotation required by the DPE, *i.e.* a premultiplication of \mathbf{y}_e by \mathbf{Q}^H or $\tilde{\mathbf{Q}}^H$. Therefore, the controls \mathbf{C} are stored in memory for use by the DPE. The resulting combined QR decomposition and data rotation structure is shown in Figure 2.

It should be noted that the channel values are fed to the systolic array one row at a time. The precision achieved is determined by the length of the CORDIC pipeline, and for a given length P_P , each individual cell presents a latency of $2P_P$ as processing is performed on complex values. Therefore, in a full QRD engine, the data path for the M^{th} column contains $2MP_P$ registers, and given the $N + M$ rows of the extended channel matrix, the total latency for a QR decomposition is

$$L_{\text{QRD}} = P_P (2M + (N + M - 1)), \quad (9)$$

which for our system results in $15P_P$ clock cycles.

B. Lattice Reduction engine

As aforementioned, transformation of the original detection lattice into a reduced one, less prone to noise-enhancement errors, requires finding a suitable matrix \mathbf{T} by means of a so-called lattice reduction algorithm. Among several alternatives, LLL algorithm [2] is arguably the most popular. In our case, given the original factorization matrix \mathbf{R} , LLL successively performs a *size-reduction* step to the columns by carrying out an integer-restricted Gram-Schmidt orthogonalization. This attempts to make columns as orthogonal as possible. After

TABLE I
MODIFIED LLL ALGORITHM

```

Input:  $\mathbf{R}, \mathbf{H}_e$ ; Output:  $\mathbf{T}, \tilde{\mathbf{H}}_e = \mathbf{H}_e \mathbf{T}$ 
1 Initialization  $\tilde{\mathbf{R}} := \mathbf{R}, \tilde{\mathbf{H}}_e := \mathbf{H}_e, \mathbf{T} := \mathbf{I}$ 
2 for  $j = 1$  to  $S$ 
3   for  $k = 2$  to  $M$ 
4     for  $l = k - 1$  to  $1$  step  $-1$ 
5        $[\tilde{\mathbf{R}}, \tilde{\mathbf{H}}_e, \mathbf{T}] = \text{size reduction}(\tilde{\mathbf{R}}, \tilde{\mathbf{H}}_e, \mathbf{T}, k, l)$ 
6     end
7      $[\tilde{\mathbf{R}}, \tilde{\mathbf{H}}_e, \mathbf{T}] = \text{basis reduction}(\tilde{\mathbf{R}}, \tilde{\mathbf{H}}_e, \mathbf{T}, k)$ 
8   end
9 end

```

the size-reduction, there might be a column swap in the subsequent *basis-reduction* step, followed by a rotation which re-introduces zeros in the appropriate positions of \mathbf{R} .

The main drawback with LLL lies in its a priori unknown run-time, as it depends on the input matrix. This variable execution time causes problems for a real-time implementation. A modified version is presented in [3] with a run-time known in advance. While the standard LLL algorithm is able to move back and forth between columns until full reduction is achieved, the modified version only allows steps forward in the column counter, and hence may result in a not fully reduced lattice. However, the fixed-complexity approach has been validated by simulation and shown to offer excellent performance. Pseudo-code for the *modified LLL algorithm* is shown in Table I, where S is a pre-defined number of sweeps through the fixed complexity lattice reduction step. Detailed pseudo-code for size and basis reduction steps are given in Tables II and III respectively. A block diagram containing all the necessary steps for a single reduction sweep in a $N = M = 4$ system is shown in Figure 3.

It is observed in Table II that size reduction implementation requires a division operation, expensive in terms of hardware, in order to calculate the orthogonalization coefficient μ . However, simulations have shown that restriction of the possible values of μ to the set $\mu \in \{-1, 0, +1\}$ incurs a negligible performance loss provided enough sweeps are performed. This allows us to replace the division operation with a much simpler addition and comparison stage, resulting in a latency for size reduction of just 2 clock cycles.

Regarding the basis reduction step, Givens rotations are performed in the same fashion as in the QRD engine. In parallel, the column swap condition is evaluated and depending on the result an output multiplexer selects either an unmodified or swapped and rotated set of columns. If the same CORDIC pipeline length as for QRD is employed, the latency for basis reduction is $3P_P$ clock cycles.

Therefore, a lattice reduction sweep comprising $M(M-1)/2$ size reduction steps and $M-1$ basis reduction steps presents a total latency of

$$L_{\text{LRE}} = (M-1)(M+3P_P), \quad (10)$$

which in our case yields $L_{\text{LRE}} = 9P_P + 12$ clock cycles.

C. Data Processing engine

The first stage of DPE contains the already described data rotation engine which obtains $\mathbf{Q}^H \mathbf{y}_e$ for MMSE or $\tilde{\mathbf{Q}}^H \mathbf{y}_e$ for RL-MMSE. Thereafter, a scaling factor $1/\alpha$ is applied if necessary.

Finally, the symbol estimation can be obtained by solving a linear system of the form $\mathbf{R}\mathbf{x} = \mathbf{Q}^H \mathbf{y}$. Given the triangular form of the matrix, this can be performed by back-substitution. A shift-subtract implementation based on CORDIC processors is chosen for the divisions, requiring for the complex valued system $2M-1$ stages. If a back-substitution pipeline length of P_{BS} (considering quantization if required) is employed, the total latency is

$$L_{\text{BS}} = P_{\text{BS}}(2M-1) + 1, \quad (11)$$

including scaling, yielding $7P_{\text{BS}} + 1$ clock cycles for our system.

A final lattice transformation stage is necessary only in the RL-MMSE case and it performs a matrix multiplication by \mathbf{T} . Implementation of this stage is also based on CORDIC processors which take as inputs rotation controls from the back-substitution engine as they are generated, and therefore does not increase latency.

IV. APPLICATION TO IEEE 802.11N WIRELESS LAN

Application of the described architecture to a Wireless LAN system as specified by the upcoming IEEE 802.11n standard [6] is studied in this section. The standard considers spatial multiplexing with up to 4 independent data streams, and a channel bandwidth of 20 MHz with optional support for 40 MHz. In this paper a 20 MHz channel is considered, which contains 64 OFDM subcarriers, of which 52 are used for data, 4 are pilots and 8 are nulled to fit the spectral mask.

A packet consists of a series of training symbols for channel estimation followed by N_{OFDM} data symbols, all of which have a duration of $T_{\text{OFDM}} = 4\mu\text{s}$ including cyclic prefix. Channel estimation is performed at the receiver upon reception of the last training symbol, and pre-processing operations can begin immediately afterwards. In order to achieve real-time operation, *i.e.* without requiring to store in buffers symbols from different packets, the time required for pre-processing and data processing for all data symbols must not exceed $(N_{\text{OFDM}} + 1)T_{\text{OFDM}}$. The previous condition ensures a packet is fully processed prior to the arrival of the next one. At low clock frequencies the total time required for pre-processing typically exceeds the OFDM symbol period. However, data processing can be typically performed faster. Therefore, the pre-processing overhead can be absorbed by data processing operations for symbols containing a sufficiently high number of data packets, allowing for real time operation to be achieved.

Fixed point simulations indicate a 16-bit representation for channel and received values and a CORDIC pipeline length $P_P = 18$ for QR decomposition and $S = 5$ lattice reduction sweeps are able to achieve performance close to floating-point detection. Therefore, processing of up to 18 subcarriers is

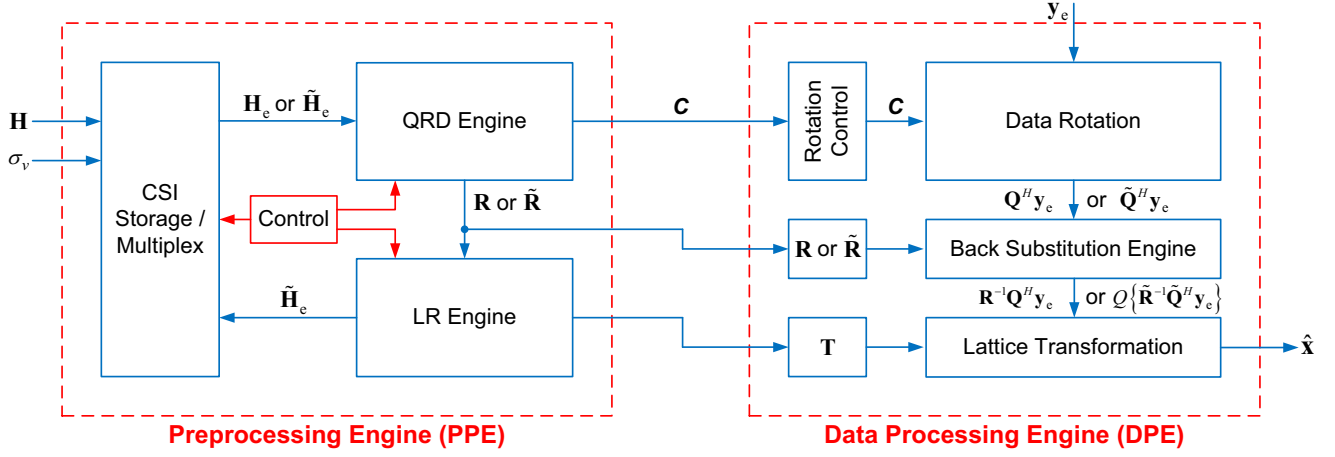


Fig. 1. Block diagram of the RL detector.

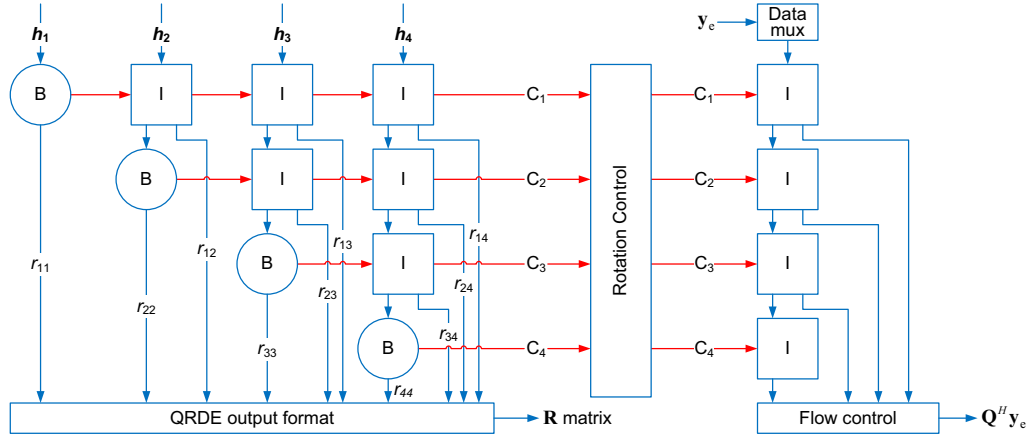


Fig. 2. QRD and Data Rotation block diagram.

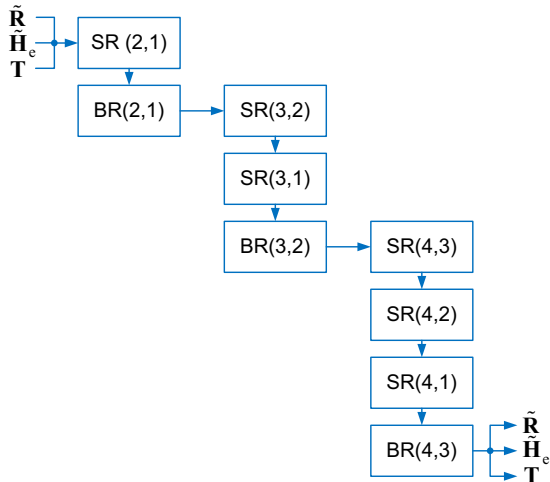


Fig. 3. Diagram of Lattice Reduction steps.

TABLE II
SIZE REDUCTION (SR)

Input: $\tilde{\mathbf{R}}, \tilde{\mathbf{H}}_e, \mathbf{T}, k, l$; Output: $\tilde{\mathbf{R}}, \tilde{\mathbf{H}}_e, \mathbf{T}$

- 1 $\mu = \left\lceil \frac{\tilde{\mathbf{R}}_{l,k}}{\tilde{\mathbf{R}}_{l,l}} \right\rceil \rightarrow \mu \in \{-1, 0, +1\}$
- 2 $\tilde{\mathbf{R}}(1:l, k) := \tilde{\mathbf{R}}(1:l, k) - \mu \tilde{\mathbf{R}}(1:l, l)$
- 3 $\mathbf{T}(:, k) := \mathbf{T}(:, k) - \mu \mathbf{T}(:, l)$
- 4 $\tilde{\mathbf{H}}_e(:, k) := \tilde{\mathbf{H}}_e(:, k) - \mu \tilde{\mathbf{H}}_e(:, l)$

TABLE III
BASIS REDUCTION (BR)

Input: $\tilde{\mathbf{R}}, \tilde{\mathbf{H}}_e, \mathbf{T}, k$; Output: $\tilde{\mathbf{R}}, \tilde{\mathbf{H}}_e, \mathbf{T}$

- 1 **if** $\delta |\tilde{\mathbf{R}}_{k-1, k-1}|^2 > |\tilde{\mathbf{R}}_{k, k}|^2 + |\tilde{\mathbf{R}}_{k-1, k}|^2$
- 2 Swap columns $k-1$ and k in $\tilde{\mathbf{R}}, \tilde{\mathbf{H}}_e$ and \mathbf{T}
- 3 $\Theta = \begin{bmatrix} \xi^* & \vartheta^* \\ -\vartheta & \xi \end{bmatrix}$ with $\xi = \frac{\tilde{\mathbf{R}}_{k-1, k-1}}{\|\tilde{\mathbf{R}}_{(k-1:k, k-1)}\|}$
 $\vartheta = \frac{\tilde{\mathbf{R}}_{k, k-1}}{\|\tilde{\mathbf{R}}_{(k-1:k, k-1)}\|}$
- 4 $\tilde{\mathbf{R}}(k-1:k, k-1:M) := \Theta \tilde{\mathbf{R}}(k-1:k, k-1:M)$
- 5 **end**

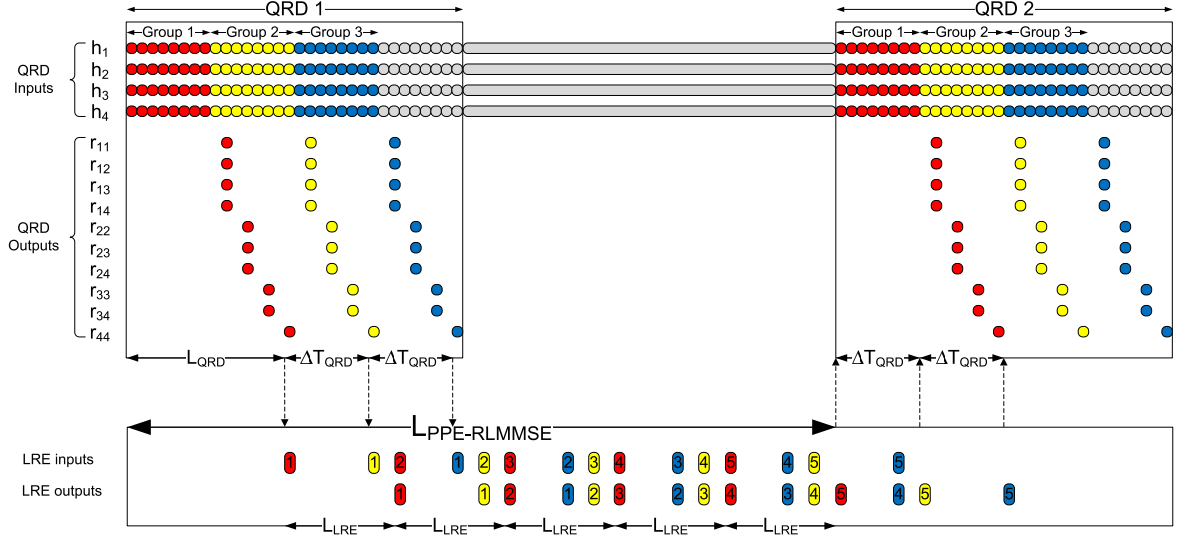


Fig. 4. Timing diagram for the pre-processing engine.

possible for the pipelined detector, and to account for all 52 active frequency tones, each OFDM symbol is divided in three groups of subcarriers. A timing diagram for the pre-processor is shown in Figure 4, where each circle at the top of the figure represents a single row of \mathbf{H}_e for a whole group of subcarriers entering the system, *i.e.* a circle is 18 clock cycles long. It takes therefore $\Delta T_{\text{QRD}} = 8P_P = 144$ cycles to enter each group of subcarriers. After $L_{\text{QRD}} = 15P_P = 270$ clock cycles, the first group of 18 \mathbf{R} matrices is ready for lattice reduction, with each subsequent group taking a further ΔT_{QRD} to be ready. The total time required for QR decomposition of all three groups of subcarriers is therefore $T_{\text{QRD}} = L_{\text{QRD}} + 2\Delta T_{\text{QRD}} + P_P = 576$ cycles.

The bottom part of Figure 4 shows the lattice reduction timing. It is observed that as soon as each full sweep through the lattice reduction pipeline is performed on each subcarrier group, the output is fed back to the input for a new iteration. When the desired number of sweeps $S = 5$ is reached, lattice-reduced channel values are fed back for a second QR decomposition. It should be noted that processing of the first data OFDM symbol in the DPE can begin at the same time as the second decomposition. The full pre-processing latency is therefore $L_{\text{PPE-RLMMSE}} = L_{\text{QRD}} + SL_{\text{LRE}}$, which in our case yields 1140 clock cycles.

Received values \mathbf{y}_e are multiplexed into the DPE in the same fashion as channel values into the QRD engine. A new OFDM symbol can enter the DPE every $3\Delta T_{\text{QRD}}$ clock cycles, and the total number of data processing cycles is therefore $T_{\text{DPE}} = 3(N_{\text{OFDM}} - 1)\Delta T_{\text{QRD}} + T_{\text{QRD}} + L_{\text{BS}}$. The back substitution CORDIC pipeline length determined by simulation is $P_{\text{BS}} = 8$, yielding $T_{\text{DPE}} = 432(N_{\text{OFDM}} - 1) + 633$. For a given clock frequency $f_{\text{clk}} = 1/T_{\text{clk}}$ and considering the previous latency values, N_{OFDM} must satisfy the inequality

$$(L_{\text{PPE}} + T_{\text{DPE}})T_{\text{clk}} \leq (N_{\text{OFDM}} + 1)T_{\text{OFDM}} \quad (12)$$

for real time RL-MMSE detection. As N_{OFDM} is known by the receiver prior to data processing, it is easy to switch to MMSE mode, where real time operation is easily achieved, if it is smaller than required for RL detection.

The described detector was coded in VHDL and implemented in an Altera EP2S180 FPGA. Evaluation was performed using a link-level IEEE 802.11n simulator employing TGn channel model B [7]. Measured bit-error-rate (BER) curves for both modes of operation and different modulation formats are shown in Figure 5, where the great performance gains achieved by reduced lattice detection are appreciated. It is observed that the linear MMSE detector requires over 10 dB higher signal-to-noise ratio (SNR) compared to the reduced lattice detector in order to achieve an average BER of 10^{-2} .

The maximum clock frequency reported by the FPGA implementation prior to any optimization is 153.82 MHz which, according to the previous latency results, requires at least $N_{\text{OFDM}} = 4$ symbols per packet for real-time reduced lattice detection. The actual number of data bytes per symbol depends on the modulation format and coding scheme employed. Figure 6 shows the required data payload to satisfy the real-time requirement for several configurations employing 4 transmit antennas. It is observed that for the FPGA clock frequency, a payload of nearly 400 bytes is necessary if the highest order modulation format (64QAM) and highest rate coding scheme (rate 5/6) considered by the standard are employed. Such configuration is able to achieve a maximum information throughput of 260 Mbps.

It is also observed that an increase in clock frequency to around 222MHz would be necessary for real-time detection of packets containing a single data OFDM symbol for all configurations supported by the standard. An ASIC implementation is expected to provide the required improvement in clock frequency with an estimated complexity of 2 Mgates including the necessary RAM storage.

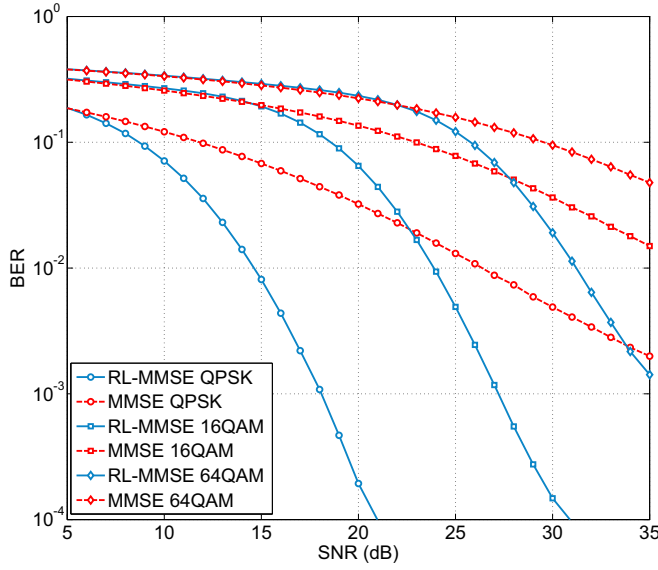


Fig. 5. BER performance for the RL and linear MMSE detectors.

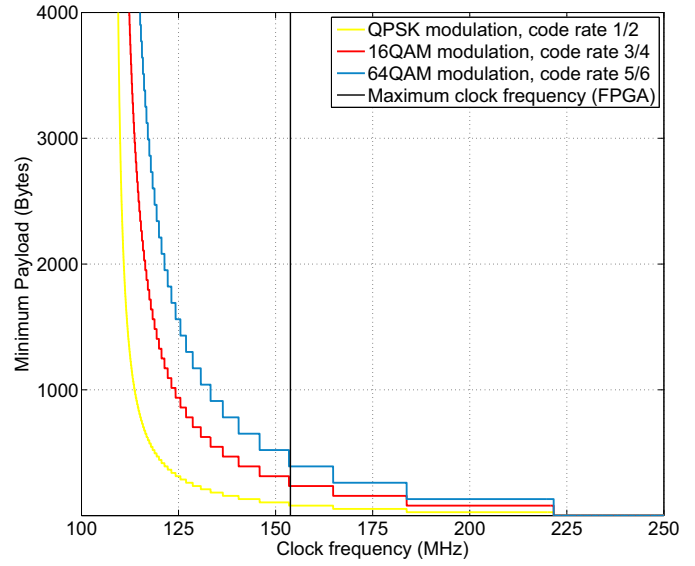


Fig. 6. Required packet payload for real-time RL-MMSE detection.

V. CONCLUSIONS

In this paper we have presented a novel VLSI implementation of a MIMO detector for OFDM systems. The proposed architecture is able to perform both linear MMSE and near-optimal reduced-lattice-aided detection, providing a practical solution for receivers requiring both high performance and flexibility. Moreover, thanks to multiple algorithm-level optimizations and a highly pipelined structure, the detector is able to achieve the throughputs required by modern wideband communications systems. In this respect, application to IEEE 802.11n wireless LAN systems has been discussed, and fixed point detection performance has been evaluated with an FPGA prototype, showing great performance gains compared to linear detection algorithms.

ACKNOWLEDGMENT

The authors would like to acknowledge the fruitful discussions with their colleagues at Toshiba Research Europe Ltd. and the support of its directors.

REFERENCES

- [1] D. Wübben, R. Böhnke, V. Kühn, and K.-D. Kammeyer, "MMSE-based lattice-reduction for near-ML detection of MIMO systems," in *Proceedings of ITG workshop on Smart Antennas*, Munich, Germany, Mar. 2004, pp. 106–113.
- [2] A. K. Lenstra, H. W. Lenstra, and L. Lovasz, "Factoring polynomials with rational coefficients," *Mathematische Annalen*, vol. 261, pp. 515–534, 1982.
- [3] H. Vetter, V. Ponnampalam, M. Sandell, and P. A. Hoeher, "Fixed complexity LLL algorithm," *IEEE Transactions on Signal Processing*, to appear in.
- [4] B. Gestner, W. Zhang, X. Ma, and D. V. Anderson, "VLSI implementation of a lattice reduction algorithm for low-complexity equalization," in *Proc. IEEE International Conference on Circuits and Systems for Communications*, May 2008, pp. 643–647.
- [5] Y. Hu, "CORDIC-based VLSI architectures for digital signal processing," *IEEE Signal Processing Magazine*, vol. 9, no. 3, pp. 16–35, July 1992.
- [6] IEEE 802.11n/D2.0, "Draft Amendment to Standard for Information Technology - Telecommunications and information exchange between systems - Local and Metropolitan networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Enhancements for Higher Throughput," IEEE Std 802.11, Feb. 2007.
- [7] V. Erceg, L. Schumacher, P. Kyritsi *et al.*, "TGn channel models," *IEEE 802.11 document 11-03/0940r4*, 2004.