

# A Low Complexity and High Throughput MIMO Detection VLSI Design for MIMO-OFDM Systems

Zhaohui Cai, Yu Hong Wang, Suttinan Chattong  
Institute for Infocomm Research, 1 Fusionopolis Way, Singapore 138632

**Abstract-** This paper presents a linear Minimum Mean Square Error (MMSE) MIMO Detector design for MIMO-OFDM systems based on Application-Specific Instrument-set Processor (ASIP). As part of the IEEE 802.11ac-compliant PHY baseband transceiver, the proposed MIMO detector offers low latency, high throughput with efficient resource utilization. The design has been synthesized with TSMC 40 nm CMOS technology, the logic gate count for each QRD engine is about 245 K gates. It is able to support 20/40/80MHz bandwidth and up to 4 spatial streams. Detection latency for 80 MHz VHT mode (234 data sub-carriers) is 750 ns.

## I. INTRODUCTION

With the techniques such as advanced channel coding, multiple-input multiple output (MIMO) and orthogonal frequency-division multiplexing (OFDM), wireless local area networking (LAN) has experienced tremendous growth in the last ten years with the proliferation of IEEE 802.11 devices. MIMO systems employ multiple antennas at both sides of the wireless link and are able to improve the data throughput significantly by transmitting multiple data streams concurrently over the same frequency band. On the other hand, OFDM is well suited to wideband systems in frequency selective fading environments. OFDM is tolerant of time synchronization errors and drastically simplifies equalization design at the receiver. Now MIMO in combination with OFDM constitutes the basis for many wireless standards [1-2].

For MIMO-OFDM systems, the pre-processing circuit to obtain the filtering matrix from the channel estimation is a crucial component of the MIMO detector and an interesting research topic. Pre-processing algorithms perform all operations necessary to compute an estimate of the transmitted symbol, which do not depend on the data symbols [3].

Many MIMO detection algorithms have been reported in the literature, such as, linear MMSE detection [3-4], SIC detectors and tree-search based detection [5-6]. As the time available for PHY layer turn-around is short due to stringent latency constraints imposed by the IEEE 802.11n/ac standard: the combined latency has to be less than 16  $\mu$ s short inter-frame space (SIFS) constraint, most 11n transceivers apply linear MMSE detection [3] due to its simplicity compared to SIC and sphere decoding.

There are two main architectures have been proposed to implement preprocessing circuits reported in the literature, namely, processor-like (ASIP-based) architectures [7-9] and pipelined architectures [3-4].

While the pipelined solutions divide the algorithm to be implemented into multiple-tasks and assign the individual tasks

to dedicated modules that are separated by pipeline stages, an ASIP-based design computes all tasks required on the same piece of hardware with one or more customized processor cores associated with dedicated instruction ROMs. All the data processing is handled by some dedicated execution units.

Compared with the pipelined architectures, one advantage of the ASIP-based architecture is its ability to adapt to algorithm/interface changes by reprogramming the instruction ROM. For example, the ASIP engine proposed for MIMO detection in this paper can be reused for other tasks such as V compression required for the steering matrix computation in the beamforming (BF) mode [2] by adding some hardware resources, as there is no MIMO detection activities for the null data packet (NDP) processing.

In this paper, as part of our IEEE 802.11ac-complaint PHY baseband transceiver design, an ASIP based VLSI architecture is implemented as the QR decomposition (QRD) engine as part of the MIMO detector. Other tasks such as channel estimation, MMSE matrix generation, bias generation and data filtering are implemented with pipelined modules (This is different from our PHY baseband transceiver designed for IEEE 802.11n, where the channel estimation and the symbol detection are also part of the ASIP engine's task, as the computation load for QRD in 11n is just roughly the half of that in 11ac).

In order to meet the 11ac latency constraint, two dedicated QRD engines are instantiated, one for odd-indexed subcarriers and one for even-indexed subcarriers, respectively. Our design offers low latency, high throughput MIMO detection with efficient resource utilization. With CMOS 40 nm processing technology, the equivalent gate-count (GE) for each QRD engine is about 245 kGE. The latency of 80 MHz VHT mode (234 data sub-carriers) is 750 ns.

The remainder of the paper is organized as follows. Section II introduces the IEEE 802.11ac system and presents the packet error rate (PER) performance result of an 11ac-compliant PHY that we have developed. The linear MMSE MIMO detection algorithm used in our design is also presented in this section. Section III discusses the VLSI architecture design for the MIMO detection in detail. The implementation result is compared with work of others in this section as well. The summary is given in section IV.

## II. THE IEEE 802.11AC SYSTEM AND LINEAR MMSE MIMO DETECTION

### A. The IEEE 802.11ac MIMO-OFDM System

IEEE 802.11ac, like its predecessor IEEE 802.11n, offers considerable advantages. 802.11ac (aka VHT, Very High Throughput) devices are required to support 20, 40, and 80 MHz channels and 1 spatial stream. Several optional features are also defined in 802.11ac, such as, 160/80 + 80 MHz bandwidth, 256 QAM, up to 8 spatial streams, Multi-User (MU) MIMO and low density parity check (LDPC) codes support. An 11ac device making use of only the mandatory parameters (80 MHz bandwidth, 1 spatial stream, and 64 QAM 5/6) will be capable of a data rate of ~293 Mbps while a device that implements all optional parameters (8 spatial streams and 256 QAM with rate 5/6 with a short guard interval) will be able to achieve almost 3.5 Gbps [2].

We have designed an IEEE802.11ac-compliant MIMO-OFDM PHY baseband transceiver with some optional features support, such as, up to 4 spatial streams support, 256 QAM, based on a systemC bit-true platform. The baseband transceiver has been synthesized with TSMC CMOS 40 nm technology. Our design supports NON-HT, HT-MF, and VHT modes. The PHY baseband transceiver block diagram is sketched in Figure 1.

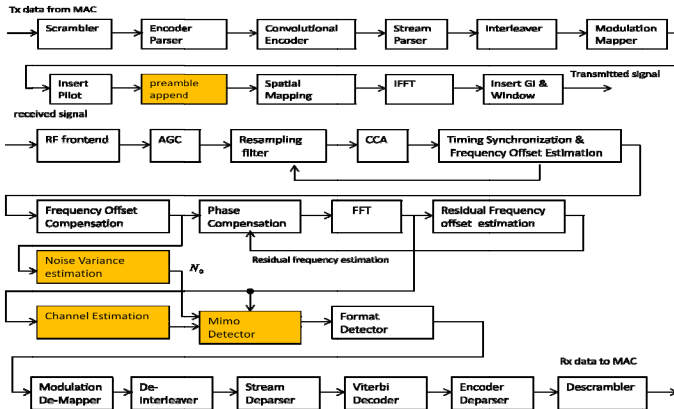


Figure 1 IEEE 802.11ac PHY baseband transceiver block diagram

With  $N_{ss} = 4$ , MCS = 9 (VHT 80 MHz mode), and 800 ns long guard interval (GI), our design is able to offer a maximum data rate of 1.56 Gbps. Using our SystemC platform, simulated PER performance for 802.11ac 4x4 VHT mode with convolutional coding and different number of spatial streams ( $N_{ss}$ ) is shown in Figure 2. In the simulation, the frequency offset of -13.75 ppm. The propagation channel is 80 MHz TGn channel B, and packet length is 1 Kbytes. The blue-solid curves and the red-dotted curves denote floating-point and fixed-point results, respectively.

One of the major parts of the PHY baseband transceiver is the MIMO detection, which usually contributes to a large percentage of the silicon area. The MIMO detection consists of data filtering and pre-processing tasks, i.e., channel estimate, post-equalization signal to interference and noise ratio (SINR) and  $G_{MMSE}$  computation. Our MIMO detector is designed to work with up to 4 transmitter antennas and 4 receiver antennas, and to support up to 4 spatial streams.

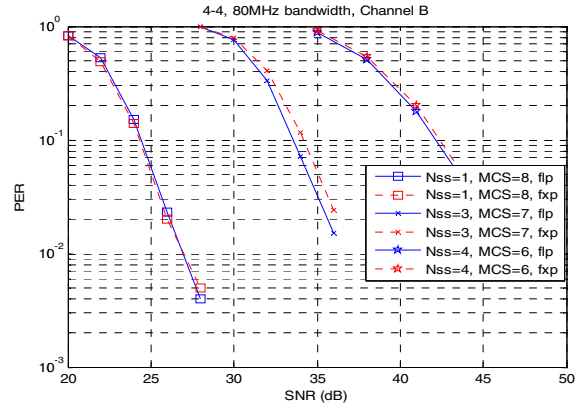


Figure 2 11ac VHT performance for Channel B

## B. The Linear MMSE MIMO Detection Algorithms

### 1) Channel Estimation

Channel estimation is performed using the L-LTF field and VHT-LTF (HT-LTF) field in frequency domain (after receiver FFT) for detection of legacy field (such as L-SIG and VHT-SIGA) and VHT field, respectively. In the following, we describe the channel estimation algorithm using VHT-LTF field. Suppose  $N_{ss}$  transmit spatial stream and  $N_{RX}$  receive antennas are used. Let  $y_{l,k}^{(i)}$  denote the received signal of  $k$ -th subcarrier of  $i$ -th received antenna during  $l$ -th ( $l=1, 2, \dots, N_{VHT}$ ) VHT-LTF symbol, and  $d_{l,k}^{(j)}$  denote transmitted frequency domain signal from  $j$ -th ( $j=1, 2, \dots, N_{ss}$ ) spatial stream during  $l$ -th VHT-LTF symbol. Then received vector  $\mathbf{y}_{l,k} = [y_{l,k}^{(1)}, \dots, y_{l,k}^{(N_{RX})}]^T$  during  $l$ -th VHT-LTF ( $l=1, 2, \dots, N_{VHT}$ ) symbol for  $k$ -th subcarrier can be written as

$$\mathbf{y}_{l,k} = \mathbf{H}_k \mathbf{d}_{l,k} + \mathbf{n}_{l,k}, \quad (1)$$

where  $\mathbf{H}_k$  denotes the  $N_{RX} \times N_{ss}$  channel matrix with the  $(p, q)$ <sup>th</sup> element being the random fading between  $p$ <sup>th</sup> receive antenna and  $q$ <sup>th</sup> transmit spatial stream.  $\mathbf{d}_{l,k} = [d_{l,k}^{(1)}, \dots, d_{l,k}^{(N_{ss})}]^T$  is the  $N_{ss} \times 1$  transmitted data vector during  $l$ -th VHT-LTF symbol.  $\mathbf{n}_{l,k}$  is the  $N_{RX} \times 1$  complex Gaussian noise vector with covariance matrix  $N_0 \mathbf{I}$ . Let  $\mathbf{Y}_k = [\mathbf{y}_{1,k}, \dots, \mathbf{y}_{N_{VHT},k}]$ , and  $\mathbf{D}_k = [\mathbf{d}_{1,k}, \dots, \mathbf{d}_{N_{VHT},k}]$ , then Least Square (LS) estimation of  $\mathbf{H}_k$  can be written as,

$$\hat{\mathbf{H}}_k = (\mathbf{D}_k^H \mathbf{D}_k)^{-1} \mathbf{D}_k^H \mathbf{Y}_k, \quad (2)$$

where  $(\cdot)^H$  is conjugate-transpose operation.  $N_{VHT}$  is number of VHT-LTF symbols in VHT-LTF field.

### 2) Noise variance estimation

To perform the MMSE detection, we need to know noise variance  $N_0$ . Noise variance estimation algorithm makes use of 2 identical L-LTFs present in the preambles of an IEEE 802.11a/n/ac packet. Let  $r_{L-LTF}^{(i)}(k)$  and  $r_{L-LTF}^{(i)}(k + N_{FFT})$  ( $k=0, 1, 2, \dots, N_{FFT}-1$ ) represent the time domain samples (before

receiver FFT) at the  $i$ -th receiver antenna of first received L-LTF symbol and second L-LTF, respectively, and  $N_{FFT}$  is number of samples in one L-LTF symbol. Then an estimate of the SNR can be written as

$$SNR = \frac{\sum_{i_R=1}^{N_{RX}} \sum_{k=0}^{N_{FFT}-1} \left| \frac{1}{2} \{ r_{L-LTF}^{(i)}(k) + r_{L-LTF}^{(i)}(k+nN_{FFT}) \} \right|^2}{2 \sum_{i_R=1}^{N_{RX}} \sum_{k=0}^{N_{FFT}-1} \left| \frac{1}{2} \{ r_{L-LTF}^{(i)}(k) - r_{L-LTF}^{(i)}(k+nN_{FFT}) \} \right|^2}, \quad (3)$$

and the estimate of  $N_0$  can be written as

$$N_0 = \frac{1}{SNR}. \quad (4)$$

### 3) Linear MMSE MIMO Detection

After the channel estimates  $\hat{\mathbf{H}}_k$  is obtained, the linear MMSE MIMO detector can be calculated. Without loss of generality, in the following section, we use  $\mathbf{H}$  to denote the channel estimation of subcarrier  $k$ . Then MMSE matrix  $\mathbf{G}_{MMSE}$  can be written as,

$$\mathbf{G}_{MMSE} = (\mathbf{H}^H + N_{SS}N_0\mathbf{I})^{-1}\mathbf{H}^H. \quad (5)$$

The estimate of the transmitted symbol vector  $\hat{\mathbf{y}}$  is

$$\hat{\mathbf{y}} = \mathbf{G}_{MMSE}\mathbf{y}. \quad (6)$$

The per-stream post-equalization SINR is given by

$$SINR = \frac{\text{bias}}{1-\text{bias}}, \quad (7)$$

where,  $\text{bias} = \text{diag}(\text{abs}(\text{real}(\mathbf{G}_{MMSE} \times \mathbf{H})))$

Most MMSE detector implementations reported in the literature apply an explicit matrix inversion of the matrix  $\mathbf{A} = \mathbf{H}^H + N_{SS}N_0\mathbf{I}$ , [10] employs direct matrix inversion with Strassen's method by exploring the Hermitian property of the matrix  $\mathbf{A}$ , while [12-13] applies QRD on  $\mathbf{A}$  as  $\mathbf{A} = \mathbf{QR}$ , and subsequently, the matrix inversion problem can be solved as  $\mathbf{A}^{-1} = \mathbf{R}^{-1}\mathbf{Q}^H$ .

Another approach is to work on the augmented channel matrix defined as

$$\mathbf{H}_{aug} = \left[ \sqrt{\frac{\mathbf{H}}{N_{SS}N_0\mathbf{I}}} \right], \quad (8)$$

and work on the QRD of  $\mathbf{H}_{aug}$  as

$$\mathbf{H}_{aug} = \mathbf{QR} = \begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \end{bmatrix} \mathbf{R}, \quad (9)$$

so that

$$\mathbf{G}_{MMSE} = \frac{\mathbf{Q}_2\mathbf{Q}_1^H}{\sqrt{N_{SS}N_0}} \quad (10)$$

This approach eliminates the need for matrix inversion. It also significantly reduces the precision requirements of the system [4].

Same as in [3] and [4], the solution based on QR Decomposition of  $\mathbf{H}_{aug}$  is adopted in our MIMO detector design and the Modified Gram-Schmidt (MGS) algorithm is applied to decompose the  $\mathbf{H}_{aug}$ . The MGS QR decomposition has an advantage in numerical stability when compared to the original Gram-Schmidt algorithm. The MGS algorithm is summarized in Algorithm 1.

#### Algorithm 1: Algorithm for QR decomposition

```

[Q, R] = MGS( $\mathbf{H}_{aug}$ )
[m, n] = size( $\mathbf{H}_{aug}$ );
Q = [ $q_1, q_2, \dots, q_{N_{SS}}$ ]  $\llcorner$   $\mathbf{H}_{aug}$ ;
R =  $0_n$ ;
for i = 1:n
     $r_{ii} = \text{sqrt}(q_i^H * q_i)$ 
     $\text{inv}_r_{ii} = 1/r_{ii}$ 
     $q_i = q_i * \text{inv}_r_{ii}$ 
    for j = i + 1 : n
         $r_{ij} = q_i^H q_j$ 
         $q_j = q_j - r_{ij}q_i$ 
    end;
end;

```

### III. PROPOSED MIMO DETECTOR VLSI ARCHITECTURE

#### A. The Latency Requirement

In 802.11n/ac the MIMO channel estimate is carried out with the HT/VHT long training field (LTF) symbols. The LTF symbols are followed immediately by OFDM data symbols, as show in the Figure 3.

This leaves very limited time for the pre-processing of the MIMO detection, i.e., channel estimation and MMSE weight computation.

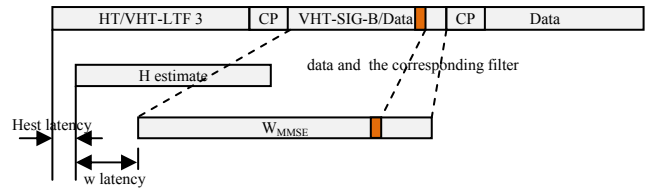


Figure 3 Target MIMO detection timing

Ideally as long as the particular subcarrier MMSE matrix is ready once the corresponding data sample for this subcarrier is available from the phase compensation module, the latency of pre-processing (channel estimation and  $\mathbf{G}_{MMSE}$  computation) added to the receiver latency is zero, i.e., there is no memory needed to buffer the data symbols; the MIMO detection latency just comes from the symbol detection.

For long GI systems with 800 ns cyclic prefix, this requires that the combined pre-processing should be less than 4  $\mu\text{s}$  for the 1<sup>st</sup> data subcarrier, and should be less than 8  $\mu\text{s}$  for the last data subcarrier.

Our design is very close to this target: the MIMO detection latency for 3<sup>rd</sup> OFDM data symbol onwards is just 750 ns (60 data samples for 80 MHz VHT mode).

#### B. The Top level Architecture Overview

The top level structure for our design is shown in Figure 4, where, the yellow color blocks are implemented using pipelined architecture, and light-blue color blocks are implemented using ASIP architecture.

It can be seen from the Algorithm 1, the MGS is an iterative process with computational intensive blocks such as complex vector dot product, square root and division. In case of 4x4 MIMO detector, the complex vector size is 8x1, and 8x1 complex vector dot product need 32 real multipliers. ASIP

based design is a solution to reuse these computational resources efficiently.

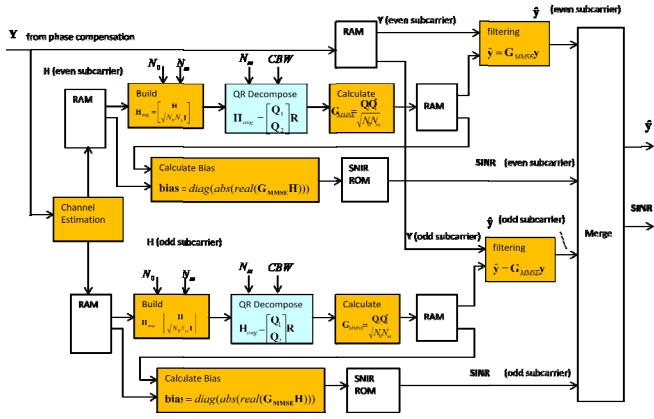


Figure 4 MIMO detector top-level architecture

The MIMO detector receives (i) vector  $\mathbf{y}$  from phase compensation module (ii) estimated noise variance (from noise variance estimator module) (iii)  $N_{SS}$  and CBW (bandwidth) as its input. Based channel estimation, an  $8 \times 4$  augmented channel matrix  $\mathbf{H}_{aug}$  is build according to eqn (8).

$\mathbf{H}_{aug}$  is pushed to the QRD processors. To improve the throughput, two QRD engines to decompose  $\mathbf{H}_{aug}$  and two MMSE weight matrix generation, two bias generation and two data filtering are instantiated, one for odd-indexed subcarriers and one for even-indexed subcarriers.

Once QRD is completed the MMSE matrix generator block reads  $\mathbf{Q}$  matrices through memory interface from QRD processors and performs per-subcarrier MMSE matrix generation according to eqn (10) and filter the received signal vector  $\mathbf{y}$  to generate  $\hat{\mathbf{y}}$  according to eqn (6). In addition to this, it also performs per-subcarrier SINR calculation as in eqn (7).

Wherever possible, we try to overlap the processing of all these blocks to reduce latency, resulting in the scheduling diagram as shown in the figure below (using  $N_{SS}=4$  as example).

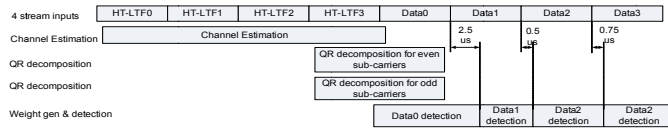


Figure 5 Implemented top-level scheduling for the MIMO detection

The most intensive part of computation occurs between the beginning of (V)HT-LTF3 and the end of the 1<sup>st</sup> data symbol (Data0), as this is where we have either 3 or all the 4 blocks being active at the same time. During this period, priority is given to MMSE matrix generation and not data detection, hence the detection latency for the first data symbol is longer compared to subsequent data symbols. From the middle of second data symbol onwards, MMSE weight generation has completed and we are only left with the detection task. Here we have plenty of cycles and resources available, so the

detection is performed at an accelerated rate in an attempt to reduce detection latency for the second and subsequent data symbols. This results in  $0.5 \mu s$  latency for the second data symbol and  $0.75 \mu s$  latency for all subsequent data symbols.

### C. The QRD Processing Engine

The QRD engine is implemented by ASIP architecture, as shown in Figure 6.

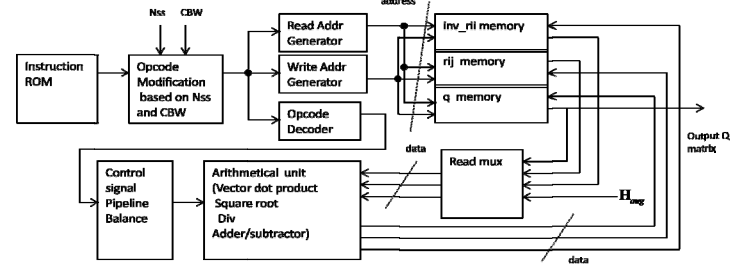


Figure 6 ASIP architecture of the QRD engine

This QRD engine comprises of

- (1) Instruction ROM – this contains custom program (sequence of instructions) which implements the function of the QRD.
- (2) Opcode Modification based on  $N_{SS}$  and CBW - since our MIMO detector need to support different bandwidth and different  $N_{SS}$ , the number of the combination of bandwidth and spatial stream is 12, and if we design a different instruction sequence for each of this combination, the instruction ROM would be very huge. This module is to online generate the modified instruction sequence based on number of spatial stream, bandwidth and basic instruction sequence.
- (3) Instruction Decoder – this module decodes each modified instruction into control signals required by the Arithmetic Unit.
- (4) Data Memory and read/write address generate –data memory stores  $inv_{ril}$ ,  $q$ , and  $r_{ij}$  in memory (refer to Algorithm 1). The data is read by the Arithmetic Unit which performs processing, and then written back to the data memory.
- (5) Pipeline Balancer – this is a shift register which acts to balance data memory access delay with the decoded control signals, thus ensuring that these signals arrive at the Arithmetic Unit at the same time.
- (6) Arithmetic Unit – this block performs the mathematical computations as dictated by the opcodes within the instruction ROM. Refer to Algorithm 1, the mathematical computations includes vector dot product, square root, div and adder and subtractor.

The program flow of the system is linear without loop control. After reset is de-asserted, the QRD module starts in idle state. It starts operation in response to a trigger pulse from the MIMO channel estimation module. The program flow is linear, meaning the address output of program counter freezes when the last instruction is reached and will only be reset again when the module is disabled or reset.

The QRD module produces a start trigger pulse to the MMSE matrix generator module, telling it when to start MMSE matrix calculation. Both MMSE matrix and SINR values are calculated only once per frame. The data detection (filtering) process is done every symbol for all sub-carriers.

#### D. The Implementation Result

The MIMO detection design, together with other blocks of an 11ac PHY baseband transceiver, has been synthesized with Synopsys<sup>®</sup> Design Compiler using 40 nm CMOS library, with a clock frequency of 320 MHz (This clock frequency constrain is imposed by the fact that free slow SRAM is used in our PHY baseband transceiver design). The logic gate count breakdown for the proposed MIMO detector is listed in Table 1 (excluding the slow SRAM). Comparison of our design with other MIMO detection architectures is presented in Table 2.

**Table 1 Logic Gate count breakdown for the proposed MIMO detector**

Module	Logic Gate (kGE)	Percentage (%)
Channel Estimator	30	5
QRD processor 0	245	42
QRD processor 1	245	42
$W_{MMSE}$ and filtering	60	11
Total	580	100

**Table 2 Comparison of the MIMO detection implementations**

	[4]	[3]	[12]	[13]	[8]	[9]	This work
Algorithm	MGS	MGS	Givens	Givens	Givens	MCMC	MGS
Architecture	Pipelined	Pipelined	Pipelined	Pipelined	ASIP	ASIP	ASIP
Dimension	4x4	4x4	4x4	4x4	4x4	4x4	4x4
Modulation	64QAM	64QAM	64QAM	64QAM	16QAM	64QAM	256QAM
Clock Freq	360 MHz	140 MHz	250 MHz	100 MHz	272 MHz	400 MHz	320 MHz
Gate count	473 K	-	66 K	152 K	0.41mm <sup>2</sup>	482 K	580 K
Technology	90 nm	FPGA	130 nm	180 nm	180 nm	65 nm	40 nm
Spec.	11n	11n	N.A.	N.A.	N.A.	N. A.	11ac
Latency	0.95 $\mu$ s	2.77 $\mu$ s	0.032 n $\mu$ s	0.04n $\mu$ s	1.41 $\mu$ s	-	0.75 $\mu$ s
Throughput	20 M	14.4 M	1000/32n	1000/40n	1/1.41n	-	29.3 M

The last row in the Table 2 is for the QRD processing throughput in Million Matrices per second, which is different from the one listed in [9]. Our design is able to process 234 sub-carriers within 8  $\mu$ s time, i.e., the throughput  $> 234/8 \mu$ s = 29.25 M matrices/s. Noted that the entries marked with ‘N.A.’ in ‘Spec.’ indicate the values (Gate count, latency and throughput) are for QRD only. Others are for entire MIMO detection and follow the corresponding specifications. To authors’ knowledge, our design is the first complete 11ac MIMO detection solution presented in public.

In this paper we have presented a VLSI architecture for MIMO detector which adopts hybrid pipelined and ASIP architecture. Our MIMO detector can support 12 different bandwidth and  $N_{ss}$  combinations. Instead of designing a different instruction sequence for each of this combination, we use a novel method which only uses one basic instruction ROM and another Opcode modification module to online modify instruction based on  $N_{ss}$  and bandwidth. And total ASIC gates count is 580 kGE. The latency of 80 MHz VHT mode (234 data sub-carriers) is 750 ns.

By adding COordinate Rotation Digital Computer (CORDIC) engine into the ASIP Arithmetic Unit, our ASIP is able to support the compression of the  $\mathbf{V}$  matrix in the beamforming mode of the 11ac, as the  $\mathbf{V}$ -compression essentially consists of a series of Givens rotations of the orthogonal matrix  $\mathbf{V}$  derived from the singular value decomposition (SVD) of the channel  $\mathbf{H}$ .

This MIMO detector is part of an IEEE 802.11ac-compliant PHY baseband transceiver. Our PHY baseband transceiver design can support up to  $N_{ss} = 4$  and 256 QAM, with a maximum data rate of 1.56 G bps.

#### REFERENCES

- [1] IEEE Std 802.11n-2009 Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Oct. 2009.
- [2] IEEE P802.11ac, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Enhancements for Very High Throughput for Operation in Bands Below 6GHz, Dec. 2013.
- [3] H S Kim, W Zhu, J Bhatia, K Mohammed, A Shah, and B Daneshrad, “A practical, hardware friendly MMSE detector for MIMO-OFDM-based systems”, *EURASIP Journal on Advances in Signal Processing*, pp1-14, Jan 2008.
- [4] C C S D Senning, Energy efficient VLSI circuits for MIMO-WLAN, *PhD dissertation, EPFL*, 2014
- [5] C Studer, Iterative MIMO decoding: algorithms and VLSI implementation aspects, *Ph.D. dissertation, ETH Zürich, Switzerland, Series in Microelectronics*, vol. 202, Hartung-Gorre Verlag Konstanz, 2009.
- [6] E M Witte, F Borlenghi, G Ascheid, R Leupers, and H Meyr, “A scalable VLSI architecture for soft-input soft-output single tree-search sphere decoding”, *IEEE Trans. Circ. Systems II*, vol. 57, no. 9, pp. 706–710, Sept. 2010.
- [7] J P Lüthi, VLSI Circuits for MIMO Preprocessing. *Ph.D. dissertation, ETH Zürich, Switzerland, Series in Microelectronics*, vol. 203, Hartung-Gorre Verlag Konstanz, 2010.
- [8] C Studer, P Blösch, P Friedli, and A Burg. “Matrix decomposition architecture for MIMO systems: Design and implementation trade-offs”, *Proc. IEEE ASIOMAR*, pp. 1986–1990, Pacific Grove, USA, Nov. 2007.
- [9] X Chen, A Minwegen, S B Hussain, A Chattopadhyay, G Ascheid, and R Leupers, “Flexible, efficient multimode MIMO detection by using reconfigurable ASIP”, *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol.23, no.10, pp.2173-2186, Oct. 2015.
- [10] I LaRoche and S Roy, “An efficient regular matrix inversion circuit architecture for MIMO processing”, *IEEE ISCAS*, pp. 4819–4822, Island of Kos, Greece, May 2006.
- [11] A Burg, S Haene, D Perels, P Luethi, N Felber, and W Fichtner, “Algorithm and VLSI architecture for linear MMSE detection in MIMO-OFDM systems”, *IEEE ISCAS*, pp 4102- 4105, Island of Kos, Greece, 2006.
- [12] J H Yoon, D Shin, J Park, “A low-complexity composite QR decomposition architecture for MIMO detector”, *IEEE ISCAS*, pp. 1692-1695, Melbourne, Australia, Jun. 2014.
- [13] Z Huang, P Tsai, “Efficient Implementation of QR Decomposition for Gigabit MIMO-OFDM Systems”, *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol.58, no.10, pp.2531-2542, Oct. 2011.