

Low Complexity Hardware Interleaver for MIMO-OFDM based Wireless LAN

Rizwan Asghar and Dake Liu

Dept. of Electrical Engineering, Linköping University
Linköping, Sweden
{rizwan, dake} @ isy.liu.se

Abstract—A low complexity hardware interleaver architecture is presented for MIMO-OFDM based Wireless LAN e.g. 802.11n. Novelty of the presented architecture is twofold; 1) Flexibility to choose interleaver implementation with different modulation scheme and different size for different spatial streams in a multi antenna system, 2) Complexity to compute on the fly interleaver address is reduce by using recursion and is supported by mathematical formulation. The proposed interleaver architecture is implemented on 65nm CMOS process and it consumes 0.035 mm² area. The proposed architecture supports high speed communication with maximum throughput of 900 Mbps at a clock rate of 225 MHz.

I. INTRODUCTION

Multiple data streams are transmitted in parallel using different antennas to increase the throughput and range in MIMO-OFDM based Wireless LAN. Newly introduced IEEE 802.11n standard [1] is the first wireless LAN standard based on MIMO-OFDM. Significant improvement in range-rate performance as compared to WLAN (IEEE-802.11-a/b/g) [2] and WIMAX (IEEE-802.16e) [3] makes MIMO-OFDM the ideal solution not only for wireless LAN, but also for home entertainment networks.

The performance of MIMO-OFDM is coupled with good interleaver design. Most of the literature available [4]–[9] covers the performance and evaluation of interleaver design for a high speed communication system, however few papers [10]–[12] focus on architecture design for high throughput. Among these, only [12] aims at the hardware architecture of interleaver for MIMO-OFDM Wireless LAN supporting multi antenna streams and all the rest of literature focus on WLAN or WiMAX.

The transmission in 802.11n can be distributed among four spatial streams as shown in Fig. 1. After encoding, a parser sends blocks of bits to different spatial streams with block size N_{CBPS} being number of codes bits per sub-carrier. These bits are interleaved by a block interleaver (one for each spatial stream) to achieve the performance benefits from both spatial diversity and frequency diversity. Considering more than one spatial stream transmission, the interleaving address for one spatial stream is independent of other. Thus using four interleaver blocks in parallel the total interleaving can be considered as 3D memory block with more than one address generation units as shown in Fig 2. The challenge is to simplify the address computation for all blocks and at the same time meeting the high throughput requirements which is mandatory for MIMO-OFDM system. This paper presents the idea of transformation of basic interleaving equations to 2-dimensional equations and also introduces recursive computing to reduce the implementation complexity. In this paper, section III and IV present 2-dimensional transformation and further simplification steps for spatial and frequency interleaving. The architecture overview and implementation results are presented in section V and VI respectively followed by conclusion.

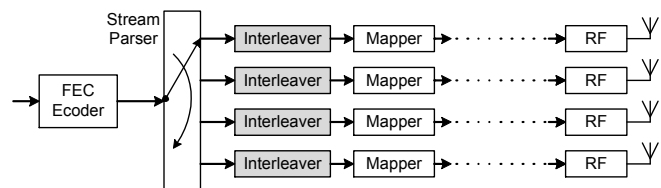


Fig. 1: Use of Interleaver in 802.11n spatial streams

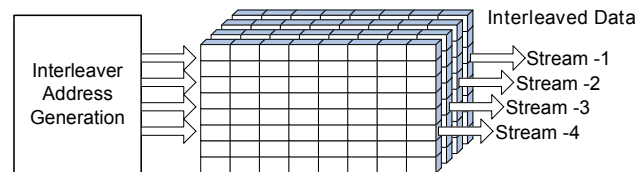


Fig. 2: 3D view of interleaver memory configuration for MIMO-OFDM Wireless LAN

II. INTERLEAVER FOR 802.11N

The interleaving in 802.11n is based on a block interleaver consisting of spatial interleaving and frequency interleaving. Spatial interleaving hereafter called basic interleaving is expressed in the form of a set of two equations for two steps of permutations. The first step ensures that adjacent coded bits are mapped onto non-adjacent subcarriers, while the second step ensures that adjacent coded bits are mapped alternately onto less or more significant bits of constellation, thus avoiding long runs of lowly reliable bits. The two permutations for index k are as follows:

$$M_k = \left(\frac{N}{d} \right) * (k \% d) + \left\lfloor \frac{k}{d} \right\rfloor \quad (1)$$

$$J_k = s * \left\lfloor \frac{M_k}{s} \right\rfloor + \left(\left(M_k + N - \left\lfloor d * \frac{M_k}{N} \right\rfloor \right) \% s \right) \quad (2)$$

Here N is the block size corresponding to number of coded bits per allocated sub-channels per OFDM. d is taken as no of columns, which has different values for 20MHz and 40 MHz case. The parameter s is defined as $s = \max \{1, N_{BPSCS}/2\}$ where N_{BPSCS} is number of coded bits per sub-carrier, (i.e. 1, 2, 4 or 6 for BPSK, QPSK, 16-QAM or 64-QAM respectively). The operator $\%$ is the modulo function computing the remainder and the operator $\lfloor x \rfloor$ is the floor function i.e. rounding x towards zero.

If more than one spatial streams exists, a frequency rotation is applied to the output of the second, third and fourth permutation, which is given by the following equation. i_{ss} is the index of the spatial stream and N_{ROT} is the parameter used for frequency rotation.

$$R_k = \left\lfloor J_k - \left\{ \left((i_{ss} - 1) * 2 \right) \% 3 + 3 \left\lfloor \frac{i_{ss} - 1}{3} \right\rfloor \right\} * N_{ROT} * N_{BPSCS} \right\} \% N \quad (3)$$

The set of eq. (1) – (3) provide the permuted address for the incoming data stream. A direct implementation of these permutation equations in hardware cannot be hardware efficient due to complexity of functions like floor function and modulo function. Next section provides the realization of basic interleaver functions as a single 2-dimensional function, to reach the hardware efficient solution.

III. 2D REPRESENTATION OF INTERLEAVER EQUATIONS FOR HARDWARE SIMPLICITY

Due to presence of floor function and modulo function in eq. (1) – (3), the algebraic derivation to find new expressions is not possible all the time. That is why we used to re-structure the equations by analysis so that the equations become hardware efficient. The verification of newly structured equations is done using MATLAB by comparing with the results from original equations.

One approach to achieve on the fly address computation for interleavers is through recursions. Eq. (1) and (2) provide the basic interleaver functionality for 802.11n, but they do not appear like a recursive expression. In the following sub-sections we try to represent the interleaver expressions as recursion to reduce the implementation complexity and at the same time avoid the use of hardware intensive components e.g. dividers, multipliers etc.

A. BPSK-QPSK Interleaver

The parameter s is 1 for both BPSK and QPSK, as N_{BPSCS} is 1 and 2 for BPSK and QPSK respectively, thus both have the same basic interleaver function. Eq. (2) simplifies to $J_k = M_k$ for index k , and it can be written as:

$$J_k = \left(\frac{N}{d}\right) * (k \% d) + \left\lfloor \frac{k}{d} \right\rfloor \quad (4)$$

In 802.11n standard, parameter d is defined as the total number of columns N_{COL} and parameter N/d is used as total number of rows N_{ROW} , but we swap the column and row definition hereafter and take parameter d as total number of rows and parameter N/d is taken as total number of columns. The functionality still remains the same, with the benefit that it ends up with the recursive expression for all the modulation schemes. Further it also helps to reach the optimal hardware solution by avoiding use of multiplier for on the fly address computation. According to new definitions, the term $(k \% d)$ provides the behavior of row counter and the term $\lfloor k/d \rfloor$ provides the behavior of column counter. Thus we can introduce here 2 variables i and j as 2 dimensions for which j increments when i expires. The ranges for i and j are mentioned below:

$$i = 0, 1, \dots, (d - 1) \text{ which satisfies against } k \text{ when } i = (k \% d) \quad (5)$$

$$j = 0, 1, \dots, (N/d - 1) \text{ which satisfies against } k \text{ when } j = \left\lfloor \frac{k}{d} \right\rfloor \quad (6)$$

Defining total number of columns as $C = N/d$, and writing in recursive way, Eq. (4) can be written as:

$$J_{i,j} = C * i + j \quad (7)$$

$$J_{(i+1),j} = J_{i,j} + C$$

$$\text{or } J_{i,j} = J_{(i-1),j} + C \quad (8)$$

To avoid the exception i.e. $i - 1 = -1$, for $i = 0$, and at the same time satisfy the algorithm requirement the above eq. can be rewritten as:

$$J_{i,j} = \begin{cases} j & ; \text{if } (i = 0) \\ J_{(i-1),j} + C & ; \text{otherwise} \end{cases} \quad (9)$$

Defining row counter i as $i = R_c$ and column counter j as $j = C_c$, the hardware for eq. (9) is shown in Fig.3a.

The case of BPSK and QPSK is the simplest one due to parameter $s=1$, and it does not carry any specific inter-row or inter-column permutation pattern. Therefore, we end up with relatively simple case; however, it provides the basis for analysis for 16-QAM and 64-QAM which are more complicated.

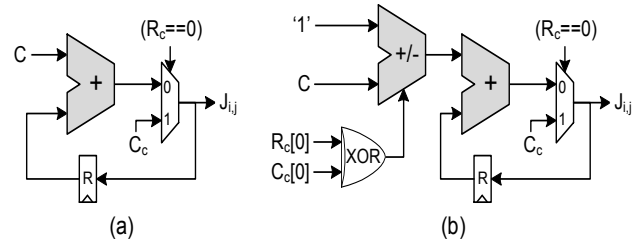


Fig. 3: Interleaver address generation for (a) BPSK-QPSK (b) 16-QAM

B. 16-QAM Interleaver

The parameter s is 2 for 16-QAM, thus eq. (2) becomes:

$$J_k = 2 * \left\lfloor \frac{M_k}{2} \right\rfloor + \left((M_k + N + \left\lfloor \frac{d * M_k}{N} \right\rfloor) \% 2 \right) \quad (10)$$

The term M_k is given by eq. (1). Like BPSK/QPSK case, we cannot proceed by using only algebraic steps due to presence of complex functions in eq. (10). By analyzing all block sizes for 16-QAM case using MATLAB, and testing for different structures, we present the following structure which is equivalent to eq. (10) and also resembles the structure of eq. (4).

$$J_k = \left(\frac{N}{d}\right) * (k \% d) + \left\lfloor \frac{k}{d} \right\rfloor + r_k^2 \quad (11)$$

As the interleaver construction for 16-QAM is not as simple as that of BPSK/QPSK, therefore an extra term r_k^2 appears in the above mentioned equation. This additional term is defined below and it provides the permutation pattern for $s=2$.

$$r_k^2 = [(1 - (k \% 2)) - (k \% 2)] \left\{ 1 - \left\lfloor \frac{k}{d} \right\rfloor \% 2 \right\} + [((k \% 2) - 1) + (k \% 2)] \left\{ \left\lfloor \frac{k}{d} \right\rfloor \% 2 \right\} \quad (12)$$

Considering the 2-dimensions i and j with ranges as mentioned in eq. (5) and (6), the behavior of term $k \% 2$ is same as that of $i \% 2$, when i is the row counter. Eq. (11) can be written in 2D representation as follows:

$$J_{i,j} = \begin{cases} j & ; \text{if } (i = 0) \\ J_{(i-1),j} + C + r_{i,j}^2 & ; \text{otherwise} \end{cases} \quad (13)$$

where

$$r_{i,j}^2 = [(1 - (i \% 2)) - (i \% 2)] \{ 1 - (j \% 2) \} + [(i \% 2) + (1 - (i \% 2))] \{ j \% 2 \} \quad (14)$$

The above expression provides a sequence of (+1,-1) for even columns and (-1,+1) for odd columns. It can further be reduced to a smaller expression but using this form as it is, ends up with efficient hardware realization. The modulo terms can be implemented by using the LSB of row counter R_c and column counter C_c . Generation of (+1,-1) or (-1,+1) sequence is achieved with the help of an XOR gate and an adder as shown in Fig. 3b.

C. 64-QAM Interleaver

The parameter s is 3 for 64-QAM, thus eq. (2) becomes

$$J_k = 3 * \left\lfloor \frac{M_k}{3} \right\rfloor + \left((M_k + N + \left\lfloor \frac{d * M_k}{N} \right\rfloor) \% 3 \right) \quad (15)$$

This term involves modulo function $x \% 3$, which makes it more complicated to reach at some simplified mathematical expression. Again MATLAB appeared to be very useful for trying different structures supporting all block sizes in 64-QAM. We directly present here the new structure, similar to eq. (9) and (13) for 2-dimensions i and j with ranges as mentioned in eq. (5) and (6).

$$J_{i,j} = \begin{cases} j & ; \text{if } (i = 0) \\ J_{(i-1),j} + C + r_{i,j}^3 & ; \text{otherwise} \end{cases} \quad (16)$$

Defining $i' = (i \% 3)$ and $j' = (j \% 3)$, $r_{i,j}^3$ is given by:

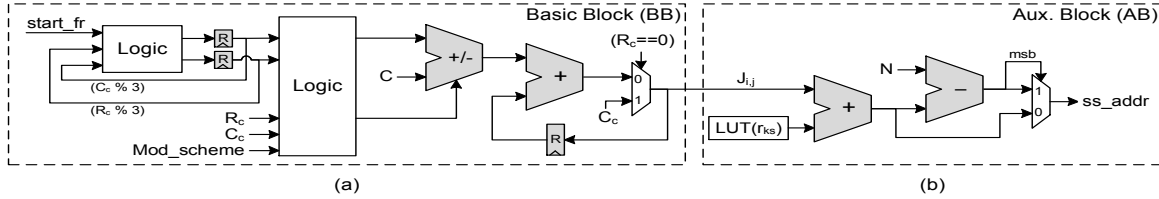


Fig. 4: (a) Combined HW for Interleaver address generation, (b) HW for Frequency Rotation in 802.11n

$$r_{i,j}^3 = \left\{ \left[(1-j') + \frac{j'(j'-1)}{2} \right] \left\{ 2 \left((1-i') + \frac{i'(i'-1)}{2} \right) - \left(i' - \frac{i'(i'-1)}{2} \right) \right\} + \left[j' - j'(j'-1) \right] \left\{ 2(i' - i'(i'-1)) - ((1-i') + i'(i'-1)) \right\} + \left[\frac{j'(j'-1)}{2} \right] \left\{ 2 \left(\frac{i'(i'-1)}{2} \right) - \left(1 - \frac{i'(i'-1)}{2} \right) \right\} \right\} \quad (17)$$

The term $r_{i,j}^3$ provides the inter-row and inter-column permutation for $s=3$ against row counter i and column counter j . The expression for $r_{i,j}^3$ looks very long and complicated but eventually, we get a hardware efficient solution as the terms inside braces are easier to generate through a very small lookup table.

We can describe the recursive expression for interleaver in a generic way for all modulation schemes as mentioned below:

$$J_{i,j} = \begin{cases} j & ; \text{if } (i = 0) \\ J_{(i-1),j} + C + r_{i,j}^s & ; \text{otherwise} \end{cases} \quad (18)$$

Where parameter s distinguishes for different modulation schemes. For BPSK/QPSK $r_{i,j}^1 = 0$, and for 16-QAM and 64-QAM, $r_{i,j}^2$ and $r_{i,j}^3$ is given by eq. (14) and (17) respectively. The hardware realization supporting all modulation schemes is shown in Fig. 4a.

IV. FREQUENCY ROTATION

The frequency rotation applies when there are more than one spatial streams. The expression for frequency rotation is given as eq. (3) where N_{ROT} is the parameter which defines different rotation for 20 MHz and 40 MHz case. The frequency rotation also depends on index of the spatial stream i.e. i_{SS} thus each spatial stream faces different frequency rotation. We define J_{ROT} as the term providing the rotation in eq. (3).

$$J_{ROT} = \left\{ \left(\left((i_{SS} - 1) * 2 \right) \% 3 + 3 \left\lfloor \frac{i_{SS} - 1}{3} \right\rfloor \right) * N_{ROT} * N_{BPPSCS} \right\} \quad (19)$$

$$\text{Therefore} \quad R_k = (J_k - J_{ROT}) \% N \quad (20)$$

The range for the term $(J_k - J_{ROT})$ can go larger than $2N$ thus direct implementation cannot be low cost. However analyzing the two terms $[J_k \% N]$ and $[(-J_{ROT}) \% N]$ separately, the second term provides the starting point for computing the rotation R_k as J_k or $(J_{i,j}) = 0$ for $k=0$ or $(i=0, j=0)$. As the rotation is fixed for a specific spatial stream, thus the starting value $[r_{ks} = (-J_{ROT}) \% N]$ also holds for the later computations. Eq. (20) can be written as:

$$J_{i,j}^{i_{SS}} \equiv R_k = (J_k + r_{ks}) \% N \quad (21)$$

Here $J_{i,j}^{i_{SS}}$ is the final address against row index i , column index j and spatial stream index i_{SS} . The starting values r_{ks} for all the modulation schemes with different spatial stream index i_{SS} for 802.11n standard are provided in Table 1. It is also noted that $(r_{ks} < N)$ for all the cases which depicts that the term $(J_k + r_{ks})$ cannot be larger than $2N$. Therefore using a small look up table which provides the starting value r_{ks} , the rotation can be computed by using two adders with a compare and select logic as shown in Fig. 4b.

V. MULTI-STREAM INTERLEAVER STRUCTURE

We call the basic interleaver address generation block shown in Fig. 4a as Basic Block (BB) and the frequency rotation block as shown in Fig. 4b as Auxiliary Block (AB). Both these blocks

TABLE 1. STARTING VALUE (r_{ks}) FOR FREQUENCY ROTATION FOR DIFFERENT SPATIAL STREAMS IN 802.11n

Modulation Scheme	Stream Index (i_{SS})	(20 MHz) $N_{ROT}=13; C=13$		(40 MHz) $N_{ROT}=29; C=18$	
		N	r_{ks}	N	r_{ks}
BPSK ($N_{BPPSCS}=1$)	0	52	0	108	0
	1	52	30	108	50
	2	52	41	108	79
	3	52	19	108	21
QPSK ($N_{BPPSCS}=2$)	0	104	0	216	0
	1	104	60	216	100
	2	104	82	216	158
	3	104	38	216	42
16-QAM ($N_{BPPSCS}=4$)	0	208	0	432	0
	1	208	120	432	200
	2	208	164	432	316
	3	208	76	432	84
64-QAM ($N_{BPPSCS}=6$)	0	312	0	648	0
	1	312	180	648	300
	2	312	246	648	474
	3	312	114	648	126

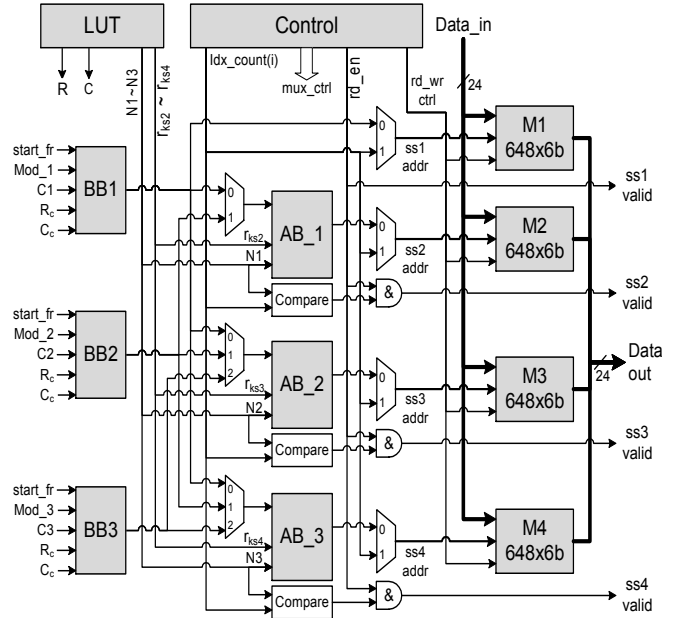


Fig. 5: Complete hardware for quad stream interleaver.

combine to form a complete address generation circuit for one spatial stream. A quad stream implementation may require 4 times the replication of complete address generation circuits shown in Fig. 4, but looking at all the mandatory and optional combinations for different spatial streams, maximum types of modulation schemes needed are 3. Thus the design can be optimized by using 3 basic blocks and 3 auxiliary blocks, still providing address for 4 spatial streams. The complete hardware for quad-stream implementation is shown in Fig. 5.

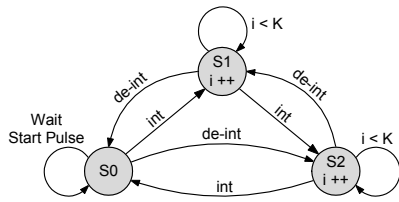


Fig. 6: Control FSM for memory read-write sequence control.

In order to control the sequence of memory read and write for interleaver or de-interleaver a small controller having just three states is used. The state diagram for the control FSM is shown in Fig. 6. This control FSM also provides the frame synchronization with external world to start and finish the frame.

VI. IMPLEMENTATION

Based on the proposed multi stream hardware interleaver for 802.11n shown in Fig. 5, we implemented the hardware in 65nm standard CMOS technology to assess the area consumption and to properly estimate the throughput. Four memory cores with size 648x6bit are used in the design to support the maximum block size of 2592 and soft bit processing in the decoder upto 6 bits. The corresponding implementation results for our design are summarized in Table 2 along with comparison with reference design [12]. The address generation (ADG) core includes the computing core, the control FSM and LUT block. Proposed multi stream (de) interleaver architecture occupies 0.035 mm² including memory and the chip layout is shown in Fig. 7.

The design can run at 225 MHz, and consumes 4mW power in total. Having 4 bit parallel processing for 4 spatial streams, the throughput can reach upto 900 Mbps. Proposed design does not incorporate any specific pipelining, thus performance and hence throughput can also be increased by adding pipeline stages. The throughput can also be increased by just adding the memory in parallel and keeping the same HW for address generation, e.g. to have the throughput of 2.7Gbps we need to just increase the memory by 3 times. However, this will increase the total power consumption of the circuit, thus a tradeoff between power consumption and throughput applies. A direct comparison of silicon cost with reference design cannot be made due to different target technology and different level of parallelism. However, using simple estimate of area cost of our design for a throughput of 2.7 Gbps comes out to be 0.085 mm² which is still 8 times lower than the reference design.

VII. CONCLUSION

This paper presents a low cost solution for interleaver implementation for MIMO-OFDM based Wireless LAN communication like 802.11n. It is flexible enough to support other interleaver implementations like WLAN (802.11a/b/g) and WiMAX (802.16e). Use of recursion for on the fly address computation with minimal hardware usage has provided the basis to achieve low silicon cost solution. Proposed architecture also facilitates to increase the throughput by increasing memory size or adding pipeline stages, which makes it a suitable choice for any high speed MIMO-OFDM based Wireless Communication.

REFERENCES

[1] IEEE P802.11n/D2.0: "Draft Standard for Information Technology-Telecomm. and Info. 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," Feb. 2007.

[2] IEEE 802.11-2007: "Standard for local and metropolitan area networks, Part 11: WLAN Medium Access Control (MAC) and Physical Layer (PHY) Sepcs," Rev. of IEEE Std. 802.11-1999.

[3] IEEE 802.16e-2005: "IEEE Standard for local and metropolitan area networks, Part 16: Air Interface for Fixed Broadband Wireless Access Systems - Amendment 2: Medium Access Control Layers for Combined Fixed and Mobile Operations in Licensed Bands."

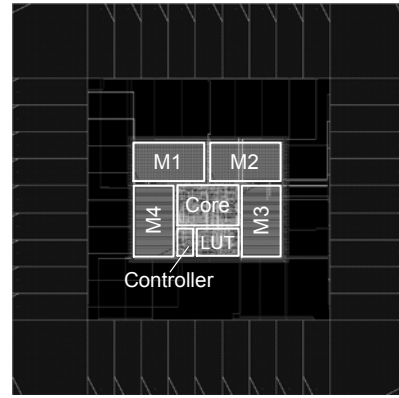


Fig. 7: Layout of interleaver core for 802.11n

TABLE 2. IMPLEMENTATION RESULTS AND COMPARISON

Parameter	Ref. Design [12]	This Design
Target Technology	0.18μm	65 nm
Parallelism	24bit parallel	4 bit parallel
Clock Rate	200 MHz	225 MHz
Throughput	2.4 Gbps	900 Mbps
Pipelined	YES	NO
Memory Configuration	24 b x 6 x 12 36 b x 108 x 8	648 x 6 b x 4
Total Memory	~32.8 Kbit	~15.6 Kbit
Memory Area	564587 μm ²	25136 μm ²
ADG Core / Logic Area	168658 μm ²	9690 μm ²
Total Area	733245 μm ²	34824 μm ²
Power Consumption	---	4 mW

[4] X. F. Wang, Y. R. Shayan and M. Zeng, "On the code and interleaver design of broadband OFDM systems," *Communication Letters, IEEE*, vol. 8, Nov. 2004, pp. 653-655.

[5] R. Van Nee, V. K. Jones, G. Awater, A. Van Zelst, J. Gardner and G. Steele, "The 802.11n MIMO-OFDM standard for Wireless LAN and beyond," *Wireless Personal Communications: An Int. Journal*, v.37, May 2006, pp.445-453.

[6] H. Niu, X. Ouyang, C. Ngo, "Interleaver design for MIMO-OFDM based Wireless LAN," *IEEE Wireless Comm. and Networking Conf.*, vol. 4, Las Vegas, USA, 2006, pp. 1825-1829.

[7] J. Baltersee, G. Fock, H. Meyr, "Achievable rate of MIMO channels with data-aided channel estimation and perfect interleaving," *IEEE Journal on Selected. Areas in Communication*, vol. 19, issue 12, Dec, 2001, pp. 2358-2368.

[8] S. Ramseier, "Shuffling bits in time and frequency: an optimum interleaver for OFDM," in *IEEE International Conference on Communications*, 2003. ICC '03, May 2003, pp. 3418 - 3422.

[9] V. D. Nguyen and H. Kuchenbecker, "Block interleaving for soft decision viterbi decoding in OFDM systems," in *IEEE VTS 54th Vehicular Technology Conference*, 2001. VTC 2001 Fall, 2001, pp. 470-474.

[10] Y. N. Chang and Y. C. Ding : "A low-cost dual mode de-interleaver design," *International Conf. on Consumer Electronics*, 2007.

[11] R. Asghar and D. Liu, "Low complexity multi mode interleaver core for WiMAX with support for convolutional interleaving," *Int. Journal of Electronics, Communications and Computer Engineering*, vol. 3, issue 1, 2009, pp. 20 - 29.

[12] Y. W. Wu and P. Ting : "A high speed interleaver for emerging wireless communications," in *Proc. of 2005 International Conf. on Wireless Networks, Communications and Mobile Computing*, vol. 2, June 2005, pp. 1192 - 1197.