World Academy of Science, Engineering and Technology
International Journal of Electrical and Computer Engineering
Vol:3, No:3, 2009

2-D Realization of WiMAX Channel Interleaver for Efficient Hardware Implementation

Rizwan Asghar and Dake Liu

***Abstract*—** The direct implementation of interleaver functions in WiMAX is not hardware efficient due to presence of complex functions. Also the conventional method i.e. using memories for storing the permutation tables is silicon consuming. This work presents a 2-D transformation for WiMAX channel interleaver functions which reduces the overall hardware complexity to compute the interleaver addresses on the fly. A fully re-configurable architecture for address generation in WiMAX channel interleaver is presented, which consume 1.1 k-gates in total. It can be configured for any block size and any modulation scheme in WiMAX. The presented architecture can run at a frequency of 200 MHz, thus fully supporting high bandwidth requirements for WiMAX.

***Keywords*—**Interleaver, deinterleaver, WiMAX, 802.16e.

I. Introduction

SILICON COST of the permutation tables for interleaver implementation used in the conventional approaches can be very high if the device is supporting many variants inside a particular standard. Low cost on-the-fly address computation with supporting multiple variants has been a challenge due to presence of complex functions. Therefore a low cost and re-configurable architecture for address computation is always beneficial. IEEE 802.16e [1] called WiMAX is being used in the communication industry with many variants in channel coding, like different block sizes and different modulation schemes (e.g. BPSK, QPSK, 16-QAM and 64-QAM). System level overview for WiMAX showing use of channel interleaver is shown in Fig. 1. The type of interleaver used here is the block interleaver, in which the data is written sequentially in a memory and read in a random order after applying certain permutations. The block interleaver can also be considered as a row-column matrix. In this case, data is written row-wise in a memory configured as a row-column matrix and then read column-wise after applying certain intra-row and inter-row permutations. Some work [2] – [4] has been published for the hardware implementation of WiMAX interleaver in different scenarios, but no mathematical formulation has been proposed behind the implementation. This paper emphasizes on reduction in complexity of the address generation by 2-D transformation of the original interleaving functions. Section II of this paper introduces the interleaving functions being used for WiMAX channel interleaver, whereas section III presents the 2-D transformation steps in detail to reach to the optimum hardware. Section IV describes the implementation of complete hardware, followed by a conclusion in section V.

R. Asghar is with Department of Electrical Engineering, Linköping University, SE-58183, Linköping, Sweden (phone: +46 (0)13 28 2313; fax: +46 (0)13139 282 ; e-mail: rizwan@ isy.liu.se).
D. Liu is with Department. of Electrical Engineering, Linköping University, SE-58183, Linköping, Sweden (e-mail: dake@ isy.liu.se).

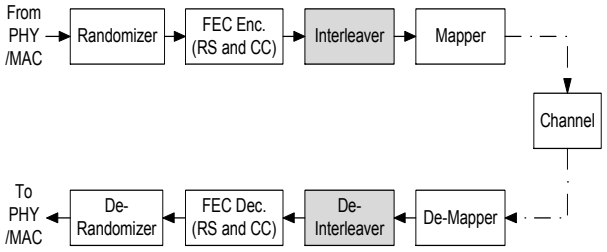

Fig. 1. Overview of encoding in WiMAX channel.

II. WiMAX Channel Interleaver

WiMAX uses Read-Solomon and convolutional encoding followed by an interleaver as shown in Fig. 1 to detect and correct errors to improve the performance of the communication system. Different interleaving patterns apply for different modulation schemes BPSK/QPSK, 16-QAM and 64-QAM. The block interleaver for channel interleaving in WiMAX 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 first permutation m_k for index k is defined by:

$$m_k = \left(\frac{N_{cbps}}{d}\right)\cdot\left(k\,\%\,d\right)+\left\lfloor\frac{k}{d}\right\rfloor \qquad (1)$$

Here N_{cbps} is the block size corresponding to number of coded bits per allocated sub-channels per OFDM and typical value for d used in WiMAX is 12 and 16. The operator % is defined as the modulo function computing the remainder and the operator $\lfloor x \rfloor$ is the floor function i.e. rounding towards zero. The second permutation j_k for index k is given by:

$$j_k = s\cdot\left\lfloor\frac{m_k}{s}\right\rfloor+\left(\left(m_k+N_{cbps}-\left\lfloor d\cdot\frac{m_k}{N_{cbps}}\right\rfloor\right)\%\,s\right) \qquad (2)$$

The parameter s is defined as $s = ceil(N_{cpc}/2)$, where N_{cpc} is number of coded bits per sub-carrier, i.e., 1, 2, 4 or 6 for BPSK, QPSK, 16-QAM or 64-QAM respectively and ceil operation is rounding towards infinity. The de-interleaver, which performs the inverse operation, is also defined by the two permutations. Let n be the index of received bits within the received block of N_{cbps} bits. The first permutation m_n for index n is defined by:

$$m_n = s \cdot \left\lfloor \frac{n}{s} \right\rfloor + \left(\left(n + \left\lfloor d \cdot \frac{n}{N_{cbps}} \right\rfloor \right) \% s \right) \quad (3)$$

The second permutation k_n for index n is given by:

$$k_n = d \cdot m_n - \left(\left(N_{cbps} - 1 \right) \cdot \left\lfloor d \cdot \frac{m_n}{N_{cbps}} \right\rfloor \right) \quad (4)$$

The range of n and k for eq. (1) to (4) is defined as 0, 1, 2, ... $(N_{cpbs} - 1)$. The next section presents the simplification steps needed to reach to hardware efficient architecture.

III. 2-D Transformation of Interleaver Functions

If we try to implement the two steps of permutations by direct computation then they are found to be quite hardware inefficient. This is due to the presence of complex functions like floor function and modulo function.

The alternate is to consider the two steps as one step and find the correlation between input and output which should be hardware efficient. We present here the idea of realizing the one dimensional equations into a joint 2-dimensional expression. It is not necessary to transform both set of equations to 2-D space and implement separately, as they are inverse of each other. Thus only one set of equations can be transformed for efficient hardware implementation and same can be used for other by just swapping the order of read and write of data into memory. The following sub-sections present the transformation steps for all kinds of modulation schemes used in WiMAX.

A. BPSK / QPSK

Due to ceil operation the parameter s is 1 for both BPSK and QPSK. Defining $N = N_{cbps}$ eq. (3) simplifies to $m_n = n + 0 = n$, and therefore eq. (4) becomes:

$$k_n = d \cdot n - \left((N-1) \cdot \left\lfloor \frac{d \cdot n}{N} \right\rfloor \right)$$

$$k_n = d \cdot \left(n - \frac{N}{d} \cdot \left\lfloor \frac{d \cdot n}{N} \right\rfloor \right) + \left\lfloor \frac{d \cdot n}{N} \right\rfloor$$

$$k_n = d \cdot \beta_n + \gamma_n \quad (5)$$

Where β_n and γ_n are defined as:

$$\beta_n = n - \frac{N}{d} \cdot \left\lfloor \frac{d \cdot n}{N} \right\rfloor \text{ and } \gamma_n = \left\lfloor \frac{d \cdot n}{N} \right\rfloor = \left\lfloor \frac{n}{N/d} \right\rfloor$$

Due to the presence of floor function, it is difficult to work out a complete algebraic solution for these equations, however looking at the behavior of different terms and verifying for all possible block sizes, we try to re-structure the equations. MATLAB is used for verification of new structures at all stages. For a simple illustration, an example case of BPSK with 2 sub channels and $d = 16, N = 32$, is taken and behavior of β_n is analyzed against the index n.

$$\beta_n = n - 2 \cdot \left\lfloor \frac{n}{2} \right\rfloor$$

$$n = 0 \rightarrow \beta_n = 0 \rightarrow \beta_n = (0 \% 2)$$

$$n = 1 \rightarrow \beta_n = 1 \rightarrow \beta_n = (1 \% 2)$$

$$n = 3 \rightarrow \beta_n = 0 \rightarrow \beta_n = (2 \% 2)$$

. . . . . .

$$n = n \rightarrow \beta_n = 1 \rightarrow \beta_n = (n \% 2)$$

After checking all cases for BPSK and QPSK (i.e. sub-channels 1,2,4,8,16), β_n can be generalized as:

$$\beta_n = \left(n \% \frac{N}{d} \right)$$

Thus for BPSK or QPSK case, eq. (5) can now be written as :

$$k_n = d \cdot \left(n \% \frac{N}{d} \right) + \left\lfloor \frac{d \cdot n}{N} \right\rfloor \quad (6)$$

Introducing 2 dimensions i and j (i.e. a two dimensional array), for which j increments when i expires, the ranges for i and j can easily be selected as mentioned below:

$$i = 0, 1, \ldots\ldots \left(\frac{N}{d} - 1 \right) \text{ which satisfies against 'n' if } i = \left(n \% \frac{N}{d} \right) \quad (7)$$

$$j = 0, 1, \ldots\ldots (d-1) \text{ with behavior against 'n' } j = \left\lfloor \frac{n}{N/d} \right\rfloor \quad (8)$$

The interleaver can now be realized as a 2D row-column matrix with size $i \times j$. Total number of columns is d, defined by the limit on j and total number of rows is N/d. Eq. (6) can be written in the form:

$$k_n \equiv k_{i,j} = d \cdot i + j \quad (9)$$

Here i and j are row and column counters respectively but at the same time, they also provide the inter-row and inter-column permutations. The case of BPSK and QPSK is the simplest one as it does not carry any specific inter-row or inter-column permutation pattern due to the parameter $s =$

1. That is why we end up with a relatively simple hardware needing just an addition and a multiplication, but it provides us the basis for analysis for 16-QAM and 64-QAM which are more complicated.

B. 16-QAM

The parameter s is 2 for 16-QAM therefore eq. (3) and eq. (4) can be written as:

$$m_n = 2 \cdot \left\lfloor \frac{n}{2} \right\rfloor + \left(\left(n + \left\lfloor \frac{d \cdot n}{N} \right\rfloor \right) \% \; 2 \right) \tag{10}$$

$$k_n = d \cdot \left(m_n - \frac{N}{d} \cdot \left\lfloor \frac{d \cdot m_n}{N} \right\rfloor \right) + \left\lfloor \frac{d \cdot m_n}{N} \right\rfloor \tag{11}$$

Two terms can again be defined as β_n and γ_n .

$$\beta_n = m_n - \frac{N}{d} \cdot \left\lfloor \frac{d \cdot m_n}{N} \right\rfloor \quad \text{and} \quad \gamma_n = \left\lfloor \frac{d.m_n}{N} \right\rfloor \tag{12}$$

Therefore $\qquad k_n = d \cdot \beta_n + \gamma_n$

After verifying for all the range for WiMAX, the parameter γ_n can be written as:

$$\gamma_n = \left\lfloor \frac{d.m_n}{N} \right\rfloor = \left\lfloor \frac{d.n}{N} \right\rfloor = \left\lfloor \frac{n}{N/d} \right\rfloor = j \tag{13}$$

However, it does not mean that m_n is equal to n all the time. This is valid only due to the presence of floor function around it. Using definitions in eq. (10) and eq. (13), β_n can be re-written as:

$$\beta_n = 2 \cdot \left\lfloor \frac{n}{2} \right\rfloor + \left(\left(n + \left\lfloor \frac{d \cdot n}{N} \right\rfloor \right) \% \; 2 \right) - \frac{N}{d} \cdot \left\lfloor \frac{d \cdot n}{N} \right\rfloor \tag{14}$$

Now we try to re-arrange this equation to find some new structure which is similar to eq. (9). For illustration purposes some steps for the 16-QAM example case with $d = 16$ and $N = 64$ are given below:

$$n = 0 \rightarrow \beta_n = 0 \rightarrow \beta_n = 0\left[1 - (0 \;\%\; 2)\right] + \left\{(0+1)\left[1 - (0 \;\%\; 2)\right] + (0-1)(0 \;\%\; 2)\right\}(0 \;\%\; 2)$$

$$n = 1 \rightarrow \beta_n = 1 \rightarrow \beta_n = 1\left[1 - (0 \;\%\; 2)\right] + \left\{(1+1)\left[1 - (1 \;\%\; 2)\right] + (1-1)(1 \;\%\; 2)\right\}(0 \;\%\; 2)$$

$$n = 2 \rightarrow \beta_n = 2 \rightarrow \beta_n = 2\left[1 - (0 \;\%\; 2)\right] + \left\{(2+1)\left[1 - (2 \;\%\; 2)\right] + (2-1)(2 \;\%\; 2)\right\}(0 \;\%\; 2)$$

$$n = 3 \rightarrow \beta_n = 3 \rightarrow \beta_n = 3\left[1 - (0 \;\%\; 2)\right] + \left\{(3+1)\left[1 - (3 \;\%\; 2)\right] + (3-1)(3 \;\%\; 2)\right\}(0 \;\%\; 2)$$

$$n = 4 \rightarrow \beta_n = 1 \rightarrow \beta_n = 0\left[1 - (1 \;\%\; 2)\right] + \left\{(0+1)\left[1 - (0 \;\%\; 2)\right] + (0-1)(0 \;\%\; 2)\right\}(1 \;\%\; 2)$$

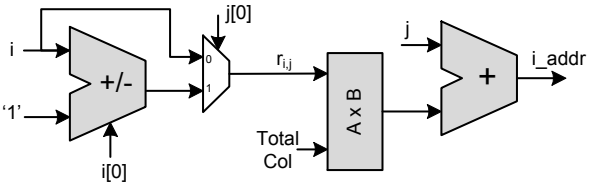

Fig. 2. HW realization for 16-QAM interleaving in WiMAX.

$$n = 5 \rightarrow \beta_n = 1 \rightarrow \beta_n = 1\left[1 - (1 \;\%\; 2)\right] + \left\{(1+1)\left[1 - (1 \;\%\; 2)\right] + (1-1)(1 \;\%\; 2)\right\}(1 \;\%\; 2)$$

. . . . . .

. . . . . .

$$\beta_n \equiv r_{i,j} = \left[1 - (j \;\%\; 2)\right] i + \left[(j \;\%\; 2)\right] \times \left\{(i+1)\left[1 - (i \;\%\; 2)\right] + (i-1)(i \;\%\; 2)\right\} \tag{15}$$

Where i and j are defined with ranges as mentioned in eq. (7) and eq. (8) i.e.

$$i = \left(n \;\%\; \frac{N}{d} \right) \quad and \quad j = \left\lfloor \frac{n}{N/d} \right\rfloor$$

Verifying for all the cases in 16-QAM, we reach to a new structure for β_n as given in eq. (15). This structure is not as simple as that of BPSK/QPSK case. The reason is the presence of permutation pattern in 16-QAM case. Considering the 2 dimensions i and j, the 2D transformation of interleaver for 16-QAM can be described as:

$$k_n \equiv k_{i,j} = d \cdot r_{i,j} + j \tag{16}$$

The parameter $r_{i,j}$ provides an intra-row permutation pattern sequence for selective columns, such that a permutation is applied for all alternate columns $(2y+1)^{th}$ and no permutation is applied for each $2y^{th}$ columns, where $y = 1,2, \ldots \ldots \ldots \ldots d/2$. Considering total number of rows as R, the required inter-row permutation for row number i $(0,1,2 \ldots \ldots \ldots R-1)$ is $i+1$ and $i-1$ for each $2i^{th}$ and $(2i+1)^{th}$ row respectively. Looking at eq. (16), the generic structure for 16-QAM is same as that of eq. (9) except the additional complexity for selective row permutation. The structure of eq. (16) is easy to implement with a row and column counter i and j. The terms with modulo function can be controlled by just the LSB of corresponding variable and the rest can be managed by a lookup table (LUT) or an adder. As number of rows in the block can be many (upto 96 for WiMAX) thus use of LUT is not efficient here. Instead we can use a 7 bit adder, which can also give the benefit of generalizing the implementation. The hardware realization for interleaver address generation for 16-QAM in WiMAX is shown in Fig. 2.

C. 64-QAM

As number of coded bits per sub-carrier are 6 for 64-QAM transmission, thus using the parameter $s = 3$, eq. (3) is written as:

0	17	2	....	29	14	31
16	1	18	....	13	30	15
32	49	34	....	61	46	63
48	33	50	....	45	62	47

(a)

0	17	34	....	29	46	15
16	33	2	....	45	14	31
32	1	18	....	13	30	47
48	65	82	....	77	94	63
64	81	50	....	93	62	79
80	49	66	....	61	78	95

(b)

Fig. 3. Examples of data interleaving for (a) 16-QAM, N=64; (b) 64-QAM, N=96.

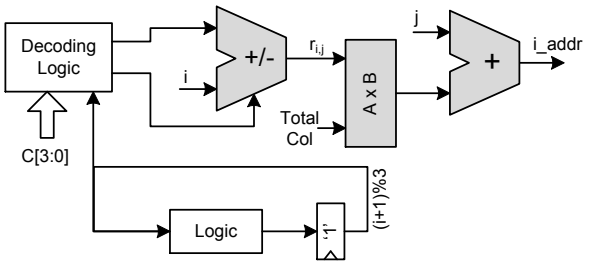

Fig. 4. HW realization for 64-QAM interleaving in WiMAX.

$$m_n = 3 \cdot \left\lfloor \frac{n}{3} \right\rfloor + \left(\left(n + \left\lfloor \frac{d \cdot n}{N} \right\rfloor \right) \% \ 3 \right) \tag{17}$$

Defining the two terms β_n and γ_n as given in eq. (12) and eq. (13) we can write expression for β_n for 64-QAM as:

$$\beta_n = 3 \cdot \left\lfloor \frac{n}{3} \right\rfloor + \left(\left(n + \left\lfloor \frac{d \cdot n}{N} \right\rfloor \right) \% \ 3 \right) - \frac{N}{d} \cdot \left\lfloor \frac{d \cdot n}{N} \right\rfloor \tag{18}$$

Again applying the same re-structuring exercise as we did for 16-QAM case, we reach to an even more complicated 2D-structure for β_n. Due to increased complexity for permutation patterns for 64-QAM the intermediate steps carry much longer terms, thus we directly present the final structure for β_n.

$$\begin{aligned} \beta_n \Rightarrow r_{i,j} = & \left[(1 - j') + \frac{j'(j'-1)}{2} \right] \cdot i \\ & + \left\{ [j' - (j'-1)] \left\{ \begin{array}{l} (i-2) \left[(1-i') + \dfrac{i'(i'-1)}{2} \right] \\ + (i+1) \left[i' - \dfrac{i'(i'-1)}{2} \right] \end{array} \right\} \right\} \\ & + \left\{ \frac{j'(j'-1)}{2} \left\{ \begin{array}{l} (i+2)[i' - i'(i'-1)] \\ + (i-1)[(1-i') + i'(i'-1)] \end{array} \right\} \right\} \end{aligned} \tag{19}$$

Here i and j are row and column count respectively, with the ranges mentioned in eq. (7) and eq.(8). The new parameters i' and j' are defined as below:

$$i' = (i+1) \ \% \ 3 \quad \text{and} \quad j' = j \ \% \ 3$$

The term β_n for 64-QAM provides the selective inter-row permutation for every $(3j+1)^{th}$ and $(3j+2)^{th}$ column.

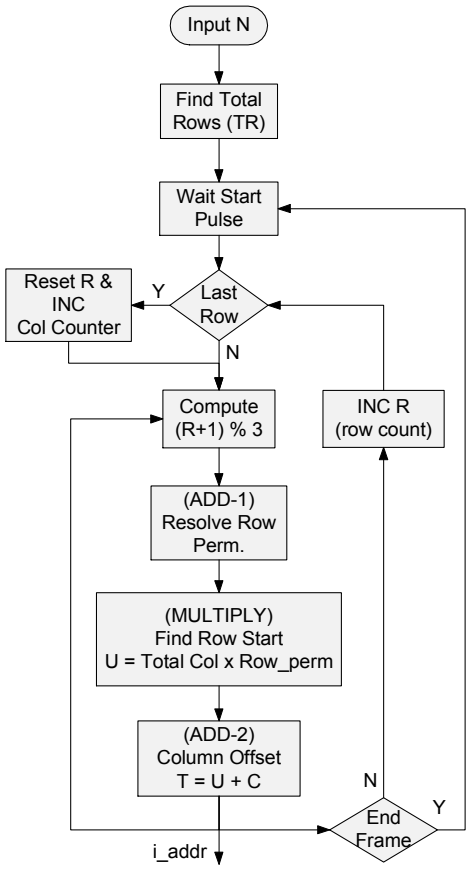

Fig. 5. Flow Graph for Channel Interleaving in WiMAX.

The permutation for all these columns is within 3 rows and afterwards it is repeated. Considering total number of rows as R, the inter-row permutation in $(3j+1)^{th}$ columns for row number i $(0,1,2 \ldots \ldots \ldots R-1)$ is $i+1$, $i+1$, $i-2$ for $3i^{th}$, $(3i+1)^{th}$ and $(3i+2)^{th}$ row respectively. The inter-row permutation for $(3j+2)^{th}$ columns is $i+2$, $i-1$ and $i-1$ for $3i^{th}$, $(3i+1)^{th}$ and $(3i+2)^{th}$ row respectively. Examples of address permutations for 16-QAM and 64-QAM with small block sizes are shown in Fig. 3, which also correspond to the permutation patterns described here.

Although eq. (19) looks very long and complicated, but eventually, we get a hardware efficient solution. Additionally we stick to the generic interleaver hardware for all types of modulation schemes. The implementation of modulo terms $j\%3$ and $(i+1)\%3$ and some other terms inside braces are easier to generate through a very small lookup table. Eq. (16) holds for 64-QAM case as well with $r_{i,j}$ given by eq. (19). The hardware realization for 64-QAM interleaver is shown in Fig. 4.

IV. Implementation

The implementation of interleaver hardware for 64-QAM can covers most of the computational requirements for other modulation schemes with small glue logic. This is due the fact that the final implementation equations for all the cases have the same structure. Combining the interleaver structure

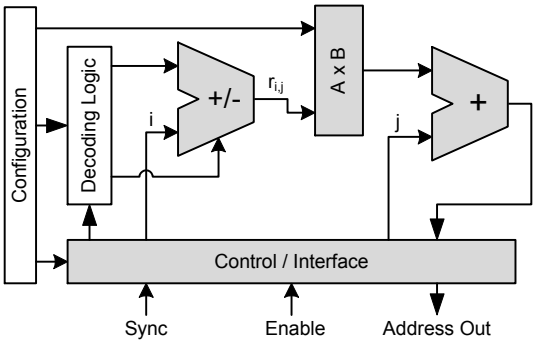

Fig. 6. Complete HW for Address Generation.

for all the cases, the 2D single step generic interleaver function $k_{i,j}$ can be described as:

$$k_{i,j} = d \cdot r_{i,j} + j \quad (20)$$

Where $r_{i,j} = i$ for BPSK/QPSK and it is defined by eq. (15) and eq. (19) for 16-QAM and 64-QAM respectively. Other permutation values with addition and subtraction can be implemented with the help of a multiplexer and an adder. In addition to hardware required for address generation a small control logic is also needed to take care of row-column counter and synchronization with external world. The flow graph for address computation including some control steps is shown in Fig. 5 and the complete hardware for fully re-configurable architecture is shown in Fig. 6. The use of multiplier can be avoided but it is kept there for the generality of design. By using the multiplier, the proposed architecture provides maximum flexibility to be used for any number of columns in the block interleaver.

The validation of address generation is verified and the design is synthesized using 0.12 µm standard CMOS technology. It consumes 1.1 k-gates in total and can run at a frequency of 200 MHz. Direct comparison of this architecture with reported designs [2] – [4] cannot be made due to different configurations used, however, the reduction in complexity of the interleaver address generation with the help of 2-D transformation provides the basis to compute the address on-the-fly. Along with reduced complexity the presented architecture also supports high throughput requirements associated with WiMAX standard.

V. Conclusion

This work provides the mathematical transformation of the one dimensional WiMAX interleaver equations mentioned in the standard to two dimensional space. This 2D transformation leads to optimized hardware architecture for address generation of the WiMAX interleaver. Due to presence of modulus and floor operators within the interleaver functions use of standard algebraic rules does not work always. Thus the structural analysis along with progressive generation of the equivalent set of equations for 2D space is used to reach to the low cost solution.

References

[1] 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."

[2] Y. N. Chang and Y. C. Ding: "A Low-Cost Dual Mode De-interleaver Design," Int. conf. on Consumer Electronics, 2007.

[3] Y. N. Chang: "A Low-Cost Dual Mode De-interleaver Design," IEEE Transaction on Consumer Electronics, vol. 54, no. 2, May 2008, pp. 326 – 332.

[4] Y. W. Wu and P. Ting: "A High Speed Interleaver for Emerging Wireless Communications," Proc. of International Conf. on Wireless Networks, Communications and Mobile Computing, vol. 2, June 2005, pp. 1192 – 1197.