

DESIGN AND IMPLEMENTATION OF ADDRESS GENERATOR FOR WiMAX DEINTERLEAVER ON FPGA

Mallikarjun Naykodi¹, G.S. Biradar²

¹Dept. of VLSI Design & Embedded System, VTURC Gulbarga, Karnataka, India

²Dept. Of Electronics and Communication, PDA College Of Engineering, Gulbarga, Karnataka, India

Abstract

This Paper enumerates efficient method of address generator for WiMAX Deinterleaver using verilog coding. It is a low-complex, high speed and resource efficient method because it eliminates the requirement of floor function. The use of an internal multiplier of FPGA and the sharing of resources for quadrature phase-shift keying (QPSK), 16-quadrature-amplitude modulation (QAM), and 64-QAM modulations along with all possible code rates makes our approach to be novel and highly efficient when compared with conventional look-up table-based approach. The proposed approach exhibits significant improvement in the use of FPGA resources.

Keywords: Digital circuits, error correction, field programmable gate arrays (FPGAs), wireless systems

1. INTRODUCTION

BROADBAND wireless access (BWA) is continuously becoming a more challenging competitor to the conventional wired last mile access technologies [1]. IEEE has developed standards for mobile BWA (IEEE 802.16e) popularly referred to as mobile WiMAX [2]. The channel interleaver employed in the WiMAX transceiver plays a vital role in minimizing the effect of burst error. In this brief, a novel, low-complexity, high-speed, and resource-efficient address generator for the channel deinterleaver used in the WiMAX transceiver eliminating the requirement of floor function is proposed [2].

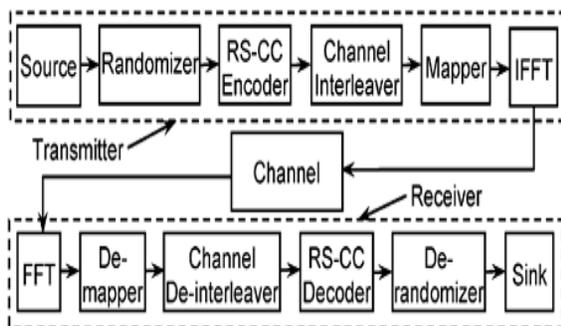


Fig:1 Block diagram of the WiMAX transceiver.

The mandatory blocks of a WiMAX transceiver are shown in Fig. 1. Data stream received from a source is randomized before being encoded by two forward error correction (FEC) coding techniques, namely, Reed-Solomon (RS) and convolutional coding (CC). The channel interleaver permutes the encoded bit stream to reduce the effect of burst error. When convolutional turbo code (CTC) is used for FEC, being optional in WiMAX, the channel interleaver is not required,

since CTC itself includes an interleaver within it[5]. Modulation and construction of orthogonal frequency-division multiplexing symbols are performed by the two subsequent blocks, namely, mapper and inverse fast Fourier transform of Fig. 1. In the receiver, the blocks are organized in the reverse order enabling the restoration of the original data sequence at the output.

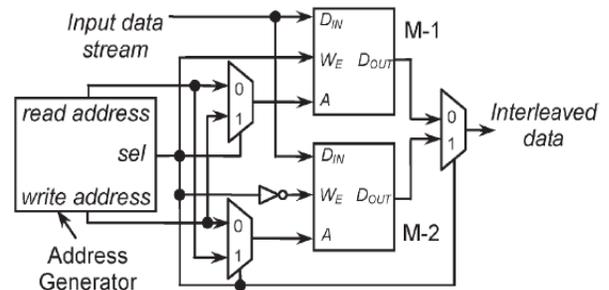


Fig:2 Block diagram of interleaver/deinterleaver structure.

Two-dimensional block interleaver/deinterleaver structure, which is used as a channel interleaver/deinterleaver in the WiMAX system, is described in Fig. 2. Hardware Structure for Address Generator for WiMAX Using Different modulation techniques

2. INTERLEAVING IN WiMAX SYSTEM

The block interleaver/deinterleaver exploits different depths/ncbps to incorporate various code rates and modulation schemes (see Table I) for IEEE 802.16e. The data stream received from the RS-CC encoder is permuted by using the two-step processes described by (1) and (2). These steps ensure mapping of coded bits onto nonadjacent subcarriers

and alternate less/more significant bits of the modulation constellation, respectively. Thus,

$$m_k = (N_{cbps}/d) \cdot (k \% d) + [k/d] \quad (1)$$

$$j_k = s \cdot [m_k/s] + (m_k + N_{cbps} - [d \cdot m_k / N_{cbps}]) \% s \quad (2)$$

Here, the number of columns is represented by d ($= 16/12$ for WiMAX); m_k and j_k are the outputs after the first and second steps, respectively; and k varies from 0 to $N_{cbps} - 1$. s is a parameter defined as $s = N_{cpc}/2$, where N_{cpc} is the number of coded bits per subcarrier, i.e., 2, 4, or 6 for QPSK, 16-QAM, or 64-QAM, respectively [9]. Modulo and floor functions are signified by percent and $[]$, respectively. The deinterleaver, which performs the inverse operation, is also defined by two permutations, i.e., (3) and (4). Let m_j and k_j define the first and second level of permutations for the deinterleaver, where j is the index of received bits within a block of N_{cbps} bits. As per [10], (3) and (4) perform inverse operation of (2) and (1), respectively. Thus

$$m_j = s \cdot [j/s] + (j + [d \cdot j / N_{cbps}]) \% s \quad (3)$$

$$k_j = d \cdot m_j - (N_{cbps} - 1) \cdot [d \cdot m_j / N_{cbps}] \quad (4)$$

3. ALGORITHM FOR DEINTERLEAVER

Here, the proposed algorithm for address generator of the WiMAX deinterleaver along with its mathematical background has been described. A MATLAB program is developed using (3) and (4) for all modulation schemes and code rates. Due to the presence of a floor function in (3) and (4), their direct implementation on an FPGA chip is not feasible. Table shows the deinterleaver addresses for the first four rows and five columns of each modulation type. As $d = 16$ is chosen, the number of rows are fixed ($= d$) for all N_{cbps} , whereas the number of columns are given by N_{cbps}/d .

Table-1: Address generation of different schemes

DETERMINATION OF CORRELATION BETWEEN ADDRESSES

Row no. (j)	Column no. (i) →	0	1	2	3	4
0	$N_{cbps} = 96$ -bits, $\frac{1}{2}$ code rate, QPSK	$d \cdot 0 + 0 = 0$	$d \cdot 1 + 0 = 16$	$d \cdot 2 + 0 = 32$	$d \cdot 3 + 0 = 48$	$d \cdot 4 + 0 = 64$
1		$d \cdot 0 + 1 = 1$	$d \cdot 1 + 1 = 17$	$d \cdot 2 + 1 = 33$	$d \cdot 3 + 1 = 49$	$d \cdot 4 + 1 = 65$
2		$d \cdot 0 + 2 = 2$	$d \cdot 1 + 2 = 18$	$d \cdot 2 + 2 = 34$	$d \cdot 3 + 2 = 50$	$d \cdot 4 + 2 = 66$
3		$d \cdot 0 + 3 = 3$	$d \cdot 1 + 3 = 19$	$d \cdot 2 + 3 = 35$	$d \cdot 3 + 3 = 51$	$d \cdot 4 + 3 = 67$
0	$N_{cbps} = 192$ -bits, $\frac{1}{2}$ code rate, 16-QAM	$d \cdot 0 + 0 = 0$	$d \cdot 1 + 0 = 16$	$d \cdot 2 + 0 = 32$	$d \cdot 3 + 0 = 48$	$d \cdot 4 + 0 = 64$
1		$d \cdot 1 + 1 = 17$	$d \cdot 0 + 1 = 1$	$d \cdot 3 + 1 = 49$	$d \cdot 2 + 1 = 33$	$d \cdot 5 + 1 = 81$
2		$d \cdot 0 + 2 = 2$	$d \cdot 1 + 2 = 18$	$d \cdot 2 + 2 = 34$	$d \cdot 3 + 2 = 50$	$d \cdot 4 + 2 = 66$
3		$d \cdot 1 + 3 = 19$	$d \cdot 0 + 3 = 3$	$d \cdot 3 + 3 = 51$	$d \cdot 2 + 3 = 35$	$d \cdot 5 + 3 = 83$
0	$N_{cbps} = 576$ -bits, $\frac{3}{4}$ code rate, 64-QAM	$d \cdot 0 + 0 = 0$	$d \cdot 1 + 0 = 16$	$d \cdot 2 + 0 = 32$	$d \cdot 3 + 0 = 48$	$d \cdot 4 + 0 = 64$
1		$d \cdot 1 + 1 = 17$	$d \cdot 2 + 1 = 33$	$d \cdot 0 + 1 = 1$	$d \cdot 4 + 1 = 65$	$d \cdot 5 + 1 = 81$
2		$d \cdot 2 + 2 = 34$	$d \cdot 0 + 2 = 2$	$d \cdot 1 + 2 = 18$	$d \cdot 5 + 2 = 82$	$d \cdot 3 + 2 = 50$
3		$d \cdot 0 + 3 = 3$	$d \cdot 1 + 3 = 19$	$d \cdot 2 + 3 = 35$	$d \cdot 3 + 3 = 51$	$d \cdot 4 + 3 = 67$

A close examination of the addresses in Table reveals that the correlation between them. The mathematical foundation of the correlation between the addresses, as derived in this brief, is represented by (5)–(7), i.e.,

$$k_{n,QPSK} = \{ d * i + j \text{ for } \forall j \text{ and } \forall i \} \quad (5)$$

$$k_{n,16-QAM} = \begin{cases} d * i + j & \text{for } j \% 2 = 0 \text{ and for } \forall i \\ d * (i + 1) + j & \text{for } j \% 2 = 1 \text{ and} \\ & \text{for } i \% 2 = 0 \\ d * (i - 1) + j & \text{for } j \% 2 = 1 \text{ and} \\ & \text{for } i \% 2 = 1 \end{cases} \quad (6)$$

$$k_{n,64-QAM} = \begin{cases} d * i + j & \text{for } j \% 3 = 0 \text{ and for } \forall i \\ d * (i - 2) + j & \text{for } j \% 3 = 1 \text{ and} \\ & \text{for } i \% 3 = 2 \\ d * (i + 1) + j & \text{for } j \% 3 = 1 \text{ and} \\ & \text{for } i \% 3 \neq 2 \\ d * (i + 2) + j & \text{for } j \% 3 = 2 \text{ and} \\ & \text{for } i \% 3 = 0 \\ d * (i - 1) + j & \text{for } j \% 3 = 2 \text{ and} \\ & \text{for } i \% 3 \neq 0 \end{cases} \quad (7)$$

Where $j = 0, 1, \dots, d - 1$ and $i = 0, 1, \dots, (N_{cbps}/d) - 1$ represent the row and column numbers, respectively, in Table-1[5]. In addition, k_n represents the deinterleaver addresses. General validity of (5)–(7) to represent the correlation between addresses of Table has formally been proven using the algebraic analysis, which lacks the involvement of (5)–(7). The outcome of this analysis using (5)–(7) provides the same result, as shown in Table. Thus, (5)–(7) play the pivotal role in establishing formal mathematical foundation of our proposed algorithm.

4. PROPOSED MODEL

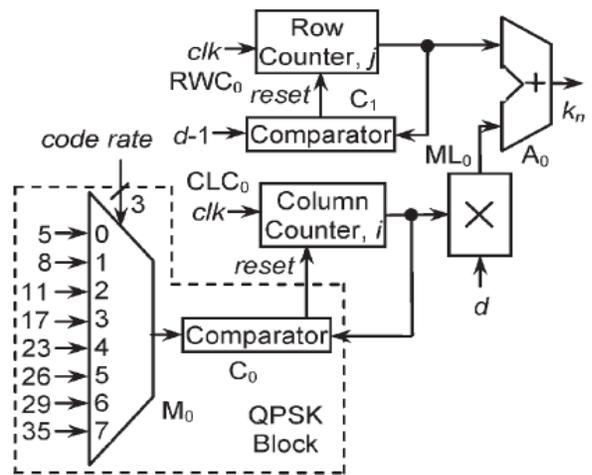


Fig:3a The address generator for QPSK.

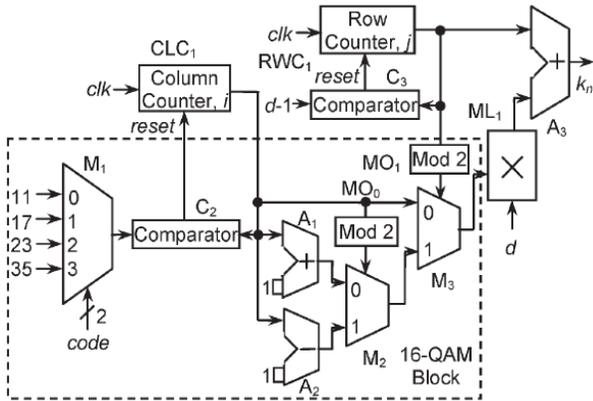


Fig.3b The address generator for 16-QAM.

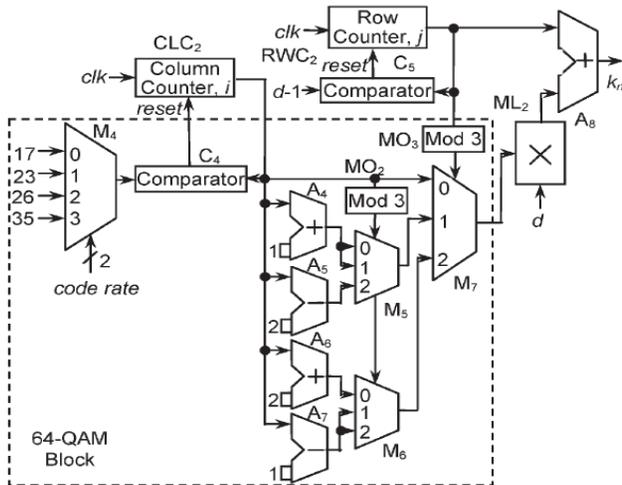


Fig.3c The address generator for 64-QAM.

In order to test the proposed algorithms for the address generator of the WiMAX deinterleaver with all modulation schemes, transformation of these algorithms into digital circuits are made and are shown in Fig. 3(a)–(c). The QPSK hardware shown in Fig. 3(a) has a row counter RWC0 to generate row numbers between 0 and $d - 1$. A column counter CLC0 with multiplexer M0 and comparator C0 generate the variable column numbers to implement permissible N_{cbps} . A multiplier ML0 and an adder A0 perform the desired operations to implement (5). The address generator for 16-QAM follows a similar structure, such as that of QPSK with few additional modules. These modules are designed with an incrementer, a decremter, two modulo-2 blocks, and two multiplexers, as shown in Fig. 3(b). As per in the 64-QAM modulation scheme, the address generator has to implement three different progressive patterns for the column numbers. The design procedure used in 16-QAM is extended in 64-QAM to meet this requirement with the use of additional hardware[6] and is shown in Fig. 3(c). A simple up counter generates the read addresses for the 2-D deinterleaver.

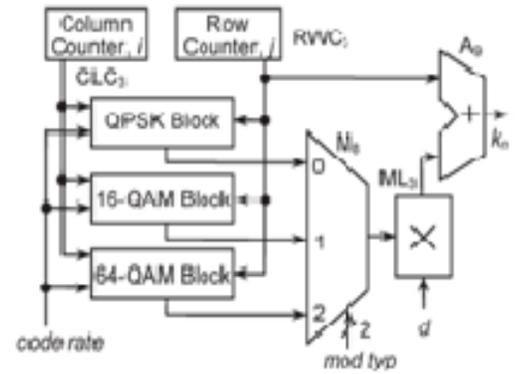


Fig.4: Top-level view of complete deinterleaver address generator.

The top-level structure of the deinterleaver address generator is shown in Fig. 4. Logic circuits shown inside the dashed line in Fig. 3(a)–(c) are presented here as QPSK block, 16-QAM block, and 64-QAM block, respectively. Our design is optimized in the sense that common logic circuits such as multiplier, adder, row counter, and column counter are shared while generating addresses for any modulation type. In addition, the design also shares the incrementer and the decremter required in 16-QAM and 64-QAM blocks.

5.FPGA IMPLEMENTATION RESULT

The Verilog program developed for the proposed WiMAX deinterleaver address generator is downloaded on the XilinxSpartan-3 (Device XC3S400) FPGA. The two blocks, namely, MO0 and MO1of Fig. 3(b), are implemented using the mod $2n$ function of Verilog. The requirement of imod 3 (MO2) and $j \text{ mod } 3$ (MO3)functions in 64-QAM circuit are fulfilled by designing two small ROMs of dimension 16×3 -bit and 64×3 -bit, respectively, as the MOD 3 function is not supported in Verilog. The use of the rest of the logic circuits is obvious in the design. As the FPGA-based implementation of the WiMAX deinterleaver address generator has not been found in the literature, direct comparison of the results of our proposed work could not be carried out. However, implementation of the conventional LUT-based technique of address generation for the WiMAX2-D deinterleaver on the same FPGA. In the latter case, the LUTs are modeled using FPGA’s embedded memory, i.e., block RAM, to reduce the memory access time. For fairness of comparison, three block RAMs are used, i.e., one for each modulation scheme to house the address LUT of various interleaver depths[7].Efficient use of block RAMs is made possible by exploring the feature that, within a modulation scheme, the address LUT of a smaller N_{cbps} is the subset of the address LUT of the larger N_{cbps} . In spite of the smart use of block RAM in LUT-based approach, this brief results in a significant reduction in occupancy of FPGA slices (by 80.24%), flip flops (by 35.9%), and four input LUTs (by 80.47%). This comparison clearly proves the low

complexity and hardware efficiency of our design over the conventional technique. Furthermore, to make the design more hardware efficient, embedded multiplier of the Xilinx Spartan-3 FPGA is used to implement the ML3 block of Fig. 4. In addition, the address generator using the proposed technique can work 48.69% faster than the latter. Furthermore, based on the equivalence drawn between FPGA and application specified integrated circuit implementations, our work is again compared with the FPGA-equivalent implementation. This comparison also shows almost 48.69% improvement in

our work with respect to operating frequency over as FPGA-equivalent maximum frequency of the latter is found to be 62.5 MHz. The reasons behind these improvements are low-complexity, optimized, and shared hardware design and the use of FPGA's embedded multiplier, which, in turn, reduces interconnection delay inside FPGA. Both CTC and long-term evolution interleavers do not use floor function for FPGA implementation, whereas this brief involves the use of floor function for such realization in an efficient manner.

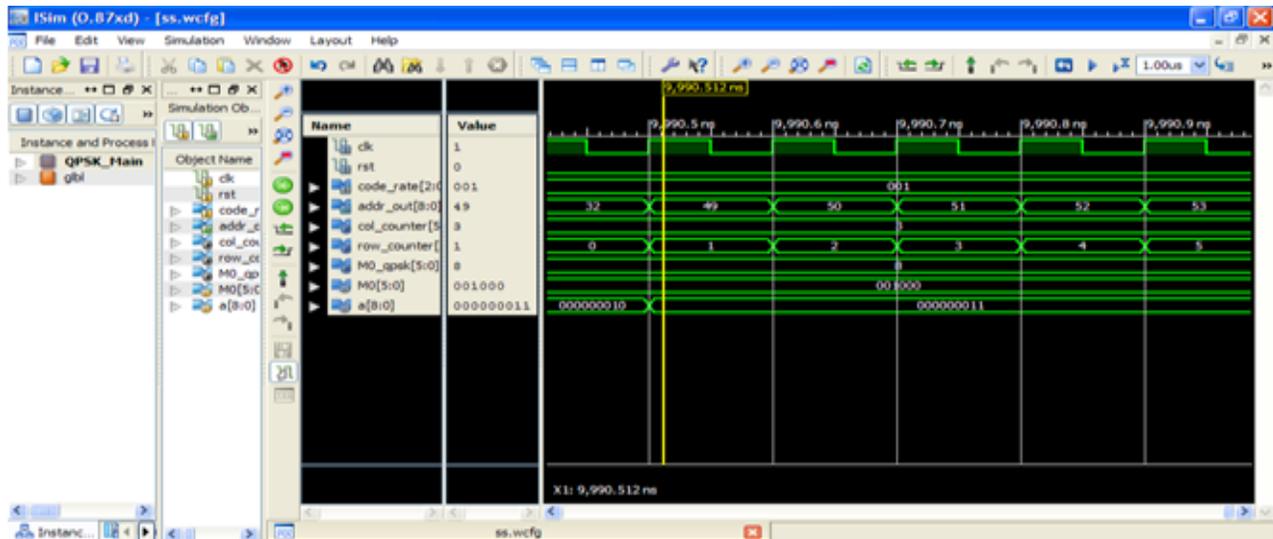


Fig 5 Simulation result

6. CONCLUSIONS

In this novel algorithm along with its mathematical formulation, including proof for address generation circuitry of the WiMAX channel deinterleaver supporting all possible code rates and modulation patterns as per IEEE802.16e. The proposed algorithm is converted into an optimized digital hardware circuit. The hardware is implemented on the Xilinx FPGA using Verilog. Comparison of our proposed work with a conventional LUT-based method and also with a recent work show significant improvement on resource utilization and operating frequency.

REFERENCES

- [1]. W.Konhauser, "Broadband wireless access solutions-Progressive challenges and potential value of next Generation", *Wirelesspers commun.*, vol.37,no.3/4pp.243-259,May 2006.
- [2]. B. Li, Y. Qin, C. P. Low, and C. L. Gwee, "A survey on mobile WiMAX", *IEEE Commun. Mag.*, vol. 45, no. 12, pp. 70–75, Dec. 2007.
- [3]. Y. N. Chang and Y. C. Ding, "A low-cost dual mode in de-interleaver design," *Proc Int. Conf. Consum. Electron.*, 2007, pp. 1–2.

- [4]. A. A. Khater, M. M. Khairy, and S. E.-D. Habib, "Efficient FPGA implementation for the IEEE 802.16e interleaver," in *Proc. Int. Conf. Microelectron.*, Marrakech, Morocco, 2009, pp. 181–184.
- [5]. B.K.Upadhyaya, I.S.Misra, and S.K.Sanyal, "Novel design of address generator for WiMAX multimode interleaver, using FPGA based finite state machine," in *Proc.13th Int.Conf.Comput. Inf. Technol.*, Dhaka, Bangladesh, 2010, pp.153–158.
- [6]. R.Asghar and D.Liu, "2 Drealization of WiMAX channel interleaver for efficient hardware implementation," in *Proc.WorldAcad.Sci.Eng.Field Programm. Gate Arrays*, Monterey, CA, USA, 2006, pp. 21–30.
- [7]. *XilinxSpartan3FPGAFamily:CompleteDataSheet*, Xilinx, Inc., San Jose, CA, USA, 2012.
- [8]. B.K.Upadhyaya and S.K.Sanyal, "An improved LUT based reconfigurable multimode interleaver for WLAN application," *Int.J.Recent Trends Eng.Tech.*, ACEEE, vol.6,no.2,pp.183–188,2011.
- [9]. I.Kuon and J.Rose, "Measuring the gap between FPGAs," in *Proc.Int.Symp. Field Programm. GateArrays*, Monterey, CA, US A, 2006, pp.21–30.