ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

Volume 3, Special Issue 3, March 2014

2014 International Conference on Innovations in Engineering and Technology (ICIET'14)
On 21st & 22nd March Organized by

K.L.N. College of Engineering, Madurai, Tamil Nadu, India

FPGA Implementation Of Reconfigurable Address Generator Of Deinterleaver For WiMAX Applications

S.Muruganantham1,K.Kalyani2,S.Rajaram3

Department of ECE, Thiagarajar college of engineering, Madurai, Tamilnadu, India.

Department of ECE, Thiagarajar college of engineering, Madurai, Tamilnadu, India.

Department of ECE, Thiagarajar college of engineering, Madurai, Tamilnadu, India.

ABSTRACT-WiMAX (Worldwide interoperability for Microwave Access) is a technology used to provide wireless internet connection over 30 miles. In order to increase the security strength in wireless internet, interleaving and deinterleaving is performed in transmitter and receiver receptively. The block deinterleaver is one of the deinterleaving methods which takes less computation time and consumes less power. But the floor function presents in the address generator increase the hardware complexity of deinterleaver. In addition conventional LUT-based technique is found to be unattractive because it increases the computation time and inefficient in terms of resource utilization. Besides different modulation schemes employ different interleaving patterns. Thus a mathematical algorithm for address generator which eliminates the floor function of deinterleaver is developed and the same address generator is made reconfigurable to support three modulation techniques such as 16-QAM, 64-QAM and QPSK for WiMAX applications.

KEYWORDS— Worldwide interoperability for Microwave Access (WiMAX), QPSK, QAM, Interleaver and Deinterleaving.

I.INTRODUCTION

The world is passionately accepting the new wireless communication methods and services. The global demand for multimedia data services has grown at a remarkable pace which has led to the expansion of system capacity in terms of the number of subscribers supported, higher data rate and unique coverage everywhere with high mobility. **B**ROADBAND 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 transreceiver plays a vital role in minimizing the effect of burst error. In this brief, a unique, less latency with high performance and resource-efficient address generator for the channel deinterleaver used in the WiMAX transreceiver eliminating the requirement of floor function is proposed. The work in [3] demonstrates the grouping of incoming data streams into the block to reduce the frequency of memory access in a deinterleaver using a conventional look-up table

(LUT)- based CMOS address generator for WiMAX. Khater *et al.* [4] has described the process of implementing address generator using VHDL for IEEE 802.16e channel interleaver with only a 1/2 code rate. In [5], the authors have described a finite-state machine (FSM)-based address generator of the same interleaver for all permissible code rates and modulation schemes. Both [4] and [5] are tested on the field-programmable gate array (FPGA) platform.

Asghar and Liu in [6] have made 2-D translation of the functions used in WiMAX channel deinterleaver to claim efficient hardware architecture. However, the derivations in [6] do not clearly explain the difficulties in design, especially for 64-quadrature-amplitude modulation (QAM). Hardware implementation of floor function is very complex and consumes abnormally large amount of resources [6]. WiMAX can be classified into Fixed WiMAX [10] and Mobile WiMAX. Fixed WiMAX is based upon Line Of Sight (LOS) condition in the frequency range of 10-66GHz whereas Mobile WiMAX is based upon Non-Line of Sight (NLOS) condition that works in 2-11 GHz frequency range, PHY layer for mobile WiMAX which is IEEE-802.16e standard [7] has scalable FFT size i.e. 128-2048 point FFT with OFDMA, Range varies from 1.6 to 5 Km at 5Mbps in 5MHz channel BW, supporting 100Km/hr speed. Conventional LUT-based technique is found to be unattractive from many aspects such as operating at low speed, large logic resources requirement leading to inefficiency in resource utilization, etc. A compact and user-friendly mathematical representation and subsequent algorithm is proposed [12]. A simple algorithm along with its mathematical background developed in the proposed approach, eliminates the requirement of floor function and allows low-complexity FPGA implementation [14].

2The paper is organized as follows: Section II explains the eneral interleaving/deinterleaving concept. In Section III, a proposed algorithm for Reconfigurable address generator with its mathematical formulation. Hardware implementation of the algorithm is discussed in Section IV. Section V shows the simulation result and in Section VI the FPGA implementation is discussed. Finally, this proposal is concluded in Section VII.

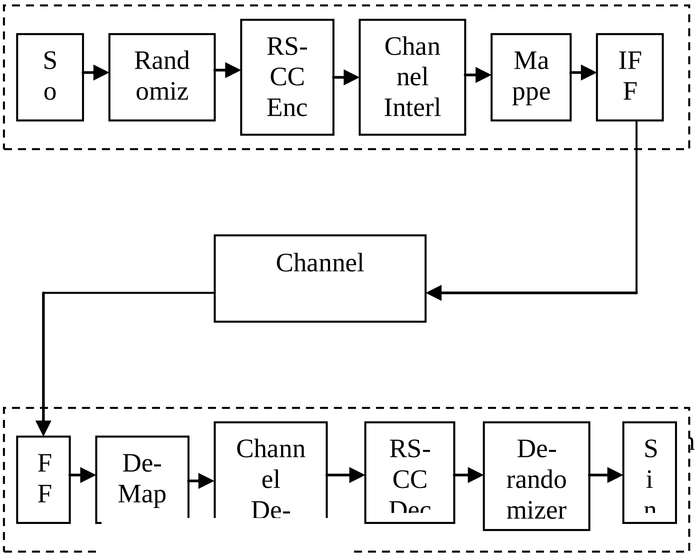

Fig.1 Block diagram of WiMAX transreceiver

II. WIMAX CHANNEL INTERLEAVER

Generalized block diagram of WiMAX transreceiver is shown in Fig 1. Randomizer is basically scrambling of data to generate random sequence in order to improve coding performance and data integrity of the input bits. RS codes are non-binary cyclic codes that add redundancy to the data. This redundancy is basically addition of parity bits into the input bit stream that improves the block errors. Where the CC codes introduce redundant bits in the data stream with the use of linear shift registers. Interleaving is the process of distributing

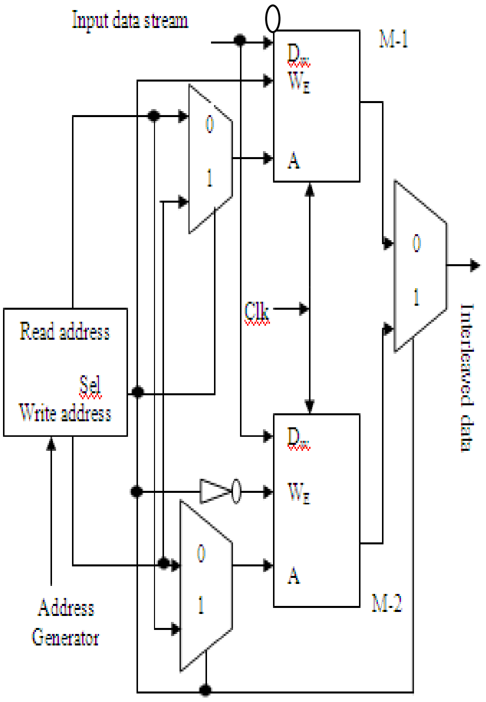

Fig.2 Block diagram of Interleaver/deinterleaver

Interleaving, data is mapped onto non-adjacent subcarriers to overcome the effects of multipath distortion and burst errors. Mapper involves mapping of digital information onto analog form such that it can be transmitted over the channel. An Inverse Fast Fourier transform converts the input data stream from frequency domain to time domain representing OFDM Subcarrier as the channel is basically in time domain. In the receiver, the blocks are organized in the reverse order enabling the restoration of the original data sequence at the output.

The interleaver memory block comprises of two memory modules (RAM-1 and RAM-2), three MUX and an inverter as shown in figure 2. In block interleaving when one memory block is being written the other one is read and vice versa. Each memory module receives either write address or read address with the help of the MUX connected to their address inputs (A) and sel line. RAM-1 at the beginning receives the read address and RAM-2 gets the write address with write enable (WE) signal of RAM-2 active. After a particular memory block is read or written up to the desired location, the select input gets toggled and the operation is reversed. The MUX at the output of the memory modules routes the interleaved data stream from the read memory block to the output.

The block interleaver for channel interleaving in WiMAX is expressed in the form of a set of two equations for two steps of permutations as shown in the equation 1 and 2. The first step ensures that coded bits in sequence are mapped onto different shuffled subcarriers, while the second step ensures that adjacent coded bits are mapped alternately onto less or more significant bits of constellation, thus avoiding long ones and zeros.

The first permutation m_k for index k is defined by

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

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

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

TABLE I
Deinterleaver Sample Addresses for Three Modulation Types

N_{cbps}, Code rate and Modulation type	De-interleaver Address				
N_{cbps} = 96-bits, ½ Code rate, QPSK	0	16	32	48	64
	1	17	33	49	65
	2	18	34	50	66
	3	19	35	51	67
N_{cbps} = 192-bits, ½ Code rate, 16-QAM	0	16	32	48	64
	17	1	49	33	81
	2	18	34	50	66
	19	3	51	35	83
N_{cbps} = 288-bits, ½ Code rate, 64-QAM	0	16	32	48	64
	17	33	1	65	81
	34	2	18	82	50
	3	19	35	51	67

The parameter s is defined as s= ceil(N_{cbps}/2), where N_{cbps} 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 deinterleaver,

which performs the inverse operation, is also defined by the two permutations as shown in the equation 3 and 4. 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

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

The second permutation k_n for index n is given by

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

The range of n and k for above equation is defined as 0, 1, and 2 . . . (Ncbps – 1).

III. Proposed Algorithm for Reconfigurable Address Generator

The direct computation of two step permutations 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. Table I 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*.

A close examination of the addresses in Table I reveals that the correlation between them follows the manner, as shown in Table II. The mathematical formulation of the correlation between the addresses, is represented by (5)–(7), i.e.,

K_n, QPSK=

$$\{d * i + j \text{ for } \forall j \text{ and } \forall \text{i} \quad \rightarrow (5)$$

K_n, 16-QAM=

$$\begin{cases} \text{d * i + j} \quad \text{for j\%2} = 0 \text{ and for } \forall i \\ d * (i+1) + j \quad \text{for j\%2} = 1 \text{and} \\ \qquad \text{for i\%2} = 0 \qquad \rightarrow (6) \\ d * (i-1) + j \quad \text{for j\%2} = 1 \text{and} \\ \qquad \text{for i\%2} = 1 \end{cases}$$

K_n, 64-QAM =

$$\begin{cases} d*i+j \ \text{for} j\%3=0 \ \text{and for} \ \forall i \\ d*(i-2)+j \,\text{for}\ j\%3=1 \,\text{and} \\ \qquad \text{for}\, i\%3=2 \\ d*(i+1)+j \,\text{for}\ j\%3=1 \,\text{and} \\ \qquad \text{for}\, i\%3\neq 2 \qquad \rightarrow (7) \\ d*(i+2)+j \,\text{for}\ j\%3=2 \,\text{and} \\ \qquad \text{for}\, i\%3=0 \\ d*(i-1)+j \,\text{for}\ j\%3=2 \,\text{and} \\ \qquad \text{for}\, i\%3\neq 0 \end{cases}$$

Where j = 0, 1. . . d − 1 and i = 0, 1, . . , (Ncbps/d) –1 represent the row and column numbers, respectively, in Table II. In addition, K_n represents the deinterleaver addresses. General validity of (5)–(7) to represent the correlation between addresses of Table II has formally been proven using the algebraic analysis in [6], which lacks the involvement of (5)–(7). The outcome of this analysis using (5)–(7) provides the same result, as shown in Table II. Thus, (5)–(7) play the pivotal role in establishing formal mathematical foundation of our proposed algorithm.

From Table II and the mathematical representation by (5)–(7), following for the three modulation schemes are proposed. These algorithms eliminate the requirement of floor function while generating write addresses .Algorithm for the Reconfigurable Address Generator is as follows,

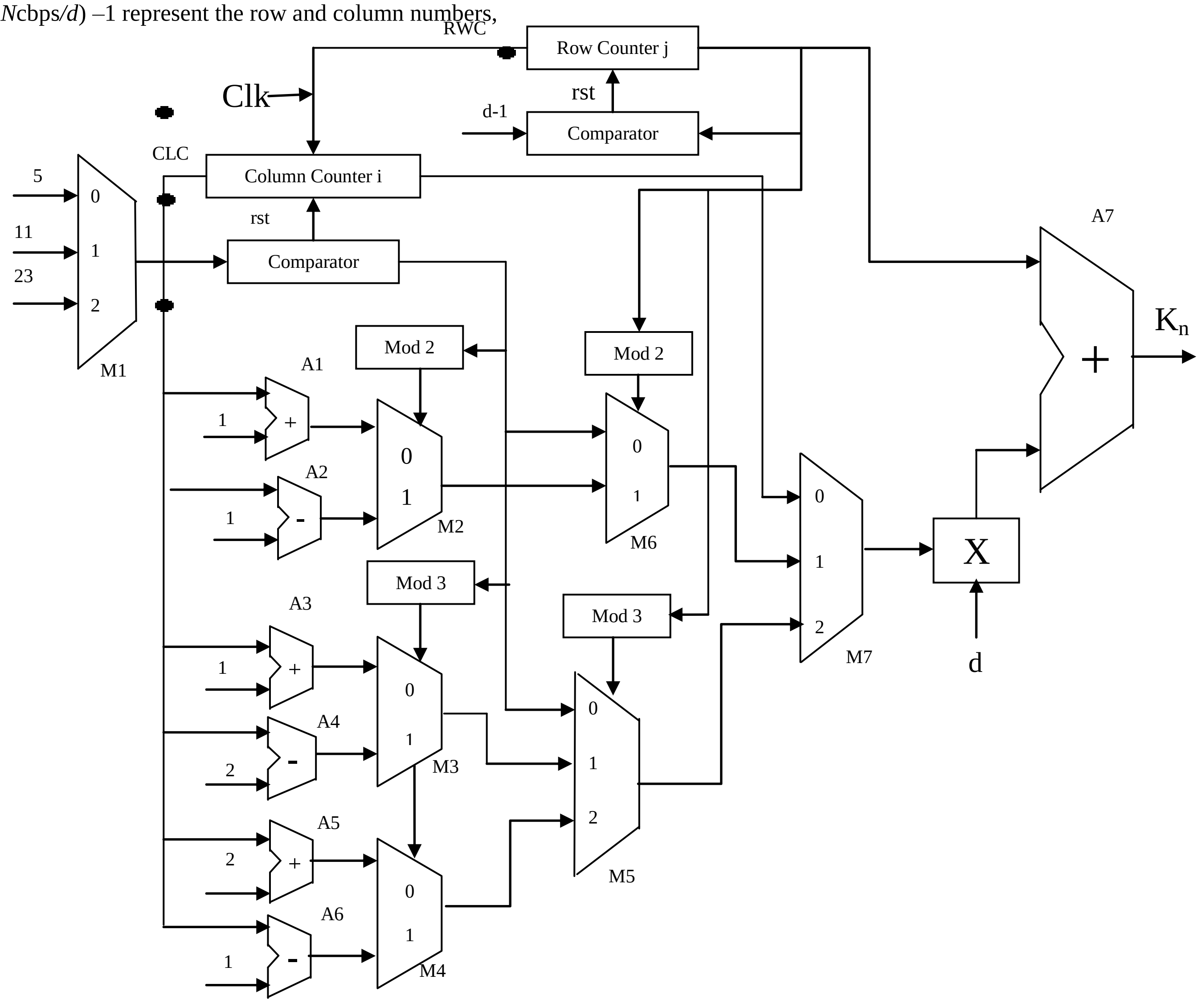

Fig.3 Reconfigurable Address Generator for WiMAX application

```
Initialize $N_{cbps}$ and d for respective modulation schemes
for j= 0 to d-1, j ++
    for i = 0 to ($N_{cbps}$/d)-1, i++
        if (sel = 0 )
            K n = d* i+ j
        end if
        if (sel = 1 )
            if (j mod2 =0)
```

```
             K n = d* i+ j
     else if (i mod2 =0)
             K n = d* (i+1)+ j
     else
            K n = d* (i-1) + j
           end if
        end if
     end if
         if (sel = 2 )
            if (j mod3=0)
                 $K_n$= d* i + j
     else  if (j mod3=1)
              if (i mod3=2)
                   $K_n$= d*( i-2) + j
                 else
                     K $_n$= d*( i+1) + j
         end if
     else if (j mod3=2)
               if (i mod3=0)
                   $K_n$= d*( i+2) + j
                 else
                    $K_n$= d*( i-1) + j
              end if
          end if
       end if
     end for
   end for
```

IV. CIRCUIT IMPLEMENTATION

The proposed algorithms for the Reconfigurable address generator of the WiMAX deinterleaver with modulation schemes, digital circuit's transformation of these algorithms are made and are shown in Fig.3. The hardware shown in Figure has a row counter RWC to generate row numbers between 0 and $d - 1$. A column counter CLC with multiplexer M and comparator C generate the variable column numbers to implement permissible N_{cbps}. A multiplier ML and an adder A perform the desired operations to implement (5, 6 and 7). The
Reconfigurable address generator modules are designed with an incrementer, a decrementer, two modulo-2 blocks, two modulo-3 blocks and seven multiplexers. As per Table I, the Reconfigurable address generator has to implement three different progressive patterns for the column numbers and 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. A simple up counter generates the read addresses for the 2-D deinterleaver of reconfigurable address generator.

TABLE II
Determination of Correlation between Addresses

Row no.(j)	Column no.(i) →	0	1	2
0	N_{cbps}= 96-bits, ½ code rate, QPSK	d.0+0=0	d.1+0=16	d.2+0=32
1		d.0+1=1	d.1+1=17	d.2+1=33
2		d.0+2=2	d.1+2=18	d.2+2=34
3		d.0+3=3	d.1+3=19	d.2+3=35
0	N_{cbps}= 192-bits,½ code rate, 16-QAM	d.0+0=0	d.1+0=16	d.2+0=32
1		d.1+1=17	d.0+1=1	d.3+1=49
2		d.0+2=2	d.1+2=18	d.2+2=34
3		d.1+3=19	d.0+3=3	d.3+3=51
0	N_{cbps}= 288-bits,½ code rate, 64-QAM	d.0+0=0	d.1+0=16	d.2+0=32
1		d.1+1=17	d.2+1=33	d.0+1=1
2		d.2+2=34	d.0+2=2	d.1+2=18
3		d.0+3=3	d.1+3=19	d.2+3=35

V. SIMULATION RESULT

The proposed hardware of the Reconfigurable address generator is converted into a VHDL program using the Xilinx ISE and simulated for all permissible modulation types and code rates using ModelSim XE-III. The Fig. 4. (a) Shows the simulation result for m = 0 (i.e.) for QPSK for row (j = 5 & 6). The Fig. 4. (b) Shows the simulation result for m = 1 (i.e.) for 16-QAM for row (j = 4). The Fig. 4. (c) Shows the simulation result for m = 2 (i.e.) for 64-QAM for row (j = 5).

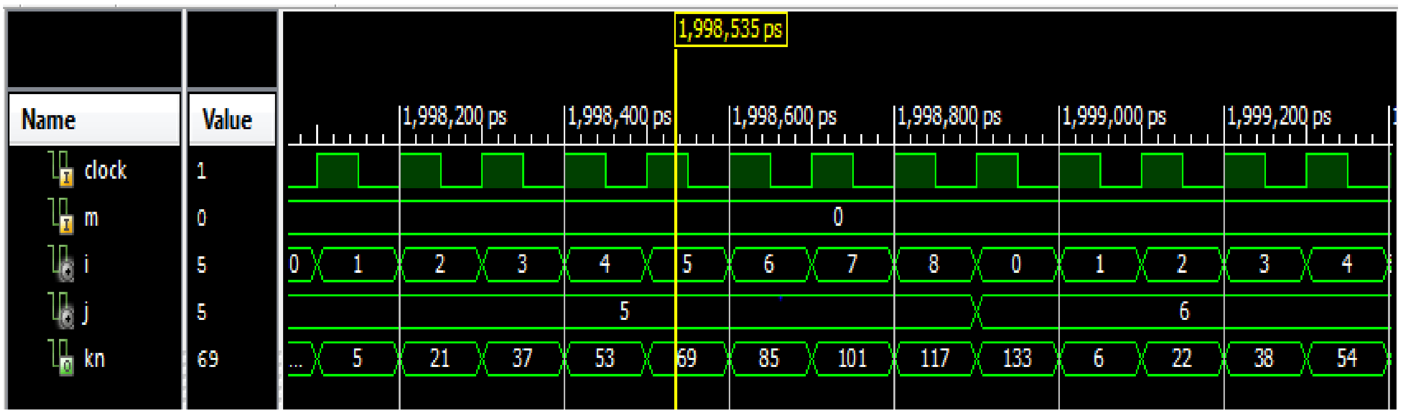

Fig.4. (a) Simulation results showing the address of the Fifth row (j = 5) and the First portion of Sixth row (j = 6) for m = 0 i.e. QPSK, ½ code rate.

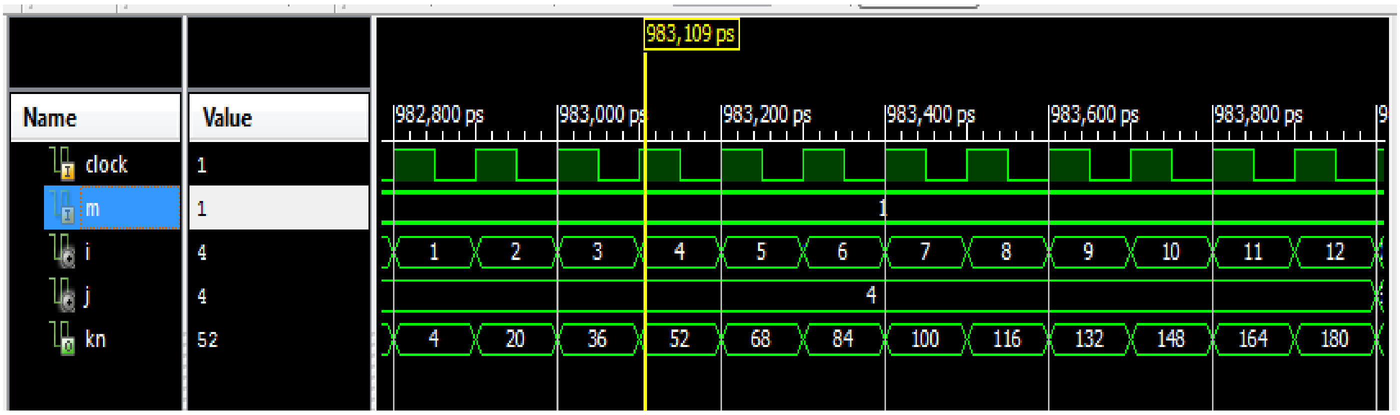

Fig.4. (b) Simulation results showing the address of the Fourth row (j = 4) for m = 1 i.e. 16-QAM, ½ code rate.

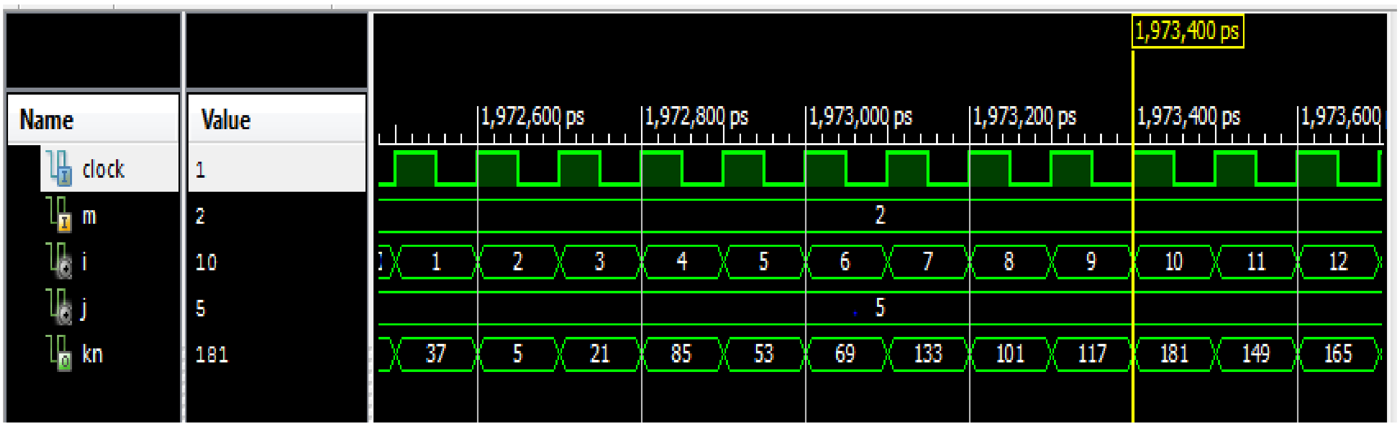

Fig.4. (c) Simulation results showing the address of the Fifth row (j = 5) for m = 2 i.e. 64-QAM, ½ code rate.

VI. FPGA IMPLEMENTATION RESULT

The VHDL program developed for the proposed WiMAX Reconfigurable address generator is downloaded on the Xilinx Spartan-6 SP605 (Device xc6slx45t-3ffg484). Table III shows the HDL synthesis report. Table IV Result for Reconfigurable addresses Generator for WiMAX.

TABLE III
HDL Synthesis Report

Logic Circuits Used	Qty	Logic Circuits Used	Qty
7-bit adder	4	8-bit register	1
8-bit adder	9	2-bit 2-to-1 multiplexer	2
4-bit sub tractor	2	6-bit 2-to-1 multiplexer	15
6-bit register	2	8-bit 2-to-1 multiplexer	8

As the FPGA-based implementation of the WiMAX Reconfigurable address generator has not been found in the literature, direct comparison of the results of our proposed work could not be carried out. However, the use of an internal multiplier of FPGA and the sharing of resources for Quadrature phase-shift keying, 16-quadrature-amplitude modulation (QAM), and 64-QAM modulations along with all possible code rates is proposed in[14]. Implementation of the conventional LUT-based technique of address generation for the WiMAX 2-D deinterleaver on the same FPGA is made in the similar manner as proposed in [12].

TABLE IV
Result for Reconfigurable address Generator for WiMAX

FPGA Resources	This Work
Number of Slice Register	24 out of 54576
Number of Slice LUTS	63 out of 27288
Number of LUT-FF pairs	16 out of 71
Number of Bonded IOBS	12 out of 296
Number of BUFG/BUFGCTRLs	1 out of 16
Frequency	199.557MHz

VII. CONCLUSION

This paper has proposed an address generation algorithm for WiMAX deinterleaving operation, with less mathematical complexity. A reconfigurable address generator circuitry for the proposed algorithm is implemented in FPGA using VHDL. This circuit supports different modulation schemes with different code rates as per IEEE 802.16e. The novelty of our approach includes higher operating frequency (199.557MHz) and better resource utilization in FPGA.

REFERENCES

[1] W. Konhauser, “Broadband wireless access solutions Progressive challenges and potential value of next generation,” Wireless Pers. Commun, vol. 37, no. 3/4, pp. 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 de-interleaver design,” in Proc Int. Conf. Consum. Electron, pp. 1–2, 2007.

[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. Micro electron, Marrakech, Morocco, pp.
181–184, 2009.

[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. 2010

[6] R. Asghar and D. Liu, “2D realization of WiMAX channel interleaver for efficient hardware implementation,” in Proc. World Acad. Sci. Eng. Technol., Hong Kong, vol. 51, pp. 25–29. 2009.

[7] IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems Amendment 2, EEE Std. 802.16e-2005, 2005.

[8] M. N. Khan and S.Ghauri, “The WiMAX 802.16e physical layer model,” in Proc. IET Int. Conf. Wireless, Mobile Multimedia Newt, Mumbai, India, pp. 117–120, 2008.

[9] J.G.Andrews, A.Ghosh, and R.Muhamed, Fundamentals of WiMAX Understanding Broadband Wireless Networking Upper Saddle River, USA: Prentice-Hall, ch. 8, 2007.

[10] Local and Metropolitan Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems, IEEE Std. 802.16-2004, 2004.

[11] Xilinx Spartan-3 FPGA Family: Complete Data Sheet, Xilinx, Inc., San Jose, CA, USA, 2012.

[12] 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.

[13] I.Kuon and J. Rose, “Measuring the gap between FPGAs and ASICs,” in Proc. Int. Symp, Field Program Gate Arrays, Monterey, CA, USA, pp. 21–30. 2006.

[14] B.K.Upadhyaya and S. K. Sanyal, “Efficient FPGA Implementation of Address Generator for WiMAX Deinterleaver’’ in IEEE Transactions on Circuits and Systems—II, 2013.