Novel design of WiMAX multimode interleaver for efficient FPGA implementation using finite state machine based address generator

Bijoy Kumar Upadhyaya, and Salil Kumar Sanyal

***Abstract*—**In this paper, we present a novel and efficient technique to model WiMAX multimode interleaver using hardware description language. The hardware model is implemented on FPGA platform. Our proposed interleaver consists of finite state machine based address generator and optimized FPGA's embedded resource based memory. The finite state machine based address generator of the interleaver shows better performance in terms of maximum operating frequency, and FPGA resource utilization compared to existing FPGA techniques. Use of FPGA's embedded memory offers advantages like reduced access time, lesser real estate occupancy of circuit board and lower power consumption than external memory based techniques. Estimated power consumption and software simulation of both address generator and the complete interleaver are also provided. Our design is truly based on IEEE 802.16e standard where all the code rates and modulation schemes for WiMAX have been incorporated. In addition our approach supports computation of interleaver addresses dynamically.

***Keywords*—** Address generator, FPGA, FSM, interleaver, VHDL, WiMAX.

I. Introduction

TREMENDOUS increase in use of internet in the last decade has put the quest of Broadband Wireless Access (BWA). BWA has emerged as the last mile access solution and a challenging competitor to existing 3G technologies [1]. It is increasingly gaining popularity as an alternative solution to Digital Subscriber Line (DSL) or cable modem for internet access. BWA has stringent requirements like high processing speed, flexibility and fast design Turn Around Time (TAT). These requirements make the designers to choose reconfigurable hardware platform like Field Programmable Gate Array (FPGA). Moreover, any new technology like Worldwide Interoperability for Microwave Access (WiMAX) needs some time to mature. Thus a product implemented on FPGA can easily be upgraded by making necessary changes in the Hardware Description Language (HDL) code and thus becomes obsolescence free. In addition, the TAT of FPGA based circuit is much shorter compared to Application Specific Integrated Circuit (ASIC). Design flexibility is another important advantage of FPGA based implementations. The proposed system could have also been implemented using software. The principal drawback of such approach is that a powerful computer is to be used to run the program for achieving high processing speed, a prerequisite for WiMAX system. Employment of such powerful computer may be a costly solution but may be detrimental to the popularity of WiMAX.

WiMAX is based on the IEEE 802.16 standard for BWA system. IEEE 802.16d, now known as, IEEE 802.16-2004 defines fixed BWA (FBWA) in the frequency band of 2 to 11GHz [2]. Amended IEEE 802.16e adds mobility support to IEEE 802.16 and defines standard for mobile BWA (MBWA) in the frequency band 2 to 6GHz.Typical data rate in IEEE 802.16e is 5 Mbps with bandwidth 1.25 to 20 MHz. Both IEEE 802.16-2004 and IEEE 802.16e permit Non Line of Sight (NLOS) connectivity [3].

Orthogonal Frequency Division Multiplexing (OFDM) [3] technique offers promising solution that has gained tremendous research interest in recent years due to its high transmission capability and also for alleviating the adverse effects of Inter Symbol Interference (ISI) and Inter Channel Interference (ICI). In an OFDM system, the data is divided into multiple parallel substreams at a reduced rate, and each is modulated and transmitted on a separate orthogonal subcarrier. This increases symbol duration and improves system robustness [4]. OFDM is achieved by providing multiplexing on users' data streams on both uplink and downlink transmissions. OFDM is the fundamental building block of the IEEE 802.16 standard.

Interleaving plays a vital role in improving the performance of Forward Error Correction (FEC) codes in terms of Bit Error Rate (BER). Interleaving is the process to rearrange code symbols so as to spread burst of errors into random like errors and thereafter FEC techniques could be applied to correct them. In conventional block interleaver [5], the bits received from the encoder are stored row wise in the interleaver's memory and read column wise. WiMAX uses a special type of block interleaver in which the Interleaver Depth (ID) and pattern vary depending on the code rate and modulation type.

Use of multiplexers (MUXs) to model the various specifications of WiMAX interleaver into a single circuit is an

Manuscript received April 13, 2012.

Bijoy Kumar Upadhyaya is with Tripura Institute of Technology, Narsingarh, Tripura, India, Pin 799009 (phone: +91-94365-02613; fax: +91-381-2342330; e-mail: bku.agt@gmail.com).

Salil Kumar Sanyal is with the Electronics & Telecommunication Engineering Department, Jadavpur University, Kolkata, India, Pin 700032. (e-mail: s_sanyal@ieee.org).

efficient approach leading to multimode interleaver. In view of the various code rates and modulation schemes in WiMAX, multimode interleaver is the ideal solution from implementation point of view. A few implementation of interleaver with different specifications are found in the literature. Efficient implementation techniques of interleaver for Digital Video Broadcasting (DVB) have been discussed in [6, 7]. Design and implementation issues of Turbo Code interleavers are addressed in [8]. Ahmad Sghaier et. al. [9] has described a look up table based method for address generation of the interleaver used in IEEE 802.11 wireless LAN (WLAN). A technique of converting 1-dimensional interleaver equation into 2-dimentional equations is proposed in [10]. The authors in [10] claim the hardware architecture to be tested on 0.12μm CMOS technology. Recently Khater et. al. [11] described a VHDL based implementation of interleaver address generation circuitry for IEEE 802.16e interleaver with ½ code rate.

The aim of this work is to present a novel FSM based multimode, high speed and hardware efficient technique to implement the WiMAX interleaver based on IEEE 802.16e standard on FPGA platform. The interleaver consists of Finite State Machine (FSM) based address generator and optimized FPGA's embedded resource based memory module. The implementation is different in the sense that it involves full FPGA based design. As far as address generator is concerned our work shows approximately 30% betterment in terms of maximum clock frequency, nearly 46% lesser FPGA flip flop consumption at the marginal cost of Logic Cell (LC) utilization (less than 3% over use) compared to existing implementation [11]. Comparison is made with conventional Look-Up Table (LUT) based address generation technique by implementing it on the same FPGA. It has shown efficiency in utilizing resources like LC and flip flop over our proposed technique, but on the issue of very important FPGA resources like internal memory block and maximum operating frequency, our proposed technique perform much better. Conventionally, use of FPGA's embedded memory offers advantages like reduced access time, reduced real estate occupancy of circuit board and lower power consumption than external memory based techniques. So far no work utilizing FPGA's embedded memory to model the WiMAX multimode interleaver has been found in the literature to the best of the knowledge of the authors. As per IEEE 802.16e standard [12], ½, ⅔ and ¾ are the allowed code rates where as QPSK, 16-QAM and 64-QAM are the permitted modulation schemes. Unlike [11], our design is truly based on IEEE 802.16e standard where all the code rates and modulation schemes for WiMAX have been incorporated. In addition our approach supports on the fly (dynamic) computation of interleaver memory addresses.

The paper is organized as following: section II provides overview of IEEE 802.16e standard for WiMAX system. Section III presents the inside detail of block interleaving in WiMAX system. Proposed hardware modeling of the interleaver along with address generator circuitry is described at length in section IV. Simulation result of the hardware model is explained in Section V. Section VI performs the critical analysis of FPGA implementation result. Concluding remarks are given in Section VII.

II. System Description

The system level overview of IEEE 802.16e based WiMAX system is described in Fig.1. In this system, the input binary data stream obtained from a source is randomized to prevent a long sequence of 1s and 0s, which will cause timing recovery problem at the receiver. Psudeo Random Binary Sequence (PRBS) is used in which randomization is done by modulo 2 addition of the data with the output of the PBRS itself [13]. The randomized data bits are thereafter encoded using Reed Solomon (RS) encoder followed by Convolutional Coder (CC). The former is suitable for correction of burst type of error whereas the later is for random error. After RS-CC encoding all encoded data bits shall be interleaved by a block interleaver. In traditional block interleaver bits received from the encoder are stored row wise in the interleaver's memory. As soon as the memory is completely filled, the bits are read column-by-column manner [5]. In the block interleaver of WiMAX system, data is written in the memory in a predefined manner based on certain permutation [14] and read in a sequential manner. After interleaving, data passes through the mapper block in which modulation takes place. The resulting data symbols are used to construct one OFDM symbol by performing Inverse Fast Fourier Transform (IFFT). Cyclic Prefix (CP) is used to reduce ISI and ICI [3].

In the receiver, inverse blocks are applied which perform DFT, de-mapping, deinterleaving, decoding and de-randomizing operations in sequential manner to get back the original data bits.

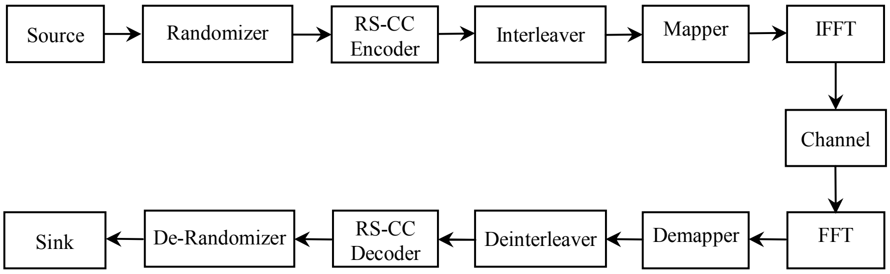

Fig. 1 Overview of WiMAX PHY layer system

I. Interleaving in Wimax

The block interleaver used in WiMAX system have different interleaving pattern for different code rates and modulation schemes [13]. Various interleaver depths are required to incorporate various code rates and modulation schemes. Table I describes permitted interleaver depths in IEEE 802.16e [12] for all code rates and modulation schemes. The encoded bits are interleaved using two step process. The first step ensures that the adjacent coded bits are mapped onto nonadjacent subcarriers, which provides frequency diversity and improves the performance of the decoder. The second step ensures that adjacent bits are alternately mapped to less and more significant bits of the modulation constellation to avoid long run of lowly reliable bits.

Let N_{cbps} is the block size corresponding to the number of coded bits per allocated sub-channels per OFDM, d represents number of columns of the block interleaver which is typically chosen to be 16 for WiMAX. m_k is the output after first level of permutation 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 sub-carrier, i.e., 2, 4 or 6 for QPSK, 16-QAM or 64-QAM respectively [14]. The first and second levels of permutation are given by (1) and (2) respectively are as follows:

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

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

where % and $\lfloor\ \rfloor$ signify modulo and floor functions respectively.

II. Hardware Modeling of Address Generator

The proposed hardware model of WiMAX interleaver consists of two sections: address generator and interleaver memory as shown in Fig. 2. The address generator is basically the simultaneous implementation of (1) and (2) which is the write address along with provision for generation of read address for interleaver memory. Conventional block interleaver uses two memory blocks out of which one memory block is written and the other is read based on the value of select (SEL) signal. In this paper we design the memory block using dual port internal memory module of FPGA by partitioning it into two sub-modules.

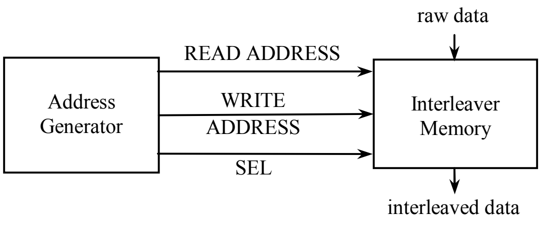

Fig. 2 Top level view of WiMAX interleaver

TABLE I.
PERMITTED INTERLEAVER DEPTH IN IEEE 802.16E FOR ALL CODE RATES AND MODULATION SCHEMES

Modulation Scheme	QPSK (s=1)		16-QAM (s=2)		64-QAM (s=3)		
Code Rate	½	¾	½	¾	½	⅔	¾
Interleaver Depth, N_{cbps} in bits	96	144	192	288	288	384	432
	192	288	384	576	576	-	-
	288	432	576	-	-	-	-
	384	576	-	-	-	-	-
	480	-	-	-	-	-	-
	576	-	-	-	-	-	-

A. Address Generator

The values of j_k obtained from (2) are the write addresses of interleaver memory. On evaluation of (1) and (2) with N_{cbps} = 96 (½ code rate, QPSK), N_{cbps} = 288 (¾ code rate, 16-QAM) and with N_{cbps} = 384 (⅔ code rate, 64-QAM) we find all the values of j_k, out of which first 32 values are listed in Table II. On careful examination of the values of j_k, it has been found that the subsequent addresses are not equally spaced for all cases. Within a modulation scheme, the increment values follow a fixed type of pattern irrespective of coding rate. Encoding of ID, Modulation Type (MOD_TYP) and increment values from implementation point of view are presented in Table III. In case of QPSK (with s = 1) the increments are linear having values like 6 for N_{cbps} = 96, 9 for N_{cbps} = 144 and so on. 16-QAM (with s = 2) and 64-QAM (with s = 3) have nonlinear increments like 13, 11 for N_{cbps} = 192 and 20, 17, 17 for N_{cbps} = 288 respectively.

TABLE II.
FIRST 32-PERMUTATION SAMPLE ADDRESSES FOR THREE CODE RATES AND MODULATION SCHEMES

N_{cbps}=96 bits, ½ code rate, QPSK	0	6	12	18	24	30	36	42
	48	54	60	66	72	78	84	90
	1	7	13	19	25	31	37	43
	49	55	61	67	73	79	85	91
N_{cbps}=288 bits, ¾ code rate, 16-QAM	0	19	36	55	72	91	108	127
	144	163	180	199	216	235	252	271
	1	18	37	54	73	90	109	126
	145	162	181	198	217	234	253	270
N_{cbps}=384 bits, ⅔ code rate, 64-QAM	0	26	49	72	98	121	144	170
	193	216	242	265	288	314	337	360
	1	24	50	73	96	122	145	168
	194	217	240	266	289	312	338	361

TABLE III.
INCREMENT VALUES FOR VARIOUS INTERLEAVER DEPTHS AND MODULATION SCHEMES WITH THEIR ENCODING

Modulation	MOD TYP	Interleaver Depth	ID	Increment values	Whether equally spaced
QPSK	00	96	000	6	Yes
		144	001	9	Yes
		192	010	12	Yes
		288	011	18	Yes
		384	100	24	Yes
		432	101	27	Yes
		480	110	30	Yes
		576	111	36	Yes
16-QAM	01	192	X00	13,11	No
		288	X01	19,17	No
		384	X10	25,23	No
		576	X11	37,35	No
64-QAM	1X	288	X00	20,17,17	No
		384	X01	26,23,23	No
		432	X10	29,26,26	No
		576	X11	38,35,35	No

Our proposed design of address generator block is described in the form of schematic diagram in Fig. 3. Unlike [11], our design includes all possible code rates and modulation type permitted under IEEE 802.16e. Bulk of the circuitry is used for generation of write address. As shown in Fig. 3, the schematic contains three levels of multiplexer (MUX). The first level MUXs implement the unequal increments required in 16-QAM and 64-QAM modulations. The four interleaver depths of 16-QAM modulation as shown in Table III are implemented by the first four MUXs in the first level. The select inputs of these four MUXs are tied together and are driven by a T-flipflop named QAM16_SEL. Similarly, the last four MUXs are for 64-QAM modulation. The select inputs like the former case are also tied together and are driven by a mod-3 counter named QAM64_SEL. The second level MUXs basically pick up one of the interleaver depth within the modulation scheme based the values of ID. Encoding of ID for various interleaver depths is also listed in Table III. The topmost MUX in level 2 implements the eight interleaver depths of QPSK modulation scheme available by concatenation of sub-channels [13]. The second and third MUXs in level 2 are for 16-QAM and 64-QAM respectively. The level 3 MUX sends one of the increment values to the next section of the circuit based on the type of modulation chosen (MOD_TYP).The 7-bit output from the third level MUX acts as one input of the 10-bit adder after zero padding. The other input of the adder comes from accumulator, which holds the previous address. After addition a new address is written in the accumulator. It has been observed experimentally that the adder output does not exceed 10-bit. The operation of the circuit begins with CLR = 1, which reset the accumulator, preset logic circuits, read address generator and SEL signal generator as shown in Fig. 3. So the first address generated is always 0. In the event of first clock the sum of MUX output and previous content of accumulator (which is 0) is stored in the accumulator which is the second address of the interleaver. Subsequent clock events generate consecutive addresses of the interleaver by adding present output of accumulator with a constant value chosen by the MUXs.

The read addresses are linear in nature and are generated using a ten bit up counter as shown in Fig. 3. The counter is reset whenever it reaches to the terminal count for a desired modulation scheme and ID. For example, in case of 16-QAM with N_{cbps} = 288, the counter counts from 0 to 287 and then repeats. The SEL generator is basically a T- flipflop used to generate the select (SEL) signal to switch between write and read memory block and is initialized to zero using CLR input. The preset logic is a FSM whose principal function is to generate the correct beginning write addresses for all subsequent iterations and is described at length in the next section.

Preset logic as finite state machine: The Preset Logic block of Fig. 3 is the heart of the address generator of WiMAX interleaver. It is basically a hierarchical FSM and the state diagram is shown in Fig. 4. This block contains a 4-bit counter keeping the track for end of states during the iteration. The FSM enters into the first state (S_F) with CLR=1. Based on the value in MOD_TYP it makes transition to one of the three possible next states (S_{MT0}, S_{MT1} or S_{MT2}). Each state in this level represents one of the possible modulation schemes. The FSM thereafter makes transition to one of the next level states (S_{ID0} to S_{ID7} from S_{MT0}, S_{ID0} to S_{ID3} from S_{MT1} or S_{MT2}) based on the value in ID. The various states of this level signify one of the interleaver depths. From these states it branches to next level of states based on the value in the accumulator. When the FSM at this level reaches to the terminal value of that iteration (e.g. 90 in S_{ID0} of S_{MT0}), it makes transition to a state (e.g. S_{000}) in which it loads the accumulator with the initial value (e.g. Preset=1) of the next iteration. This process continues till all the interleaver addresses are generated for the selected MOD_TYP and ID. If no changes take place in the values of MOD_TYP and ID, the FSM will follow the same route of transition and the same set of interleaver addresses will be continually be generated. Any change in MOD_TYP and ID value causes the interleaver to follow a different path. In order to facilitate the address generator with on the fly address computation feature, we have made the circuit to respond to CLR followed by MOD_TYP and ID inputs at any stage of the FSM. With CLR=1 it comes back to S_F state irrespective of its current position and there after transits to desired states in response of new values in MOD_TYP and ID.

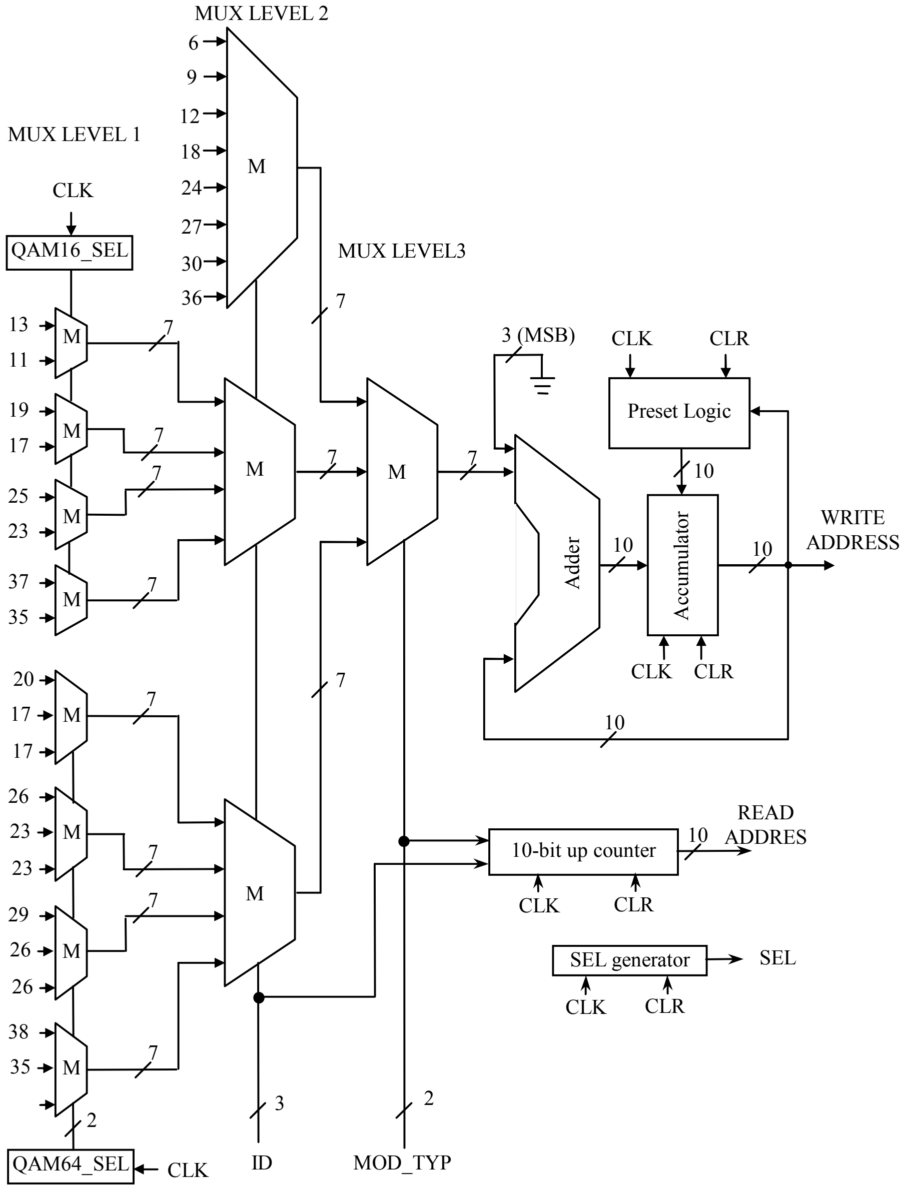

Fig. 3 Address generator of interleaver

B. Interleaver Memory

The interleaver memory block comprises of two memory modules (RAM-1 and RAM-2), three MUXs and an inverter as shown in Fig. 5. 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 (W_E) signal of RAM-2 active. After a particular memory block is read / written up to the desired location, the status of SEL changes 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.

In this work the memory requirement of block interleaver has been modeled using a dual port BRAM of FPGA. The use of memory is optimized in the sense that one dual port memory has been utilized to model the two memory blocks required in

C. Modeling Memory in FPGA

Static Random Access Memory (SRAM) based FPGAs [11] offer embedded storage for potential applications like local storage, FIFO, data buffers, stack etc. We used Xilinx Spartan-3, device XC3S400 FPGA in our experimentation. It contains total 288K bits of such embedded memory called Block RAM (BRAM) arranged in 16 blocks of 18K bit each. Out of 18K bits, 16K bits are used for data and rest 2K bits are used for parity purpose. Each block can be organized in various manner like 16K X 1bit, 8K X 2bit, 4K X 4bit etc. The maximum memory required for WiMAX interleaver is 576 bits. Two identical memory blocks each of capacity 576 bits are required for implementation of the block interleaver.

IEEE 802.16e based block interleaver. As shown in Fig. 6 when one port is in write mode the other is in read mode and vice versa. The first memory module occupies 0-575 bit locations whereas the other module is placed from 1024 to 1599 bit location in the 16K X 1 dual port BRAM.

IV. Simulation Results

The simulation results are obtained in the form of timing diagram using ModelSim Xilinx Edition-III version 6.0a software. In order to have a clear picture of the proposed technique the simulation result of the address generator and the complete interleaver have been presented separately.

A. Address Generator

Simulation results of address generator are described in Fig. 7(a), (b) and (c). Fig. 7(a) is for MOD_TYP = 00 and ID = 000 i.e. QPSK with N_{cbps} = 96. In Fig. 7(b) simulation result of 16-QAM with N_{cbps} = 288 (MOD_TYP = 01 and ID=001) is presented. Similarly address generation for 64-QAM with N_{cbps} = 384 (MOD_TYP = 10 and ID=001) is shown in Fig. 7(c). In all the figures, initially CLR = 1 to ensure that the counter in preset logic and accumulator are reset. In order to maintain clarity, only first two iterations for the three situations have been presented. Addresses generated in Fig. 7(a), (b) and (c) clearly conform to Table II. The authors have simulated and tested the address generation circuitry for all other values of ID and MOD_TYP, however to restrict the length of the paper other situations are not shown.

B. Complete Interleaver

Figures 8(a), (b) and (c) explain the interleaving operation of our proposed interleaver for WiMAX system. In these figures the modulation types and interleaver depths chosen are identical with Fig. 7(a), (b) and (c) respectively. The raw data input (data_in) into the interleaver in Fig. 8(a), (b) and (c) are held high for first 16 consecutive bit duration and made low thereafter to have clear view of the interleaving operation. As seen in figures these consecutive bits are dispersed by a predefined interval which is 6 in Fig. 8(a), 19, 17 in Fig. 8(b) and 26, 23, 23 in Fig. 8(c) and conforms to Table III.

The interleaving operation is verified by developing a Simulink model of the same using general block interleaver in MATLAB. The parameter block of the Simulink interleaver is configured as per the addresses obtained shown in Table II.

V. Critical Analysis of FPGA Implementation Results

The proposed hardware model of WiMAX interleaver is implemented and tested on Xilinx Spartan-3 (Device: XC3S400) FPGA platform in the laboratory. The FPGA implementation of the interleaver is carried out in two phases; firstly the address generator and thereafter the complete interleaver and presented accordingly.

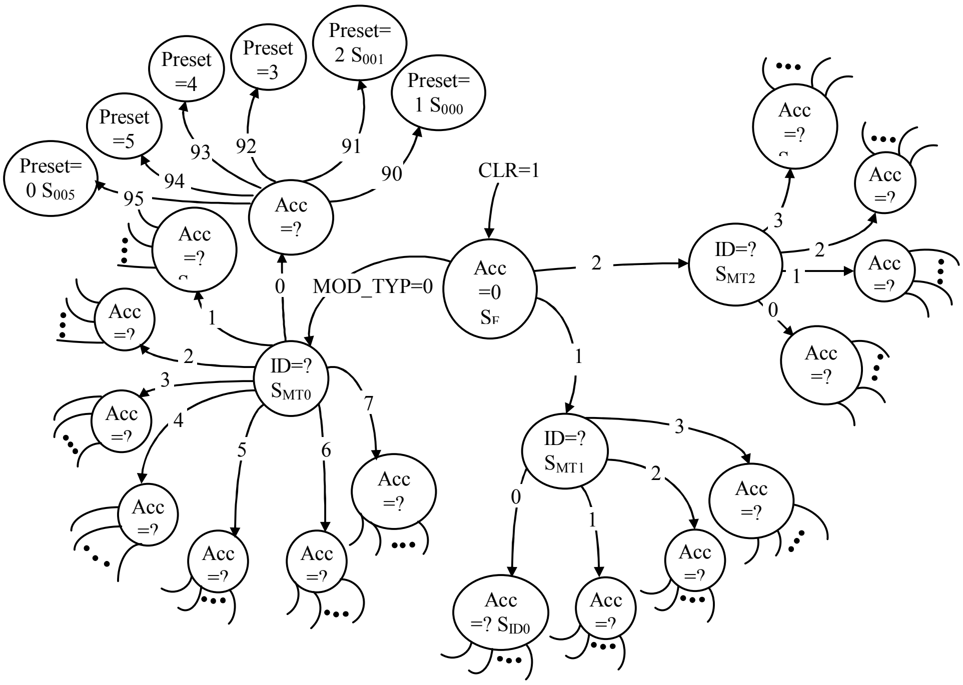

Fig. 4 States in preset logic

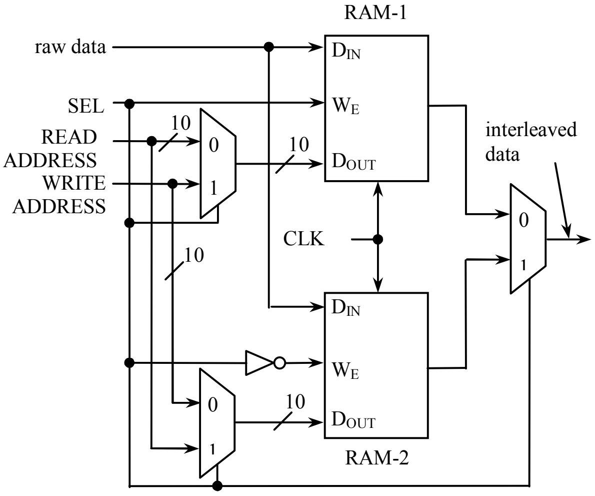

Fig. 5 Schematic view of interleaver memory block

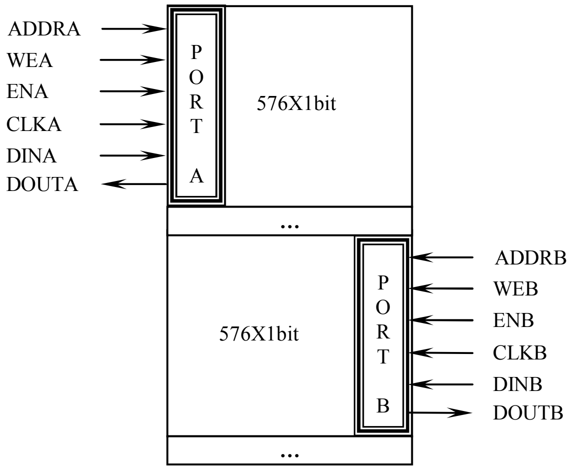

Fig. 6 Modeling of interleaver memory using dual port BRAM in FPGA

A. Address Generator

The VHDL model of the proposed FSM based address generator is prepared using Xilinx ISE 8.1i and implemented in the said FPGA. Direct comparison of our work with [10] is not possible as the two implementations are in different platform. In order to make comparative analysis we have also designed and implemented address generator circuitry for the interleaver depths listed in [11] and result is presented in Table IV. Our approach shows approximately 30% improvement in terms of maximum operating clock frequency, approximately 46% improvement in FPGA flipflop used with negligible (less than 3%) loss in terms of LCs used. Compact and efficient design of the preset logic provides this improvement. Table V shows the HDL synthesis report corresponding to the implementation of the circuit shown in Fig. 3. In addition we also implemented the conventional LUT based approach and comparison with our proposed approach in terms of FPGA resources and maximum operating frequency is presented in Table VI. The LUT based approach shows better results in terms of LCs and FPGA flipflops used where as the FSM based technique is free from using BRAM for address generation purpose and conserves it for other requirements. Most importantly, our approach can allow maximum operating frequency of 191.05MHz against 140.33MHz of the other technique showing 36.14 % improvement. Higher operating frequency is an important advantage in WiMAX system. The estimated power consumption of the circuit is found to be 56mW using Xilinx XPower I.25.

The address generator circuit when implemented on recent FPGAs like Virtex 4 shows further betterment in terms of operating frequency (278.30MHz) but at the cost of increased power consumption (224mW). As these FPGAs offer a large number of resources the utilization percentage as shown in Table VI further goes down.

B. Complete Interleaver

This section makes the critical analysis of FPGA implementation results of the entire interleaver including the proposed FSM based address generator. The HDL synthesis report of the complete interleaver is presented in Table VII. It shows additional requirement of few flipflops/latches and multiplexers which are used in designing the memory module of the interleaver.

TABLE IV.
COMPARATIVE ANALYSIS OF SIMILAR IMPLEMENTATIONS OF ADDRESS GENERATOR

Implementation Technique	No. of LCs used	No. of flip-flops used	Improvement in flip-flop used	Max. Clock Freq. (MHz)	Improvement in Max. Clock Freq.
Khater et. al. [11]	105	54	45.95%	147.9	29.14%
This work	108	37		191.05	

TABLE V.
HDL SYNTHESIS REPORT OF ADDRESS GENERATOR

Logic Circuit used	Quantity
8x7-bit ROM	1
10-bit adder	2
2-bit adder	1
4-bit up counter	1
Flipflops	18
10-bit latch	1
7-bit latch	1
7-bit 4-to-1 multiplexer	2
7-bit 8-to-1 multiplexer	1

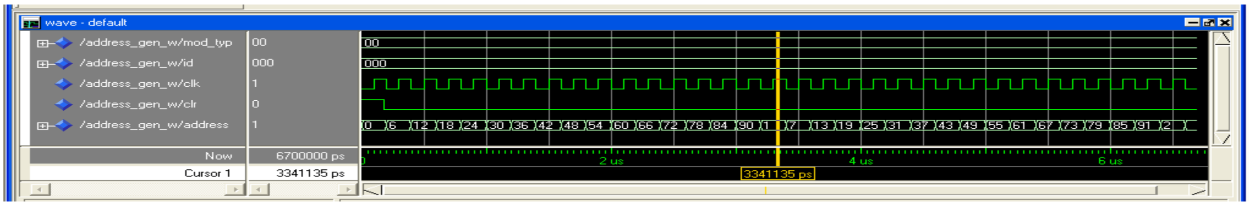

Fig. 7(a) Generation of first 32 write addresses with MOD_TYP = 00, ID = 000

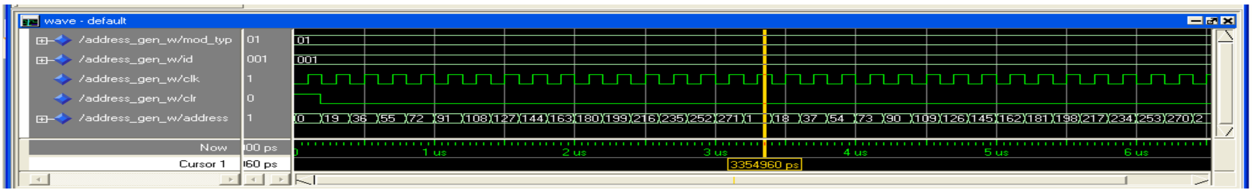

Fig. 7(b) Generation of first 32 write addresses with MOD_TYP = 01, ID = 001

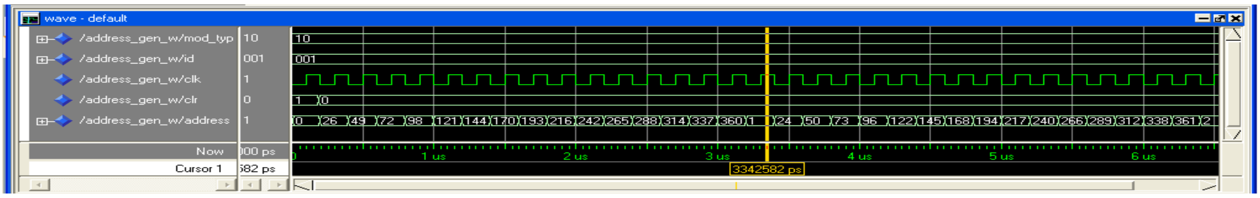

Fig. 7(c) Generation of first 32 write addresses with MOD_TYP = 10, ID = 001

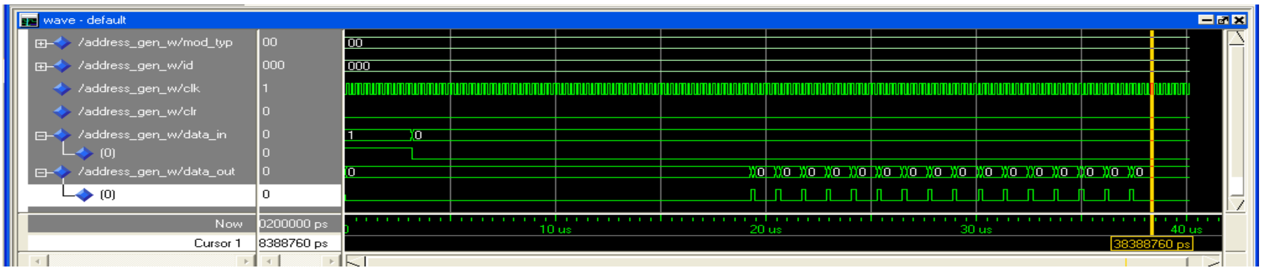

Fig. 8(a) Interleaving operation with MOD_TYP = 00, ID = 000

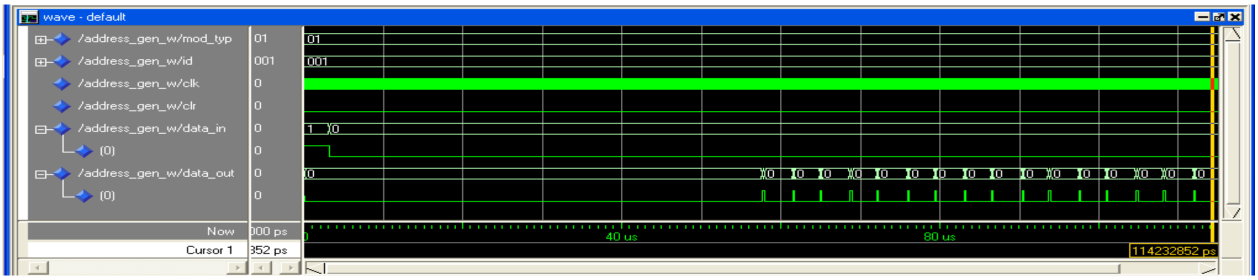

Fig. 8(b) Interleaving operation with MOD_TYP = 01, ID = 001

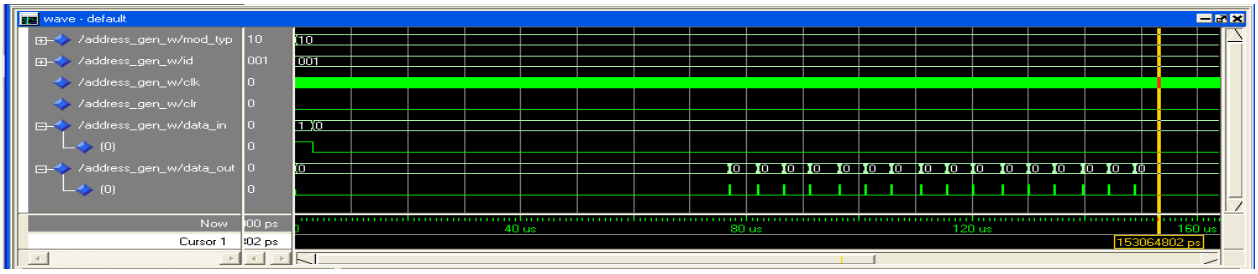

Fig. 8(c) Interleaving operation with MOD_TYP = 10, ID = 001

TABLE VI.
COMPARISON OF DEVICE UTILIZATION SUMMARY OF ADDRESS GENERATOR BETWEEN FSM AND LUT BASED METHODS

FPGA Parameters	Performance of FSM based Technique	Performance of LUT based Technique	Comparison
Logic Cells used	242 out of 3584 (6.75 %)	74 out of 3584 (2.01 %)	168 nos. over use by FSM
Flip Flops used	48 out of 7168 (0.67 %)	27 out of 7168 (0.38 %)	21 nos. over use by FSM
BRAM used	0 out of 16 (0 %)	6 out of 16 (37.5 %)	6 nos. over use by LUT
Maximum Operating Frequency	191.05MHz	140.33MHz	FSM based 50.72MHz faster than LUT

TABLE VII.
HDL SYNTHESIS REPORT OF INTERLEAVER

Logic Circuit used	Quantity
8x7-bit ROM	1
10-bit adder	2
2-bit adder	1
4-bit up counter	1
Flipflops	23
10-bit latch	1
7-bit latch	3
1-bit latch	1
7-bit 4-to-1 multiplexer	2
7-bit 8-to-1 multiplexer	1
1-bit 4-to-1 multiplexer	5

Device utilization summary of the complete interleaver implementation is described in Table VIII. The utilization percentage of LCs and flipflops are marginally increased because of the associated circuitry in the memory module of the interleaver. Number of Input Output Blocks (IOBs) has been dropped by 8 as because the 10-bit address output lines of address generator have been replaced by 2 lines; one carrying raw input data and the other sending out the interleaved data. The interleaver utilizes two BRAMs which is 12.5% of the available BRAM blocks in Spartan-3 FPGA. Minimum propagation delay and maximum operating frequency of the FPGA based interleaver is found to be 7.442ns and 134.381MHz respectively. Due to efficient modeling, the interleaver circuitry uses very few FPGA resources thereby making room for other associated circuitry like randomizer, encoder etc to be implemented on the same FPGA chip. Because of the presence of floor and mod function in (1) and (2), direct implementation of the address generation circuitry is very complex and consumes large amount of logic resources. Instead, our state machine based approach provides a faster and resource efficient implementation of WiMAX interleaver on FPGA platform.

TABLE VIII.
DEVICE UTILIZATION SUMMARY OF INTERLEAVER

FPGA Resources	Utilization in Number	Utilization in %
Number of LCs	267 out of 3584	7.45
Number of Flip Flops	54 out of 7168	0.75
Number of Bonded IOBs	9 out of 141	06.38
Number of GCLKs	2 out of 8	25.00
Number of BRAMs	2 out of 16	12.50

VII. CONCLUSION

This paper presents a full FPGA implementation of WiMAX multimode interleaver. It proposes a novel finite state machine based address generator used for generation of write and read addresses for the interleaver memory. The interleaver memory is implemented using dual port Block RAM of Xilinx Spartan-3 FPGA. The presented circuit supports all the code rates and modulation schemes permitted under IEEE 802.16e standard. The simulation results endorse the correct operation of both address generator and interleaver as a whole. The novelty of our approach includes higher operating frequency and better resource utilization in FPGA.

REFERENCES

[1] K. H. Teo, Z. Tao and J. Zhang, “The Mobile Broadband WiMAX Standard,” *IEEE Signal Processing Magazine*, September, 2007, pp. 144-148.

[2] A. Ghosh, D. R. Wolter, J. G. Andrews and R. Chen, “Broadband Wireless Access with WiMAX/802.16: Current Performance Benchmarks and Future Potential,” *IEEE Commun. Mag.*, vol. 43, Feb. 2005, pp. 129–36.

[3] U. S. Jha and R. Prasad, *OFDM Towards Fixed and Mobile Broadband Wireless Access*, Artech House Publisher, London, 2007.

[4] B. Li, Y. Qin, C. P. Low and C. L. Gwee, “A Survey on Mobile WiMAX,” *IEEE Communication Magzine*, December, 2007, pp. 70-75.

[5] B. Sklar, *Digital Communications: Fundamentals and Applications, Prentice-Hall*, Englewood Cliffs, NJ, USA, 2nd edition, 2001.

[6] J. B. Kim, Y. J. Lim, and M. H. Lee, “A Low Complexity FEC Design for DAB,” *IEEE ISCAS, Sydney, Australia*, pp. 522-525, 2001.

[7] B. K. Upadhyaya and S. K. Sanyal, “VHDL Modeling of Convolutional Interleaver- Deinterleaver for Efficient FPGA Implementation,” *International Journal of Recent Trends in Engineering*, Academy Publisher, Finland, Vol 2, No. 6, November, 2009, pp. 66-68.

[8] S. Vafi and T. A. Wysocki, “Application of Convolutional Interleaver in Turbo Code with Unequal Error Protection,” *Journal of Telecommunication and Information Theory*, Jan, 2006, pp. 17-23.

[9] A. Sghaier, S. Ariebi, and B. Dony, “A pipelined implementation of OFDM transmission on reconfigurable platforms”, *CCECE08 Conference*, Dec, 2007, pp. 801-804.

[10] R. Asghar, and D. Liu, “2D realization of WiMAX channel interleaver for efficient hardware implementation” in *Proc. World Academy of Sc. Engineering and Technology*, vol 51., Hong Kong, 2009, pp. 25-29.

[11] A. A. Khater, M.M. Khairy and S. E.-D. Habib, “Efficient FPGA Implementation for the IEEE 802.16e Interleaver”, *International Conference on Microelectronics*, Morocco, 2009, pp. 181-184.

[12] *IEEE 802.16e-2005*, IEEE standard for local and metropolitan area networks—part 16: air interface for fixed broadband wireless access systems—amendment 2, 2005.

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

[14] J G. Andrews , A. Ghosh , R. Muhamed, Fundamentals of WiMAX: Understanding Broadband Wireless Networking (Prentice Hall Communications Engineering and Emerging Technologies Series), Prentice Hall PTR, Upper Saddle River, NJ, 2007.

Bijoy Kumar Upadhyaya received the B. Tech degree in Electronics and Communication Engineering from North Eastern Regional Institute of Science and Technology (NERIST), Itanagar, India in 1998 and the Master of Electronics and Tele-communication Engineering (M. E. Tel. E.) from Jadavpur University, Kolkata, India in 2007. He joined Tripura Institute of Technology, TIT (erstwhile known as Polytechnic Institute), Narsingarh, Tripura, India as Lecturer in 2000. He is presently working as Assistant Professor and Head in Electronics and Tele-communication Engineering Department of TIT, Narsingarh. He is Life member of Institution of Engineers (India) (IEI) and Institution of Electronics and Telecommunication Engineering (IETE). He has several publication in peer reviewed journal and international and national conferences. His research area is error control coding and its application in digital communication and digital system design using hardware description languages.

Dr. Salil K. Sanyal received his Ph.D (Engineering) Degree from Jadavpur University, Kolkata – 700032, West Bengal, India in 1990. He joined the Department of Electronics and Telecommunication Engineering, Jadavpur University as Lecturer in 1982 where currently he holds the post of Professor. He is the immediate past Head of the Department of Electronics and Telecommunication Engineering of Jadavpur University. He has published more than 130 Research Papers in reputed peer reviewed Journals and International/National Conference Proceedings. He is the co-author of the Chapter “Architecture of Embedded Tunable Circular Microstrip Antenna” in the book entitled “Large Scale Computations, Embedded Systems and Computer Security ” edited by Fedor Komarov and Maksim Bestuzhev (NY, Nova Science Publishers Inc. 2009). He is a Senior Member of IEEE and also the past Chair of IEEE Calcutta Section. His current research interests include Analog/Digital Signal Processing, VLSI Circuit Design, Wireless Communication and Tunable Microstrip Antenna.