An Amendment to FSM based Interleaver for WLAN and WiMAX in VHDL

Jaikaran Singh
Assoc. Professor, ECE
Shri Satya Sai Institute of Science & Technology
Sehore (M.P)

Mukesh Tiwari
Assoc. Professor, ECE
Shri Satya Sai Institute of Science & Technology
Sehore (M.P)

Deepak Dehriya
M. Tech Scholar (VLSI)
Shri Satya Sai Institute of Science & Technology
Sehore (M.P)

ABSTRACT

In Wireless communication is one among the foremost vivacious analysis areas within the communication field these days. This paper presents the implementation of FSM based WLAN and modified FSM based WiMAX interleaver in VHDL. For WLAN the implemented interleaver is compared with the available works. A modification in the FSM of address generator for WiMAX interleaver provides a significant 35.8% enhancement in terms of logic cells and 22% enhancement in terms of slice flip flops used, as compared to available work [41]. The circuit parameters and simulation results obtained using ModelSim XE II software are also presented.

General Terms

Interleaver, Address Generator, FPGA.

Keywords

WLAN, FSM, WiMAX, VHDL, Xilinx, ModelSim.

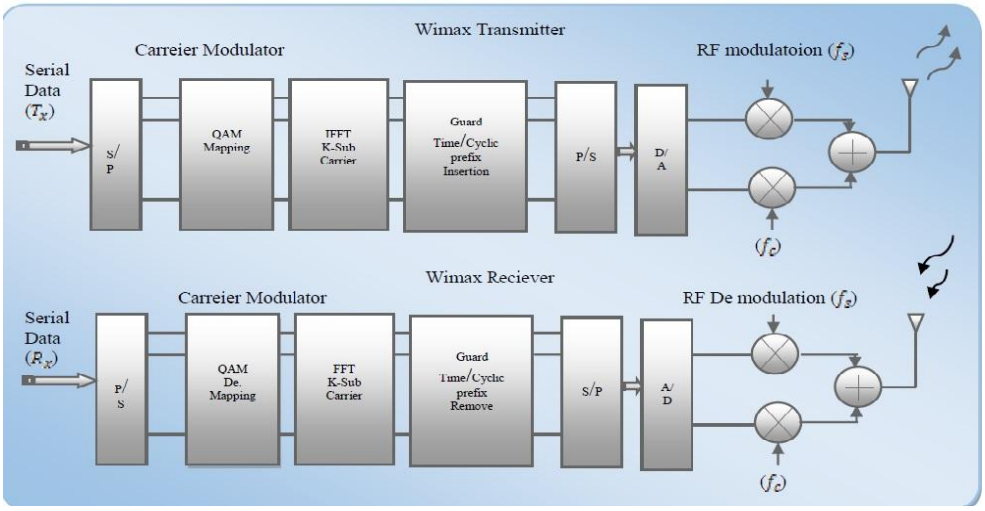
1. INTRODUCTION

IEEE 802.16 standard describes the air interface for immobile Broadband Wireless Access (BWA) systems to be used in WLAN (Wireless Local Area Network) and WMANs (Wireless Metropolitan Area Networks), generally referred to as WiMAX (Worldwide Interoperability for Microwave Access) [1]. WLAN plays as significant role as a complement to the existing or planned cellular networks. The market penetration of WLAN has been extensive due to its easy and low cost deployment, wide interoperability and inherent flexibility. They provide connectivity for slow mobility with high throughput for both indoor and outdoor environments. WiMAX is a non-profit corporation formed by equipment and component suppliers to promote the adoption of IEEE 802.16–compliant equipment by operators of BWA systems [4]. It also ensures the compatibility and interoperability of broadband wireless access equipments. Both these standards use OFDM PHY layer that greatly increase their performance. The WiMAX physical layer (PHY) is based on orthogonal frequency division multiplexing (OFDM), a scheme that offers good resistance to multipath, and allows WiMAX to operate in conditions where physical obstructs are present.

A WiMAX system consists of two major parts:

- A WiMAX base station.
- A WiMAX receiver.

A general block diagram for the transmitter and receiver of WiMAX is shown below:



. **Figure 1: Generalized block diagram of WiMAX transmitter and receiver**

The figure shows WiMAX transmitter and receiver block diagram. In the transmitter, the serial to parallel converted data is mapped in the modulator. Also, the fast Fourier transform (IFFT) realizes the time domain WiMAX signal by modulating each data symbol onto a unique carrier frequency. The remaining steps are A/D and serial to parallel conversions.

2. PROPOSED METHODOLOGY

2.1 Interleaver

Interleaver being one of the sub blocks of WLAN and WiMAX based system. It plays a vital role in improving the performance of Forward Error Correcting (FEC) codes in terms of Bit Error Rate (BER) over wireless channel. Moreover, wireless channel introduces burst errors and hence need of an interleaver becomes essential, as it is an important and powerful technique to combat these burst errors. Basically, interleaver rearranges the code symbols, so as to spread burst of errors into random like errors, which can be corrected by FEC techniques. In view of the various modulation schemes in OFDM based WLAN and WiMAX, multimode block interleaver is the ideal solution for implementation purpose. The major challenges in implementation of the interleaver are reduced access time, lesser occupancy of circuit board and lower power consumption.

The interleaver is defined by a two-step permutation. The first ensures that adjacent coded bits are mapped onto nonadjacent subcarriers. The second permutation ensures that adjacent coded bits are mapped alternately onto less or more significant bits of the constellation thus avoiding long runs of lowly reliable bits. 'd' represents number of columns of the block interleaver which is typically chosen to be 16. Mk is the output after first level of permutation and k varies from 0 to Ncbps -1. S is a parameter defined as:

$$S = \max\{1, Ncpc/2\}$$

Where Ncpc is the number of coded bits per subcarrier [1].

$$m_k = \left[\frac{N_{cbps}}{d}\right](k \,\%\, d) + \left[\frac{k}{d}\right]$$

$$j_k = s \times \left[\frac{m_k}{s}\right] + \left[m_k + N_{cbps} - \left(\frac{d \times m_k}{N_{cbps}}\right)\right] \% s$$

Where % is signify modulo function.

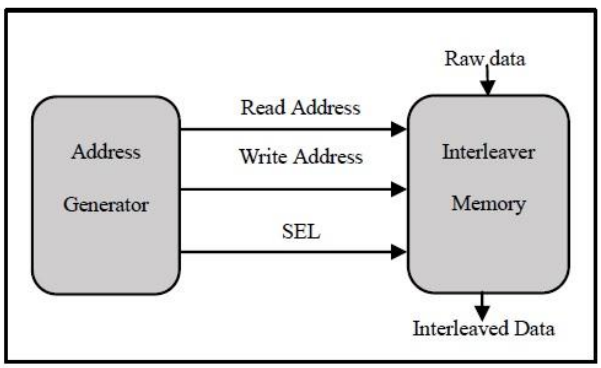


Figure 2: Architecture of WiMAX interleaver

2.2 Implementation of Interleaver for WLAN

2.2.1 Address Generator

The address generator circuit is used to generate (1) write addresses (2) read addresses (3) sel signals. The complete circuit is shown in figure 2.

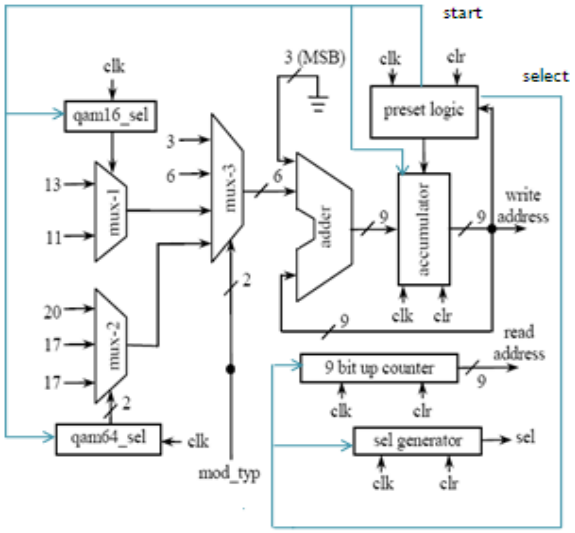


Figure 3: Address generator for WLAN

The circuit consist of stages of multiplexers which implement equal or unequal increments for various modulation schemes as shown in Table 1.

Table 1: Permitted modulation schemes with incremented values

Modulation Type	Increment values	Whether equally spaced or not
BPSK	3	Yes
QPSK	6	Yes
16-QAM	13,11	No
64-QAM	20,17,17	No

Bulk of the circuitry is used for generation of write address. It contains three multiplexers (MUX): MUX-1 and MUX-2 implements the unequal increments required in 16-QAM and 64-QAM whereas MUX-3 routes the outputs received from MUX-1 and MUX-2 along with equal increments of BPSK and QPSK.

The select input of MUX-1 is driven by a T-flip-flop named QAM16_SEL whereas that of MUX-2 is driven by a counter named QAM64_SEL. The two lines of MOD_TYP (modulation type) are used as select input of MUX-3. The 6-bit output from the MUX-3 acts as one input of the 9-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.

The 9 bit up counter is used to generate read addresses for a particular modulation scheme. For example in BPSK, the counter count from 0 to 47 after 47th address it needs to be set to 0 again. We need to provide terminal limit for each modulation scheme to the read address generator. We can provide a select signal from preset which simply defines the upper limit for each modulation scheme. This technique reduces the circuitry and also increases the frequency of operation. The sel generator block is simply a T-FF. It needs some input to change its output as one FEC block is completed. This can be provided by the same select signal from preset block. The start signal from preset logic to the QAM16_SEL and QAM64_SEL block provide correct functional output for 16-QAM and 64-QAM modulation schemes.

2.2.2 Preset Logic as FSM

The preset logic is a hierarchical FSM whose principal function is to generate the correct beginning addresses for all subsequent iterations and is shown in the form of state diagram in figure 4 [41]. This block contains a 4-bit counter keeping track of end of states during the iteration. The FSM enters into the first state (SF) with CLR = 1. Based on the value in MOD_TYPE it makes transition to one of the four possible next states (SMT0, SMT1, SMT2 or SMT3). Each state in this level represents one of the possible modulation schemes. The FSM thereafter makes transition to the next level of states (e.g. S000, S001 and so on) based on the value in the accumulator.

When the FSM at this level reaches to the terminal value of that iteration (e.g.45 in SMT0), it makes transition to a state (e.g. S000) 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. If no changes take place in the values of MOD_TYP, 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 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 input followed by MOD_TYP inputs at any stage of the FSM. With CLR=1 it comes back to SF state irrespective of its current position and there after transits to the desired states in response of new value in MOD_TYPE.

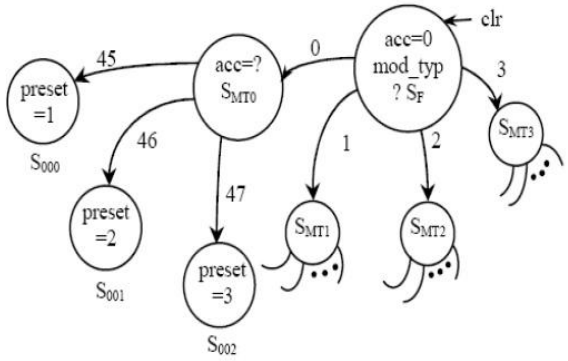


Figure 4: State diagram of preset logic

2.2.3 Interleaver Memory for WLAN

The interleaver memory block comprises of two memory modules (RAM-1 and RAM-2), three MUX and an inverter as shown in figure 5 [41]. 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 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.

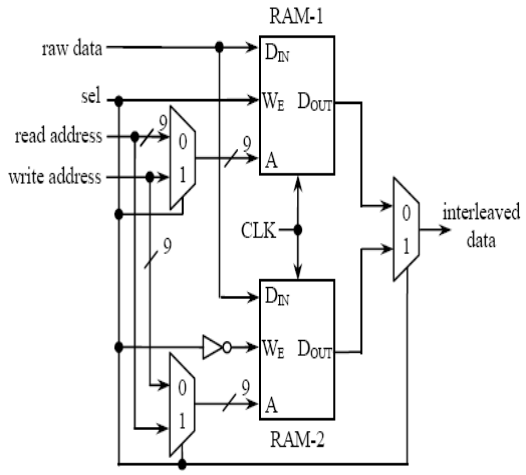


Figure 5: Interleaver memory for WLAN

2.2.4 Implementation of Interleaver for WiMAX for Address Generator

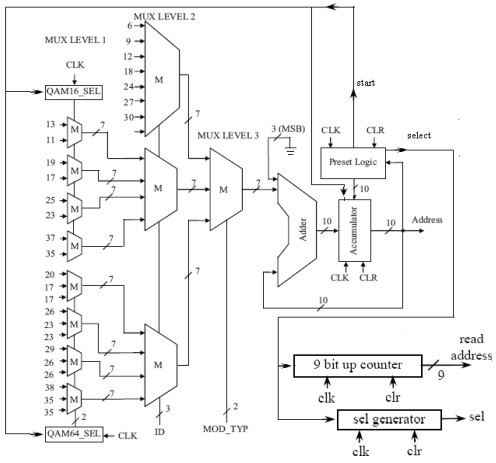


Figure 6: Address generator for WiMAX

The address generator circuit is used to generate (1) write addresses (2) read addresses (3) sel signals. The address generation concept of the proposed block interleaver is described in the form of schematic diagram as shown in figure 6 [40]. This design includes all possible code rates and modulation type permitted under IEEE 802.16e. The circuit

consists of stages of multiplexers which implement equal or unequal increments for various modulation schemes as shown in Table 2.

Table 2: Permitted modulation schemes with incremented values

Modulation	MOD_TYPE	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	26,233,23	NO
		432	X10	29,26,26	NO
		576	X11	38,35,35	NO

As shown in figure 6, the design concept contains three levels of multiplexer (MUX). The first level of MUXs implements the unequal increments required in 16-QAM and 64-QAM. The four interleaver depths of 16-QAM as shown in Table 2 are implemented by the first four MUXs from the top in level 1. The select inputs of these four MUXs are tied together and are driven by a T-flip-flop named QAM16_SEL. Similarly the last four MUXs are for 64-QAM modulation. The select inputs are driven by a MOD-3 counter QAM64_SEL. The second level MUXs basically pick up one inputs based on the values of ID.

The topmost MUX in level 2 implements the eight interleaver depths of QPSK modulation scheme available by concatenation of sub-channels. The second and third MUXs in level 2 are for 16-QAM and 64-QAM respectively. The outputs from level 2 MUXs are routed to the next section by level 3 MUX based on MOD_TYP value. The 7-bit output from the level 3 MUX acts as one input to the 10-bit adder circuit 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. The preset logic is a FSM whose principal function is to generate the correct beginning addresses for all subsequent iterations.

2.2.5 Preset Logic as FSM

The preset logic block is the heart of the address generator for WiMAX interleaver. It is basically a hierarchical FSM and the state diagram as shown in figure 7 [40]. This block contains a 4-bit counter keeping the track of end states during the iteration. The FSM enters into the first state (SF) with CLR=1. Based on the value in MOD_TYP, it makes transition to one of the three possible next states (SMT0, SMT1 or SMT2). Each state in this level represents one of the possible modulation schemes. The FSM thereafter makes transition to one of the next level states (SID0 to SID7 from SMT0, SID0 to SID3 from SMT1 or SMT2) 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 SID0 of SMT0), it makes transition to a state (e.g. S000) 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 ID and MOD_TYP. If no changes take place in the values of ID and MOD_TYP, the FSM will follow the same route of transition and the same set of interleaver addresses will be continually be generated. Any change in ID and MOD_TYP 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 ID and MOD_TYP inputs at any stage of the FSM. With CLR=1 it comes back to SF state irrespective of its current position and there after transits to desired states in response of new values in ID and MOD_TYP.

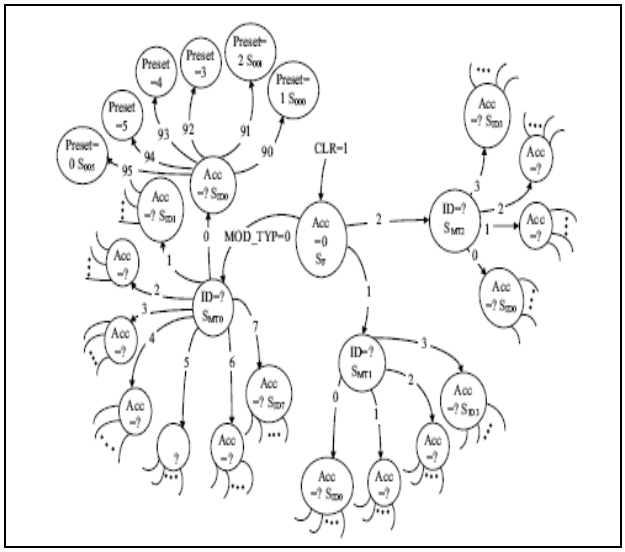


Figure 7: Preset logic as FSM.

2.2.6 Improved Preset Logic as FSM

A careful examination of Table 2 and address generated for different modulation schemes with different code rate shows that a improvement can be made in the preset logic. From Table 2 some interleaver depths (N_{cbps}=288, 384, 432, 576) will generate the same terminal address for both QPSK and 64-QAM modulation schemes. From figure 8 when FSM enters into the first state (SF) with CLR=1, based on the value

in MOD_TYP it makes transition to one of the three possible next states (SMT0, SMT1 or SMT2). Each state in this level represents one of the possible modulation schemes.

The FSM thereafter makes transition to one of the next level states (SID0 to SID7 from SMT0, SID0 to SID3 from SMT1 or SMT2) based on the value in ID. If MOD_TYPE is 64-QAM and ID is X00 (N_{cbps}=288) then this state is directly transferred to the ID-3 state of QPSK modulation as shown in Figure 8. This will generate the same pattern of addresses. If no changes take place in the values of ID and MOD_TYP, the FSM will follow the same route of transition and the same set of interleaver addresses will continually be generated. If at any stage a change in ID and MOD_TYP is needed then the circuit will follow a different path. The circuit responds to CLR followed by ID and MOD_TYP inputs at any stage of the FSM. With CLR=1 it comes back to SF state irrespective of its current position and there after transits to desired states in response of new values in ID and MOD_TYP. Similarly the ID states X01, X10 and X11 (Ncbps=384, 432, 576) for 64-QAM modulation scheme will be directly transferred to the 100, 101and 111 respective ID states for QPSK modulation schemes and will follow the same route as for N_{cbps}=288.

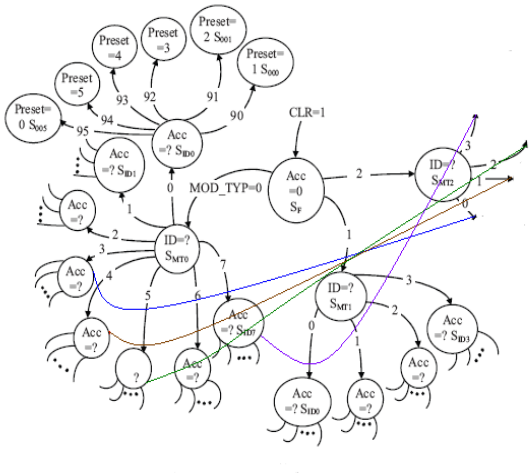


Figure 8: Improved preset logic as FSM

2.2.7 Interleaver Memory for WiMAX

The interleaver memory architecture is as shown in Figure 9 [40]. It comprises of two memory modules RAM-I and RAM-II. When one memory block RAM-I is being used to write over, the other one is used to read from. After completing one FEC block, the memory modules exchange their operations i.e. now RAM-II is used to write over and RAM-I is used to read from. The multiplexers are used to provide correct read and write addresses while inverter is used to provide correct write and read enable signals. The MUX at the output of the memory modules routes the interleaved data stream from the read memory block to the output.

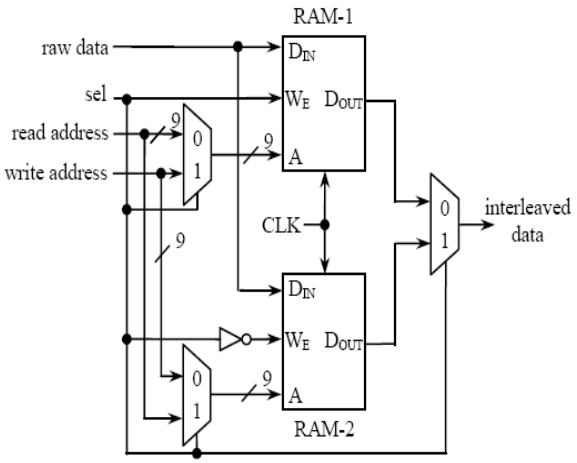


Figure 9: Interleaver memory for WiMAX

3. SIMULATION RESULTS

3.1 Results for WLAN Interleaver

Table 3: Result for WLAN interleaver

FPGA Resources	Slices	LUT	Bonded IOBs	BRAM	Freq.(MHZ)	Slice FF
Ref [41]	61	108	6	2	154.8	31
Proposed work	58	103	6	2	156.5	38

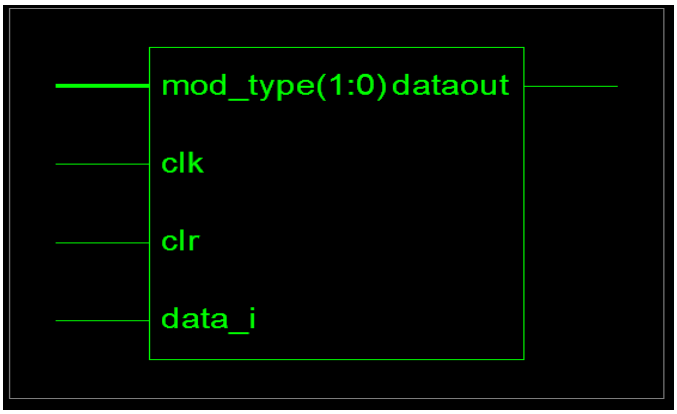


Figure 10: Schematic of possible inputs and outputs

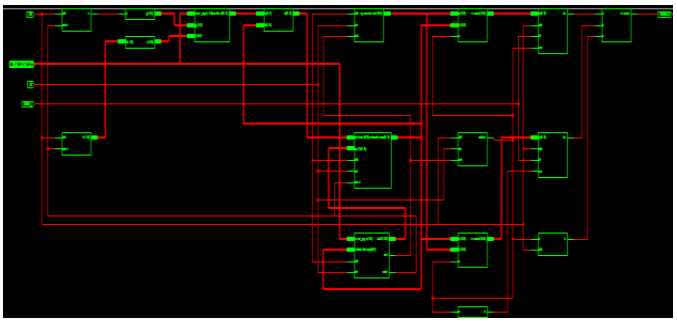


Figure 11: Detailed RTL schematic of WLAN interleaver

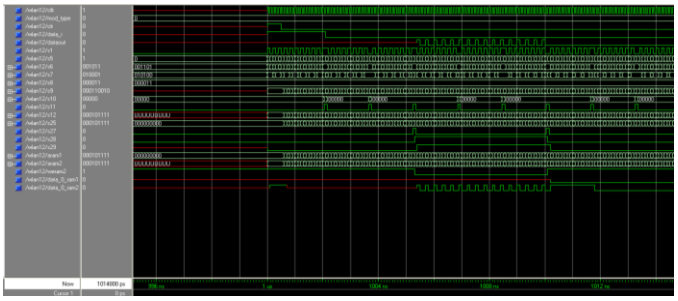


Figure 12: Simulation waveform for BPSK modulation scheme (Ncbps=48)

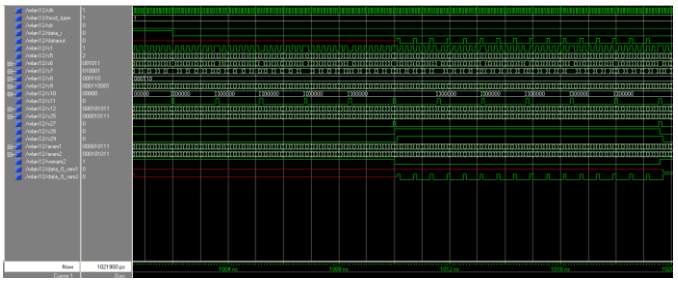


Figure 13: Simulation waveform for QPSK modulation scheme (Ncbps=96)

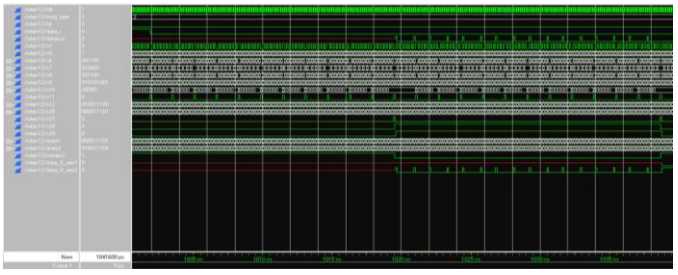


Figure 14: Simulation waveform for 16-QAM modulation scheme (Ncbps=192)

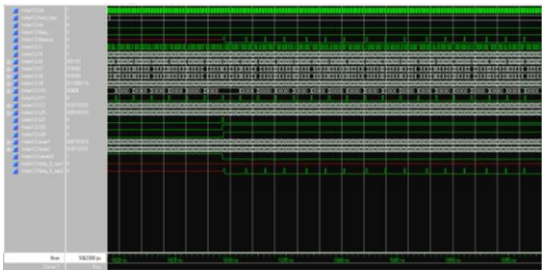


Figure 15: Simulation waveform for 64-QAM modulation scheme (Ncbps=288)

3.2 Result for WiMAX Interleaver

Figure 16: Schematic of possible inputs and outputs

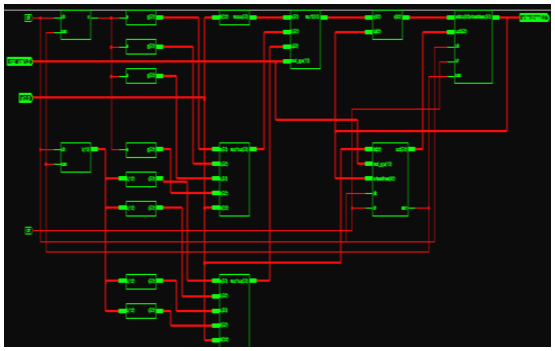


Figure 17: Detailed RTL schematic of WiMAX address generator

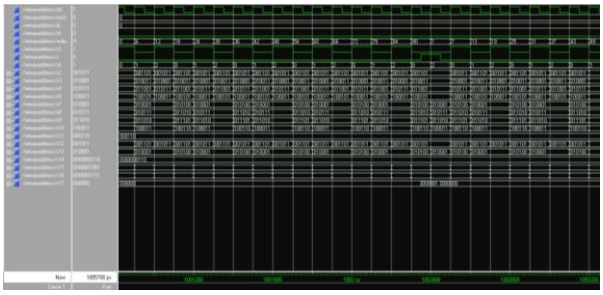


Figure 18: Simulation waveform for QPSK modulation scheme (Ncbps=96)

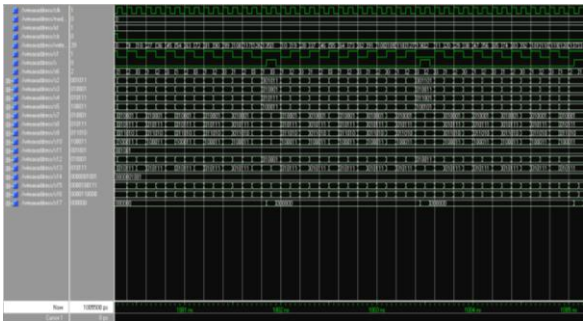


Figure 19: Simulation waveform for QPSK modulation scheme (Ncbps=144)

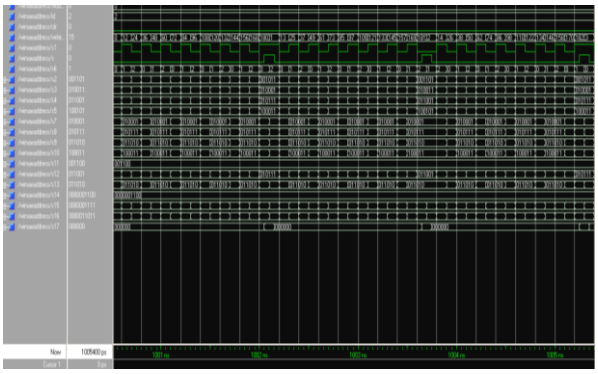


Figure 20: Simulation waveform for QPSK modulation scheme (Ncbps=192)

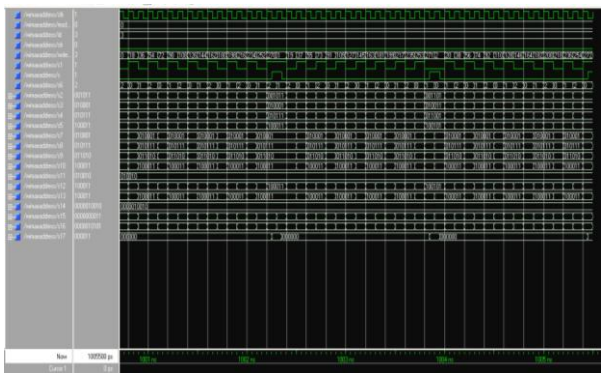
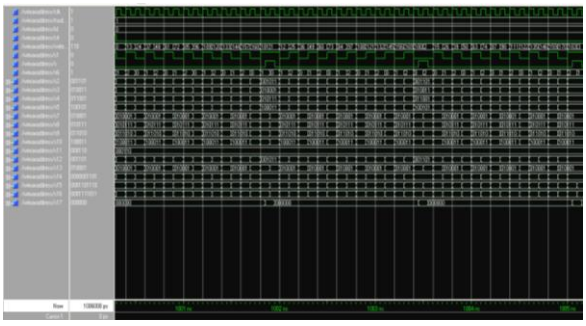


Figure 21: Simulation waveform for QPSK modulation scheme (Ncbps=288)



Figurer 22: Simulation waveform for 16-QAM modulation scheme (Ncbps=192)

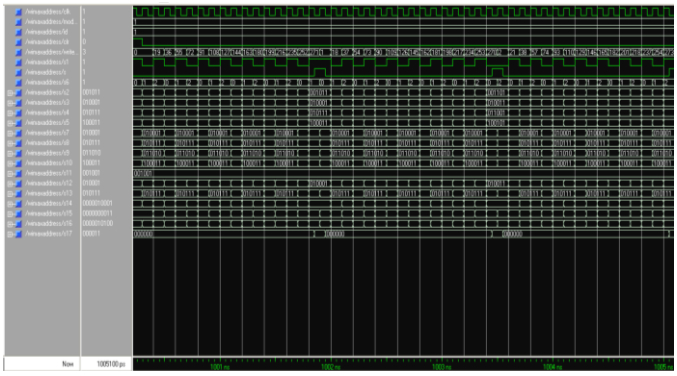


Figure 23: Simulation waveform for 16-QAM modulation scheme (Ncbps=288)

4. CONCLUSION

In this paper the FSM based multimode block interleaver for WLAN is implemented in VHDL on Xilinx Spartan-3 FPGA platform and results are compared with the available literature. The complete interleaver has been divided into two sub modules; address generator and interleaved memory. A modification, in the FSM of address generator for WiMAX standard has reduced the number of slices by 82 and number of slice flip flops by 11 which show a 35.8% and 22% improvement respectively as compared to available literature. Also, using this modified address generator an efficient block interleaver for WiMAX has been implemented with all permitted code rates and different modulation schemes. The circuit parameters and simulation results for WLAN and WiMAX interleaver are obtained using ModelSim XE II software.

In continuation to this work, further research can be done in the area of physical implementation, programmability and improving the memory sub-system partitioning.

5. ACKNOWLEDGEMENT

The authors are grateful to the reviewers for their valuable suggestions. These have helped the treatment to be concise and effective.

REFERENCES

[1] Jha, U. S. and Prasad, R., 2007. OFDM towards fixed and mobile broadband wireless access. London, Artech House Publisher, pp. 1-67.

[2] Konhauser, W., 2006, Broadband wireless access solutions progressive challenges and potential value of next generation mobile networks, International Conference on Wireless Personal Communications, vol 37, May 2006, pp.243-259.

[3] Andrews, J. G., Ghosh, A. and Muhammad, R., 2007. Fundamentals of WiMAX: Understanding broadband wireless networking. Upper Saddle River, NJ (Prentice Hall Communications Engineering and Emerging Technologies Series), Prentice Hall PTR.

[4] Ghosh, A., Wolter, D. R., Andrews, J. G. and Chen, R., 2005. Broadband wireless access with WiMAX/802.16: current performance benchmarks and future potential. IEEE Communication Magazine, vol. 43, pp. 129–36, Feb. 2005

[5] Neubaus, Andre., Freudenberges, Jurgen and Kuhn, Volker, 2007. Coding theory, algorithms, architectures and application. Wiley & sons inc.

[6] Haykin, Simon, Communication systems. 4th edition. New York. John Wiley & sons Inc.

[7] Wicker S. B., 1995. Error control system for digital communication and storage, Englewood Cliffs, Prentice-Hall, Inc.

[8] Tse, D. and Viswanath, P., 2004. Fundamentals of wireless communication. Cambridge University press.

[9] Hanna, S. A., 1993, Convolutional interleaving for digital radio communications, Second IEEE International Conference on Personal Communications: Gateway to the 21stmCentury, vol. 1, pp. 443-447, 1993.

[10] Engels, M., 2002. Wireless OFDM Systems: How to Make Them Work? Springer-Verlag.

[11] Sghaier, A., Ariebi, S. and Dony, B., 2007, A pipelined implementation of OFDM transmission on reconfigurable platforms, CCECE08 Conference, Dec, 2007.

[12] Chang, K., Sobelman, G., Saberinia, E. and Tewfik, A., 2004, Transmitter architecture for pulsed OFDM, in the proc. of the 2004 IEEE Asia-Pacific conf. on circuits and systems, Vol. 2, Issue 6-9, Tainan, ROC, Dec. 2004

[13] Garcia J., 2005. FPGA-Based hardware implementations of OFDM modules for IEEE 802 standards: A common design, Tonantzintla, Mexico, Thesis Report.

[14] Brown, S. and Rose, J., 1996, FPGA and CPLD Architectures: A Tutorial", IEEE design & test of computers, summer 1996, pp. 42-57.

[15] Xilinx, 2004, Spartan-3 FPGA Family: Complete Data Sheet, available at www.xilinx.com.

[16] Shin, M.C. and Park, I.C. 2003, Processor-based turbo interleaver for multiple third-generation wireless standards. IEEE Communications Letters, vol. 7, no. 5, pp. 210–212, 2003.

[17] Ampadu, P. and Kornegay, K., 2003, An efficient hardware interleaver for 3G turbo decoding, in Proceedings of IEEE Radio and Wireless Conference (RAWCON '03), pp. 199–120, August 2003.

[18] Wang, Z. and Li, Q., 2007. Very low-complexity hardware interleaver for turbo decoding. IEEE Transactions on Circuits and Systems II, vol. 54, no. 7, pp. 636–640, 2007.

[19] Asghar, R. and Liu, D., 2008, Very low cost configurable hardware interleaver for 3G turbo decoding, in Proceedings of the 3rd International Conference on Information and Communication Technologies: From Theory to Applications (ICTTA '08), pp. 1–5, Damascus, Syria, April 2008.

[20] Asghar, R. and Liu, D., 2008, Dual standard re-configurable hardware interleaver for turbo decoding, in Proceedings of the 3rd International Symposium on Wireless Pervasive Computing (ISWPC '08), pp. 768–772, Santorini, Greece, May 2008.

[21] Asghar, R., Eilert, J. and Liu, D. Memory conflict analysis and implementation of a re-configurable interleaver architecture supporting unified parallel turbo decoding. Journal of Signal Processing Systems. In press.

[22] Horvath, L., Dhaou, I. B., Tenhunen, H. and Isoaho, J., 1999, A novel, high-speed, reconfigurable demapper-symbol deinterleaver architecture for DVB-T, in Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS '99), vol. 4, pp. 382–385, Orlando, Fla, USA, May-June 1999.

[23] Kim, J. B., Lim, Y. J. and Lee, M. H., 2001, A low complexity FEC design for DAB, in Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS '01), vol. 4, pp. 522–525, Sydney, Australia, May 2001.

[24] Chang, Y. N., 2005, Design of an efficient memory-based DVB-T channel decoder, in Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '05), vol. 5, pp. 5019–5022, Kaohsiung, Taiwan, May 2005.

[25] Afshari, H. and Kamarei, M., 2006, A novel symbol interleaver address generation architecture for DVB-T modulator, in Proceedings of the International Symposium on Communications and Information Technologies (ISCIT '06), pp. 989–993, Bangkok, Thailand, October 2006.

[26] Chang, Y. N. and Ding, Y. C. 2007, A low-cost dual-mode deinterleaver design, in Proceedings of IEEE International Conference on Consumer Electronics (ICCE '07), pp. 1–2, Las Vegas, USA, January 2007.

[27] Wu, Y. W., Ting, P. and Ma, H. P. 2005, A high speed interleaver for emerging wireless communications, in Proceedings of the International Conference on Wireless Networks, Communications and Mobile Computing, vol. 2, pp. 1192–1197, Maui, Hawaii, USA, June 2005.

[28] Asghar, R. and Liu, D. 2009, Low complexity hardware interleaver for MIMO-OFDM based wireless LAN, in Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS '09), pp. 1747–1750, Taipei, Taiwan, May 2009.

[29] Xilinx Inc., Interleaver/De-Interleaver, Product Specification, v5.1, DS250, March 2008. available at www.xilinx.com

[30] Altera Inc., Symbol Interleaver/De-Interleaver Core, Mega core function user's guide, ver. 1.3.0, June 2002.

[31] Lattice Semiconductor Inc., Interleaver/De-Interleaver IP Core, isp Lever Core User's Guide, ipug_61_02.5, August 2008.

[32] Dioni, L and Benedetto, S., 2003, Design of prunable S-random interleaver's, International Symposium on Turbo Codes and Related Topics, pp. 279–289, September 2003.

[33] Ferrari, M., Scalise F. and S. Bellini, 2002, Prunable S-random interleaver, IEEE International Conference on Communications (ICC), vol. 3, pp. 1711–1715,May 2002.

[34] Popovski, P., Kocarev, L. and Risteski, A., 2004. Design of flexible-length S-random interleaver for turbo codes. IEEE Communications Letters, vol. 8, no. 7, pp.461–463, July 2004.

[35] Tell, E. and Liu, D., 2004, a hardware architecture for a multimode block interleaver, ICCSC, Moscow, Russia, June 2004.

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

[37] Asghar, R. and Liu, D., 2009. Low complexity multimode interleaver core for WiMAX with support for convolutional interleaving. International Journal of Electronics, Communication and Computer Engineering, vol.1, no.1 Paris, pp. 20-29, 2009.

[38] Upadhyaya, B. K. and Sanyal, S. K., 2009. 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, pp. 66-68, 2009.

[39] Khater, A. A., Khairy M. M., and Habib, S. E. D., 2009, Efficient FPGA implementation for the IEEE 802.16e interleaver, International Conference on Microelectronics, Morocco, pp. 181-184, 2009.

[40] Upadhyaya, B. K., Misra, I. S. and Sanyal, S. K., 2010, Novel design of address generator for WiMAX multimode interleaver using FPGA based finite state machine, ICCIT 2010, AUST, Dhaka, Dec., 2010.

[41] Upadhyaya, B. K. and Sanyal, S. K., 2011, Design of a novel design of a FSM based reconfigurable multimode interleaver for WLAN Application, Devices and Communication (ICDeCOM) Mesra, IEEE conference20