

---

# An Enhanced area efficient reconfigurable multimode interleaver address generator for multistandard radios with low complexity

---

K.Satya Manuja  
Dept of ECE

**Abstract:** *Developing a reconfigurable transceiver to support multiple protocols seamlessly and efficiently is an extremely tough task. Wireless standards such as wireless local area network (IEEE 802.11a/g) and WiMAX (IEEE 802.16e) incorporate block interleaving technique to overcome the occurrence of burst errors during transmission. Field Programmable Gate Array (FPGA) implementation of floor and modulus (MOD) functions to perform the two step permutation for attaining the new index is quite complex. In this study, the authors propose a low complexity and area efficient reconfigurable architecture for multimode interleaver address generator to support multiple wireless standards. In addition, a novel MOD\_row and MOD\_column circuit are proposed to compute MOD function for row and column counter values, respectively. The proposed address generation circuitry supports BPSK, QPSK, 16-QAM and 64-QAM modulation schemes under all possible code rates. The reconfigurable address generator for various block size and modulation scheme is implemented on Xilinx Spartan XC3S400 FPGA and the functionalities are verified through simulation. The synthesis results of the proposed design shows a reduction of 60% in resource utilization and an improvement of 46% in operating frequency over the existing approaches.*

## 1 Introduction

Multistandard radios utilise distinct transceiver block for each wireless standard to process the corresponding communication signals. Hence, they result in increase of silicon cost and power consumption of the wireless

devices which operates on battery mode. To overcome this problem, the concept of the software defined radio (SDR) has been introduced [1]. Main aim of SDR is to provide a single platform consisting of a hardware layer and number of software layers which incorporates set of radios from different communication standards. The reconfigurable nature of SDR enhances an improvement in the physical layer (PHY) of the existing and upcoming modern wireless system. Mitola [2] has introduced the concept of cognitive radio (CR) in his dissertation. SDR based CRs are fully programmable wireless devices that can sense their environment and dynamically

adapt their transmission waveform, channels and spectrum reuse for dynamic spectrum access [1]. The wideband input signal of a typical CR exhibits the coexistence of radio channels based on different wireless communication standards simultaneously. Due to abundant growth in broadband wireless access wireless based electronic devices operating at high data rate becomes a more challenging task compared with wired last mile access technologies [3]. IEEE has developed standards (IEEE 802.11a/g) which are popularly known as wireless local area network (WLAN) [4, 5] and (IEEE 802.16e) known as mobile WiMAX [6]. Among the PHY layer subsystems, the forward error correction (FEC) block is the one, which consumes more silicon due to addressing/permutation tables used in the conventional approaches for interleaver and deinterleaver. To support the rapid growth in wireless technology, the interleaver address

generator hardware needs to adapt to the different interleaving standards. Hence, there is a scope for researchers to develop a reconfigurable architecture for address generator supporting multiple radio standards which reduces the silicon cost. The channel interleaver is a mandatory block in the PHY of 802.11a/g and 802.16e based transmitter. A channel interleaver for multistandard SDR has to support multiple interleaving functions.

Interleaving plays a vital role in improving the performance of FEC codes in terms of bit error rate. The basic function of interleaver block is to spread the encoded data into a random one. In the conventional approach, the received data is stored row-wise in the memory and read column-wise after applying certain permutations. To achieve more flexibility in the multimode

interleaver, an address generator has to be developed using specialised blocks with reconfigurable capabilities. Upadhyaya et al. [7], have implemented the conventional interleaver on field programmable gate array (FPGA) with reduction in area. In [8], an address generating circuit, for 802.11 interleaver based on the conventional look up table (LUT) method is presented. In [9], a simple technique is developed to implement

the one dimensional interleaver equation in matrix form (2D). Khater et al. [10] carried out the hardware implementation of the address generator for 1/2 code rate WiMAX channel interleaver. Chang [11] implemented an efficient dual mode deinterleaver for both IEEE 802.16e (block deinterleaver) and outer de-interleaver for digital video broadcasting standards. Yu et al. [12] presented a high speed block interleaver/deinterleaver to provide arbitrary column wise permutation by adopting a low-power first in first out bank structure and a simple finite state machine (FSM). In [13], the authors have derived 2D translation of the functions for WiMAX channel deinterleaver

which is more complex for 64-QAM. Upadhyaya et al. have tested an address generating circuit on FPGA for WiMAX multimode interleaver/deinterleaver in [14, 15] and WLAN multimode interleaver in [16] for all permissible code rates and modulation schemes based on FSM. Asghar, et al. [17], proposed a twofold interleaver architecture for different spatial stream application in 802.11n. Zhang et al. [18] had presented, a low complexity architecture for interleaver/ deinterleaver suitable for MIMO application in 802.11a/g/n wireless LAN. Upadhyaya et al. [19] have designed and implemented separate hardware for QPSK, 16-QAM

In this paper, we propose a reconfigurable architecture of multimode interleaver address generator suitable for multistandard SDR by exploring the redundant hardware between different modulation schemes. In addition, we proposed a novel MOD\_row and a MOD\_column circuits to compute MOD function for row and column counter values. Our work differs from the approach

[19] in the style of implementation, which eliminates the redundant hardware resources and the ROM required to store the modulus (MOD) values as well. Moreover we have derived the mathematical expression of address generator for different modulation schemes and the Boolean expression for the

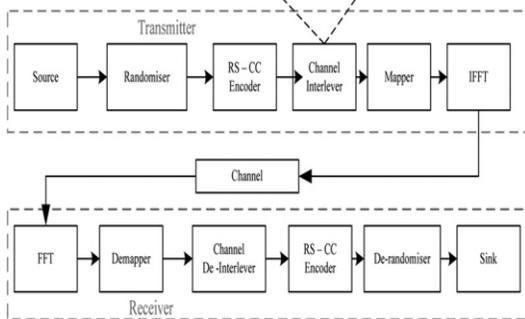
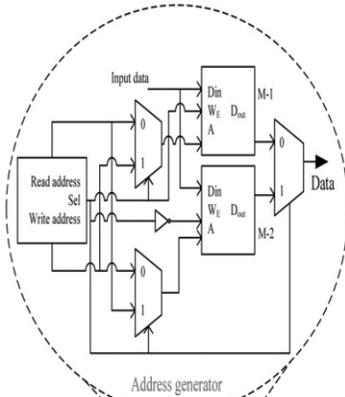
MOD\_column and MOD\_row circuit. A detailed analysis of the proposed architecture is performed and the results are compared with prevailing techniques which show an improvement in terms of logic utilisation and operating frequency. The rest of the paper is organised as follows:

discusses the interleaving techniques in WLAN/WiMAX wireless standards. In Section 3, a reconfigurable algorithm is proposed for multimode interleaver address generations. Section 4 explains the

implementation of reconfigurable architecture for the multimode interleaver. The performance analysis of the proposed architecture is presented in Section 5. Finally, the paper is concluded in Section 6.

## 2 Interleaving in WLAN/WiMAX standards

The wireless LAN standards 802.11a [4], 802.11g [5] and WiMAX standard 802.16e [6] supports the conventional block interleaving schemes. The channel interleaving is a process of rearranging code symbols to reduce the effect of burst error. It processes one block of encoded bits at a time with block size equal to one OFDM symbol depending on the modulation scheme for a specific code rate.



where,  $mn$  and  $kn$  are the first and second level permutation outputs, respectively;  $n$  represents the index of the coded bit in the un-permuted source block which varies from 0 to  $(N_{cbps} - 1)$ ;  $s$  is the parameter defined as  $\max(N_{bps}/2, 1)$ , where  $N_{bps}$  represents the number of coded bits per sub-carrier, that is, 1, 2, 4, 6 for

BPSK, QPSK, 16-QAM and 64-QAM, respectively [13].  $d$

represents the number of columns and  $N_{cbps}$  is the block size corresponding to the number of coded bits per allocated sub-channels. Table 2 shows the theoretical values of  $N_{bps}$ ,  $N_{cbps}$  and number of rows in interleaver memory for different signal constellations [16]. The modulo and floor functions are denoted as % respectively. To generate various permutations values for all modulation schemes and code rates, a MATLAB program is developed using (1) and (2). As  $d$  is chosen as 16.

### 3.1 BPSK/QPSK

For both BPSK/QPSK the value of parameter 's' is 1. Therefore, (2) reduces to  $k_n = m_n + 0$

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

Introducing the 2D array in which  $i$  increments when  $j$  expires, the range of  $i$  and  $j$  as [13]. Now, the interleaver can be realised as a 2D row-column matrix with size  $j \times i$ . Therefore, (2) can be rewritten as

$$k_{n,BPSK/QPSK} = \frac{N_{cbps}}{d} * i + j \text{ for } \forall i \text{ and } \forall j \quad (4)$$

where,  $k_n, j = 0, 1, \dots, (N_{cbps}/d) - 1$  and  $i = 0, 1, \dots, (d - 1)$  represents the interleaver address, row and column values, respectively.

### 3.2 16-QAM

The parameter 's' is 2 for 16-QAM as the number of coded bits per subcarrier is 4. Therefore, (1) and (2) can be rewritten as

$$m_n = \left( \frac{N_{cbps}}{d} \right) * (j \% d) + \left\lfloor \frac{j}{d} \right\rfloor \quad (5)$$

$$k_n = 2 * \left\lfloor \frac{m_n}{2} \right\rfloor + \left( m_n + N_{cbps} - \left\lfloor \frac{d * m_n}{N_{cbps}} \right\rfloor \right) \% 2 \quad (6)$$

Considering the 2D  $i$  and  $j$ , the 2D transformation of interleaver for

**Table 2** Theoretical values of  $N_{bps}$ ,  $s$ ,  $N_{cbps}$  for different signal constellations

Constellation	$N_{bps}$	$s$	$N_{cbps}$	No. of rows
BPSK	1	1	48	3
QPSK	2	1	96	6
16-QAM	4	2	288	18
64-QAM	6	3	384	24

**Table 3** First 32 permutation sample address for four code rates and modulation schemes

Parameters	Interleaver addresses							
$N_{cbps} = 48$ -bits, 1/2 code rate, BPSK	0	3	6	9	12	15	18	21
	24	27	30	33	36	39	42	45
	1	4	7	10	13	16	19	22
	25	28	31	34	37	40	43	46
$N_{cbps} = 96$ -bits, 1/2 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
$N_{cbps} = 192$ -bits, 1/2 code rate, 16-QAM	0	13	24	37	48	61	72	85
	96	109	120	133	144	157	168	181
	1	12	25	36	49	60	73	84
	97	108	121	132	145	156	169	180
$N_{cbps} = 384$ -bits, 2/3 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

16-QAM can be described as:

$$k_{n,16-QAM} = \begin{cases} \frac{N_{cbps}}{d} * i + j & \text{for } i \% 2 = 0 \forall j \\ \frac{N_{cbps}}{d} * i + (j + 1) & \text{for } i \% 2 = 1 \ \& \ \text{for } j \% 2 = 0 \\ \frac{N_{cbps}}{d} * i + (j - 1) & \text{for } i \% 2 = 1 \ \& \ \text{for } j \% 2 = 1 \end{cases} \quad (7)$$

### 3.3 64-QAM

For 64-QAM transmission, the number of coded bits per sub-carrier is 6. Thus, using  $s = 3$  in (1) and (2), it can be rewritten as

$$m_n = \left( \frac{N_{cbps}}{d} \right) * (j \% d) + \left\lfloor \frac{j}{d} \right\rfloor \quad (8)$$

$$k_n = 3 * \left\lfloor \frac{m_n}{3} \right\rfloor + \left( m_n + N_{cbps} - \left\lfloor \frac{d * m_n}{N_{cbps}} \right\rfloor \right) \% 3 \quad (9)$$

The 2D transformation of the interleaver for 64-QAM can be

#### Algorithm 1

```

initialise  $N_{cbps}$ ,  $mod\_type$ , code rate and  $d$ 
for  $j = 0 : 1 : (N_{cbps}/d) - 1$  do
  for  $i = 0 : 1 : d - 1$  do
    if ( EX-OR ( $mod\_type$ ) or  $i \text{ MOD } 2$  or  $i \text{ MOD } 3 ) = 0$ 
       $k_n = \frac{N_{cbps}}{d} * i + j$ 
    elseif (  $i \text{ MOD } 2$  or  $i \text{ MOD } 3 ) = 1$ 
      if OR (  $j \text{ MOD } 2$  or  $j \text{ MOD } 3 ) = 0$ 
        if (  $mod\_type[0]$  ) = 0
           $k_n = \frac{N_{cbps}}{d} * i + (j + 2)$ 
        else
           $k_n = \frac{N_{cbps}}{d} * i + (j + 1)$ 
        endif
      else
         $k_n = \frac{N_{cbps}}{d} * i + (j - 1)$ 
      endif
    elseif (  $i \text{ MOD } 2$  or  $i \text{ MOD } 3 ) = 2$ 
      if (  $\text{MOD } j [1]$  ) = 0
         $k_n = \frac{N_{cbps}}{d} * i + (j + 1)$ 
      else
         $k_n = \frac{N_{cbps}}{d} * i + (j - 2)$ 
      endif
    endif
  endif

```

```

endif
endif
endif
end for
end for

```

**Fig. 2** Proposed reconfigurable algorithm for multimode interleaver address generator

described as

$$k_{n,64-QAM} = \begin{cases} \frac{N_{cbps}}{d} * i + j & \text{for } i \% 3 = 0 \ \forall j \\ \frac{N_{cbps}}{d} * i + (j - 2) & \text{for } i \% 3 = 2 \ \& \ \text{for } j \% 3 = 2 \\ \frac{N_{cbps}}{d} * i + (j + 2) & \text{for } i \% 3 = 1 \ \& \ \text{for } j \% 3 = 0 \\ \frac{N_{cbps}}{d} * i + (j + 1) & \text{for } i \% 3 = 2 \ \& \ \text{for } j \% 3 \neq 2 \\ \frac{N_{cbps}}{d} * i + (j - 1) & \text{for } i \% 3 = 1 \ \& \ \text{for } j \% 3 \neq 0 \end{cases} \quad (10)$$

Using (6), (9) and (12) an reconfigurable algorithm is developed to eliminate the requirement of modulo and floor function as described in Algorithm 1 (see Fig. 2).

#### 4 Proposed reconfigurable address generator architecture

In this section, the architectural details of the proposed reconfigurable multimode interleaver address generator is presented. The architecture of the proposed reconfigurable multimode interleaver address generator for all modulation schemes is shown in Fig. 3. It comprises of a row counter, column counter, selection unit, multiplier and an adder. The output of column counter is compared with  $d$  to generate a fixed count value between  $0$  to  $d - 1$ . Whereas, the row counter is a variable one, which counts different levels between  $0$  to  $(N_{cbps}/d) - 1$  based on specific block size selected by multiplexers (MUXs) M6 and M7. From (6), the hardware required

to implement BPSK and QPSK is quite similar and also the addresses are equally spaced as shown in Table 4. Whereas, the addresses of 16-QAM and 64-QAM are not equally spaced. Therefore, the address generator has to produce two progressive patterns for 16-QAM and three for the 64-QAM modulation scheme. Thus, the design procedure

adapted for QPSK can be extended to 16-QAM and 64-QAM with additional components such as incrementer, decremter and a separate MOD circuit for

row and column counter respectively. The structure shown inside the dashed line in Fig. 3 represents the selection unit for different modulation schemes such as BPSK, QPSK, 16-QAM and 64-QAM block. Each modulation scheme and their corresponding block size are encoded with specific binary values as tabulated in Table 4. To ensure flexibility in generating address for different modulation schemes, MUX based logic is realised to select incremented/decremented values of row counter based on specific  $mod\_type$ ,  $MOD\_row$  and  $MOD\_column$  values. The requirement of MOD functions for column and row count values in the address generation of 16-QAM and 64-QAM are satisfied by the proposed  $MOD\_column$  and  $MOD\_row$  circuits as shown in Figs. 4 and 5, respectively. From

Table 4, the modulation schemes can be categorised into two groups (BPSK, QPSK and 16-QAM, 64-QAM) based on the address spacing in the increment values. In hardware perspective, a simple EX-OR

gate is enough to classify the two sets of modulation schemes. The MUX M1 selects the row counter output directly if the EX-OR of  $mod\_type$  is '0' else it selects the incremented/decremented value of the row counter. Likewise, if the  $MOD\_column$  value is '0', the row counter output is directly selected in M5 for all modulation schemes as given in (6), (9) and (12). If  $MOD\_column$  value is '1', the MUX M3

should select decrement by 1 logic else the output of MUX M2. Whereas, M2 generates two different outputs, that is, increments by 1 for 16-QAM and increments by 2 for 64-QAM based on  $mod\_type$  [0]. While performing 64-QAM, the  $MOD\_column$  value reaches a value of 2. In this case, the MUX M4 selects the row counter output incremented by 1 based on  $MOD\_row$ [1]

= '0' else decrement by 2. In our design, we have reduced the size of the MUXs compared with the logic in [19] by having special kind of select signal. Moreover, the control signals are not generated externally, but generated internally from encoded binary values.

#### 4.1 Working of MOD\_column

As, the number of column (d) is chosen as 16, the column counter values goes between 0 to (d - 1), that is, maximum value of 15.

Hence, a 4-bit MOD\_column circuit is proposed to compute the MOD value of column counter as shown in Fig. 4. with 4-bit input as C3, C2, C1 and C0 from MSB to LSB. The final MOD2 or MOD3 value for the column counter is obtained by performing the successive steps as shown in Table 5.

4.1.1 Logical expression for MOD\_column: In general, for a 2-bit ripple carry adder with two 2-bit input numbers (B1, B0, A1, B0), that is, totally 4-bit and the outputs are 2-bit sum (Sum1, Sum0) and a carry bit (Cout). The general Boolean expression for sum and carry of a 2-bit ripple carry adder is represented as

$$\text{Sum}_1 = \overline{B_1} \cdot \overline{B_0} \cdot A_1 + \overline{B_1} \cdot A_1 \cdot \overline{A_0} + \overline{B_1} \cdot B_0 \cdot \overline{A_1} \cdot A_0 + B_1 \cdot \overline{B_0} \cdot \overline{A_1} + B_1 \cdot \overline{A_1} \cdot \overline{A_0} + B_1 \cdot B_0 \cdot A_1 \cdot A_0 \quad (11)$$

$$\text{Sum}_0 = \overline{B_0} \cdot A_0 + B_0 \cdot \overline{A_0} \quad (12)$$

$$C_{\text{out}} = B_0 \cdot A_1 \cdot A_0 + B_1 \cdot A_1 + B_1 \cdot B_0 \cdot A_0 \quad (13)$$

The four bit inputs (C<sub>3</sub>, C<sub>2</sub>, C<sub>1</sub>, C<sub>0</sub>) are grouped into two 2-bit numbers. To obtain the Boolean expression of the sum (s<sub>1</sub>, s<sub>0</sub>) and

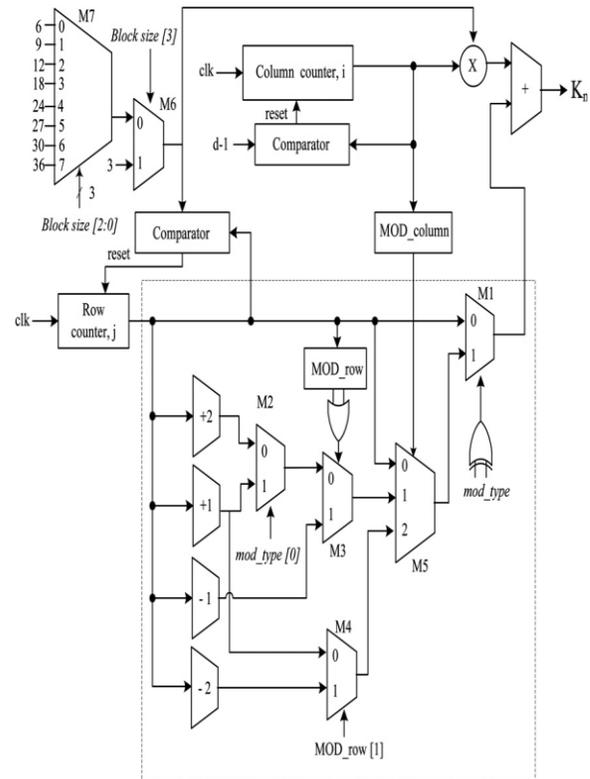


Fig. 3 Proposed reconfigurable architecture of multimode Interleaver address generator supports IEEE 802.11a/g and 802.16e wireless standards

carry (c<sub>0</sub>) of the first 2-bit adder in Fig. 4, the variables (B<sub>1</sub>, B<sub>0</sub>, A<sub>1</sub>, A<sub>0</sub>) in (11), (12) and (13) are replaced with (C<sub>3</sub>, C<sub>2</sub>, C<sub>1</sub>, C<sub>0</sub>).

$$s_1 = \overline{C_3} \cdot \overline{C_2} \cdot C_1 + \overline{C_3} \cdot C_1 \cdot \overline{C_0} + \overline{C_3} \cdot C_2 \cdot \overline{C_1} \cdot C_0 + C_3 \cdot \overline{C_2} \cdot \overline{C_1} + C_3 \cdot \overline{C_1} \cdot \overline{C_0} + C_3 \cdot C_2 \cdot C_1 \cdot C_0 \quad (14)$$

Table 4 Encoded binary values for various interleaver depths and modulation schemes of IEEE 802.11a/g and IEEE 802.16e

Modulation	mod_type [1:0]	Interleaver depths (N <sub>cbps</sub> )	Block size [3:0]	Increment values
BPSK	0 0	48	1 0 0 0	3
		96	0 0 0 0	6
		192	0 0 1 0	12
		288	0 0 1 1	18
QPSK	1 1	96	0 0 0 0	6
		144	0 0 0 1	9
		192	0 0 1 0	12
		288	0 0 1 1	18
		384	0 1 0 0	24
		432	0 1 0 1	27
16-QAM	0 1	480	0 1 1 0	30
		576	0 1 1 1	36
		192	0 0 1 0	13,11
		288	0 0 1 1	19, 17
64-QAM	1 0	384	0 1 0 0	25, 23
		576	0 1 1 1	37, 35
		288	0 0 1 1	20,17,17
		384	0 1 0 0	26, 23, 23
		432	0 1 0 1	29, 26, 26
		576	0 1 1 1	38, 35, 35.

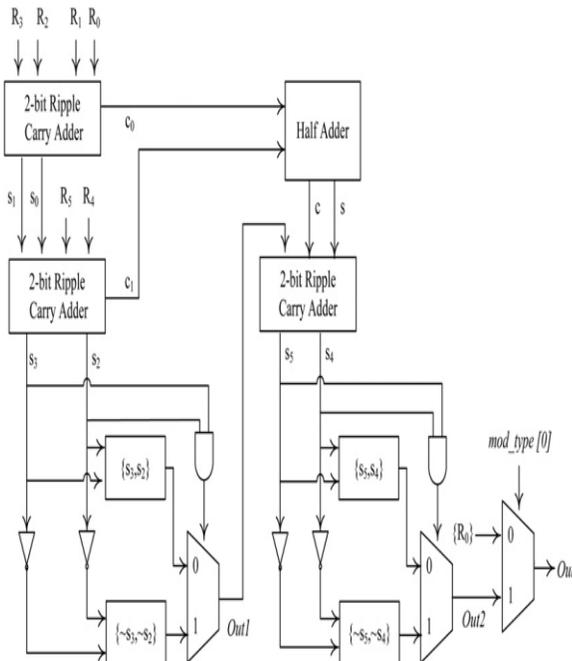


Fig. 5 Proposed architecture of MOD\_row circuit with 6-bit input

Table 5 Working of proposed MOD\_column circuit

STEP	For the count value of 12.	Output	Operation
1	12 = '1100' as 11 00	$C_1 C_0 = 00$ $C_3 C_2 = 11$	Group in to two bits each
2	$C_1 C_0 = 00$ $+ C_3 C_2 = 11$ 0) 11	$s_1 s_0 = 11$ (sum) $c_0 = 0$ (carry)	2-bit ripple carry addition
3	$s_1 s_0 = 11$ $+ 0 c_0 = 00$ 0) 11	MSB is fixed to '0' $s_3 s_2 = 11$ (sum)	2-bit ripple carry addition
4	If $s_3 s_2 = 11$ , complement each bit; Else, pass the same.	$Out1 = 00$ , that is, MOD3 value	Complement
5	$C_0$ represents MOD2 value	$C_0 = 0$ , that is, MOD2 value	
6	Final MUX with the selection line as $mod\_type[0]^a$	$mod\_type[0] = 0$ for 64-QAM $Out = C_0 = 0$	MOD_column Output, that is, $Out = C_0 = 0$

<sup>a</sup>From Table 4, the four modulation schemes (mod\_type) are assigned with binary values, that is,  $mod\_type[0] = 0$  for (BPSK, 64-QAM) and  $mod\_type[0] = 1$  for (QPSK, 16-QAM).

In this section, the performance analysis of the proposed MOD function circuit and the reconfigurable address generator with existing architectures are discussed in detail.

### 5.1 Mod circuit analysis and synthesis results

Several implementation techniques for computing modulo function with varying input and modulo bit width on FPGA and ASIC platform are discussed in [20–22]. In [20], Sivakumar et al. discusses the implementation of X mod m architecture with the MOD values of 3, 5, 6, 7, 9 and 10 in an ASIC chip using 3  $\mu$ m CMOS technology. Since, our proposed MOD circuit for address generation is implemented on FPGA, it is difficult to make a direct comparison with [20].

In [21], Butler, et al. have developed two methodologies for computing x mod z with z fixed as 3, where x is the n-bit input and z is the MOD value. First approach is based on combinational circuits and the second design uses cascade of LUTs. To have better comparison with [21], we have also performed the synthesis of the proposed MOD circuit on Altera Stratix I EP4SE530F43C3NES FPGA using ALTERA Quartus II tool with the input bit width of 8 and 16. From Table 8, it is inferred that the proposed MOD circuit is much faster and consumes fewer resources than the first approach in [21]. It shows an average reduction of 65% in total registers and 28% improvement in operating speed with inputs of 8 and 16 bit width. From the results, it is concluded that for higher input bit width, the proposed circuit provides further reduction in resource utilization and improvement in operating speed. Compared with the second design using LUTs, our proposed design also consumes less logic resources. From Table 9, it is observed that the proposed design achieves an average reduction of 44% in total registers. Since, the LUT approach adapts LUT in

## 5 Performance analysis of the proposed architecture

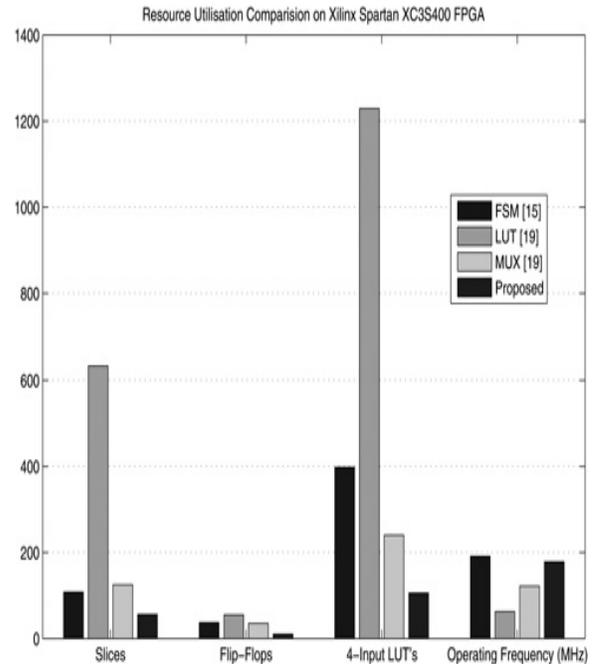
each stage rather than combinational logic, the design able to operate at high speed with increase in latency. The trade-off between the complexity and latency were discussed in [21]. In [22], Gorodecky has developed a circuit for the calculation of  $X \bmod P$  with the Boolean expressions are represented in Reed–Muller XOR polynomial form. It uses XOR and AND operators for the computation. For comparison of the proposed MOD circuit with [22], we have extended the logic behind our proposed MOD 3 circuit to develop a MOD 7 function with input of 10-bit width and the synthesis are performed on both Virtex 7 XC7V285t and Spartan 3 XC3S1000 FPGA. The results are tabulated in Table 10 to show its better performance in terms of LUTs and propagation time.

## 5.2 Reconfigurable address generator analysis and results

The proposed reconfigurable address generator architecture is developed using Verilog HDL [23] and the functionalities are verified using Xilinx ISE. The whole design is implemented on Xilinx Spartan XC3S400 FPGA [24] device to have a better comparison of the proposed work with the existing approaches in [15], [19].

5.2.1 FPGA simulation results: The Xilinx ISE simulation results

for Ncbps = 48-bits, 3/4 code rate, BPSK; Ncbps = 96-bits, 1/2 Code rate, QPSK; Ncbps = 192-bits, 1/2 code rate, 16-QAM and 64-QAM modulation scheme for Ncbps = 384-bits, 2/3 code rate



### 5.2.2 FPGA synthesis results:

In the case of FSM based method [15], for each block size, a unique MUX is used for all modulation schemes which results in increase of hardware and decrease in critical path delay. From Table 11, it is evident that the proposed technique shows reduction of 48% in FPGA slices, 72% in flip flops, 73% in 4-input LUT's and with 6% lesser operating frequency compared with FSM method [15]. Compared with LUT based approach [19], our work eliminates the conventional usage of block RAMs to house the address for each interleaver depths which results in significant reduction in slices (by 91%), in flip flops (by 81%), in 4-input LUT's (by 91%) and the design operates at 1.85 times faster rate. Our work shows an reduction of 55% in terms of logic slices, 71% in terms of flip flops and 56% in terms of 4-input LUT's and operates at 46% faster in terms of operating speed compared with MUX based approach in [19]. For a better comparison of the implementation results, a bar chart of resource utilisation for various methods against the

proposed method is shown in Fig. 7.: In the case of FSM based method [15], for each block size, a unique MUX is used for all modulation schemes which results in increase of hardware and decrease in critical path delay. From Table 11, it is evident that the proposed technique shows reduction of 48% in FPGA slices, 72% in flip flops, 73% in 4-input LUT's and with 6% lesser operating frequency compared with FSM method [15]. Compared with LUT based approach [19], our work eliminates the conventional usage of block RAMs to house the address for each interleaver depths which results in significant reduction in slices (by 91%), in flip flops (by 81%), in 4-input LUT's (by 91%) and the design operates at 1.85 times faster rate. Our work shows an reduction of 55% in terms of logic slices, 71% in terms of flip flops and 56% in terms of 4-input LUT's and operates at 46% faster in terms of operating speed compared with MUX based approach in [19]. For a better comparison of the implementation results, a bar chart of resource utilisation for various methods against the proposed method is shown in Fig. 7.

## 6 Conclusion

In this paper, a reconfigurable address generator architecture of multimode interleaver for IEEE 802.11a/g and 802.16e wireless standards supporting all possible code rates and modulation schemes has been presented. The complete hardware was developed using Verilog HDL and implemented on Xilinx Spartan 3 FPGA. In addition, a novel MOD\_row and a MOD\_column circuits were proposed to compute MOD function for row and column counter values. The synthesis results of the proposed MOD circuit shows significant reduction in resource utilisation and improvement in operating speed compared with existing modulo algorithms. MATLAB simulation results endorse the functionality of the address

generator for various interleaver depths. A detailed analysis of the implementation results has been made to show the performance of the proposed method is improved compared with the existing methods. The proposed work shows an average of 60% reduction in resource utilization and an improvement of 46% in operating frequency compared with the existing approaches. Thus, the proposed reconfigurable address generator architecture can be used to support various modulation schemes in multimode interleaver of multistandard SDR devices.

## 8 References

- 1 Mitola, J., Maguire, G.Q.: 'Cognitive radio: making software radios more personal', IEEE Pers. Commun., 1999, 33, (4), pp. 13–18
- 2 Mitola, J.: 'Cognitive radio – an integrated agent architecture for software defined radio'. PhD thesis, Royal Institute of Technology (KTH), 2000
- 3 Konhauser, W.: 'Broadband wireless access solutions–Progressive challenges and potential value of next generation', Wireless Pers. Common., 2006, 37, (4), pp. 243–259
- 4 IEEE 802.11a: 'Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: high-speed physical layer in the 5 GHz band', 1999
- 5 IEEE 802.11g: 'Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: further higher data rate extension in the 2.4 GHz band', 2003
- 6 IEEE 802.16e: 'Local and metropolitan networks–part 16: air interface for fixed broadband wireless access systems', 2005
- 7 Upadhyaya, B.K., Sanyal, S.K.: 'VHDL modeling of convolutional interleaver–deinterleaver for efficient FPGA implementation', Int. J. Recent Trends Eng., 2009, 2, (6), pp. 66–68
- 8 Sghaier, A., Arieibi, S., Dony, B.: 'A pipelined implementation of OFDM

- transmission on reconfigurable platforms'. Canadian Conf. on Electrical and Computer Engineering, December 2007, pp. 801–804
- 9 Asghar, R., Liu, D.: 'Low complexity multimode interleaver core for WiMAX with support for convolutional interleaving', Int. J. Electron. Commun. Comput. Eng., 2009, 1, (1), pp. 20–29
- 10 Khater, A.A., Khairy, M.M., Habib, S.E.D.: 'Efficient FPGA implementation for the IEEE 802.16e interleaver'. Proc. Int. Conf. Microelectron., Marrakech, Morocco, December 2009, pp. 181–184
- 11 Chang, Y.N., Ding, Y.C.: 'A low-cost dual mode de-interleaver design'. Proc Int. Conf. Consum. Electron., Las Vegas, USA, January 2007, pp. 1–2
- 12 Yu, C., Yen, M.H., Hsiung, P.A.: 'Design of a high-speed block interleaving/deinterleaving architecture for wireless communication applications'. Proc. Int. Conf. Consumer Electronics, Las Vegas, USA, January 2009, pp. 1–2
- 13 Asghar, R., Liu, D.: '2D realization of WiMAX channel interleaver for efficient hardware implementation'. Proc. World Academy of Science and Engineering Technology, Hong Kong, April 2009, pp. 25–29
- 14 Upadhyaya, B.K., Misra, I.S., Sanyal, S.K.: 'Novel design of address generator for WiMAX multimode interleaver using FPGA based finite state machine'. Proc. 13th Int. Conf. Computer and Information Technology, Dhaka, Bangladesh, December 2010, pp. 153–158
- 15 Upadhyaya, B.K., Sanyal, S.K.: 'Novel design of WiMAX multimode interleaver for efficient FPGA implementation using finite state machine based address generator', Int. J. Commun., 2012, 6, (2), pp. 27–36
- 16 Upadhyaya, B.K., Sanyal, S.K.: 'Design of a novel FSM based reconfigurable multimode interleaver for WLAN application'. Proc. Int. Conf. on Devices and Communications, Mesra, India, February 2011, pp. 1–5
- 17 Asghar, R., Dake, L.: 'Low complexity hardware interleaver for MIMO-OFDM based wireless LAN'. IEEE Int. Symp. on Circuits Syst. Circuits and Systems, Taiwan, May 2009, pp. 1747–1750
- 18 Zhang, Z., Wu, B., Zhou, Y.M., et al.: 'Low-complexity hardware interleaver/deinterleaver for IEEE 802.11a/g/n WLAN', Hindawi VLSI Design, 2012, 2012. doi:10.1155/2012/948957
- 19 Upadhyaya, B.K., Sanyal, S.K.: 'Efficient FPGA implementation of address generator for WiMAX deinterleaver', IEEE Trans. Circuits Syst. II, 2013, 60, (8), pp. 492–496
- 20 Sivakumar, R., Dimapolos, N.J.: 'VLSI architecture for computing  $X \bmod m$ ', IEE Proc. Circuit Syst., 1995, 142, (5), pp. 313–320
- 21 Butler, J.T., Sasao, T.: 'Fast hardware computation of  $x \bmod z$ '. Proc. 18th Reconfigurable Architectures Workshop (RAW 2011), Anchorage, Alaska, USA, May 2011, pp. 294–267
- 22 Gorodecky, D.A.: 'Reed-muller realization of  $X \pmod{P}$ ', CoRR, 2015, abs/1504.04773,- arxiv.org
- 23 Michael, D.C.: 'Advanced digital design with the Verilog HDL' (Prentice-Hall, 2010, 2nd edn.)
- 24 Xilinx: 'DS099 Spartan-3 FPGA family data sheet', [http://www.xilinx.com/support/documentation/data\\_sheets/ds099.pdf](http://www.xilinx.com/support/documentation/data_sheets/ds099.pdf)