Analysis and Design of Coding and Interleaving in a MIMO-OFDM Communication System

Zafar Iqbal, Saeid Nooshabadi, *Senior Member*, IEEE, and Heung-No Lee, *Member*, IEEE

Abstract — *Use of Wireless communications for Metropolitan Area Network (MAN) in consumer electronics has increased significantly in the recent past. This paper, presents the performance analysis of four different channel coding and interleaving schemes for MIMO-OFDM communications systems. A comparison is done based on the BER, hardware implementation resources requirement, and power dissipation. It also presents a memory-efficient and low-latency interleaver implementation technique for the MIMO-OFDM communication system. It is shown that among the four coding and interleaving schemes studied, the cross-antenna coding and per-antenna interleaving performs the best under all SNR conditions and for all modulation schemes. It is also the best scheme as far as the hardware resource implication and power dissipation are concerned, which are particularly important in the context of consumer electronics. Next, using the proposed interleaver, a MIMO-OFDM based transmitter employing a double data stream 2×2 MIMO spatial multiplexing system is built*[1].

Index Terms — Channel Coding, Interleaving, MIMO-OFDM, IEEE 802.16, FPGA.

I. Introduction

One of the fastest growing areas of consumer electronics is multimedia applications based on Wireless communications for Metropolitan Area Network (MAN) [1]-[5]. It is a rapidly evolving field with ever increasing data rates to support consumer's demands for new features, advanced functionality, and services for multimedia content provision. Orthogonal frequency division multiplexing (OFDM) with multiple-input multiple-output (MIMO) feature is mainly used in the standard for high speed data communications Worldwide Interoperability for Microwave Access (WiMAX) [6].

In the recent past, MIMO-OFDM has been studied at the algorithmic, system design and implementation levels for consumer [1]-[5], and other wireless systems [7], [8].

The forward error correction (FEC) mechanisms play an important role in the performance of MIMO-OFDM systems. One aspect of the MIMO-OFDM system that has not been investigated adequately is the effect of using different combinations of the convolutional encoder and interleaver on the system performance. Yu et al. [1] adopted per-antenna coding (separate encoder for each data stream) with cross-antenna interleaving (combined interleaving for all the data streams), and Haene et al. [8] used cross-antenna coding (combined encoder for all the data streams) with cross-antenna interleaving. On the other hand, Boher et al. [9] employed per-antenna coding with per-antenna interleaving (separate interleaving for each data stream), while Muller-Weinfurtner [7] used cross-antenna coding with per-antenna interleaving. However, cited works focus on a specific FEC mechanism and do not compare their schemes with other alternatives among the four possible schemes. In this paper, to the best of authors' knowledge, for the first time, the performance and computational complexity of the four different convolutional encoding and interleaving schemes are analyzed. It shows that the cross-antenna convolutional coding with per-antenna interleaving is superior to the other schemes in terms of bit error rate (BER). It will also show that the hardware implementation of per-antenna interleaving systems demand the least amount of resources for the same processing rate. In addition, this paper presents an efficient interleaver design for IEEE 802.16 system on FPGA with a focus on the four different FEC schemes presented by Iqbal and Nooshabadi [10]. The goal is to achieve minimum memory usage, faster interleaving, and increased speed of the overall system, while maintaining the best BER performance.

The paper is organized as follows. Section II presents an overview of the MIMO-OFDM system. Section III discusses the simulation results and analysis of the four different coding and interleaving schemes. Section IV presents the implementation of the whole MIMO-OFDM transmitter, and puts an emphasis on an innovative design of the encoder and interleaver. Section V presents the resource requirement and power dissipation of the MIMO-OFDM transmitter on an FPGA platform and focuses on the significance of the interleaving scheme choice on the overall system performance. Section VI concludes the paper.

II. System Description

The basic architecture of the OFDM communication system is shown in Fig. 1. The FEC blocks include convolutional encoding, puncturing, and interleaving.

[1] This work was supported by the National IT Industry Promotion Agency of Korea and National Research Foundation of Korea (NRF) grant funded by the Korean government (MEST) (Do-Yak Research Program, No. 2011-0016496 and Haek Sim Research Program, No. 2011-0027682).

Z. Iqbal and H.-N. Lee are with the School of Information and Communications, Gwangju Institute of Science and Technology, Gwangju 500-712, South Korea (e-mail: {zafar, heungno}@gist.ac.kr).

S. Nooshabadi, is with the Dept. of Electrical and Computer Engineering, Michigan Technological University, Houghton, MI 49931-1295, USA. (e-mail: saeid@mtu.edu).

Contributed Paper
Manuscript received 07/01/12
Current version published 09/25/12
Electronic version published 09/25/12.

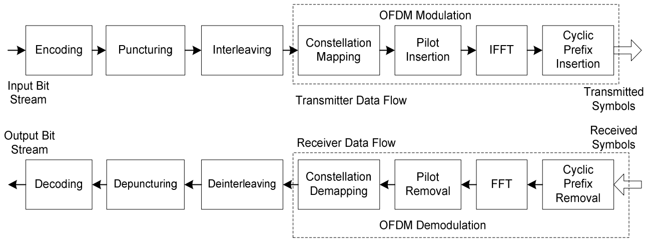


Fig. 1. OFDM communication system

The input bit stream is first encoded using punctured convolutional codes with constraint length K=7, and then interleaved to leverage frequency diversity. This is followed by constellation mapping which is BPSK, QPSK, 16-QAM, or 64-QAM depending on the signal-to-noise ratio (SNR) at the receiver. Next, the symbols are assembled, pilot symbols, and null symbols are inserted. A 256-point IFFT forms the OFDM symbol with 192 data, 8 pilots, and 56 null subcarriers forming the frequency guard bands [6]. The IFFT block computes a 256-point IFFT to form an OFDM symbol. This is the most computationally complex part of the system. A cyclic prefix (CP) is inserted at the start of every OFDM symbol to avoid inter-symbol interference in the case of any delay at the receiver. CP is the end fraction (T_g) of the OFDM useful symbol period (T_b) that is copied to its beginning and is used to collect multipath while maintaining the orthogonality of the tones. CP varies between 1/4, 1/8, 1/16, and 1/32 depending on the bandwidth used, which can vary from 1.5 to 28 MHz. The completed symbol corresponding to 320 points is then transmitted over the channel.

For the analysis and implementation in this paper, four different FEC schemes of double data stream MIMO systems are used, which are categorized as follows. Details of these schemes have been discussed by Iqbal and Nooshabadi [10].

1. Case 1: Cross-antenna convolutional coding with per-antenna interleaving (C-A-P-A), shown in Fig. 2.
2. Case 2: Per-antenna convolutional coding with per-antenna interleaving (P-A-P-A), shown in Fig. 3.
3. Case 3: Cross-antenna convolutional coding with cross-antenna interleaving (C-A-C-A), shown in Fig. 4.
4. Case 4: Per-antenna convolutional coding with cross-antenna interleaving (P-A-C-A), shown in Fig. 5.

In all these cases, the input data is first encoded using a convolutional encoder followed by puncturing. For this analysis, a coding rate of 1/2 is used for BPSK modulation, while coding rate of 3/4 is used for QPSK, 16-QAM, and 64-QAM. Next step is interleaving, which is implemented using a block interleaver, whose size varies according to the modulation scheme used and the system configuration [6]. The receiver performs these functions in reverse order to retrieve the data as shown in Fig. 1. A memoryless AWGN channel and an ideal channel gain of unity for each subcarrier are used, which eliminates the need for channel estimation and carrier recovery.

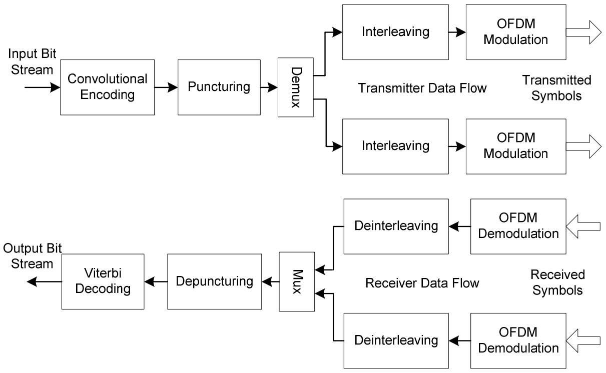


Fig. 2. Cross-antenna coding with per-antenna interleaving

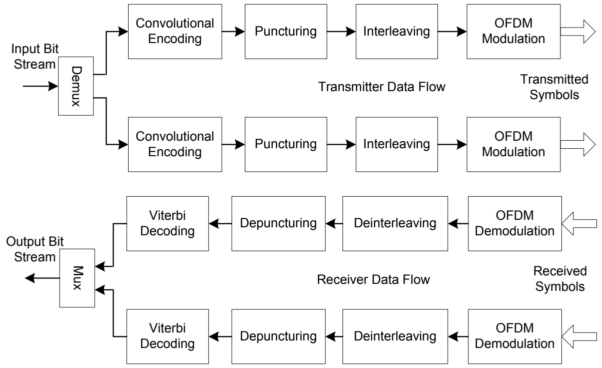


Fig. 3. Per-antenna coding with per-antenna interleaving

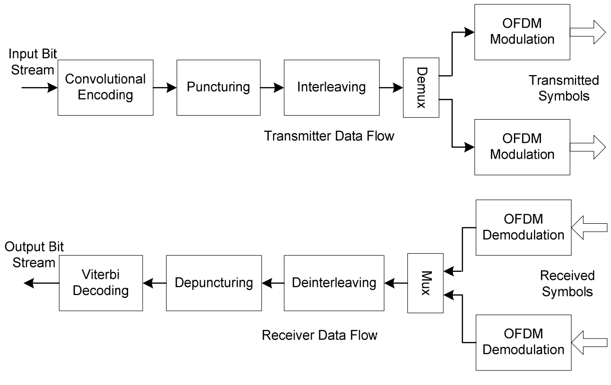


Fig. 4. Cross-antenna coding with cross-antenna interleaving

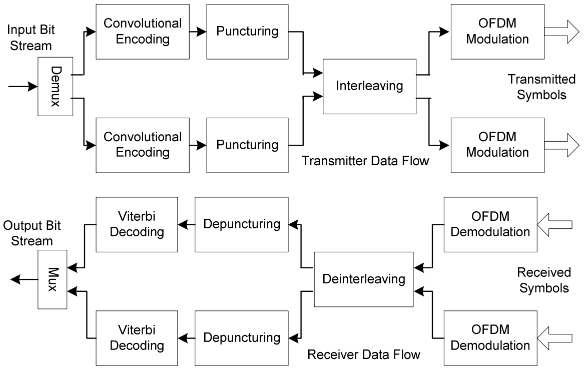


Fig. 5. Per-antenna coding with cross-antenna interleaving

TABLE I
BLOCK SIZES OF THE BIT INTERLEAVER

Mod. Scheme	16 sub-channels	8 sub-channels	4 sub-channels	2 sub-channels	1 sub-channel
			N_{cbps}		
BPSK	192	96	48	24	12
QPSK	384	192	96	48	24
16-QAM	768	384	192	96	48
64-QAM	1152	576	288	144	72

TABLE II
OFDM SYMBOL PARAMETERS USED IN SIMULATION

Parameter	Bandwidth (BW)	Useful Symbol Time (T_b)	OFDM Symbol Period (T_s)	Cyclic Prefix (T_g)
Value	20 MHz	11.11 μs	13.89 μs	1/4 T_b

The encoded data is interleaved by a block interleaver with a block size of N_{cbps}. Table I shows the bit interleaver size as a function of modulation and coding. The interleaver is defined by a two step permutation [6]. The first step ensures that adjacent coded bits are mapped onto nonadjacent subcarriers, while the second step ensures that adjacent coded bits are mapped alternately onto less or more significant bits of the constellation to avoid long runs of low reliable bits. The first bit out of the interleaver maps to the MSB in the constellation [6]. The 16-subchannel system is implemented and hence, the corresponding interleaver block sizes are used.

III. SIMULATION RESULTS AND ANALYSIS

This section, presents the analysis and comparison of the BER performance of the four schemes shown in Fig. 2 to Fig. 5 and their associated complexities for implementation on a reconfigurable FPGA hardware. Table II shows the simulation parameters used.

Fig. 6 shows the fixed-point (16 total and 14 fractional bits) simulation results for all four schemes. Each scheme was simulated over a range of SNRs for all the four types of modulations (BPSK, QPSK, 16-QAM, and 64-QAM) that are used in WiMAX (IEEE 802.16) [6]. As seen, the cross-antenna coded, per-antenna interleaved scheme of Fig. 2 performs best in terms of BER at the higher SNR. The second best scheme is the per-antenna coded, per-antenna interleaved scheme of Fig. 3, while the cross-antenna interleaved schemes of Fig. 4 and Fig. 5 perform worse with a degradation of about 1 to 1.5 dB. So, the performance plots for the pair of schemes with the same interleaver closely follow each other; with the pair with the per-antenna interleaver showing a significant improvement over the pair with the cross-antenna interleaver. Also, it can be seen that using the higher constellation mapping, interleaving plays a bigger role than encoding. As seen from Fig. 6 for the higher SNR values, for a given BER, the performance difference between the per-antenna interleaved and cross-antenna interleaved schemes is wider for the higher constellation mapping.

The important factor in the complexity of the system is its decoding throughput requirement. It is first noted that the data rates for the 2×2 system are doubled with respect to the single data stream system. Next from Fig. 2 to Fig. 5, it can be seen that the interleaver block sizes for the per-antenna interleaved systems are half that of the cross-antenna interleaved systems, which plays a role in improved BER performance of the former, specially for higher constellation systems at the higher SNR values. By the same token, the decoder throughput requirement for the cross-antenna coded system is twice higher than the per-antenna coded system as a single decoder is used to decode two data streams of incoming symbols, making them computationally more complex.

IV. SYSTEM IMPLEMENTATION

To analyze the hardware implication of various coding and interleaving schemes considered in this paper, in this section, the IEEE 802.16-2009 (WiMAX) [6] transmitter is modeled in VHDL and implemented on an FPGA platform. However, before presenting the details of the implementation, a brief presentation of the existing trends in MIMO-OFDM systems implementation is in order.

There have been several FPGA based implementations targeting various functional modules in MIMO-OFDM transceiver [4], [8], [9], [11], [12]. Except for the implementation by Boher et al. [9], that employs cross-antenna coding with cross-antenna interleaving, other works do not cover the design of FEC. However, the work by Boher et al. [9] does not clearly describe the interleaver design and the role it plays in reducing latency. The work by Haene et al. [8] is the only FPGA implementation of MIMO-OFDM with per-antenna coding with cross-antenna interleaving that describes the design of interleaver and deinterleaver. However this design is based on complicated dual-port RAMs that allow concurrent storage and retrieval, which is different from the simple implementation given in this paper, using single-port RAM with a single read or write access at a time.

A. Convolutional Encoder

As shown in Fig. 7, convolutional encoder is implemented using a 6-bit long shift register and *XOR* gates. Two outputs, *X* and *Y* are formed as *modulo2* sums and generated using *XOR* operations as described in IEEE Std. 802.16-2009 [6].

B. Puncturing

Puncturing is implemented using shift registers. For QPSK, *X* and *Y* outputs of the encoders feed two 3-bit shift registers. From each shift register one bit is punctured every 3 clock cycles to create two 2-bit symbols. Each symbol is sent on each data stream for QPSK mapping. For 16-QAM, the same procedure is employed using 6-bit shift registers to *X* and *Y* outputs of the encoders. The puncturing drops two bits from each shift register every 6 clock cycles. Two 4-bit symbols are sent to two data streams for 16-QAM mapping. For 64-QAM, the same procedure is used with 9-bit shift registers as we need 12 bits at the output to generate two 6-bit symbols to send on each data stream. Fig. 7 shows the shift register length for each modulation scheme used. A '1' in the register shows the bit position which is sent to the next block and a '0' shows

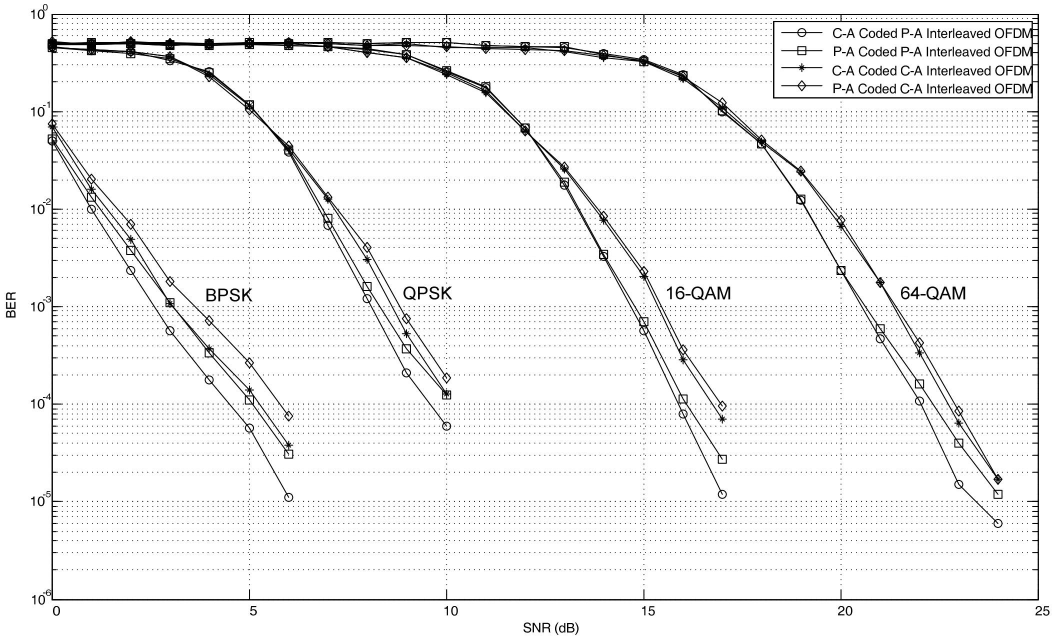


Fig. 6. BER performance of the four systems vs modulation schemes

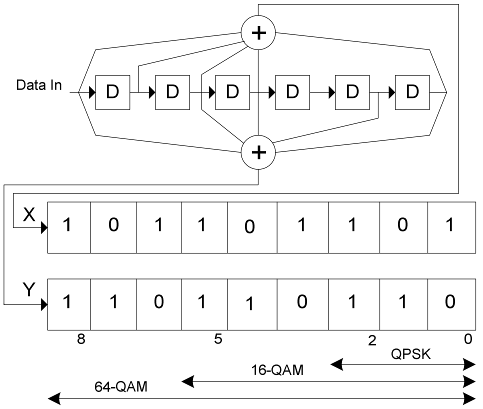


Fig. 7. Convolutional encoding and puncturing block

the punctured bit position. Note that there is no puncturing for BPSK modulation as the coding rate is always rate 1/2.

C. Interleaver

An interleaver design method has been proposed by Chang [12], which employs a divided memory bank architecture for the implementation of interleaver for IEEE 802.16e. In this paper, the interleaver is implemented using the dedicated RAM blocks (BRAM) or distributed RAM (DisRAM) on the FPGA fabric plus a state machine for the address generator for read/write operations. Double buffering technique is used to implement the interleaver to eliminate the delay in the interleaving process. Compared to the work by Chang [12], this method provides a simple write and read logic with no overhead of extra memory usage and complex circuitry [13].

After the first block of symbols is stored in the buffer set (one buffer for each block of symbols), the address generator starts generating read addresses to read data from the buffer set. In the meantime, the second buffer is filled with incoming data and the interleaver will start reading from the second buffer after the first one is read out completely. This technique only incurs an initial latency equal to the incoming time for one block of symbols.

The main problem in implementing the bit-interleaver with multi-port memory using the FPGA on-chip memory is that the synthesizer duplicates the used memory blocks according to the number of ports. In order to avoid this wastage of memory resource, the interleaver is designed in a way that it only uses single-port memory with one-bit write and read to/from each buffer at a time.

Table III shows the buffer sizes in bits, for different interleaving schemes used in the system. The number of buffers increases with the modulation symbol size, so that it can write/read multiple bits simultaneously to/from multiple RAM blocks. Note that the size of interleaver is doubled for cross-antenna systems because a single block interleaver is used to buffer the data for two streams.

TABLE III
BUFFER SIZES FOR DIFFERENT INTERLEAVERS

Mapping Type	BPSK		QPSK		16-QAM		64-QAM	
Interleaver Type	P-A	C-A	P-A	C-A	P-A	C-A	P-A	C-A
Buffer Size	384	768	384	768	384	768	384	768
No. of Buffers	1	1	2	2	4	4	6	6

1) Interleaver for BPSK Mapping

For BPSK mapping, the double buffer interleaver is implemented using a single memory block of double the required size. For example, an interleaver of size 192 is implemented using a buffer of 384 bits for per-antenna interleaving and an interleaver of size 384 is implemented using a buffer of 768 bits for cross-antenna interleaving as shown in Table III. Incoming bits are first stored in RAM until 192 bits are filled and then are read-out. A state machine generates write addresses for the RAM block. For the read the RAM block is partitioned into 12 logical partitions. Partition 0 ≤ k < 12 corresponds to addresses of the form *address%12* = k. Partitions are read out to the end one at a time, sequentially. During the read of one half of the RAM block, the write process continues for the next 192 bits on the other half.

2) Interleaver for QPSK Mapping

For QPSK mapping, the interleaver is implemented using two memory blocks (RAM1 & RAM2) to perform simultaneous writes of two consecutive bits from the data stream. Similar to BPSK, after 192 writes to each RAM the read out starts, while the other half of the RAM blocks are filled. Each RAM block is logically partitioned into 6 partitions. Partition 0 ≤ k < 6 corresponds to addresses of form *address%6* = k. Partitions from RAM1 and RAM2 are read out alternatively to implement the interleaver. That is, partition 0 of RAM1 is read out first completely, followed by the partition 0 of RAM2. This process continues with other partitions from RAM1 and RAM2. The pair of successive reads is used to generate a 2-bit symbol for QPSK mapping.

3) Interleaver for 16-QAM Mapping

For 16-QAM mapping, the interleaver is implemented using four memory blocks (RAM1 to RAM4). Each RAM block has 3 partitions. Partition 0 ≤ k < 3 corresponds to addresses of *address%*3 = k. Data from the k^{th} partitions in RAM1 to RAM4 are read successively. The group of four successive reads is used to generate a 4-bit symbol for 16-QAM mapping.

4) Interleaver for 64-QAM Mapping

For 64-QAM mapping, the interleaver is implemented using six memory blocks (RAM1 to RAM6), which are logically partitioned into two partitions. Partition 0 ≤ k < 2 corresponds to addresses of *address%2* = k. The group of six successive reads is used to generate a 6-bit symbol for 64-QAM mapping. A memory realization of the interleaver structure is shown in Fig. 8 for the 64-QAM mapped data. The structures for the other modulation schemes are similar. As explained above, six memory blocks partitioned into two logical partitions are used. The gray-background indices are RAM addresses generated by the address generator to write data to the RAM blocks whose bit positions in the data stream are shown in the white background. After half of the double buffer for the respective interleaver is filled with data from the input data stream, the address generator generates read addresses with an increment of 2 to read 6 successive locations from each RAM block. The zeroth partition of RAM1 is read first, followed by the zeroth partition of RAM2 and this process continues until the zeroth partition of RAM6 is read. Then the same process continues for the first partition of each RAM block in the same order. The block diagram of the interleaver for 64-QAM having six RAM blocks and an address generator is shown in Fig. 9.

D. Constellation Mapper

Constellation mapping for each scheme is implemented using a ROM which stores the pre-calculated I (real) and Q (imaginary) output values for each input symbol. Two ROMs, one for each I and Q values are used, having a 16-bit output with 14 fractional bits, 1 bit for magnitude, and 1 sign bit. The constellation mapping block for each scheme, implements the mapping technique as explained in the IEEE Std. 802.16-2009 [6], and generates the output I and Q data which is then fed to the IFFT module.

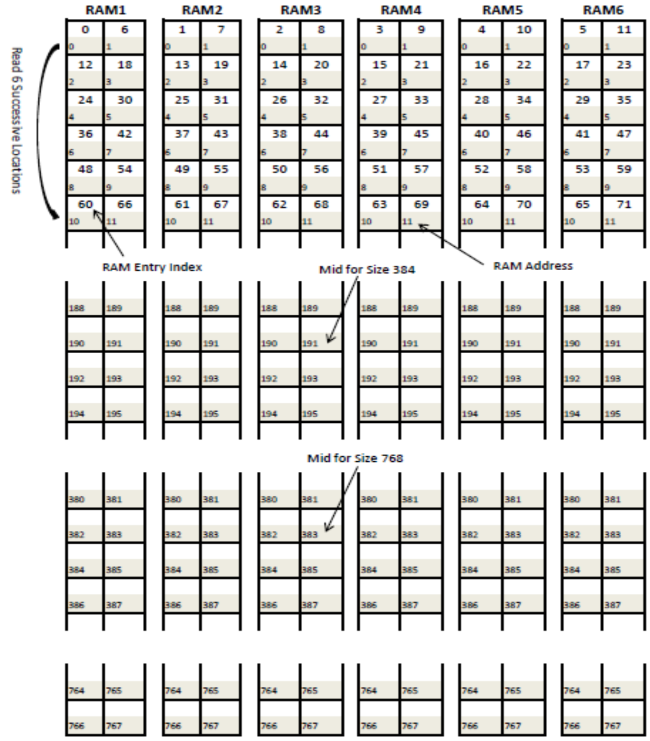


Fig. 8. Interleaver structure in memory for 64-QAM mapping

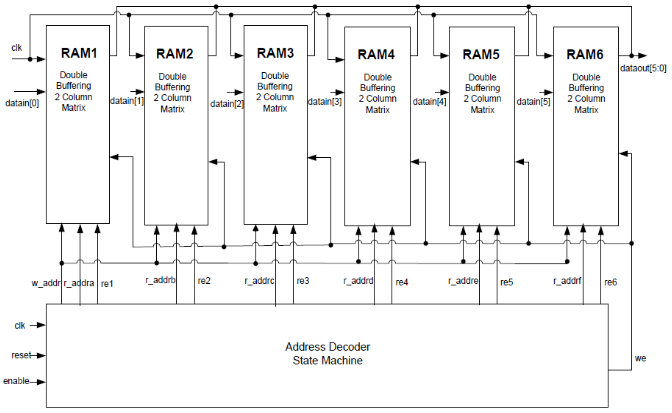


Fig. 9. Interleaver schematic for 64-QAM mapping

E. OFDM Modulator

Using the data in Table II, the OFDM symbol time T_s is given as,

$$T_s = \left(T_b = 11.11\right) + \left(T_g = 2.78\right) = 13.88\,\mu s \ . \qquad (1)$$

In order to satisfy this condition, the OFDM modulator needs to produce 320 (256 IFFT + 64 CP) symbols in 13.88 µs. The corresponding required IFFT module clock speed can then be calculated as,

$$\text{Output Rate} = 320/13.88 = 23.1 \text{ MHz } (43.2 \text{ ns}) \qquad (2)$$

This requires implementation of the blocks in two clock domains. To process the data across two clock domains, the incoming data from the interleaver should be buffered before it is consumed by the IFFT module, as 320 output symbols per every 192 input symbols should be produced. The OFDM modulator block inserts 8 pilot, 1 DC, and 55 null subcarriers, and produces a cyclic prefix of 64 symbols during the input time of 192 input symbols to this block. Thus, the IFFT module clock is *320/192 = 1.667* times faster than the clock rate of the constellation mapper from the previous system. The clock domain separation point is shown in Fig. 10.

The buffer in Fig. 10 is implemented using double buffers for both I and Q inputs to the IFFT module. It is clocked by separate clocks from two clock domains for write and read operations and a buffer of size 384×16-bit is used for each I and Q input. When the first 192 locations of the buffer are filled, this block of data is input to the IFFT module along with the insertion of pilot symbols, DC, and null subcarriers. By the time the next 192 data symbols are written to the buffer, the IFFT module is ready for the next block of data and the process is repeated. The IFFT module is implemented using an FPGA IP core using the pipelined, streaming I/O architecture.

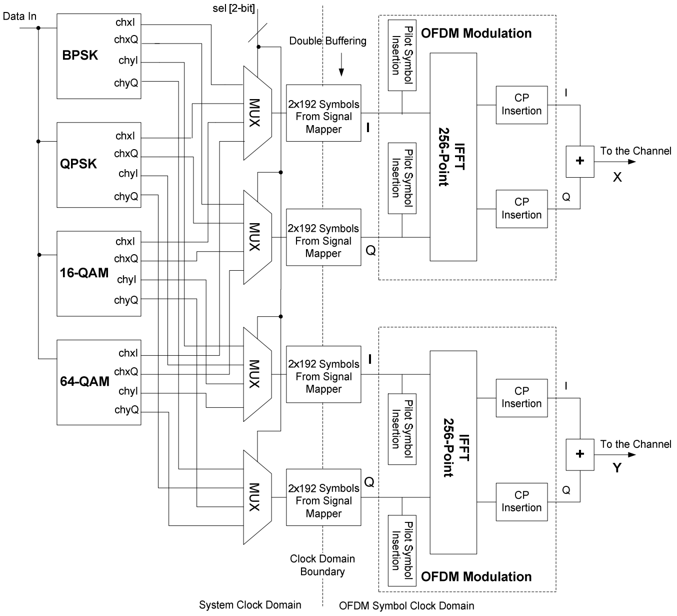


Fig. 10. Overall system architecture

F. System Architecture

Fig. 10 shows the overall system architecture. The blocks labeled as BPSK, QPSK, 16-QAM, and 64-QAM encapsulate one of the four different encoding and interleaving schemes as described in Section II. The output of these blocks is selected using the 2 MSBs of *sel* switch to choose the desired modulation scheme. The chosen output is then fed to the OFDM modulation block where pilot symbols, DC, and null symbols are inserted, IFFT is computed, and cyclic prefix is inserted to produce a 320 point OFDM output symbol.

V. Resource Requirement and Power Dissipation

A. Interleaver Memory Requirement

Table IV shows the RAM resource requirement for the different types of interleavers. Each lookup table (LUT) on FPGA contains 32 bits of RAM and the size of the BRAM is 36 Kb, which can also be partitioned in two separate 18Kb blocks. As it can be seen, the implementation is very efficient in terms of RAM resource requirement if DisRAM extraction method is used during the synthesis of the design. However, if Auto RAM extraction is used, the synthesizer uses BRAM resources to implement interleavers to improve the operating frequency of the overall system, which also saves DisRAM resources. However, in this extraction method, as seen in Table IV, only a small fraction of the instantiated BRAM bits are used. For example, for 64-QAM, for Case 3/4 only about 4% of the BRAM bits are used to implement the interleaver.

TABLE IV
RAM Resource Requirement by Interleavers

Mod. Scheme	System Type	Req. RAM Size	RAM Type Instantiated		RAM Size Instantiated	
			Auto.	Dist.	Auto.	Dist.
BPSK	Case 1/2	768 b	24 LUT	24 LUT	768 b	768 b
	Case 3/4	768 b	1 BRAM	24 LUT	18 Kb	768 b
QPSK	Case 1/2	1536 b	48 LUT	48 LUT	1536 b	1536 b
	Case 3/4	1536 b	1 BRAM	48 LUT	36 Kb	1536 b
16-QAM	Case 1/2	3072 b	96 LUT	96 LUT	3072 b	3072 b
	Case 3/4	3072 b	2 BRAM	96 LUT	72 Kb	3072 b
64-QAM	Case 1/2	4608 b	144 LUT	144 LUT	4608 b	4608 b
	Case 3/4	4608 b	3 BRAM	144 LUT	108 Kb	4608 b

B. Interleaver Resource Requirement

Table V shows the overall resource requirement, for Auto RAM and DisRAM instantiation types, for per-antenna (Case 1/2) and cross-antenna (Case 3/4) interleavers that are used in our system. As it can be seen, the larger size interleavers try to instantiate BRAM instead of DisRAM in order to improve performance and save LUT resources.

From the discussion in Section III and the data in Tables IV and V, it is obvious that the per-antenna interleaver of systems in Case 1/2 has a clear advantage in terms of both BER performance and hardware resources requirement.

Table VI shows a comparison between this implementation of the interleaver and the one in [12]. The method used here, provides a simple write and read logic with no overhead of extra memory usage and complex circuitry.

C. Overall Resource Requirement

Table VII shows the overall resource requirement by the complete system when Auto/DisRAM extraction method is used during the synthesis. As it can be seen, BRAMs are instantiated for the higher size interleavers in order to improve the operating frequency of the system. This results in significant increase in the use of BRAM resources (by more than 3 times) for higher frequency of about 13% and a minute impact on the number of slice logic. This is advantageous when there are enough BRAM resources available. In

DisRAM extraction method, there is no wastage of RAM resources but the number of slice logic requirement increases by a small amount and the operating frequency of the overall system is marginally lower. This method is advantageous when there are less RAM resources and the desired speed of the system could be achieved easily.

TABLE V
INTERLEAVERS RESOURCE REQUIREMENT

Mod. Block	Interleaver Size (bits) – RAM Type	Maximum Frequency (MHz)	RAM Requirement		Slice Logic Requirement	
			No. of BRAMs out of 132	No. of LUTs out of 12480	No. of Slice Regs out of 32640	No. of Slice LUTs out of 32640
BPSK	192-Auto	181.48	0	12	77	170
	192-Dist.	181.48	0	12	77	170
	384-Auto	175.53	1x18 Kb	0	83	161
	384-Dist.	181.02	0	24	84	190
QPSK	384-Auto	172.01	0	24	95	217
	384-Dist.	172.01	0	24	95	217
	768-Auto	177.86	1x36 Kb	0	102	176
	768-Dist.	158.37	0	48	102	236
16-QAM	768-Auto	160.5	0	48	104	216
	768-Dist.	160.5	0	48	104	216
	1536-Auto	169.73	2×36 Kb	0	111	190
	1536-Dist.	169.73	0	96	111	303
64-QAM	1152-Auto	169.19	0	72	110	255
	1152-Dist.	169.19	0	72	110	255
	2304-Auto	154.58	3×36 Kb	0	117	188
	2304-Dist.	152.76	0	144	117	355

It should be noted that due to the large resource requirement by the OFDM modulation module, the overhead of four different types of coding and interleaving systems are very similar to each other. Exception is significantly larger usage of BRAMs for bigger size of interleavers in Case 3/4 system.

D. Initial Latency and Data Rates

Table VIII shows the initial latency of interleaver and the whole transmitter system as well as the output raw data rates that can be achieved with this implementation. The initial latency for each system remains around 54.5 µs and the resulting OFDM symbol period is 13.88 µs as stated in the IEEE Std. 802.16-2009 [6]. Note that data rates reported in Table VIII are for a dual stream 2×2 MIMO and are therefore, twice as high as the specification stated in IEEE 802.16 standard for a single antenna system. The initial latency is very low as compared to the block interleaver system with latency of 2.3 ms, by Crisan et al. [14]. Due to the double buffering technique used in the interleaver design and IFFT computation, there is no latency after the system outputs the first OFDM symbol.

TABLE VI
INTERLEAVER IMPLEMENTATION COMPARISON WITH [12]

Method	Write Operation	Read Operation	Memory Locations	R/W Circuitry
[12]	Needs transposer	Needs LUT	Irregular	Complex
This paper	No transposer needed	No LUT needed	Regular	Simple increment counter

E. Power Dissipation

Table IX shows the power dissipated by transmitters for the four systems implemented. It can be seen that the power dissipation increases with the increase in the modulation symbol size from BPSK to 64-QAM. This is mainly because of the increase in operating frequency and the size of interleavers. The power dissipation also goes up from Case 1 to Case 4 when BRAM is employed in the implementation, revealing the role of memory in the power dissipation. The clocks and memory are the two main contributors to dynamic power consumption. When using DisRAM instead of BRAM extraction synthesis method, the power dissipation reduces significantly because of the smaller size of memory used for the implementation. Table IX shows how the power dissipation with DisRAM, has a significant reduction in power dissipation for Case 3 and Case 4 systems because they use DisRAM instead of BRAM. Case 1 and Case 2 systems use DisRAM in both implementations because of the smaller size of interleavers, so the power dissipation is the same in both Auto RAM and DisRAM extraction methods. It is also worth noting that in DisRAM extraction method Case 3 has lower power dissipation than in Case 2.

F. Discussion

- *Transmitter*: From the results from the previous sections, it can be concluded that the cross-antenna convolutional coding with per-antenna interleaving method presented in this paper has the best BER performance, least memory footprint, and least power dissipation for a MIMO-OFDM transmitter system. The minor impact on the hardware resources is in part due to the memory efficient interleaver design.
- *Receiver*: In this paper, a more complex hardware implementation of full MIMO-OFDM receiver was not done. However, a full end-to-end model of the transmitter receiver pair was used to simulate the results in Fig. 6 for the analysis of the multiple coding and interleaving schemes. The efficient interleaver design that was presented in Section IV-C can also be used in the receiver with the necessary modification. Since the MIMO-OFDM receiver is a very complex hardware, the conclusion drawn on the minor impact of the proposed cross-antenna convolutional coding with per-antenna interleaving method with an efficient interleaver design, equally holds on the receiver side.

TABLE VII
TRANSMITTER RESOURCE REQUIREMENT

RAM Extraction	System	Maximum Frequency (MHz)	RAM Requirement		Slice Logic Requirement		DSP48 Blocks out of 288
			No. of BRAMs out of 132	No. of LUTs out of 12480	No. of Slice Regs out of 32640	No. of Slice LUTs out of 32640	
Auto	Case 1	160.28	3 (2%)	904	7755 (23%)	9480 (29%)	18 (6%)
	Case 2	158.76	3 (2%)	904	8012 (24%)	9705 (29%)	18 (6%)
	Case 3	169.22	10 (7%)	592	7571 (23%)	8706 (26%)	18 (6%)
	Case 4	169.22	10 (7%)	592	7739 (23%)	8808 (26%)	18 (6%)
Dist.	Case 1	160.28	3 (2%)	904	7755 (23%)	9480 (29%)	18 (6%)
	Case 2	158.76	3 (2%)	904	8012 (24%)	9705 (29%)	18 (6%)
	Case 3	147.72	3 (2%)	904	7572 (23%)	9069 (27%)	18 (6%)
	Case4	147.72	3 (2%)	904	7740 (23%)	9171 (28%)	18 (6%)

TABLE VIII
INTERLEAVER AND TRANSMITTER OUTPUT LATENCY AND DATA RATES

Mod. Scheme	System Type	Initial Latency				OFDM Symbol Period (µs)		Raw Data Rate (Mbps)
		Interleaver		Transmitter				
		Clock Cycles	Time (µs)	Clock Cycles	Time (µs)	$T_b + T_g$	T_s	
BPSK	Case 1	192	1.28	1516	54.57	11.11+2.78	13.82	13.88
	Case 2	197	1.31	1526	54.92	11.11+2.78	13.82	13.88
	Case 3	390	2.60	1522	54.79	11.11+2.78	13.82	13.88
	Case 4	395	2.63	1529	55.04	11.11+2.78	13.82	13.88
QPSK	Case 1	581	3.87	4544	54.53	11.11+2.78	13.82	41.66
	Case 2	588	3.92	4558	54.71	11.11+2.78	13.82	41.66
	Case 3	1167	7.78	4548	54.57	11.11+2.78	13.82	41.66
	Case 4	1177	7.84	4562	54.75	11.11+2.78	13.82	41.66
16-QAM	Case 1	1160	7.73	9081	54.49	11.11+2.78	13.82	83.33
	Case 2	1170	7.80	9103	54.62	11.11+2.78	13.82	83.33
	Case 3	2331	15.53	9088	54.53	11.11+2.78	13.82	83.33
	Case 4	2341	15.60	9103	54.62	11.11+2.78	13.82	83.33
64-QAM	Case 1	1739	11.59	13622	54.49	11.11+2.78	13.82	125
	Case 2	1752	11.67	13643	54.57	11.11+2.78	13.82	125
	Case 3	3495	23.29	13622	54.49	11.11+2.78	13.82	125
	Case 4	3505	23.36	13643	54.57	11.11+2.78	13.82	125

TABLE IX
POWER DISSIPATION – ALL SYSTEMS

System	Type	Power Consumption (mW) Auto RAM Extraction				Power Consumption (mW) Distributed RAM Extraction			
		BPSK	QPSK	16-QAM	64-QAM	BPSK	QPSK	16-QAM	64-QAM
Case 1	Clocks	90.23	96.94	105.34	113.45	90.23	96.94	105.34	113.45
	Logic	1.39	6.29	12.27	16.83	1.39	6.29	12.27	16.83
	Signals	1.65	6.32	12.44	18	1.65	6.32	12.44	18
	IOs	0.92	3.32	5.33	6.87	0.92	3.32	5.33	6.87
	BRAMs	3.27	3.67	3.69	3.71	3.27	3.67	3.69	3.71
	DSPs	0.03	0.2	0.37	0.4	0.03	0.2	0.37	0.4
	Total Dynamic Power	97.5	116.74	139.45	159.29	97.5	116.74	139.45	159.29
	Total Quiescent Power	844.6				844.6			
Case 2	Clocks	105.6	113.56	123.88	133.63	105.6	113.56	123.88	133.63
	Logic	1.33	5.99	11.88	16.35	1.33	5.99	11.88	16.35
	Signals	1.69	6.36	12.63	18.34	1.69	6.36	12.63	18.34
	IOs	0.93	3.22	5.39	7.04	0.93	3.22	5.39	7.04
	BRAMs	3.28	3.67	3.69	3.71	3.28	3.67	3.69	3.71
	DSPs	0.03	0.2	0.37	0.44	0.03	0.2	0.37	0.44
	Total Dynamic Power	112.86	133	175.75	179.52	112.86	133	175.75	179.52
	Total Quiescent Power	845.1				845.1			
Case 3	Clocks	102.75	115.29	125.54	142.47	91.9	100.19	111.44	126.52
	Logic	2.36	5.58	10.85	15.42	2.39	5.57	10.83	15.4
	Signals	2.6	5.94	11.19	17.14	2.45	5.52	11.08	16.87
	IOs	1.85	2.29	3.54	4.59	1.91	2.29	3.47	4.46
	BRAMs	3.98	7.25	18.27	36.62	3.66	3.58	3.59	3.6
	DSPs	0.08	0.18	0.33	0.4	0.08	0.18	0.33	0.39
	Total Dynamic Power	113.62	136.52	170.02	216.63	102.39	117.32	140.74	167.24
	Total Quiescent Power	845.5				844.7			
Case 4	Clocks	114.4	118.54	144	161.92	108.11	119.03	135.01	148.73
	Logic	2.26	5.29	10.45	14.88	2.23	5.32	10.4	14.78
	Signals	2.64	5.86	11.52	17.62	2.82	6.59	13.41	20.06
	IOs	2	2.44	3.73	4.81	1.73	2.14	3.26	4.2
	BRAMs	3.98	7.25	18.26	36.62	3.65	3.57	3.59	3.6
	DSPs	0.07	0.17	0.31	0.38	0.07	0.17	0.32	0.38
	Total Dynamic Power	125.34	139.54	188.27	236.23	118.61	136.83	166.03	191.75
	Total Quiescent Power	846				845.4			

VI. Conclusion

As discussed in Sections III and V, the Case 1, Cross-antenna convolutional coding with per-antenna interleaving system (C-A-P-A), wins in all aspects of the system performance such as BER, power dissipation, and hardware resource requirement. Hardware resource requirement is almost the same because of the large size of the OFDM modulation block that takes up most of the system hardware resources. The implemented systems show a consistent improvement in the BER performance and an increase in the hardware resource utilization, power dissipation, and initial latency as the constellation size increases.

This paper also provides an efficient way to design the IEEE 802.16 system for FPGA. A special double-buffering design method is used to implement the interleaver with minimum memory requirement and initial latency. The data rate of the standard is doubled with the help of efficient design methodologies and optimization. This approach can also be used to design other high-speed communication systems or to improve their speeds.

As a further extension, this design can take advantage of the adaptive modulation for grouped subcarriers [5], or as IEEE 802.16 standard [6] supports both Alamouti transmit diversity and spatial multiplexing, one can use the adaptive space-time coding/spatial multiplexing switching techniques [11], [15] in combination with the proposed system to further improve the BER performance.

References

[1] H-G Ryu, “System design and analysis of MIMO SFBC CI-OFDM system against the nonlinear distortion and narrowband interference,” *IEEE Trans. Consumer Electron.*, vol. 54, no. 2, pp. 368 –375, May 2008.

[2] Y. Houand and T. Hase, “New flexible OFDM structure for consumer electronics communication systems,” *IEEE Trans. Consumer Electron.*, vol. 55, no. 1, pp. 191–198, Feb. 2009.

[3] H. Yu, M.-S. Kim, E. young Choi, T. Jeon, and S. Kyu Lee, “Design and prototype development of MIMO-OFDM for next generation wireless LAN,” *IEEE Trans. Consumer Electron.*, vol. 51, no. 4, pp. 1134–1142, Nov. 2005.

[4] J. Soler-Garrido, D. Milford, M. Sandell, and H. Vetter, “Implementation and evaluation of a high-performance MIMO detector for wireless LAN systems,” *IEEE Trans. Consumer Electron.*, vol. 57, no. 4, pp. 1519 –1527, Nov. 2011.

[5] C.-S. Choi, Y. Shoji, and H. Ogawa, “Implementation of an OFDM baseband with adaptive modulations to grouped subcarriers for millimeter-wave wireless indoor networks,” *IEEE Trans. Consumer Electron.*, vol. 57, no. 4, pp. 1541–1549, Nov. 2011.

[6] IEEE Standard for Local and Metropolitan Area Networks – Part 16: *Air Interface for Broadband Wireless Access Systems*, IEEE Std. 802.16-2009, May 2009.

[7] S. H. Muller-Weinfurtner, “Coding approaches for multiple antenna transmission in fast fading and OFDM,” *IEEE Trans. Signal Process.*, vol. 50, no. 10, pp. 2442–2450, Oct. 2002.

[8] S. Haene, D. Perels, and A. Burg, “A real-time 4-stream MIMO-OFDM transceiver: System design, FPGA implementation, and characterization,” *IEEE J. on Sel. Areas in Commun.*, vol. 26, no. 6, pp. 877–889, Aug. 2008.

[9] L. Boher, R. Rabineau, and M. Helard, “FPGA implementation of an iterative receiver for MIMO-OFDM systems,” *IEEE J. on Sel. Areas in Commun*, vol. 26, no. 6, pp. 857–866, Aug. 2008.

[10] Z. Iqbal and S. Nooshabadi, “Effects of channel coding and interleaving in MIMO-OFDM systems,” *IEEE Int. Midwest Sym. on Cir. and Sys. (MWSCAS)*, Seoul, Korea, August 2011, pp. 1–4.

[11] J.-M. Lin, H.-Y. Yu, Y.-J. Wu, and H.-P. Ma, “A power efficient baseband engine for multiuser mobile MIMO-OFDMA communications,” *IEEE Trans. Cir. and Sys. I: Regular Papers*, vol. 57, no. 7, pp. 1779 –1792, July 2010.

[12] Y.-N. Chang, “A low-cost dual-mode deinterleaver design,” *IEEE Trans. Consumer Electron.*, vol. 54, no. 2, pp. 326–332, May 2008.

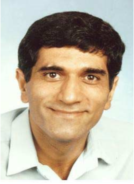
[13] Z. Iqbal, S. Nooshabadi, and H.-N. Lee, “Efficient interleaver design for MIMO-OFDM based communication systems on FPGA,” *IEEE Int. Sym. on Consumer Electron. (ISCE),* Harrisburg, PA, June 2012, pp. 62–66.

[14] N. Crisan, L. C. Cremene, E. Puschita, and T. Palade, “Spectral efficiency improvement for the under-11 GHz broadband wireless access,” in *Int. Conf. on Telecomm., (ICT)*, Athens, Greece, June 2008, pp. 1–6.

[15] W. Nurmi and S. Nooshabadi, “An adaptive space-time coding/spatial multiplexing detector on FPGA,” *IEEE Int. Sym. on Cir. and Sys. (ISCAS)*, Paris, France, May 2010, pp. 4169–4172.

Biographies

Zafar Iqbal received his undergraduate degree in computer engineering from COMSATS Institute of Information Technology, Islamabad, Pakistan, in 2005 and M.S. in information and communications from Gwangu Institute of Science and Technology (GIST), South Korea, in 2010. He was with ZTE Corporation, Shanghai R&D Center, China from 2005 to 2008. Currently, he is a researcher at the INFONET Lab in GIST. His research interests include wireless communications, digital signal processing, and design of VLSI circuits and systems. He was awarded the Korea IT Industry Promotion Agency scholarship for his M.S. study and research.



Saeid Nooshabadi (M’01–SM’07) received MTech and PhD degrees in electrical engineering from the India Institute of Technology, Delhi, India, in 1986 and 1992, respectively. Currently, he is the professor of Computer Systems Engineering, having a joint appointment, with Departments of Electrical & Computer Engineering, and Computer Science, Michigan Technological University, Michigan. Prior to his current appointment he has held multiple academic and research positions. His last two appointments were with the Gwangju Institute of Science and Technology, Republic of Korea (2007 to 2010), and with the University of New South Wales, Sydney, Australia (2000 to 2007). His research interests include VLSI information processing and low-power embedded processors.

Heung-No Lee (S’94–M’99) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of California, Los Angeles (UCLA), in 1993, 1994, and 1999, respectively. From March 1999 to December 2001, he was with the Network Analysis and Systems Department, Information Science Laboratory, Hughes Research Laboratories, Malibu, CA, where he led a number of research projects as the Principal Investigator including traffic modeling for tactical Internet (under the Defense Advanced Research Projects Agency (DARPA) Advanced Technology Office (ATO) Adaptive Signal Processing and Networks (ASPEN) Program), future tactical networking system, capacity analysis for satellite networks using realistic input traffic, and broadband wireless modem. In 2002, he joined the Department of Electrical Engineering, University of Pittsburgh, Pittsburgh, PA. Since January 2009, he has been an Associate Professor in the Department of Information and Communications, Gwangju Institute of Science and Technology, Korea. His current research interests include information and signal processing theories for wireless network and biomedical applications.