FPGA-Based Design of a Pulsed-OFDM System

Kai-Chuan Chang and Gerald E. Sobelman
Department of Electrical and Computer Engineering
University of Minnesota, Minneapolis, MN 55455 USA
{kaichang, sobelman}@umn.edu

***Abstract*—An enhancement to the MB-OFDM system, known as Pulsed-OFDM, has been proposed to reduce the complexity and power consumption of the transceiver without sacrificing performance. In this paper, we describe the detailed FPGA implementation of a complete Pulsed-OFDM transceiver. The resource requirements are given for each of the major blocks for an implementation using a Xilinx Virtex™-4 device. The entire system can be mapped onto a single FPGA chip.**

***Keywords*—Pulsed-OFDM, MB-OFDM, transceiver, FPGA.**

I. INTRODUCTION

A multi-band orthogonal frequency division multiplexing (MB-OFDM) ultra wideband (UWB) system is being considered for the physical layer of the new IEEE wireless personal area network (WPAN) standard, IEEE 802.15.3a [1][2]. The standard is targeting high data transmission rates of 110 Mb/s over 10 meters, 220 Mb/s over 4 meters and 480 Mb/s over 1 meter [3]. The IEEE 802.15.3a transceivers will be used in portable devices, such as camcorders, and laptops, as well as in fixed devices, such as TVs and desktops [4]. Therefore the complexity and power consumption of the transceivers are very important issues to be addressed.

In the multi-band OFDM system, the entire available UWB spectrum is divided into several sub-bands of a given bandwidth. In each sub-band an OFDM symbol is transmitted and then, the system switches to another sub-band [1]. The Pulsed-OFDM system is an enhancement to the MB-OFDM standard which reduces the complexity and power consumption without sacrificing performance. The system uses a novel spectrum spreading technique which up-samples the baseband digital signal within a sub-band to exploit multi-path diversity. Using this spreading technique, the Pulsed-OFDM system provides equal or superior performance in fading channels with much lower complexity [5]. In previous work, we have described the system-level architecture of the transmitter [6] and the receiver [7]. This paper provides the detailed synthesis results for a Xilinx Virtex™-4 xc4vsx35-10ff668 FPGA implementation and demonstrates that the entire transceiver system will fit onto a single chip.

The block diagram of the Pulsed-OFDM transceiver is shown in Fig. 1. At the transmitter, input data bits are first encoded by a rate 1/2 convolutional encoder as described earlier; then a reordering back to sequential format is obtained after passing through the parallel to serial converter. The inter-interleaver block permutes 150 bits around different sub-bands to exploit frequency diversity. An inner-interleaver block permutes 50 bits across data tones of each OFDM symbol within each sub-band to exploit frequency diversity within a sub-band. The mapper assigns the OPSK* mapping and generates in-phase and quadrature outputs for the corresponding input sequences. Next, several blocks apply OFDM modulation to a frame of data inputs, which are then passed onto the upsampler before digital to analog conversion. Finally, the transmitted signal is generated by an analog mixer with the assigned sub-band frequencies. At the receiver, the signals from the antenna first pass through the analog front-end to obtain the in-phase and quadrature digital signals. These signals are then demodulated by the FFT module using either sequential channel processing or parallel channel processing methods. The data is then processed by the diversity combining block to enhance the performance and robustness of the system. The QPSK de-mapper translates the symbol back into bits before being re-permuted by the de-interleavers. The de-inner-interleaver and de-inter-interleaver blocks reorder the data back into the original order. Finally, the Viterbi decoder estimates the corresponding transmitted data.

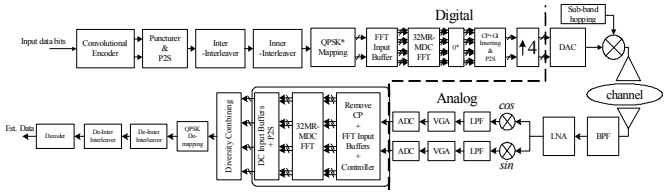

Figure 1: The Pulsed-OFDM transceiver.

II. DESIGN METHODOLOGY AND FLOW

The design methodology used to synthesize each block is shown in the Fig. 2. The algorithm of each block in the system is first verified by using *Matlab* with double precision floating point. After the algorithms are verified, the hardware implementations are obtained by constructing block diagrams in *Simulink*. For block and system simulations, this method enables us to integrate different components and validate their performance by exploring design trade-offs between different sets of parameters. VHDL and/or Verilog code can also be imported into Simulink via the Xilinx System Generator block set, which gives flexibility to the design flow. Simulink and Xilinx System Generator create bit-true and cycle-accurate hardware models which can be programmed into FPGA prototyping boards. The Xilinx Integrated Software

Environment (ISE) is used as the synthesizer in the design flow diagram shown below. ModelSim can also be used to verify the hardware simulation of the blocks by using test vectors generated by the System Generator or HDL test benches. Finally synthesis and performance results of the blocks are reported using ISE, and bitstreams are generated to program the FPGA boards.

III. TRANSMITTER BLOCKS

The synthesis results for each block within the transmitter are summarized in Table 1 with the required numbers of slices, flip-flops, look-up tables (including those used for logic, route-through and shifted registers), DSP48s and the total equivalent gate count of an ASIC implementation. Several additional blocks in the transmitter are presented in the sections below.

Table 1: Synthesis reports for the transmitter blocks.

Block Name	Slice	F.F.	LUTs			IOBs	DSP48s	RAMB16s (Total bit size)	Total Equivalent Gate count
			Logic	R.T.	S.R.				
Encoder	10	11	17	0	0	9	0	0	190
P2S	9	11	8	0	0	8	0	0	150
Inter-Interleaver CU	49	41	71	14	0	22	0	0	946
Inter-Interleaver	70	44	102	19	0	8	0	6 (300)	1181
Inner-Interleaver CU	35	34	59	6	0	20	0	0	728
Inner-Interleaver	53	38	86	6	0	8	0	10 (100)	938
Input Buffer CU	12	12	17	3	0	21	0	0	261
Input Buffer	43	20	68	6	0	25	0	16 (256)	677
MRMDC442 CU	12	11	21	0	0	11	0	0	229
MRMDC442	1329	1198	1471	84	336	186	20	10 (672)	45616
Output Buffer CU	20	18	28	6	0	13	0	0	405
Output Buffer	91	20	164	6	0	194	0	4 (2432)	1367
Conjugation	19	0	38	0	0	135	0	0	345
Upsampler	26	9	41	0	0	81	0	0	638

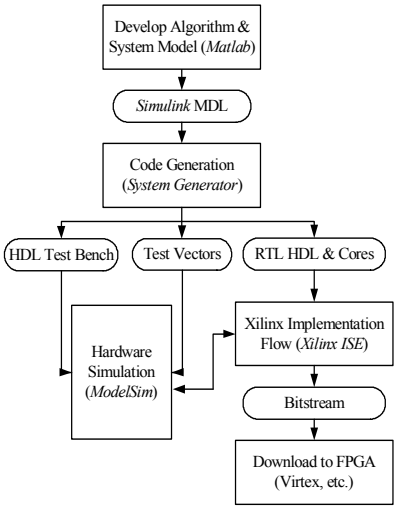

Figure 2: Design methodology and flow diagram.

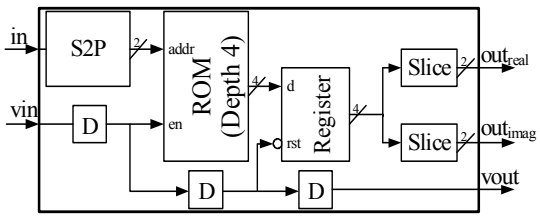

Figure 3: The QPSK* mapper block.

A. QPSK* Mapper Block

The constellation mapping module maps the interleaved bits into QPSK constellations. Since the output of this module is going to the IFFT module and we implement the IFFT with an FFT module by adding conjugating operations at the inputs and outputs of the FFT, we can instead map the interleaved bits to the corresponding conjugate directly, i.e. a QPSK* mapping. The structure of the QPSK* mapper is shown in Fig. 3, where the binary inputs are passed through a serial to parallel converter first to obtain the corresponding address for the ROM. The values of 1101, 1111, 0101, and 0111 are pre-stored in the ROM with addresses 0, 1, 2, and 3 respectively. Registers at the output of the ROM are used to reset output values to zero when the input validity signal, vin, is low. The 2 most significant bits out of register outputs represent the real (in-phase) output values, and the 2 least significant bits represent the imaginary (quadrature) output values. The signal vout indicates the validity of the outputs of the QPSK* mapper. This structure can also be applied to other 1-D or 2-D signal constellation mappings by modifying the serial to parallel converter and the values in the ROM (i.e. 1111, 1101, 0111, and 0101 for QPSK mapping).

B. BMRMDC442 FFT Blocks

As described in [6] and [7], a pipelined 32-point Buffered Mixed-Radix Multi-path Delay Commutator (MRMDC) FFT will be used in the Pulsed-OFDM system due its performance and flexibility. The QPSK* symbols are first processed by the input buffer block before passing onto the MRMDC442 computation elements, as shown in Fig. 4(a). Real and imaginary part buffers store the in-phase and quadrature parts of the input QPSK symbols. A control unit, which is constructed by a 6-bit wide free running counter with an additional finite state machine to generate the output validity signals (vout1 and vout2), groups the input symbols into a group of 4 before FFT computation.

The buffered symbols are then passed onto the MRMDC442 module for FFT computation. A noise analysis was performed on the block to determine the optimal internal wordlengths of the FFT [8]. For the Pulsed-OFDM transmitter with QPSK symbol wordlength equal to Fix_2_0 (fixed-point representation with total wordlength equal to 2 and fractional wordlength equal to 0), the coefficient wordlength of the FFT is set to Fix_12_10 to achieve the required 9-bit FFT resolution in the Pulsed-OFDM system [7]. Fig. 5 shows the average computation error weight (i.e., percentage) per output node due to different coefficient wordlengths with input values +1 or -1 (QPSK* mapping values).

The pipeline MRMDC442 outputs the FFT results in parallel and in digit-reversed order. Therefore, an output buffer block is required to reorder the outputs into the correct sequential and serial form before being processed by the upsampler. The transmitter output buffer block is shown in Fig. 4 (b). The IFFT operation of the transmitter can now be completed by passing the BMRMDC442 output buffer values through the conjugation block, which is simply constructed by using a negate circuit (computes the arithmetic negation of its input) on the imaginary input from the output buffer stage.

C. Upsampler Block

The final stage of digital processing of the Pulsed-OFDM transmitter is the upsampler, which inserts multiple zeros for every data value sampled. The upsampler block for the Pulsed-OFDM transmitter is presented in Fig. 6 with an upsampling rate equal to four. The circuit uses flip-flops to adjust the timing where the MUX switches to the data input sample at the start of the input sample period and then switches to the constant zero after the first input sample. Registers are used at the outputs and the output rate of the upsampler is four times the input rate with an internal delay of 1 clock cycle.

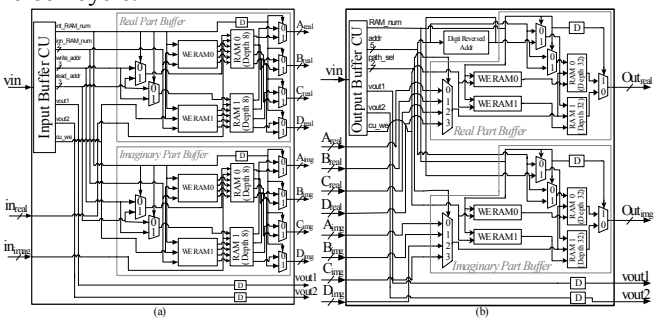

Figure 4: Transmitter (a) input and (b) output buffer blocks.

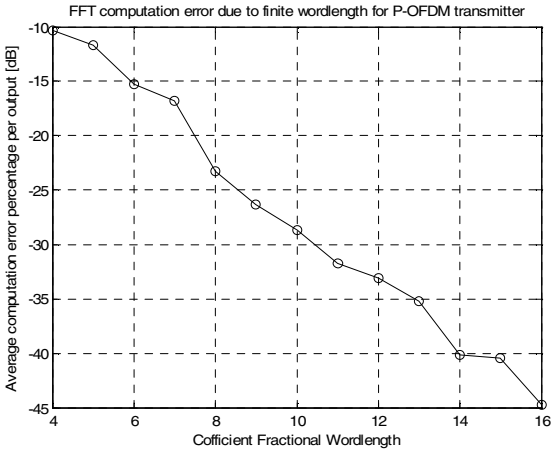

Figure 5: FFT computation error due to finite wordlength

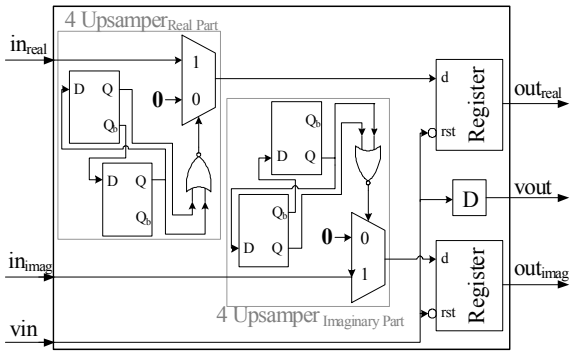

Figure 6: The upsampler block.

IV. Receiver Blocks

Implementation details of the receiver blocks are presented in this section, and the synthesis results of each block are summarized in the Table 2.

A. Receiver Input Buffer Block

The received signal can be grouped into 4 diversity branches which can be combined to deal with frequency selective fading. An implementation of four parallel FFTs will occupy significantly larger area than a single 128-point FFT. Therefore, a receiver input buffer is required to reduce four parallel FFT structures down to one. The structure of the receiver input buffer is shown in Fig. 7(a) with a receiver input control unit, together with real part and imaginary part buffering blocks.

B. Sequential and Parallel Channel Processing FFTs

The MRMDC442 FFT block in the transmitter can be reused to demodulate the received signals with the receiver input buffer block. One single MRMDC442 FFT module decodes four channels in sequence. A performance and noise analysis of the block indicate that the coefficient wordlength of the receiver FFT needs to be set to at least Fix_13_11 to achieve 9 bits of resolution.

The second structure of the receiver FFT processes the channels in parallel, where the first four outputs from the BMRMDC will be FFT results for channel 1 and the second four outputs will be results from channel 2, and so on. In this method, a smaller receiver output buffer block can be used to reduce the memory size required as shown in Table 2. However, sequential structure is chosen due to its flexibility and expandability to processing gain other than K=4 without having to change the structure of the MRMDC442 module.

C. Receiver Output Buffer Block

The final stage of the receiver FFT demodulation section is the receiver output buffer block which stores the FFT computation results from 4 channels and outputs the results in sequential order. The detailed structure of this block is shown in Fig. 7(b), which is similar to the structure of the transmitter output buffer with a control unit constructed from an 8-bit counter. Four channels of FFT computation results are passed onto the diversity combining block at the same time.

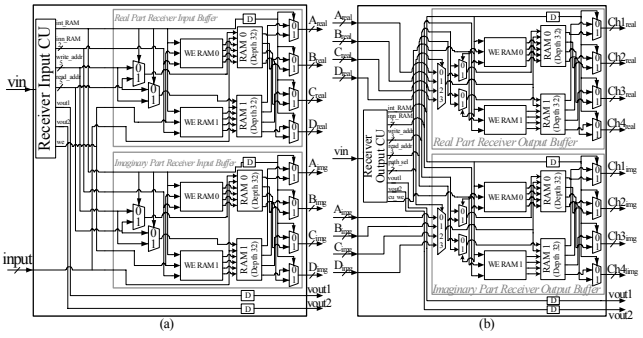

Figure 7: The receiver (a) input and (b) output buffer blocks.

D. Diversity Combining Block

After the FFT demodulation, results from four channels are processed next by the diversity combiner block as described in [7]. The maximal ratio combining (MRC) block is shown in Fig. 8(a), where FFT results and channel estimation of each channel are passed into the block as the inputs. If the sub-optimal equal gain combining (EGC) diversity technique is used instead, the required resources will be reduced tremendously. The structure of the EGC block is also shown in Fig. 8(a), where only a shifter is required instead of the CORDIC divider. As shown in Table 2, the required resources for EGC block are much less than the requirements for the MRC block with the same input wordlength. The outputs of the diversity combiner are passed into the QPSK de-mapper block before the de-interleavers. The de-mapper is constructed by using comparators and a parallel to serial converter.

E. De-interleaver blocks

The de-interleaving operations are applied in this section to put the coded message into its original order before sending it to the decoder. The de-interleaver blocks can be easily constructed by modifying the interleaver blocks' control signals. Due to the similarity of the structures between interleavers and deinterleavers, a combination block can be obtain by inserting MUXs into the control signals, as shown in the Fig. 8(b). The opmode signal sets the operation mode of the interleaver-deinterleaver block. The input signals are inter-interleaved first and then inner-interleaved before being passed to the output during the transmitter mode (opmode equal to 0). During the receiver mode with opmode equal to 1, the input signals are de-inner-interleaved first and then de-inter-interleaved to obtain the original order of the coded message.

Adding up the number of slices, flip-flops, LUTs, RAMB16s and DSP48s used in the transmitter and receiver, we find that these require 39%, 18%, 28%, 57% and 37% of the total amount of those resources on the Virtex^{TM}-4 xc4vsx35-10ff668 device, respectively. Therefore, the entire system will fit onto a single FPGA chip.

V. CONCLUSIONS

FPGA synthesis results of all of the major blocks in the Pulsed-OFDM system have been presented. We showed that the VirtexTM-4 xc4vsx35-10ff668 device has sufficient resources to accommodate the entire system. Many of the blocks could also be used in other OFDM systems with only minor modification. For example, the interleaver and de-interleaver structures can be applied to other rectangular interleaving operations by modifying the RAM sizes and the control unit. Similarly, the constellation mapper and FFT blocks can be adjusted to handle different maps and transform sizes, respectively. Thus, one possible extension of this work would be to design a parameterizable library of generic components which could be used for FPGA implementations of a wide range of OFDM-based systems.

ACKNOWLEDGEMENTS

We thank A. Tewfik and E. Saberinia for many helpful discussions. This work was supported by NSF Grant No. CCR-0313224 and by an equipment grant from Intel Corp.

REFERENCES

[1] Anuj Batra at al, "Multi-band OFDM: merged proposal #1," Merged proposal for the IEEE 802.15.3a standard, IEEE 802.15 work group website, http://grouper.ieee.org/groups/802/15/pub/2003/Jul03/, San Francisco, CA,USA, July 2003

[2] J. Balakrishnan, A. Batra and A. Dabak, "A multi-band OFDM system for UWB communication," IEEE Conference on Ultra Wideband Systems and Technologies 2003, pp. 354-358, Nov. 2003.

[3] Expected performance and attribute criteria approved for P802.15.3a Alt PHY Selection Criteria, IEEE 802.15 work group website, http://grouper.ieee.org/groups/802/15/pub/2003/Jan03/

[4] Summary of the 8 application presentations from the Study Group IEEE 802.15.3a call for applications IEEE 802.15 work group website, http://grouper.ieee.org/groups/802/15/pub/2003/Jan03/

[5] E. Saberinia, J. Tang, A. H. Tewfik and K. Parhi, "Design and implementation of multi-band Pulsed-OFDM system for wireless personal area networks" Proc. of the IEEE Conference on Communications 2004 (ICC'04), Paris, France, June 2004.

[6] K.-C. Chang, G.E. Sobelman, E. Saberinia and A.H. Tewfik, "Transmitter architecture for pusled OFDM," IEEE APCCAS, 6-9 Dec. 2004, pp 693-696.

[7] K.-C. Chang, G.E. Sobelman, E. Saberinia, and A.H. Tewfik, "Implementation of a Multi-band Pulsed-OFDM Transceiver," Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology, 2006.

[8] K.-C. Chang and G.E. Sobelman, "Noise Analysis of Optimized Mixed-Radix Structures for Pulsed-OFDM," IEEE Globecomm, 2006 (in press).

Table 2: Synthesis reports for the receiver blocks.

Block Name	Slice	F.F.	LUTs			IOBs	DSP48s	RAMB16s (Total bit size)	Total Equivalent Gate count
			Logic	R.T.	S.R.				
Rec. Input Buffer CU	27	24	36	8	0	19	0	0	537
Rec. Input Buffer	61	26	102	8	0	65	0	16 (3072)	968
Rec. FFT - sequential	1631	1462	1539	200	364	258	24	10 (782)	51984
Rec. FFT - parallel	1733	1532	1549	206	578	258	24	10 (782)	66372
Rec. Output Buffer CU	27	24	36	8	0	20	0	0	537
Rec. Output Buffer	164	26	306	8	0	325	0	16 (10240)	2312
R.O.B. CU - parallel	19	15	24	6	0	10	0	0	339
R.O.B. - parallel	148	17	280	6	0	325	0	16 (1280)	2027
Diversity Combiner (MRC)	605	801	992	22	13	177	26	0	16883
Diversity Combiner (EGC)	73	1	134	1	0	153	16	0	1626
QPSK de-mapper	28	27	36	3	0	25	0	0	472
De-inner-interleaver	73	38	126	6	0	17	0	10 (400)	1196
De-inter-interleaver	72	44	105	22	0	17	0	6 (1200)	1208
Viterbi Decoder	1552	1536	2153	151	16	12	0	4	34739

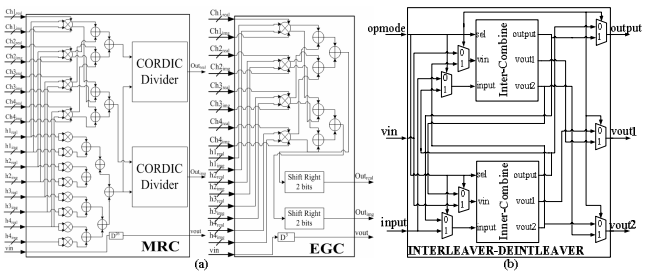

Figure 8: (a) Diversity combiner and (b) inter-deinterleaver combiner.