Indian Journal of Science and Technology, Vol 8(36), DOI: 10.17485/ijst/2015/v8i36/90317, December 2015
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Partial Reconfigurable Implementation of IEEE802.11g OFDM

S. Sivanantham*, R. Adarsh, S. Bhargav and K. Jagannadha Naidu

ASIC Design Laboratory, School of Electronics Engineering, VIT University, Vellore - 632014, Tamil Nadu, India; ssivanantham@vit.ac.in

Abstract

Today's mobile networks are moving towards creating base stations and mobile stations that are compatible with many standards simultaneously. One way of achieving it is to reconfigure and time multiplex the processing resources based on the present necessity. Field Programmable Gate Arrays have become one of the best choices for implementing digital signal processing and Software Defined Radio platforms due to advancements in VLSI over the past few decades. Partial Reconfiguration has regained its importance in the last decade and is the one of the best methodologies to implement an Software Defined Radio. This paper presents an implementation of physical layer specifications of IEEE 802.11g using dynamic partial reconfiguration on FPGA. The Orthogonal Frequency Division Multiplexing (OFDM) Physical layer is implemented with various encoding and modulation schemes to achieve different data rates. The design has been implemented in Xilinx Virtex-5 board.

Keywords: FPGA, IEEE 802.11g, OFDM, Partial Reconfiguration

1. Introduction

Software radios are evolving as flexible all-purpose radios that can implement new and different standards or protocols through reprogramming[1]. Software radio provides seamless services by time sharing the processing resources. It provides the ability to connect with various networks on a single device. Field Programmable Gate Arrays (FPGAs) can offer more bandwidth to many signal processing applications by providing hardware parallelism. FPGAs are more advantageous to implement front end designs than digital signal processors (DSP)[2]. FPGAs physical bit level programming architecture is desired for digital signal processing algorithms as it is suitable for implementing many arithmetic applications while DSPs provide a fixed multiply-and-accumulation (MAC) operation support.

Lately with the rise in the use of Partial Reconfiguration (PR) many such applications have been implemented with improved performance. Though not supported on all FPGA's, the Partial Reconfiguration is slowly gaining fame by the repeated use in demonstrating systems so basic as a basic communication system having multiple encryption schemes each used for a specific set of data or even complicated systems like those requiring video processing. Many of these applications require reconfiguration during run time. On the fly partial reconfiguration implementation of such systems facilitates faster reconfiguration and better resource utilization. Partial Reconfiguration is a very unique way of keeping the critical static processes alive while the system is being reconfigured. In this way the reconfigurable time is considerably reduced. Also the space required to contain the design is reduced as the FPGA holds only the designs necessary for a particular configuration and not all the details of the system. Hence whenever a change in configuration is required a new partial bit stream is loaded into the space and operated.

The OFDM is a multi-carrier system used in various wireless communication standards such as IEEE 802.11a/g, LTE. A multi carrier transmitter involves all of the conceptual challenges to define a configuration management system[3]. IEEE 802.11g system offers various data rates varying from 6 Mbps to 54 Mbps which involve changing the mapping scheme and encoding scheme.

*Author for correspondence

The various mapping schemes are BPSK, 64-QAM and the various coding rates are 1/2, 2/3 and 3/4. Both IEEE 802.16 and IEEE 802.11 systems differ in very few aspects such as the scrambler and an additional Reed Solomon encoder. Hence it is visible that these two standards can be implemented using Partial Reconfiguration. The design is made adaptable to various data rates by loading the bit files whenever necessary. IEEE 802.11g inspired partially reconfigurable OFDM transmitter module has been implemented on FPGA in this project. The design has been implemented on Xilinx Virtex5 board.

2. Overview of IEEE 802.11 Physical Layer

The first Orthogonal Frequency Division Multiplexing (OFDM) system offers the wireless local area network (WLAN) with communication capabilities of 6 Mbps to 54 Mbps. The OFDM consists of a length-127 frame synchronous scrambler, a convolution coder, data interleaver, symbol mapper, Inverse Fast Fourier Transform (IFFT) and cyclic prefix addition blocks[4].

Figures 1 (a) and (b) show the block diagram of the OFDM PHY layer and scrambler. The generator polynomial of the scrambler is given by $s(x)=x^7+x^4+1$. Figure 2 shows the convolution coder with generator polynomials of $g_0 = 133_8$ and $g_1 = 171_8$ with rate R = 1/2, as per the industry standard. Higher rates can be achieved by puncturing.

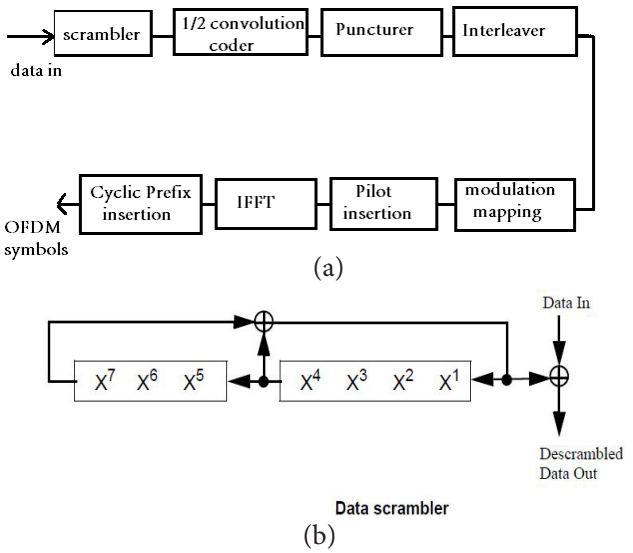


Figure 1. Block diagram of (a) OFDM PHY and (b) Data Scrambler.

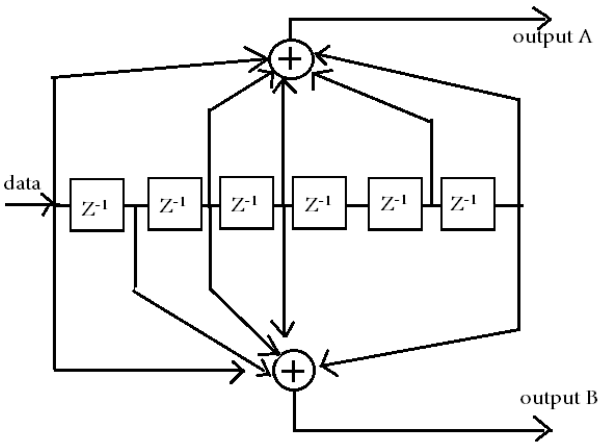


Figure 2. A ½ Rate Convolution Coder.

Data Interleaving is performed in a two step permutation.

(1) The rule of first permutation can be defined as

$$i = (N_{CBPS})(K \bmod 16) + \left\lfloor \frac{k}{16} \right\rfloor$$

where k is the index of the coded bit, N_{CBPS} describes the number of coded bits per symbol and K varies from 0 to $N_{CBPS}-1$.

(2) The rule for the second permutation can be given as

$$j = s \times \left\lfloor {}^{1}\!/_{s} \right\rfloor + \left(i + N_{CBPS} - \left\lfloor \frac{k}{16} \right\rfloor\right) \bmod s$$

where $s=\max(N_{BPSC}/2,1)$.

N_{BPSC} is defined as the number of coded bits per sub carrier and 'j' is the coded bit index after permutation. The OFDM sub-carriers can be modulated using the modulation schemes like BPSK, QPSK, 16-QAM, and 64-QAM. The sub-carriers are not actually modulated but are mapped to the corresponding constellation points which are depend on the required data rate. The constellations of BPSK and 16-QAM are shown as an example in Figure 3. The complex numbers obtained after the mapping scheme are fed to a 64 point IFFT block to convert the frequency domain information to time domain. Then the output is attested with cyclic prefix.

3. Partial Reconfiguration

FPGA technology can be suitable for on-site programming; also can be re-programmed at any number of times. The

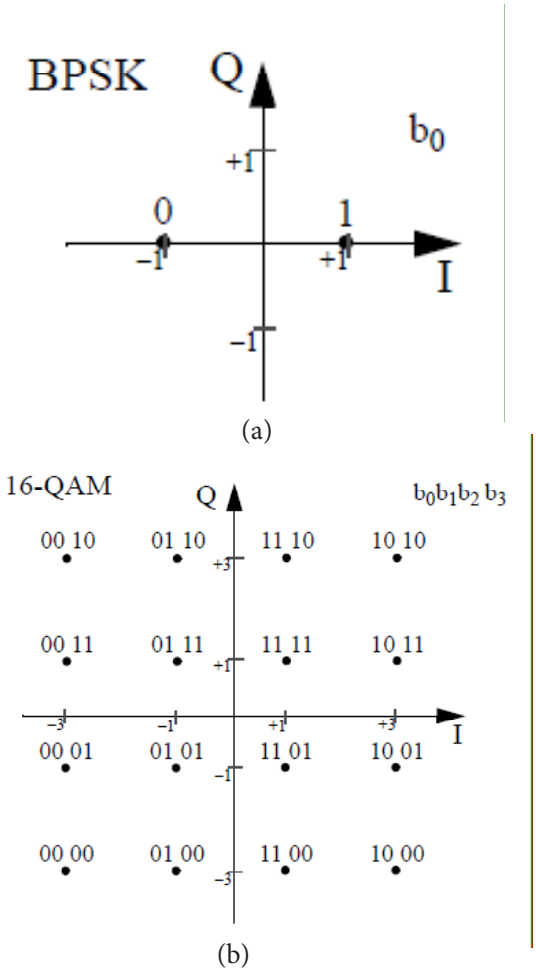


Figure 3. Constellations of (a) BPSK and (b) 16- QAM.

Partial Reconfiguration (PR) is the new feature available in the recent FPGA systems which modifies a subset of logic in a regular FPGA design[6]. The main idea is to reconfigure specific parts (or modules) of a system while the remaining part of the system remains as such and is still running. The portions of the system that are to be reconfigured are called reconfigurable modules and the parts that do not change are called static modules.

Partial reconfiguration can be done in two ways, i.e., modular based PR and difference based PR. In modular based PR, bit files for each module are loaded for during reconfiguration. However, a single partial bit stream is loaded in difference based PR which contains information about the difference between the current system and the new system. Difference based partial reconfiguration is particularly very useful when small changes have to be made to the design. Modular based design is implemented in this design.

4. Implementation and Results

4.1 Design Flow and Testing of Partial Reconfigurations

The static top module with black boxes of the reconfigurable module is synthesized and the corresponding netlist is generated. Each reconfigurable module is separately synthesized and the netlists are generated. Using the PlanAhead tool with Partial Reconfiguration License, the PR project is made. A suitable rectangular area is selected to suffice each reconfigurable partition. Area group constraints are then added to the partitions. A design run check is made before the actual implementation to make sure that there are not any critical issues in the design. The design is then promoted if design runs made are satisfactory. 'PR verify' is used to identify if the design is compatible on the hardware. Bit files are then generated.

The proposed system was described using Veriog HDL and implemented on Xilinx® Virtex5 board. The target device selected for the hardware implementation is Xilinx Virtex5- xupv5lx110t. This device supports partial reconfiguration and is compatible with the design flow of the Xilinx Plan Ahead tool. The FFT/IFFT IP cores are optimized for high performance and reduced area. The transmitter gets input from the pre-stored data in a ROM. The ROM is generated using Xiinx LogiCore distribute memory generator[5]. Figure 4 shows the 16-bit deep, 1-bit wide read only memory (ROM) which can be created by

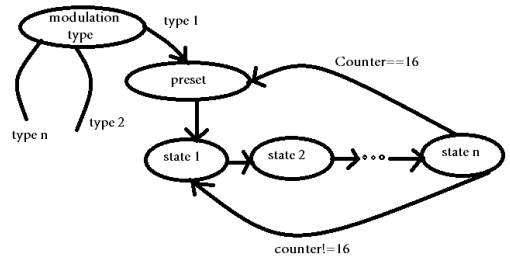


Figure 5. State diagram for the Interleaver block.

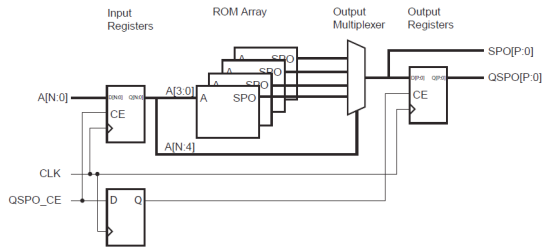


Figure 4. Organization of ROM.

Distributed Memory Generator which uses the look-up table (LUT) based ROMs. The configuration file (i.e.COE file) defines the initial contents of the memory which consists of two parameters: (i) memory initialization radix and (ii) memory initialization vector. The initial radix value can be selected as 2, 10, or 16 time of the memory initialization vector. The row of the memory elements is described with its binary equivalent which indicates whether an individual memory element along the row width is set to 0 or 1 . The separation between the memory initialization rows, up to the depth of the memory can be specified by the characters, a comma or white space, at the same time, negative values are not to be considered.

The scrambler has been implemented with Linear Feedback Shift-Register (LFSR) approach as described in the IEEE standards. The convolution coder with bit puncturing for respective rates (½, 2/3 or ¾) has been implemented using the Intellectual Property (IP) core provided by the Xilinx. The design for the coder without bit puncturing has been implemented using shift- registers and XOR gates as per the standard. The Implementation of the interleaver has been followed from the design specified in[6]. The design is based on the regularity in the addresses generated in each modulation type based on the design rules[4] specified in the standards.

A finite state machine (FSM) is used to implement the interleaver block as shown in Figure 5 and IFFT block is implemented with FFT IP core. The interleaver contains 16 rows. The regularity or periodicity of the addresses generated can be viewed from Table 1. It is to be noted that *j and k* values in subsequent addresses are not equally spaced for all cases. Within a modulation scheme, the increment values follow a fixed type of pattern irrespective of coding rate. Table 1 shows the interleave pattern formation for different modulation schemes with different increment values. For QPSK scheme with s = 1, the increments are linear for values like 6 for Ncbps = 96, 9 for Ncbps = 144

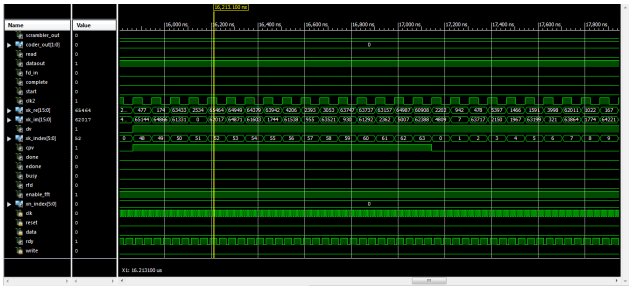
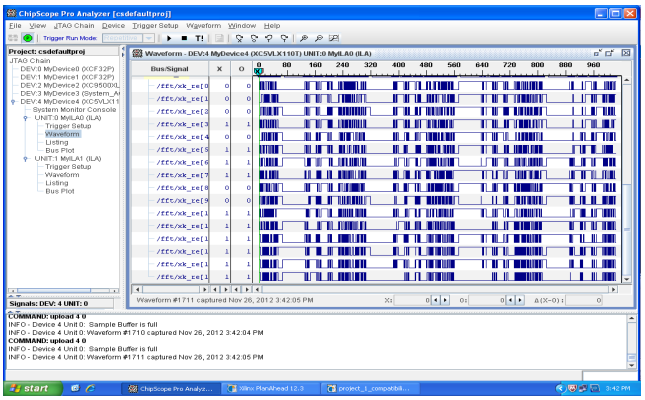


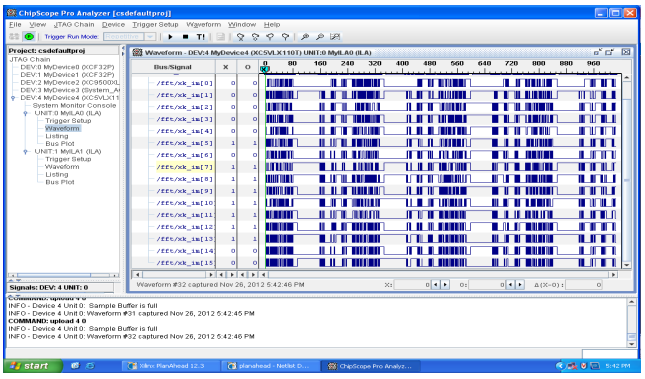
Figure 5. Simulation result of 2/3 rate QAM modulated OFDM System.

Table 1. Interleave pattern formation

NCBPS =48 (BPSK) (mod_typ=00)	0	3	6	9	12	14	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
NCBPS =96 (QPSK) (mod_typ=01)	0	6	12	18	24	30	36	42
	48	54	60	66	72	78	84	90
	1	7	13	19	25	31`	47	43
	49	55	61	66	73	79	85	91
NCBPS =192 (16-QAM) (mod_typ=10)	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
NCBPS =288 (64-QAM) (mod_typ=11)	0	20	37	54	74	91	108	128
	145	162	182	199	216	236	253	270
	1	18	38	55	72	92	109	126
	146	163	180	200	217	234	254	271



(a)



(b)

Figure 6. On-chip verfication of OPDM transmitter. (a) Real Part and (b) Imaginary Part.

and so on. For 16-QAM with s = 2, and 64-QAM with s = 3 have nonlinear increments like 13, 11 for Ncbps = 192 and 20, 17, 17 for Ncbps = 288 respectively.

The interleaver memory is generated registers on the FPGA. The FSM generates the address for the input data in the memory and the memory is consecutively filled. Then a complete signal is generated from the interleaver to enable reading data sequentially from the RAM. The data output is fed to the mapper circuit. Constellation Mapper maps the incoming bits onto separate sub-carriers. Mapper is synthesized as a black box and is one of the reconfigurable partitions. Mapper generates outputs from the input data after receiving some data samples. The number of data per symbol is based on the modulation scheme used. QPSK uses 2 data samples per symbol. QAM-16 uses 4 data samples per carrier and QAM-64 uses 6 data samples per carrier. Mapper outputs the type of modulation type and I component and Q component of the data. Direct mapping output is stored in separate DRAMS before being fed to the FFT engine to add pilot carriers. The FFT core takes I component and Q component of the samples serially and are read from the DRAMS when the all the samples have been loaded into the memory.

The reconfigurable partitions are the convolution coder and symbol mapper. The reconfigurable modules for each partition are the convolution coders with respective coding rates and the symbol mapper corresponding to a modulation scheme and the respective black boxes. Xilinx provides a reconfigurable FFT IP core but the design does not need the IFFT module to be reconfigurable. The modules are separately synthesized for partial reconfiguration. The desired data rates and modulation scheme for a particular configuration can be achieved by using the appropriate combination of convolution coder and symbol mapper. The static modules are the scrambler, interleaver, the control unit and IFFT block.

A locally synchronous globally asynchronous methodology has been applied. A master clock of 200MHz is used. Various clocks are generated from the master clock for different blocks based on the modulation scheme to reach convergence in the overall system. A reset after every change in the configuration is advisable. A Control unit is designed to maintain the synchronization among the blocks and to apprehend all the signals coming out of the IP core blocks. The design is simulated using the ISIM in the Xilinx ISE. The PlanAhead software is used to perform the 'place and route' and 'floor planning' operation and create the reconfigurable project.

4.2 Synthesis Results

The Synthesis report of configurations without Partial reconfiguration is given below:

Number of slice LUTs used: 16%
Number of slice registers used: 15%
Number of occupied slices: 27%
Number of BRAM used: 19%

Convolution coder IP block uses nearly 41 LUT-FF pairs for ¾ coding rate and 9 LUT-FF pairs for ½ coding rate. The FFT IP core block uses nearly seven 18k block RAMs and 12 XtremeDSP slices. Simulation result of 2/3 rate 64-QAM OFDM is presented in the Figure 5.

4.3 On-Chip Debugging

We also set up the Chipscope IP cores for the on-chip verification. Chipscope is a tool provided by Xilinx that allows us to provide trigger conditions for the system and simulate them to obtain results by operation on the board. This tool eliminates the need for I/O pins and gives full internal access to the FPGA. This tool uses the global clock assigned in the UCF file which has been previously added to the project during the PR design process. Though it uses the clock source from the UCF file we need to set the clock as an input clock trigger and set the input data to trigger ports. The trigger ports are monitored and the results are obtained and displayed as waveforms. We can add as many trigger conditions as we want to view the outputs. Chipscope Integrated Controller (ICON) is an internal IP core which can be used for the design. The ICON tool can be used to configure the number of control ports and also the boundary scan instance. The Chipscope Integrated Logic Analyzer (ILA) is an IP core that helps us in analyzing a particular design with the help of the sources. The clock and trigger ports are configured to respective inputs by using this tool. The outputs can be viewed via JTAG port.

We configure the output signal from the ROM as an input for the trigger port of ILA and the clock sources is added as a clock port. The outputs of the IFFT block are configured as the respective outputs to be viewed. These signals are continuously monitored and presented in the way we wish to view them. The results of the Chipscope simulation can be seen in Figures 6 (a) and (b). The result depicts the real and imaginary parts of the outputs obtained from the IFFT block. These outputs are 16 bit arrays and hence we can view up to 32 signals as output

when expanded. These outputs are checked when the data valid signal is high. We can also observe that we get 80 samples as output when data valid goes high. This is due to the 64 point FFT and the 16 bit cyclic prefix that is added at the end. These 80 samples are read as output from the system.

5. Conclusion

Partial Reconfiguration was successfully demonstrated for the IEEE 802.11g OFDM PHY on the Virtex 5 FPGA kit. The reconfigurability was observed in the encoder for obtaining varied code rates and also in the symbol mapper for providing different modulation schemes. Multi functionality has been observed on a single chip which is a key feature of Software Defined Radio. This feature can also be extended to various other system's PHY layer. Also Partial Reconfiguration can be extended to the Tier 3 and 4 SDR systems.

6. References

1. Srikanteswara S, Reed JH, Athanas P, Boyle R. A soft radio architecture for reconfigurable platforms. IEEE Comm Mag. 2000; 38(2):140, 147.
2. Meyer-Baese U. Digital signal processing with field programmable gate arrays. 3rd ed. Springer; 2007.
3. Delahaye J-P, Moy C, Leray P, Palicot J. Managing dynamic partial reconfiguration on heterogeneous SDR platforms. SDR Forum Technical Conference 5; 2005.
4. Clause 17, IEEE Std 802.11a, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer(PHY) Specifications, High-Speed Physical Layer in the 5 GHz Band; 2007.
5. Xilinx Partial Reconfiguration User Guide. Available from: www.xilinx.com
6. Upadhyaya, BK, Sanyal SK. Design of a novel FSM based reconfigurable multimode interleaver for WLAN application. 2011 International Conference on Devices and Communications (ICDeCom); 2011 Feb 24–25. p. 1–5.