Candidate Waveforms for Wireless Communications: Analysis via Hardware Software Co-Design on Zynq SoC

Student Name: Sasha Garg

IIIT-D-MTech-ECE
July , 2017

Indraprastha Institute of Information Technology
New Delhi

Thesis Committee

Dr. Sumit Jagdish Darak (Advisor)
Mr. Prateek Sikka (External Advisor, NXP Semicondustors)
Dr. Sneh Saurabh (Internal Reviewer)
Dr. Rahul Shrestha (External Reviewer, IIT Mandi)

Submitted in partial fulfillment of the requirements
for the Degree of M.Tech. in Electronics & Communication with specialization in
Communication & Signal Processing

Keywords: Hw-SW Co-design, Zynq, Software defined radio, OFDM, Filtered-OFDM, WOLA, FBMC

Certificate

This is to certify that the thesis titled "**Candidate Waveforms for Wireless Communications: Analysis via Hardware Software Co-Design on Zynq SoC**" submitted by **Sasha Garg** for the partial fulfillment of the requirements for the degree of *Master of Technology in Electronics and Communication Engineering* is a record of the bonafide work carried out by her under my guidance and supervision at Indraprastha Institute of Information Technology, Delhi. This work has not been submitted anywhere else for the reward of any other degree.

July, 2017

Dr. Sumit J Darak
Assistant Professor
Department of Electronics and Communication
Indraprastha Institute of Information Technology Delhi
New Delhi, 110020

Abstract

Upcoming wireless communication networks are expected to support the wide variety of services ranging from low data rate applications such as wireless sensor networks to high data rate delay sensitive multimedia services. To bring such networks to life, wireless transceivers should evolve from existing homogeneous to heterogeneous architectures capable of adapting their transmission parameters on-the-fly to meet the desired quality of service. For such heterogeneous transceivers, the design and analysis of various waveforms such as Orthogonal Frequency Division Multiplexing (OFDM) and its variants such as windowed overlap and add OFDM (WOLA-OFDM) and filtered OFDM (F-OFDM), filter bank multi carrier (FBMC) etc., is one of the popular and important research area today.

From transceiver implementation perspective, Zynq System on Chip (ZSoC) from Xilinx provides an efficient solution for implementation of heterogeneous reconfigurable systems and is superior to conventional two chip solutions. ZSoC is an integrated system consisting of an ARM processor which comprises the processing system (PS) and a reconfigurable field programmable gate array (FPGA) forming the programming logic (PL) of the architecture. The design and implementation of 802.11a based transceiver using various waveforms on the ZSoC platform along with the detailed performance and complexity analysis are the main focus of the work presented in this thesis.

The first contribution of this thesis is the design of the 802.11a based transceiver architecture using OFDM waveform. The architecture is then divided into two sections, one for PL and other for PS. Such co-design approach gives the flexibility to choose which part of the system to realize in PL and which in PS. Different configurations of the architecture are analyzed to identify the units best suited to be implemented respectively on PS and PL.

The second contribution is to replace OFDM based transceiver with WOLA-OFDM and F-OFDM waveforms. In WOLA-OFDM, some portion of the symbol is appended at the start and end overlapping with adjacent symbols. In addition, time domain windowing is applied via root raised cosine filtering. In F-OFDM, a linear phase finite impulse response filter is used to further improve the out-of-band attenuation of the OFDM.

Next, the functionality of these architectures with different configurations of PS and PL is verified by implementing them on ZSoC platform using hardware software co-design workflow of MATLAB and Simulink. Based on out-of-band attenuation plots for different bandwidths, it is observed that WOLA-OFDM and F-OFDM offer better performance than OFDM for any given bit-to-error ratio. Among them, F-OFDM offers significantly better out-of-band attenuation than WOLA-OFDM. Implementation complexity comparison shows that F-OFDM complexity is slightly higher than others. However, in all the configurations, the utilization of the number of flip-flops and look-up-tables of PL section is less than 7% and 35% respectively leaving enough resources for higher layers.

Acknowledgments

It gives me immense pleasure to express my heartily gratitude to everyone who supported and guided me in the completion of my thesis.

Foremost, I would like to express my sincere gratitude to my advisors Dr. Sumit J Darak and Mr. Prateek Sikka. Without their excellent guidance, encouragement and support, I would never be able to finish my thesis work. They have been a great source of inspiration and I feel extremely fortunate to work with them.

I would like to thank Shannon lab technical staff, Mr. Khagendra Joshi and Mr. Rahul Gupta for providing me quick access to all instruments whenever I needed them.

I would like to acknowledge my parents and friends for encouraging and supporting me. They have been a source of moral support to me and have extended their helping hands without fail.

Contents

1 Introduction **1**

1.1 Motivation 1

1.2 Objective and Contribution 2

1.3 Organization 3

2 Zynq System on Chip **4**

2.1 Advantages 4

2.2 Architecture 5

2.2.1 Processing System (PS) 6

2.2.2 Programming Logic (PL) 7

2.2.3 AXI Interface 7

2.3 HW-SW Co-design Workflow 8

2.4 Summary 10

3 OFDM, WOLA-OFDM and FOFDM **11**

3.1 Orthogonal Frequency Division Multiplexing (OFDM) 11

3.1.1 Model Varients 14

3.2 Windowed Overlap and Add OFDM (WOLA-OFDM) 17

3.3 Filtered OFDM (F-OFDM) 20

3.4 Summary 22

4 Results and Discussion **23**

4.1 Out-of-Band attenuation 24

4.2 Resource Utilization and Power results 25

4.3 Timing Information 27

4.4 Summary 28

5 Conclusion and Future Works **29**

5.1 Conclusion 29

5.2 Future Work 29

A Steps for implementation of QPSK HW-SW cosimulation example **31**

A.1 Software Installation 31

A.2 Open the example and Configure MATLAB 32

A.3 Generate the Hardware-Software Model 32

A.3.1 HDL Workflow Advisor 32

A.3.2 Hardware/Software interface model 32

A.4 Run the Model 33

List of Figures

2.1 Comparison of various platforms 5
2.2 Snapshot of Xilinx ZC706 evaluation board along with its important architectural features [6]. 6
2.3 AXI Link 8
2.4 Implementation of algorithm on ZSoC via hardware software co-design approach. 9
2.5 Hardware-Software work flow for ZSoC using HDL and Embedded coders of Matlab/Simulink and Xilinx Vivado. 9

3.1 Block diagram of the OFDM based transceiver with five different versions indicating division between PS and PL. 11
3.2 Subcarrier Mapping for 650 KHz models 13
3.3 Subcarrier Mapping for 350 KHz models 13
3.4 Top level architecture for V1 15
3.5 Top level architecture for V2-V5 16
3.6 PS implementation OFDM modulation 16
3.7 PL implementation OFDM modulation 17
3.8 Block diagram of the WOLA-OFDM based transceiver with five different versions indicating division between PS and PL. 18
3.9 Time domain windowing at the transmitter in WOLA-OFDM 18
3.10 PS implementation of WOLA-OFDM tranmsitter 19
3.11 PS implementation of WOLA-OFDM receiver 19
3.12 PL implementation of WOLA-OFDM tranmsitter 20
3.13 PL implementation of WOLA-OFDM receiver 20
3.14 Block diagram of the F-OFDM based transceiver with five different versions indicating division between PS and PL. 21
3.15 PS implementation of filter 21
3.16 PL implementation of filter 21

4.1 Demonstration setup: Real-time Transmitter output waveform for OFDM & F-OFDM 23
4.2 Power spectral density comparison of OFDM, WOLA-OFDM and F-OFDM. . . 24

4.3 Power spectral density comparison of OFDM, WOLA-OFDM and F-OFDM. . . 25

List of Tables

2.1 ZC706 zynq evaluation board specifications 9

3.1 Data Transfer between PS and PL 15

4.1 Parameter Values 24
4.2 Experimental Results on ZSoC for 650 KHz bandwidth signal 26
4.3 Experimental Results on ZSoC for 350 KHz bandwidth signal 27
4.4 PS frame time for 650 KHz bandwidth signal in ms 28
4.5 PS frame time for 350 KHz bandwidth signal in ms 28

List of abbrevation

AMBA	Advanced Microcontroller Bus Architecture
ADC	Analog-to-Digital Converters
ARM	Advanced RISC Machines
ASIC	Application Specific Integrated Circuits
ASSP	Application Specific Standard Parts
AXI	Advanced eXtensible Interface
BPSK	Binary Phase Shift Keying
CLB	Configurable Logic Blocks
CRs	Cognitive Radios
DFT	Discrete Fourier Transform
DSP	Digital Signal Processing
FFs	Flip Flops
F-OFDM	Filtered OFDM
HDL	Hardware Description Language
IDFT	Inverse Discrete Fourier Transform
OFDM	Orthogonal Frequency Division Multiplexing
OOB	Out-of-band
LUTs	Look Up Tables
PS	Processing System
PL	Programming Logic
PSD	Power Spectral Density
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
SDR	Software Defined Radio
USRP	Universal Software Radio Peripheral
WOLA-OFDM	Windowed Overlap and Add OFDM
ZSoC	Zynq System on Chip

Chapter 1

Introduction

1.1 Motivation

In recent years, the field of wireless communication has evolved tremendously. The applications of wireless technology have diversified over the years ranging from cellular communication to air-traffic-management. This has lead to setting up of larger wireless networks and traffic congestion in existing networks. The exponential increase in the demand has led to introduction of various new protocols and standards. Wireless communication particularly has seen the highest rate of development evolving from 2G CDMA based communication with a data rate of maximum 9.6 kbps to LTE which now offers a downlink data rate of 50 Mbps. Lately, intensive research is going on to propose a 5G standard for multiple integrated wireless/access applications like like Internet of Things (IoT) and LDACS. New waveforms are being proposed to achieve higher out-of-band attenuation and spectral efficiency compared to existing OFDM based systems. With the coming up of new technologies, the transceiver systems must be able to keep up with evolving standards[1, 2]. Thus, a need was felt for platforms which could provide flexibility to reconfigure the framework according to the future protocols.

Xilinx has introduced a new-age technology, Zynq system on chip (ZSoC). ZSoC is a heterogeneous system which provides decision making capabilities. It allows the users to modify both the software and the hardware according to their requirement. ZSoC consists of a processing system (PS) and a reconfigurable programmable logic (PL) on a single chip. Thus, provides high level of hardware-software integration and hence is the preferred solution over two chip based platforms. The first contribution of this work is the design of OFDM based transceiver architecture for implementation on ZSoC using hardware software co-design workflow of MATLAB and Simulink. Various co-designs for the architecture are realized by dividing the model into two sections, one for PL and other for PS. Such co-design approach gives the flexibility to choose which part of the system is best suited to be implemented on PL and which on PS. Thus helps realize an optimal design so as to meet the given area, delay and power constraints.

OFDM is a multicarrier transmission technique being used in LTE/LTE-Advanced networks.

But, its sensitivity to frequency and clock offsets limits the spectral agility of the system. High out-of-band emissions in an OFDM signal causes inter-carrier-interference. Thus, in order to meet 5G requirements, new waveforms like F-OFDM and WOLA-OFDM are being proposed. The aim is to decrease the OOB emissions and thus attain better spectral efficiency compared to OFDM. The second contribution of this work is to replace OFDM based transceiver with Windowed Overlap and Add OFDM (WOLA-OFDM) and Filtered OFDM (F-OFDM) waveforms. Then, these architectures with different configurations of PS and PL are synthesized and implemented on ZSoC platform.

Like the existing standards, wireless communication standards being proposed for 5G networks need to cover a wide range of frequency bands, for e.g. CDMA, WCDMA, WiFi, bluetooth etc. In conventional multiple device system, we have a dedicated device for each frequency band. Such a system, allows optimization of individual device but suffer from loss in performance due to high switching complexity. Also, such a system is larger in size and has higher cost. Thus, a compact single device, tunable to multiple frequency bands is required. Keeping this in mind, the proposed architectures can be used for multi-standard wireless communication systems where the transmitter and the receiver can be tuned to operate at multiple frequency bands. We analyze the performance of the proposed transceiver systems for two different bandwidths. Finally, the detailed experimental results demonstrate the trade-off between the different waveforms with respect to out-of-band attenuation, area, delay and power requirements.

1.2 Objective and Contribution

The objectives of the work presented in thesis are:

1. Implement OFDM based transceiver system on ZSoC. Various variants of the architecture are realized by dividing it into two sections, one for PL and other for PS.

2. Demonstrate the flexibility offered by the HW-SW co-design approach to decide which part of the transceiver to implement on PL and which on PS to meet the given area, delay and power constraints.

3. Implement F-OFDM and WOLA-OFDM hardware-software co-designs on ZSoC. Analyze trade-off between these waveforms with respect to out-of-band attenuation, area, delay and power requirements.

Contributions of this thesis, which are under review and submission are described below:

1. S. Garg, N. Agarwal, S. J. Darak,, and P. Sikka, "Spectral Coexistence of Candidate Waveforms and DME in Air-to-Ground Communications: Analysis via Hardware Software Co-Design on Zynq SoC,"in *2016 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC)*, Orlando, Florida, USA, Sept. 2017

This paper offers detailed performance and complexity analysis of OFDM, F-OFDM and WOLA-OFDM as the candidate waveforms for L-band (960-1164 MHz) digital aeronautical communication system (LDACS) in air traffic management. The implementation is done on Zynq System on Chip (ZSoC) platform.

1.3 Organization

The thesis is organized as follows. Chapter 2 presents the detailed background on zynq architecture and the workflow followed during implementation. In Chapter 3, the three waveforms, OFDM, F-OFDM and WOLA- OFDM have been discussed. The transmitter and receiver architecture is explained in detailed along with the variations encountered in each co-design. Chapter 4 presents the comparison between the waveforms based on the OOB attenuation. In addition, the area and time complexity for the different model versions has been analyzed. Finally, Chapter 5 concludes the work done in this thesis and briefly discusses the possible future works.

Chapter 2

Zynq System on Chip

In this chapter, we explain why Zynq SDR is a preferred platform for the implementation of wireless communication standards. We compare the characteristics of ZSoC with other platforms available in the market. The chapter also explains the zynq architecture with detailed description of both the hardware and the software components. AXI interface which is required for the communication between the the components is also discussed. Next, we discuss the workflow followed for the implementation of the hardware-software co-design models on the ZC706 platform.

2.1 Advantages

Gradual process of developing a next generation reconfigurable system which provides improved performance and higher levels of integration has been going on over the years. Various devices like software defined radios (SDRs) and cognitive radios (CRs) have been introduced. They allow the users to modify various parameters like modulation type, frequency band etc. However, these platforms do not allow the user to modify the hardware component of the device. Also, many a times the developers are required to use the pre-defined software routines and thus restricts them from implementing new algorithms. Other platforms available are Application Specific Integrated Circuits (ASIC) and Application Specific Standard Parts (ASSP) platforms. As shown in Fig. 2.1, these devices are application specific and thus do not offer the flexibility to reconfigure hardware and/or software and limited in scalability. Though they lack hw-sw integration, these platforms are power efficient and offer good performance. Two-chip solutions, which have a separate chip for hardware and software allow hardware/software reconfigurability but are not power and cost efficient. With the new applications coming up, higher degrees of flexibility , reconfigurability and scalability are required. Thus, Xilink introduced Zynq-7000 all programmable system-on-chip/ software defined radios. ZSoC is a single chip platform which provide the following advantages:

1. **Higher reconfigurability/flexibility** : The platform allows the developers to program

Figure 2.1: Comparison of various platforms

both the processor (i.e PS) and the FPGA (i.e PL) separately according to the exact needs of the concerned application.

2. **Higher levels of hardware-software integration** : ZSoC integrates the processor and FPGA along with numerous input-output peripherals. This allows higher number of application support with minimum on-board components.

3. **Scalability**: ZSoC provides the flexibility to design various applications ranging from low end to high end on a single platform.

4. **Lower power consumption**: High levels of integration and lesser on-board components lead to lower power consumption as opposed to other technologies available.

5. **Lower cost**: Zsoc comes at a lower price compared to other platforms.

Other SDRs like WARP and SORA provide software programmability for wireless communication standards, thereby providing limited flexibility as compared to ZSoC. Ettus Research Universal Software Radio Peripheral (USRP)is another software defined radio extensively used by researchers. USRP provide higher dynamic range and higher bandwidth but unlike ZSoC, don't allow hardware reconfigurabitility.

2.2 Architecture

ZSoC platform by Xilinx enables an efficient implementation of heterogeneous reconfigurable systems [9, 10]. ZSoC is a unique modern platform which offers hardware, software as well as

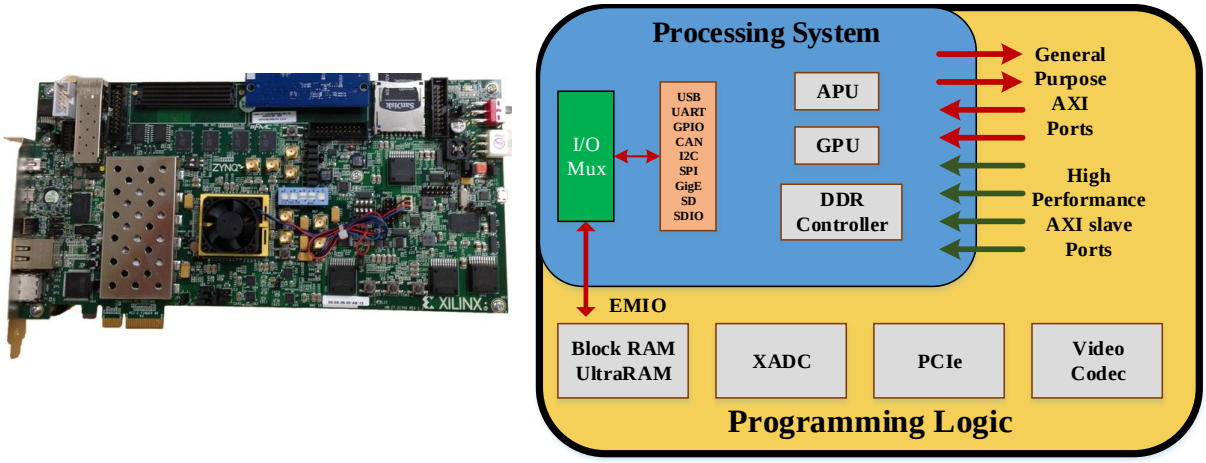

Figure 2.2: Snapshot of Xilinx ZC706 evaluation board along with its important architectural features [6].

input-output programmability on a single SoC, thereby leading to fewer on-board components. This helps achieve better performance and lower power consumption compared to two chip approach based platforms. ZSoC platform has wide range of applications such as wired and wireless communications, medical imaging, driver assistance etc. ZSoC consists of PS (the software component) and PL (the hardware component), which communicate with each other via Advanced eXtensible Interface (AXI) protocol. ZC706 Zynq evaluation board used in the proposed work has dual core cortex A9 Advanced RISC Machines (ARM) as PS and a Xilinx 28nm Kintex 7-series field programmable gate array (FPGA) as PL (Fig.2.2).

2.2.1 Processing System (PS)

ZSoC has a processor centric architecture where, PS always boots first and is fully autonomous to the PL. PS consists of the Application processor unit (APU), memory interfaces, I/O peripherals (IOP) and interconnect [6, 10]. The APU comprises of the dual core ARM Cortex-A9. It is capable of working at three different operating frequencies:667 MHz (-1); 800 MHz (-2); 1 GHz (-3). In our work, we are working on 800 MHz. Since ZC706 has a dual core processor, it has the ability to work in three different processing modes depending on the algorithm requirement. The three processing modes are single processor, symmetric dual processor, and asymmetric dual processor modes. The PS has a dual ported 256 KB on-chip RAM. The on-chip memory is accessible by both the CPU and the PL. Using the on-chip memory allows low latency access of data from the CPU, thereby increasing the speed of operation. In addition to the on-chip memory, the PS also has 1 GB of dynamic memory. The external static memories including RAM, NOR flash and NAND flash memories are handled by 8-bit parallel data bus , 8-bit parallel NOR flash interface, ONFi 1.0 NAND flash support and a Quad-SPI flash interface respectively. The dynamic memory interface include a dynamic memory controller which supports DDR3, DDR3L, DDR2, and LPDDR2 memories. The DDR memory controller of ZC706 allows 16-bit or 32-bit wide access to the 1 GB of dynamic memory. The DDR memory controller is multi-

ported that allows both the PS and PL to have access to a shared memory. It has four 64-bit AXI slave ports, out of which two ports are dedicated to the PL, one to PS and one is shared by all the other AXI masters. In addition to the memory elements, PS also has a total 130 IO port out of which 76 ports are dedicated for DDR . The remaining 54 ports are for multiplexed IO (MIO) which are shared by static/flash memory interfaces as well as peripherals such as USB,CAN, SD card, I2C, UART etc.

2.2.2 Programming Logic (PL)

On the other hand, PL is similar to conventional FPGA consisting of configurable logic blocks (CLBs) like look-up tables, flip-flops and cascadeable adders [6, 10]. There are 8 LUTs, 16 flip flops and two 4-bit cascadeable adders per CLB. The memory LUTs can be configured as 32X2 bit or 64 bit shift registers. In addition to CLB,PL also has digital signal processing (DSP) blocks. Each DSP slice has 18X25 signed multipliers and 48-bit adder. Other components of the PL are 36 Kb Block RAM, programmable I/O blocks, digital signal processing (DSP) blocks, serial transceivers, PCI interface etc. The 36 Kb BRAM can be configured as dual 18 Kb block RAMs. It supports data width of up to 36 bit and has an inbuilt error detection and correction logic and FIFO controller for synchronous or multirate operations. The I/O blocks consist of high performance SelectIO resources, high frequency decoupling capacitors and digitally controlled impedance that can be tri-stated. ZC706 is classified as high rage (HR) supporting a wide range of voltage from 1.2V to 3.3V. All the I/O ports support double data rate (DDR). The inputs and outputs can experience delays implemented as IDELAY and ODELAY. The PL of ZSoC also has a low power serial transceiver for ultra-fast data transmission. There are 16 transceiver circuits each with a combined transmitter and receiver. PL also has a integrated block for PCI express designs that can be configured as an Endpoint or Root Port. It can operate 1, 2, 4, or 8 lanes at the 2.5 Gb/s and 5.0 Gb/s data rates. PL also has two 12-bit 1 MSPS ADCs that can support up to 17 external analog input channels. It also has on-chip temperature and power supply sensors. The measurements are stored in dedicated registers and can be accessed using JTAG connection.

2.2.3 AXI Interface

Modern day application designs consist of a large number of intellectual properties (IP) and require extensive communication between the PS and PL. Since, PS and PL are independent of each other and each is design to perform assigned task differently, we need proper communication standard or protocol for efficient implementation. Such protocol makes the maintenance, debugging, re-use and future upgradation of the design easy. One of the several standards developed like Core-connect and WishBone, Xilinx adopted Advanced eXtensible Interface (AXI) for Zynq architecture. AXI has been developed by ARM. Advanced Microcontroller Bus Architecture (AMBA) based AXI based interface allows synchronized data transfer between PS and PL. Any AXI contains two parts: AXI master and AXI slave. The master initiates any read/write

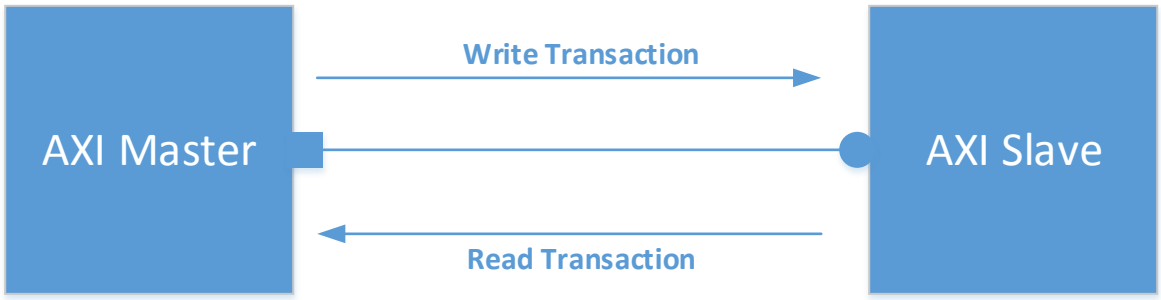

Figure 2.3: AXI Link

transaction (Fig. 2.3). The slave responds to the master's request. In ZSoC, there are 9 AXI ports between PS and PL out of which PS acts as master on 7 ports and PL acts as master on 2 ports. The reader may refer to [11] for more details. Each of the port can be configured in one of the three modes: AXI memory-mapped, AXI-Lite and AXI-Stream. AXI memory-mapped mode allows high performance burst transfer capabilities. For simple low-throughput memory-mapped requirements, AXI-Lite is used. For high speed streaming burst data transfer AXI-stream is best suited. In this paper, the discussion is limited to AXI-Stream mode as it is simple to implement and sufficient for the throughput requirement between PS and PL of the transceiver.

2.3 HW-SW Co-design Workflow

In this section, we discuss the steps of hardware software co-design for the implementation of any given algorithm on ZSoC. Such co-design approach is important to exploit the heterogeneity of PS and PL architectures. Using PL increases system performance and reduces power consumption, while PS makes decision making operations easier along with faster memory access. Hence, understanding of which blocks to be implemented on PL and PS is critical for an efficient algorithm implementation.

To do this, we need Matlab/Simulink 2016 or higher version, hardware description language (HDL) coder and HDL verifier toolboxes, support packages for ZSoC platform and Xilinx Vivado with software development kit (SDK). The hardware setup includes a Zynq ZC706 evaluation board and a host computer. A JTAG and an Ethernet cables is connected between the board and the host PC. The specifications of the zynq board are given in the Table. 2.1 . Depending on the functionality of the algorithm, additional toolboxes might be needed.

Workflow

1. The first step is to realize the given algorithm using the Matlab/Simulink and verify the functionality. Note that all simulink blocks can not be synthesized on hardware and hence, the necessary care should be taken to avoid the use of such blocks. Simulink provides

Table 2.1: ZC706 zynq evaluation board specifications

Device	Z-7045
Processor	ARM Cortex 9
FPGA	Kintex-7
LUTs	218,600
Registers	437,200
DSP Slices	900
BRAM blocks	545

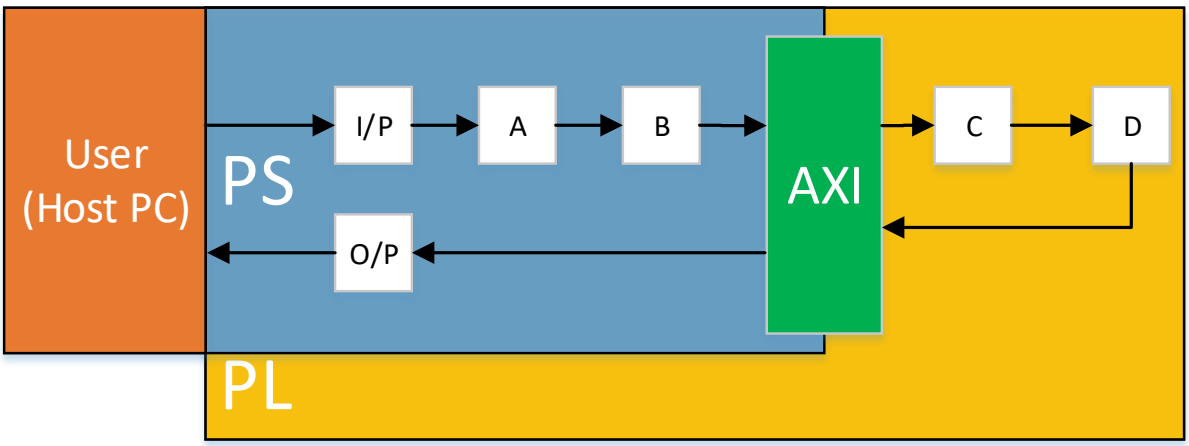

Figure 2.4: Implementation of algorithm on ZSoC via hardware software co-design approach.

a set of hardware description language (HDL) optimized blocks which can be used for hardware implementation. The data type should be specified as fixed point for hardware compatibility.

2. The design is then divided into two blocks one each for PS and PL. Since the PS works in

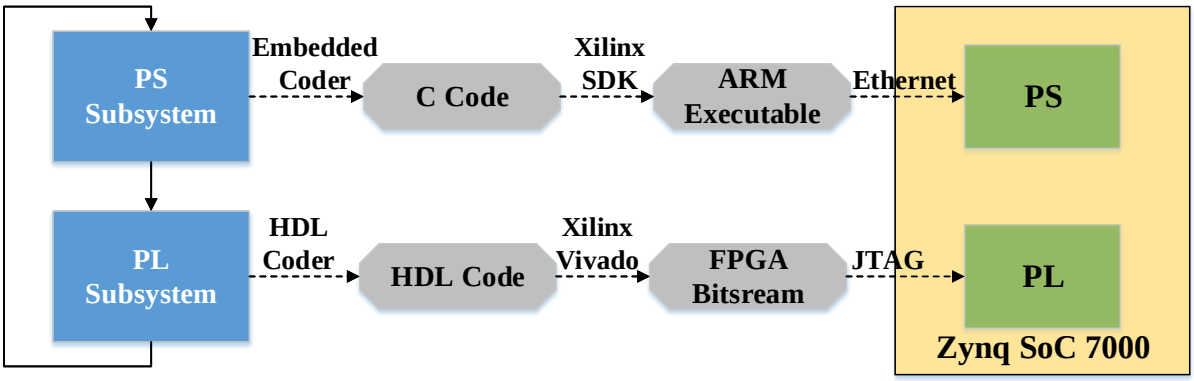

Figure 2.5: Hardware-Software work flow for ZSoC using HDL and Embedded coders of Matlab/Simulink and Xilinx Vivado.

frame mode while PL works in sample mode, appropriate serial-to-parallel and parallel-to-serial data conversion is required at the PL-PS boundary. Also, the sample time for blocks targeted to be implemented in PL and those in PS need to specified accordingly to avoid any data loss. As shown in Fig. 2.4, the algorithm consists of four functional blocks, A, B, C and D. For illustration, we consider the requirement that A and B to be realized on PS while C and D to be realized on PL. This approach helps the user to identify operations best suited to be implemented on PS and PL.

3. For PL, HDL coder is used which converts the blocks to be implemented on PL into corresponding synthesizable HDL representation like verilog or VHDL code. This is then used by Xilinx Vivado for bit file generation which configures the PL as shown in Fig. 2.5. The bit stream is loaded onto the PL using a JTAG connection. HDL coder facilitates the generation of platform specific code using corresponding libraries. It also allows verification of functionality between original and HDL representation of the algorithm via various in-built checks. HDL coder allows the developer to optimize the model architecture along with pipelining and delay balancing options. Hardware resource utilization estimation reports can also be generated using HDL coder. A major advantage of HDL coder is code-to-model and model-to-code traceability between each component in original and HDL representation of the algorithm which makes it easier to debug the design.

4. For PS, embedded coder is used to generate feasible C code for the remaining blocks. This code is then converted to ARM executable code by Xilinx Vivado SDK. In addition, a software interface model is generated which contains the PS and PL blocks interconnected by AXI interface. This interface aids user to visualize data flow between PS and PL in Matlab/Simulink when algorithm is running on the ZSoC platform. The interface model when run in external mode programs the processor with the ARM executable using Ethernet connection. Embedded coder also allows code verification by using software-in-the-loop (SIL) and processor-in-the-loop (PIL) simulation.

Appendix A gives the steps to implement a given HW-SW co-design on a zynq platform.

2.4 Summary

In this chapter the ZSoC architecture is presented followed by hardware software co-design work flow. Heterogeneous Zynq System on Chip (ZSoC) platform, consisting of a processing system (PS) and a reconfigurable programmable logic (PL) on a single chip, is the preferred platform for to meet the quality of service desired by the new age multi-standard transceiver systems. With the help of various softwares and tools like MATLAB/Simulink and Xilinx Vivado, one can easily divide the transceiver functionalities between the PS and PL and generated the bit files and ARM executables to be loaded onto the board. This chapter explains the process in detail. In the next chapter, we will discuss the proposed waveforms and their architectural details.

Chapter 3

OFDM, WOLA-OFDM and FOFDM

In this chapter, the architectural design of the transceiver system using OFDM, WOLA-OFDM and FODM along with the details regarding their implementation on ZSoC are presented. The chapter focuses on the co-design variants for each waveform.

3.1 Orthogonal Frequency Division Multiplexing (OFDM)

Orthogonal frequency division multiplexing (OFDM)[12–14] is multi-carrier digital modulation scheme. The carriers in an OFDM signal are mutually orthogonal to one another over a given time interval thereby reducing inter-carrier-interference. This allows the The sub-carriers to occupy the spectral zero crossing positions of other sub-carriers and hence increasing the spectral efficiency. Due to high spectral efficiency OFDM is being used for data transfer at high rates in 4G wireless networks. An OFDM modulated signal can be generated by performing Inverse Discrete Fourier Transform (IDFT) on the digitally modulated signal and correspondingly a DFT operation in the demodulator. In OFDM, the bandwidth W is equally divided among all the subcarriers. Thus, each subcarrier acts as a narrowband signal but the overall OFDM signal

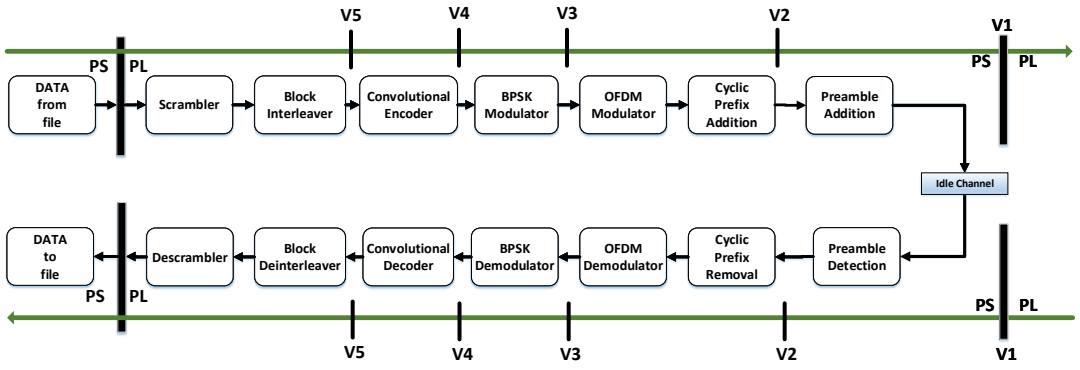

Figure 3.1: Block diagram of the OFDM based transceiver with five different versions indicating division between PS and PL.

will be wideband signal thus making the signal immune to frequency selective fading. In our work, the OFDM transceiver system based on the IEEE 802.11a standard.

Transceiver System

The OFDM implementation of ZSoC has been discussed in [7] and the proposed OFDM transceiver is an extension of [7]. The OFDM based transceiver consists of blocks such as scrambler, convolutional encoder, interleaver, binary phase shift keying (BPSK) modulator, fast Fourier transform (FFT) and cyclic prefix adder in the transmitter as shown in Fig. 3.1. The scrambler rearranges the input bits according to a pre-defined scrambling sequence followed by a convolutional encoder and an interleaver block. This is followed by modulation via BPSK with -1 and $+1$ as the constellation points. Similarly, other modulation schemes can be used. Next steps include serial-to-parallel conversion, 64-point fast Fourier transform (FFT), parallel-to-serial conversion using OFDM modulation block. As per 802.11a specifications, 64 sub-carriers are used out of which 48 are data sub-carriers and the remaining are guard sub-carriers with single DC sub-carrier. Cyclic prefix of length 16 is appended to the OFDM symbol to avoid inter-block interference. At the end, preamble is added for synchronization. The preamble consists of both short training sequence (STS) for course frequency acquisition, timing acquisition and diversity selection and long training sequence(LTS) for channel estimation and fine frequency acquisition [7, 8]. Based on empirical observations, we need to repeat long training sequence twice and short training sequence 10 times for given length of 160 samples. The receiver is similar to transmitter with all the operations being performed in reverse order as shown in Fig. 3.1. Necessary care has been taken to synchronize transmitter and receiver via pre-defined constant polarity pilot sub-carriers. The implementation details for the stimulus(i.e the information source), transmitter and the receiver are explained below.

Stimulus Subsystem

The stimulus subsystem, reads the input bitstream to be transmitted from the MATLAB workspace. The input is a binary stream containing 864 bits. Out of the 864 bit, 24 bits are transmitted per OFDM frame. Thus overall 36 OFDM are transmitted. The further operations have been model for one frame. These operations are then repeated as a new set of 24 bits are read. This is done with the help of a free-running counter keeping a track of the number of frames. With the help of the selector block, we select 24 bits from the incoming stream as the input to the transmitter.

Transmitter

The 24 bit input stream is scrambled according to a predefined constant scrambling sequence. The scrambled sequence and the valid sequence are then forwarded to the convolutional encoder block. A 1/2 rate convolutional encoder with $p_1 = 133$ and $p_2 = 171$ as the generator polynomials has been used to add error detection and correction capability at the receiver. This is followed

by BPSK modulation. The pre-defined BPSK baseband modulator block of the simulink communication toolbox is used the implementation. The phase offset is set to zero. The 48 BPSK symbols are then interleaved according to user defined sequence. Finally OFDM modulation is performed. In our model we have 64 subcarriers. The 48 symbols form the 48 active subcarriers. Out of the remaining 16 subcarrier, one is define as DC subcarrier, 4 are reserved for the pilot signals and the remaining 11 form the guard subcarriers and are set to zero. This is followed by a IFFT block. The FFT length is kept to be 64, implementation mode is set to radix-2. The output is normalized by dividing it by the FFT length. Since, we are not reading the input in conjugate symmetric manner, the subcarriers are mapped accordingly (Fig. 3.2 and 3.3), with the DC signal as the first element of the sequence. Next, using a vector concatenate block, we add the last 16 samples as the cyclic prefix to the data. 4 frames each with 80 samples are added before the first data frame as the preamble. The preamble sequence is predefined and stored in LUTs. The preamble has both long and short preamble sequence. The short preamble is required for coarse frequency acquisition, diversity selection and timing acquisition while the long training sequence helps in channel estimation and fine frequency acquisition.

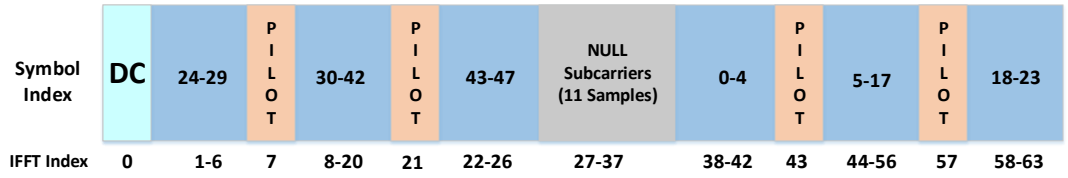

Figure 3.2: Subcarrier Mapping for 650 KHz models

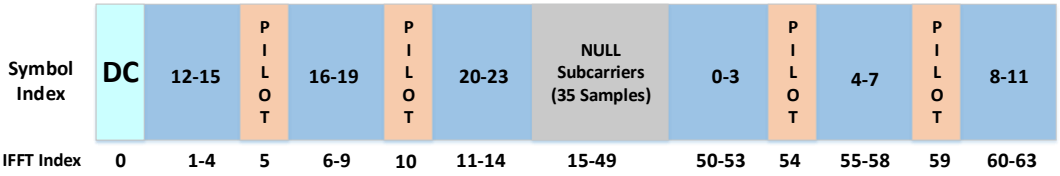

Figure 3.3: Subcarrier Mapping for 350 KHz models

Receiver

The valid signal from the transmitter now enables the receiver functionality. The first step in the receiver is preamble detection which uses auto-correlation to detect the data frames. Once the data frame is detected, it is transferred to OFDM demodulation block. For cyclic prefix removal, the first 16 samples are discarded out of the 80 incoming samples. The remaining 64 samples are given as input to the FFT block. Using a selector block, the subcarriers are mapped to the BPSK symbols. The 48 active subcarriers form the input to the BPSK demodulator block. Using the simulink BPSK baseband demodulator block we retrieve the interleaved bit stream. The bitstream is deinterleaved using the pre-defined deinteleaver sequence. Next, the viterbi decoder block from communications system toolbox is used to decode the data and predefined descrambling sequence is used to retrieve the original 24 bits. The received data is stored in

the workspace. The received bits can now be compared with the transmitted data to verify the functionality of the transceiver system.

3.1.1 Model Varients

The transceiver architecture is divided into two sections, one for PL and other for PS, for implementation on ZSoC. For example, all blocks up to modulation can be realized in PS and remaining blocks in PL. Similarly, various other versions are also possible. For illustration, we consider five versions of the transceiver. In the first version (V1), the entire transceiver is implemented on the PS. We then move the components to the PL one-by-one in the subsequent versions i.e V2-V5 as shown in Fig. 3.1. Such co-design approach gives the flexibility to choose which part of the system is best suited to be implemented on PL and which on PS. Each co-design has been discussed in detail below.

In model V1, the entire transceiver system is implemented on PS, thus there is no PL component in this particular model. The simulink model contains three subsystems, Stimulus, dut-ps and Monitor as shown in Fig. 3.4 . The output of stimulus is packed into a 32 bit signed integer and passed to the transmitter block. The first 24 bits of the integer correspond to the data, followed by a valid bit, a reset bit and zero padding. The 32 bit integer is given as the input to the next subsystem, i.e. the dut-ps. The dut-ps contains both the transmitter and the receiver operations and is targeted to be implemented on the PS of the evaluation board. In this block, the 24 bits are first extracted from the input. Next, depending on the condition of the valid bit, the transmitter operation is enabled to function followed by receiver. Thus, the output of the dut-ps block contained 80 samples each of 16 bit fixed point data type along with the valid and the reset signal being propagated. The sample time for each of the block has been kept equal to the frame time (tpf) which is $80\mu s$ in this case. The run time has been set to $43 * tpf$ to accommodate any delays encountered. To implement the model on the ZSoC, we follow the PIL verification method. In the configuration parameters dialog box, set the *Create Block* to PIL to generate the PIL block for the dut-ps block. Select the dut-ps subsystem and select deploy the subsystem on hardware from the panel. Select *Build* to build the C code for the selected subsystem. Using the generated PIL block corresponding to the dut-ps subsystem, run the model in normal mode to load the processor of the target hardware and verify the design.

In the V2 model, the preamble addition and preamble detection operations are implemented in PL while rest of the transceiver functionality are targeted to be implemented on the PS. The co-design contains three subsystems in addition to stimulus and monitor subsystem (Fig. 3.5). The dut-ps-tx and dut-ps-rx subsystems are targeted to be implemented on PS and correspond to the transmitter and receiver operations respectively. The dut-pl subsystem will be implemented on the PL. The output of the dut-ps-tx subsystem combines the 16 bit fixed point complex output of the OFDM modulator, valid and reset to 32 bit unsigned integer in order to pass the data from PS to PL via AXi interface. 80 such samples constituting 1 PS frame are converted to 80 PL samples using a unbuffer block. The dut-pl extracts the 16 bit complex OFDM sample from

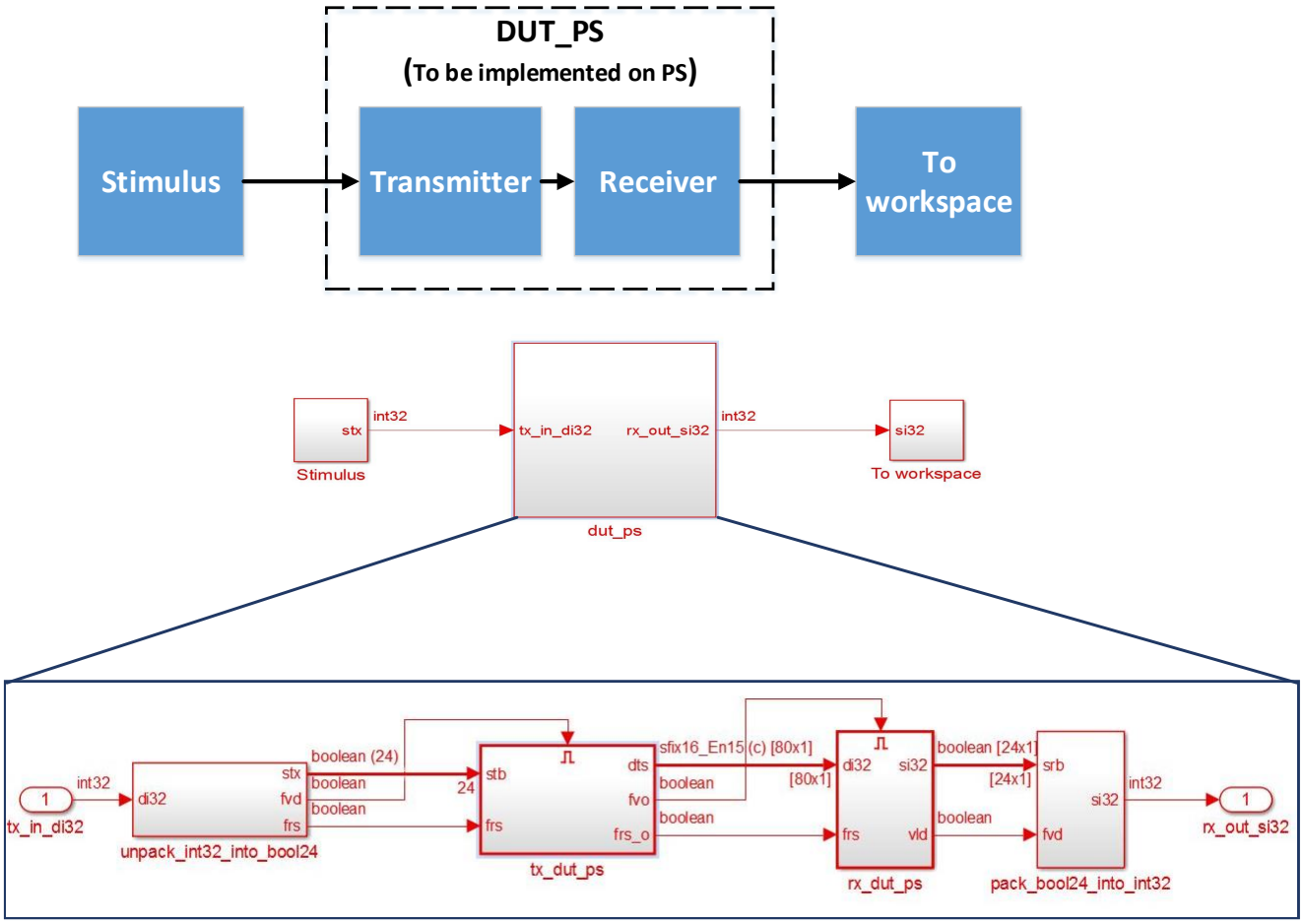

Figure 3.4: Top level architecture for V1

Table 3.1: Data Transfer between PS and PL

Model Varient	textbfData Type	textbfSize of 1 element	textbfNo. of elements
V1	Signed Fixed Point	16 bits	80
V2	Signed Fixed Point	16 bits	80
V3	Signed Fixed Point	16 bits	64
V4	Boolean	1 bit	48
V5	Boolean	1 bit	48

the input along with the valid and reset signal. The PL then implements the preamble addition and detection as explained in the previous section. The out put of the preamble detection is again packed into a 32 unsigned integer. A buffer of size 80 is used to generate a complete OFDM frame which is then passed to dut-ps-rx, The dut-ps-rx subsystem performs the OFDM demodulation, BPSK demodulation, deinterleaving, decoding and descrambling operation which will be implemented on PS. The sample time for each of the block in dut-ps-tx and dut-ps-rx

has been kept equal to the frame time (tpf) which is $80\mu s$ while the block in dut-pl have sample time of $1\mu s$. Fig. 3.7 depicts the implementation of OFDM modulation block when targeted on PS. the receiver architecture is the reverse process of the same.

In the third version, the OFDM modulation and demodulation are transfer to PL in addition to preamble addition and detection. Though, the symbol-to-subcarrier mapping targeted to be implemented in PS. Since the output from subcarrier mapping block in dut-ps-tx, there are only 64 samples per frame. We pad 16 zeros to each frame before packing the the data into

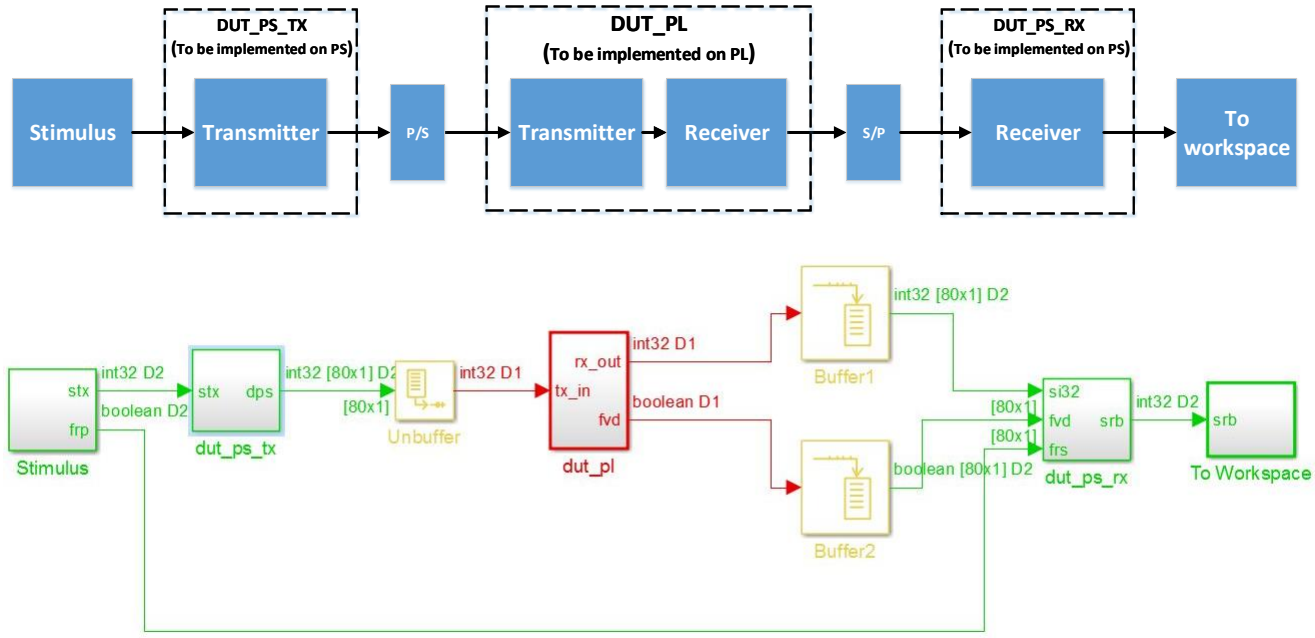

Figure 3.5: Top level architecture for V2-V5

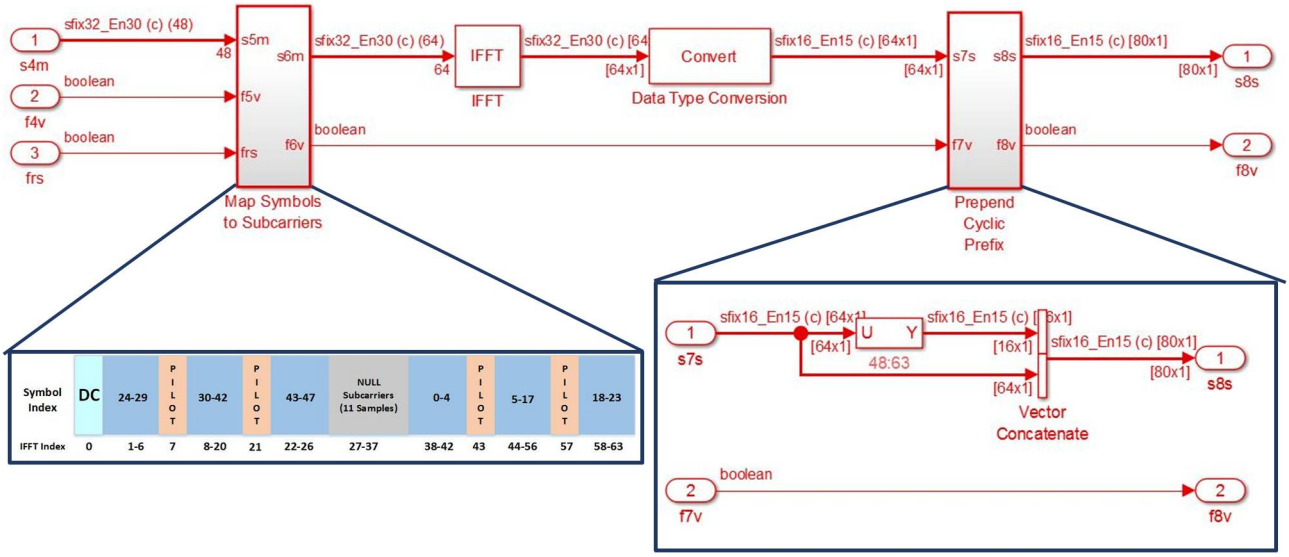

Figure 3.6: PS implementation OFDM modulation

32 bit integer for transfer to PL. The valid is set to false for these 16 sample in order to avoid their interference. Note, that since OFDM modulation and demodulation takes place in PL, we use HDL optimized IFFT and FFT blocks to maintain hardware compatibility. Since, we now only have excess to one sample in one cycle in PL, the cyclic prefix addition implementation is different in this case compare to previous models. We use multiple delays in order to implement this functionality. The rest of the operations are implemented as discussed before. Fig. 3.7 depicts the implementation of OFDM modulation block when targeted on PL. the receiver architecture is the reverse process of the same.

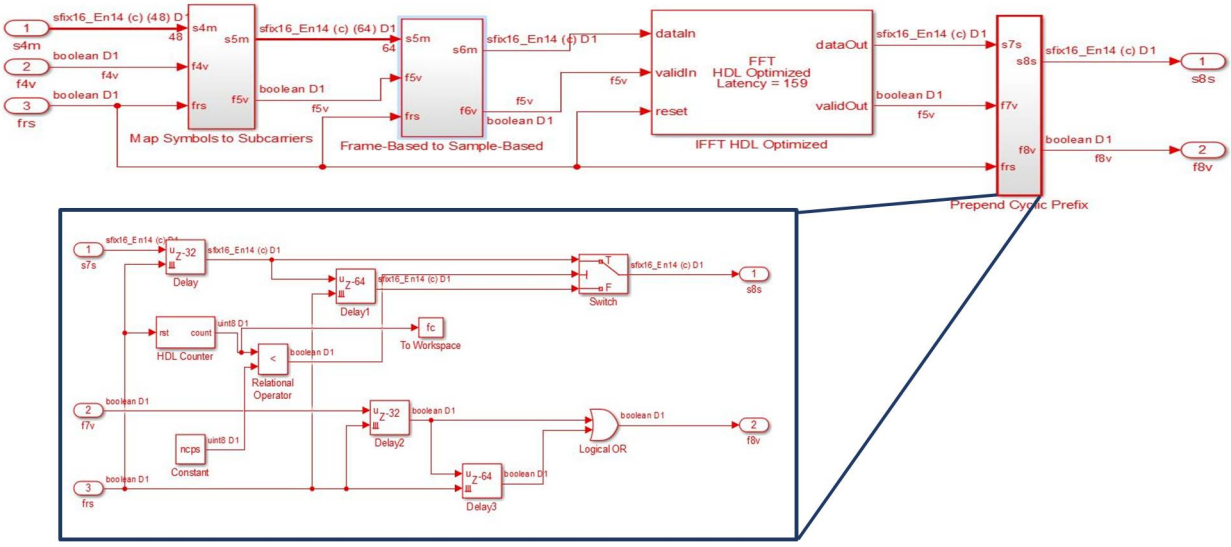

Figure 3.7: PL implementation OFDM modulation

In the version 4 co-design, we move the BPSK modulation/demodulation, symbol-to-subcarrier and subcarrier-to-symbol mapping operations to the PL subsystem. While, in V5 we add the interleaver block to PL. Since, the the subcarrier mapping has been moved to PL, this requires all the data samples per frame to be accessible by PL in the same clock cycle. The data to be transferred per frame from the PS to PL in these models is 48 bit boolean data. Thus the 48 bits are packed into 80X32 bit integers where the first and the second samples contain 24 out of the 48 bits each. This method is followed to maintain consistency in the PS-PL interface. The data is then extracted accordingly in PL and propagated further for the subsequent operations. Table. 3.1 gives the data types of the data transferred between PS an PL in each model version.

3.2 Windowed Overlap and Add OFDM (WOLA-OFDM)

WOLA-OFDM is based on conventional OFDM where some portion of OFDM symbol is appended at the start and end overlapping with adjacent OFDM symbols [5]. In addition, time domain windowing using a pulse with soft edges is done compared to rectangular pulse in OFDM. This leads to superior spectral containment and allows a smooth transition from one symbol to

the next which leads to better out-of-band attenuation when compared to OFDM.

WOLA-OFDM transceiver requires cyclic suffix addition, windowing and overlap and add blocks in addition to the OFDM architecture, as shown in Fig. 3.8. The functionality of the windowing block is depicted using Fig. 3.9. The overlapping edges are achieved by appending the last $CP + W_2$ samples at the start and the first W_1 samples as suffix. In addition, time domain windowing is applied via root raised cosine (RRC) filtering by performing point-to-point multiplication. A window of length $N + CP + W_1 + W_2$ is applied on the new OFDM symbol. In our design, $N = 64, CP = 16$ and $W_1 = W_2 = 10$, making the window length equal to 100 with the taper region of the window equal to W_1 and W_2 on left and right side, respectively. From the 100 samples of the OFDM symbol, we exclude the 20 samples of tapper region. The remaining 80 samples form the final WOLA-OFDM symbol to be transmitted. At the the receiver, the data is retrieved with the help of overlap and add operation in addition to windowing.

The realization of windowing operation in PS (V1 and V2) is easy due to frame based processing as shown in Fig. 3.10 and Fig. 3.11. For the cyclic extensions, the required samples are extracted from the OFDM frame and concatenated at the start and end to get the desired OFDM symbol.

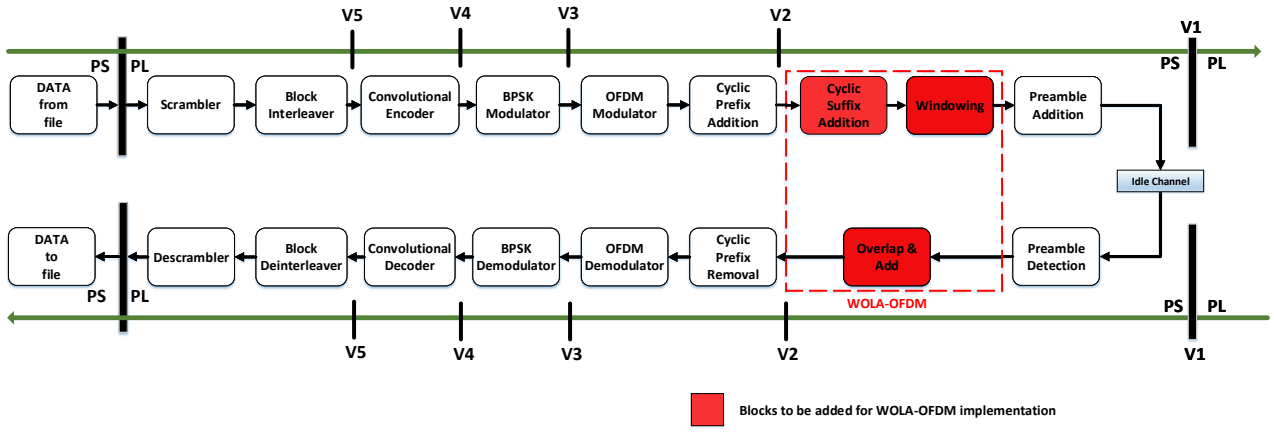

Figure 3.8: Block diagram of the WOLA-OFDM based transceiver with five different versions indicating division between PS and PL.

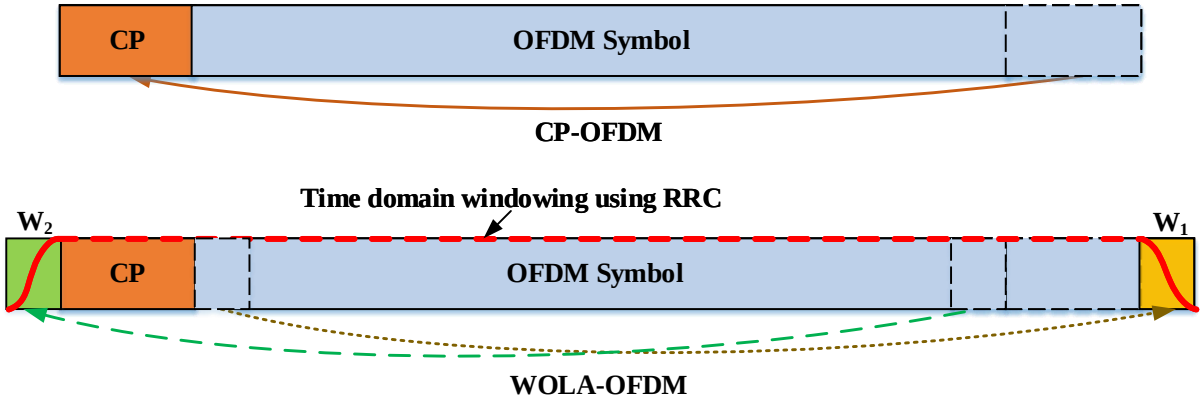

Figure 3.9: Time domain windowing at the transmitter in WOLA-OFDM

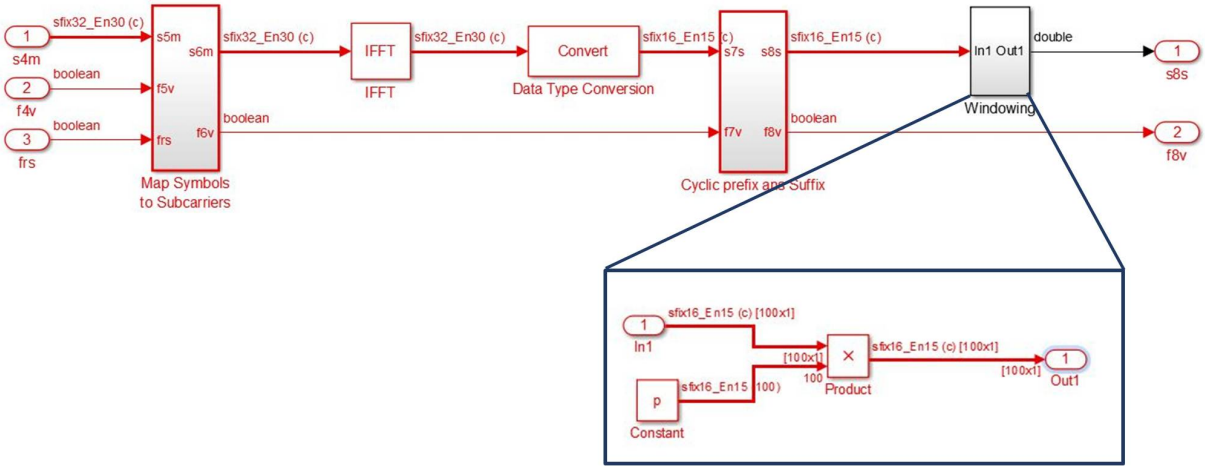

Figure 3.10: PS implementation of WOLA-OFDM tranmsitter

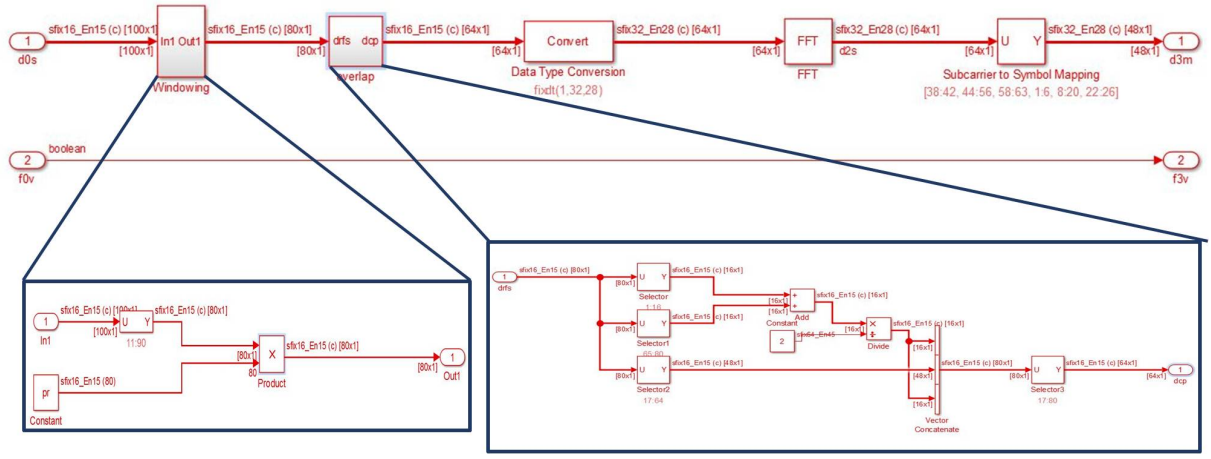

Figure 3.11: PS implementation of WOLA-OFDM receiver

Next, the OFDM symbol is multiplied with the pre-defined RRC window. The taper region of the window is kept to be equal to W.

However, when the windowing is shifted from PS to PL (V3-V5), sample mode is needed which necessitates proper delay balancing. This is achieved by using valid signal based synchronization in which active valid signal indicates the availability of correct data on data bus. Valid signal is generated by each block for its subsequent block. This makes the WOLA-OFDM architecture more complex as compared to PS implementation(Fig. 3.12and Fig. 3.13). As shown in Fig. 3.8, the overlapping samples at the start and end of the received symbol are added together at the receiver to mitigate the data loss caused due to windowing.

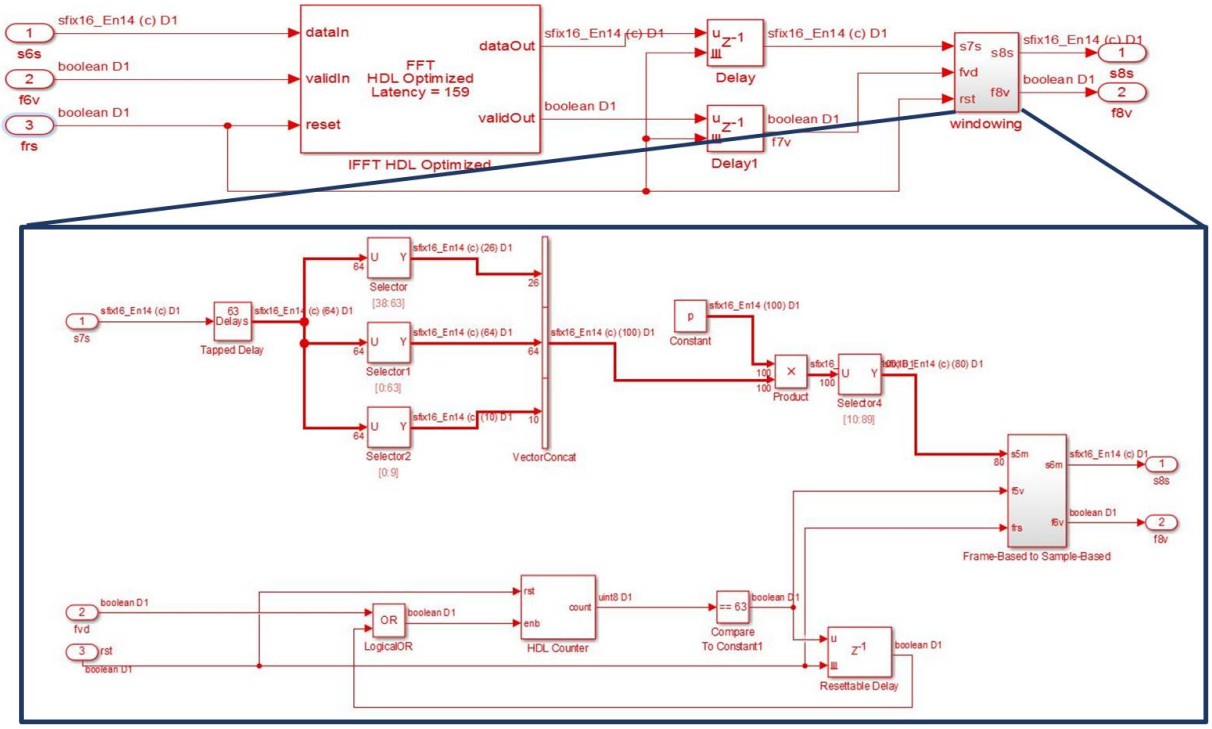

Figure 3.12: PL implementation of WOLA-OFDM tranmsitter

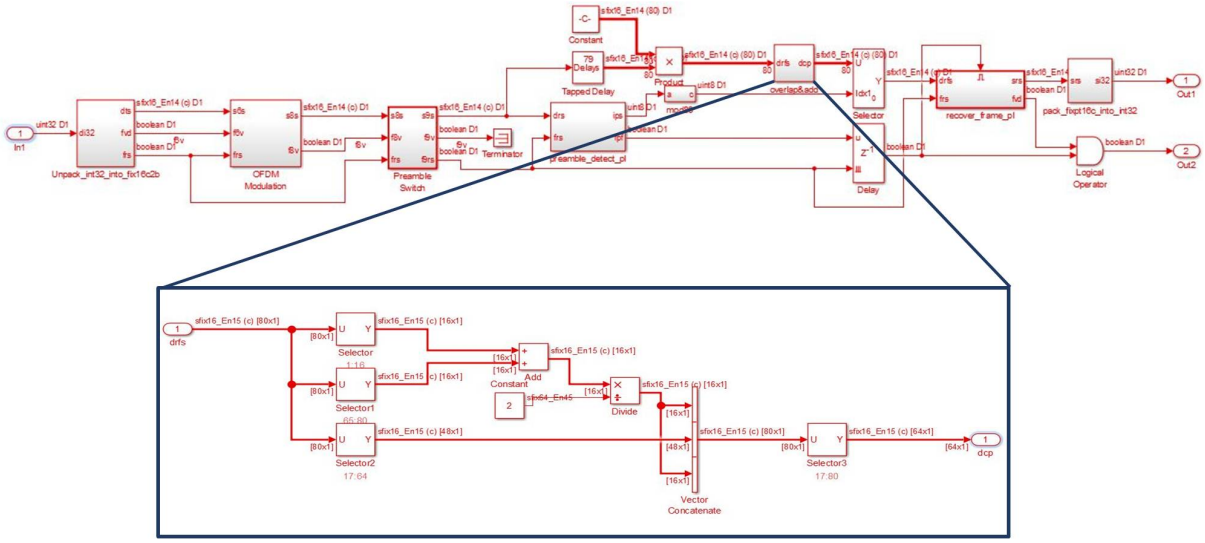

Figure 3.13: PL implementation of WOLA-OFDM receiver

3.3 Filtered OFDM (F-OFDM)

In F-OFDM, the time domain windowing in WOLA-OFDM is replaced with frequency domain filtering to achieve further improvement in out-of-band attenuation. As shown in Fig. 3.14, the F-OFDM uses linear phase finite impulse response filter after OFDM modulation. The use of F-OFDM [4, 15, 16] in enables higher transmission bandwidth compared to bandwidth limitation in WOLA-OFDM based system. In addition, sub-band filtering approach of F-OFDM allows

transmission in non-contiguous bands and sharing of adjacent frequency band among multiple asynchronous transceivers. The use of filters however leads to increase in the complexity of the transceiver when compared to WOLA-OFDM and OFDM.

In the proposed F-OFDM transceiver, we have used linear phase bandpass filter [17, 18] of order 150 with a normalized bandwidth of 0.86 and the transition bandwidth of 0.02. When the signal bandwidth is reduced, the filter passband bandwidth is also changed accordingly. A symmetric coefficient filter is used to reduce the area and power utilization. Appropriate processing mode (sample/frame based) is defined for the filter depending on whether it is being implemented on PS or PL. Filtering is convolution in time domain of the input samples and the filter coefficients. Thus, theoretically, number of output samples (N_o) is given by:

$$N_o = N_i + H - 1$$

where N_i is the number of input samples and H is the filter order. While implementing the filter in PS in the version V1, the number of output samples is equal to the number of input sample.

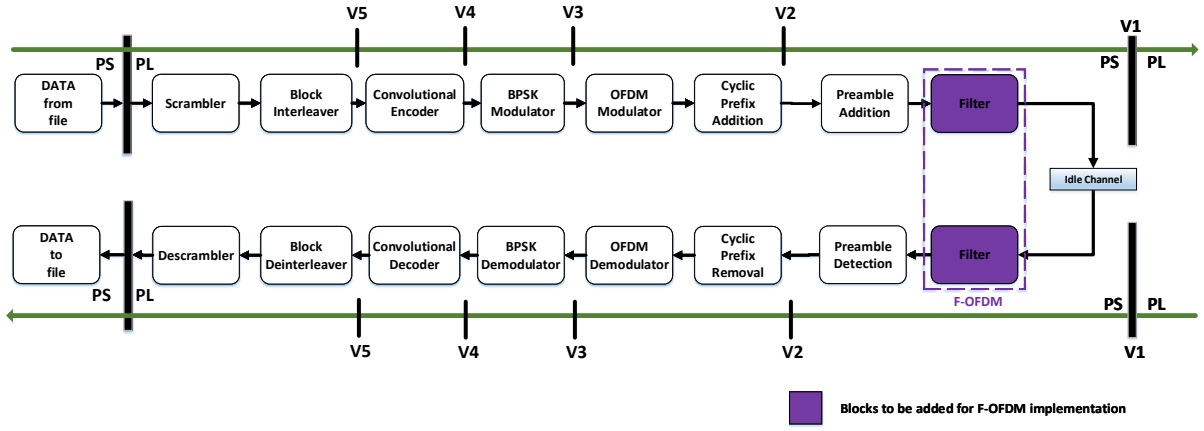

Figure 3.14: Block diagram of the F-OFDM based transceiver with five different versions indicating division between PS and PL.

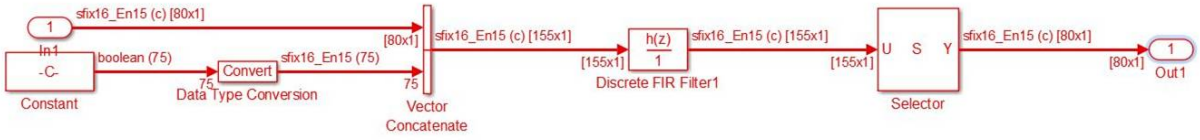

Figure 3.15: PS implementation of filter

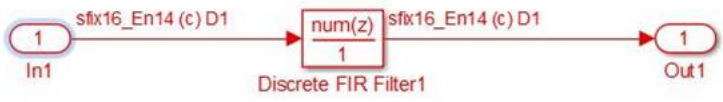

Figure 3.16: PL implementation of filter

This causes loss of data. Thus, to eradicate this error $H/2$ zeros are padded at the end of the filter input as shown in Fig. 3.15. No such processing is required in rest of the versions where the filter is implemented in PL (V2-V5). This is due to the fact that in PL implementation, the filter works in sample based mode where it takes one sample at a time as the input(Fig. 3.16).

Finally, the architecture is pipelined to improve the speed at which it can be clocked. This is done by identifying the critical path delay and inserting the registers at appropriate locations.

3.4 Summary

In this chapter, the proposed waveform designs and architectures are discussed. Various variants of the architecture are realized by dividing it into two sections, one for PL and other for PS. We demonstrate the flexibility offered by co-design approach to decide which part of the transceiver to implement on programmable logic (PL) such as FPGA and which on processing system (PS) such as ARM. The next chapter experimental results are presented.

Chapter 4

Results and Discussion

In this section, we compare the performance of various variants of transceivers using OFDM, WOLA-OFDM and F-OFDM realized on ZSoC. The setup requires a host PC, a ZC706 evaluation board and a JTAG and Ethernet cable along with softwares mentioned in the previous sections as shown in Fig. 4.1. The PL sample time (tps) is fixed to $1\mu s$ and correspondingly, the PS frame time (tpf) is $80\mu s$ since each frame consists of 80 samples.

$$tpf = tps * ns \tag{4.1}$$

where ns is the number of samples per frame.

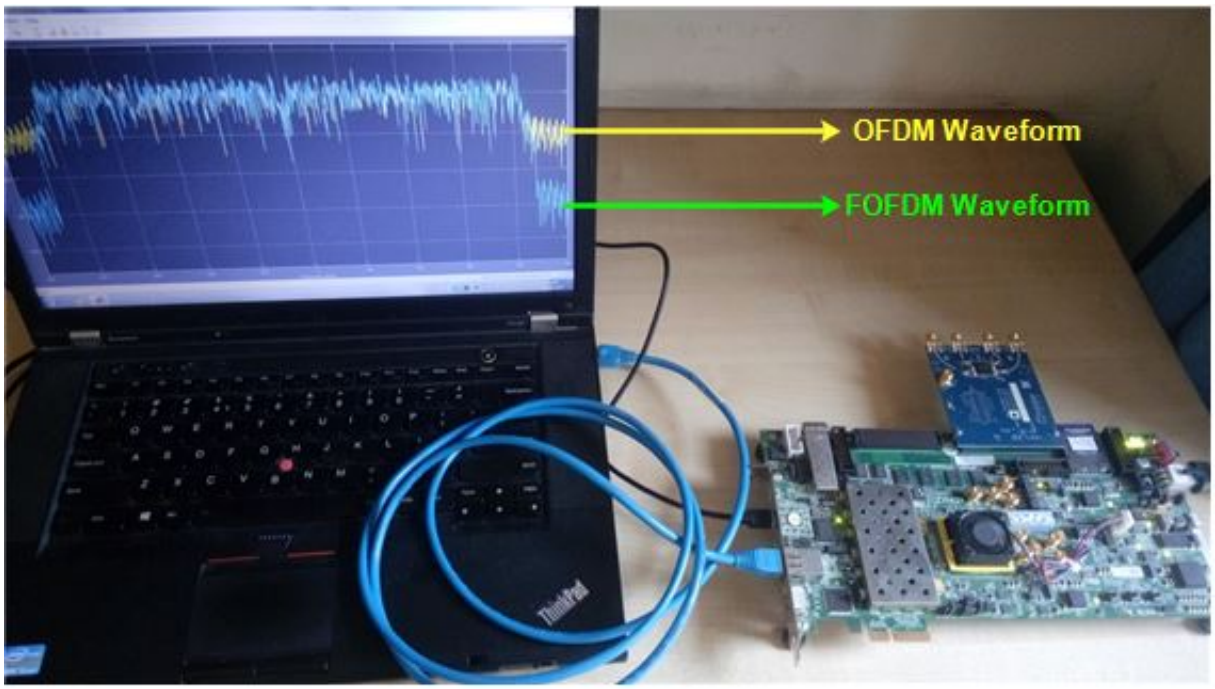

Figure 4.1: Demonstration setup: Real-time Transmitter output waveform for OFDM & F-OFDM

Table.4.1 defines the various parameter values used during the implementation.

Table 4.1: Parameter Values

Parameter	Value	Value
Symbol/Frame Time	80 μs	80 μs
Subcarrier spacing ($\Delta f = 1/tpf$)	12.5 KHz	12.5 KHz
FFT size	64	64
CP length	16	16
Number of Pilot Subcarriers	4	4
Number of Active Subcarriers(N_a)	52	28
Signal bandwidth($\Delta f * N_a$)	650 KHz	350 KHz

4.1 Out-of-Band attenuation

The performance of the proposed waveforms is compared with the traditional OFDM based system on the basis of their power spectral density (PSD) plots. The PSD plots of three waveforms for the transmission bandwidths of 650 KHz and 350 KHz are shown in Fig. 4.2 and Fig. 4.3 respectively. It can be observed that WOLA-OFDM and F-OFDM offer better performance than OFDM and only F-OFDM meets the desired out-of-band attenuation constraints at wide bandwidth of 650 KHz and narrow bandwidth of 350 KHz.

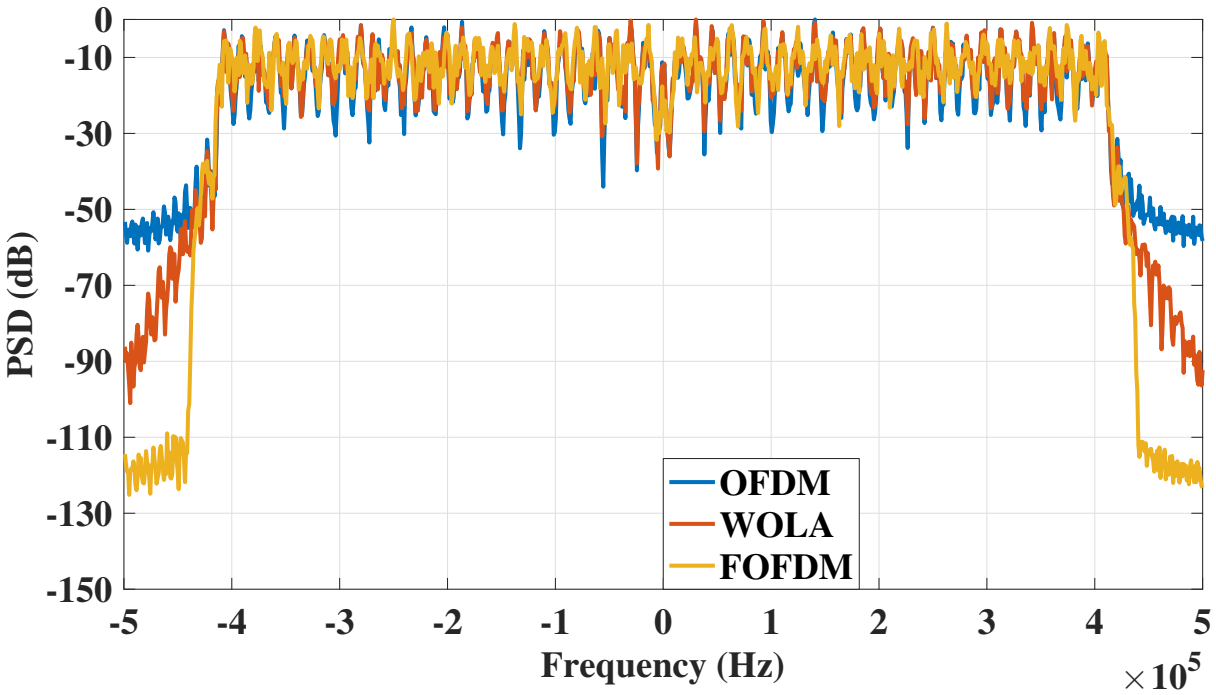

Figure 4.2: Power spectral density comparison of OFDM, WOLA-OFDM and F-OFDM.

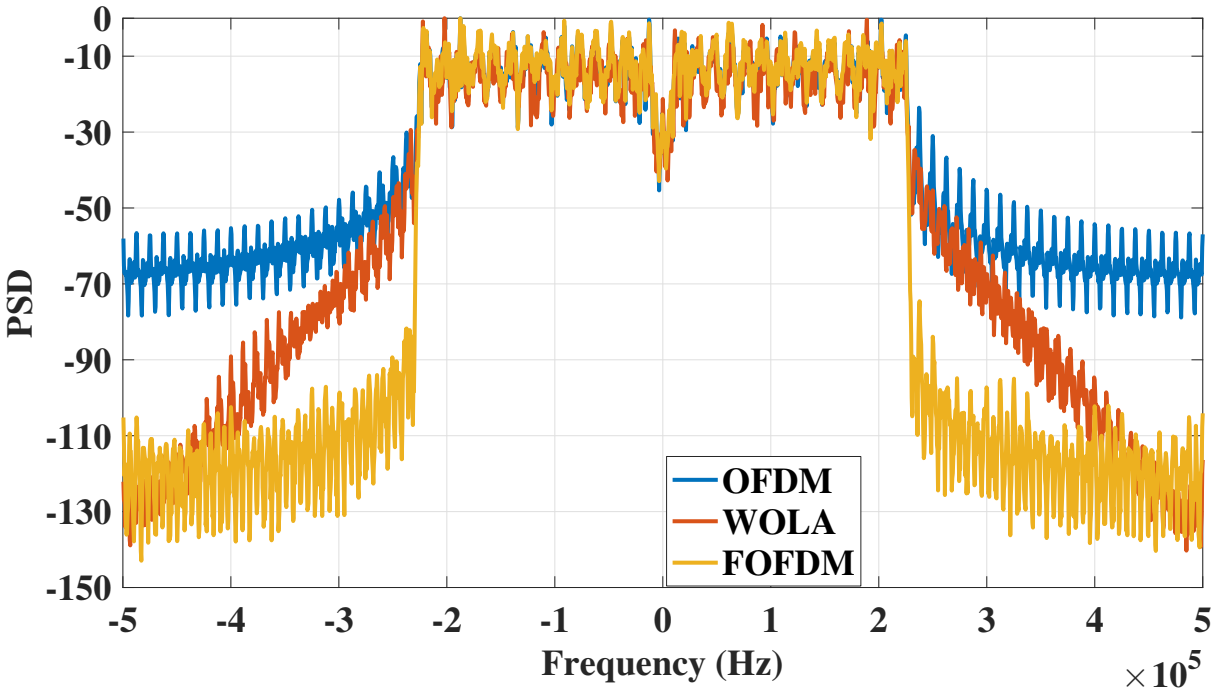

Figure 4.3: Power spectral density comparison of OFDM, WOLA-OFDM and F-OFDM.

4.2 Resource Utilization and Power results

Next, we compare the implementation results of the various versions of the transceivers as shown in Table. 4.2 and Table. 4.3. The transceiver system architectures have been optimized to attain minimum area utilization. To begin with, we compare the number of DSP48 slices used in each case. These are hardware embedded units specialized to efficiently perform various operations like multiply-accumulator, multiply-adder, counters etc. Usually, FIR filter are realized using these DSP48 elements. In the proposed architectures, we have also used DSP48 for preamble block implementation. As shown in Table. 4.2 and Table. 4.3, DSP48 utilization is around 30% higher in F-OFDM when compared to OFDM and WOLA-OFDM mainly due to filtering operation. There is slight increase in DSP48 utilization in all architectures from V2 to V3 due to FFT/IFFT operations being moved to PL.

Next, we compare the number of slices and LUTs which are the main building blocks of PL. It can be observed that F-OFDM has higher utilization for V2 while WOLA-OFDM has higher utilization for the rest of the versions due to windowing operations which is intentionally implemented without the use of DSP48. As result, the number of multiplexers utilization is higher in WOLA-OFDM due to overlap and addition operations.

Registers are mainly used in filtering and pipelining operations. Pipelining refers to the method of reducing the critical path delay of the architecture via cut-set retiming. Cut-set retiming involves identification of critical path followed by application of cut-sets and addition of delay units (hence, registers). The reduction in the critical path delay leads to increase in the clock frequency at which architecture can be clocked. As expected, the utilization of registers is

Table 4.2: Experimental Results on ZSoC for 650 KHz bandwidth signal

Parameter	Waveform	V1	V2	V3	V4	V5
No. of DSP48 units	OFDM	NA	554 (61.55%)	570 (63.33%)	570 (63.33%)	570 (63.33%)
	F-OFDM	NA	850 (94.44%)	866 (96.22%)	866 (96.22%)	866 (96.22%)
	WOLA-OFDM	NA	554 (61.55%)	570 (63.33%)	570 (63.33%)	570 (63.33%)
No. of slices	OFDM	NA	8508 (15.57%)	11086 (19.89%)	11169 (20.44%)	11562 (21.16%)
	F-OFDM	NA	10695 (19.57%)	12915 (23.63%)	13838 (25.32%)	15162 (26.01%)
	WOLA-OFDM	NA	8508 (15.57%)	13258 (24.26%)	14157 (25.90%)	14256 (26.09%)
No. of LUTs	OFDM	NA	22083 (10.102%)	28843 (13%)	29568 (14%)	30970 (14.167%)
	F-OFDM	NA	25379 (11.66%)	30703 (14.045%)	33321 (15.243%)	34782 (15.423%)
	WOLA-OFDM	NA	22083 (10.102%)	36284 (16.598%)	38613 (17.664%)	39348 (18%)
No. of Multiplexers	OFDM	NA	35	683	1144	1217
	F-OFDM	NA	35	683	1144	1217
	WOLA-OFDM	NA	35	1131	1597	1673
No. of Registers	OFDM	NA	1130	2050	2431	2432
	F-OFDM	NA	1736	2656	3037	3038
	WOLA-OFDM	NA	1130	2184	2522	2523
Dynamic Power (in Watts)	OFDM	NA	1.98	1.97	1.97	1.97
	F-OFDM	NA	2.23	2.17	2.31	2.71
	WOLA-OFDM	NA	1.98	1.97	1.97	1.97

slightly higher in F-OFDM. The critical path delay of all architecture is identical and equal to 259 ns with pipelining which corresponds the operating frequency of 3.86 MHz. Overall, power consumption in F-OFDM is slightly high due to higher number of DSP48. However, the PS power remains constant at 1.56 W for all the models.

Thus, overall, F-OFDM offers better spectral containment but at the risk of exhausting the resources and higher power consumption. It is also observed that the FPGA resource utilization will increase as we move block to PL from V1-V5, while the PS step time will decrease. However, in all the configurations, the utilization of the FPGA slices and LUTs for the PL section is less than 27% and 35% respectively leaving enough resources for higher layers. The above results have been obtained from Xilinx Vivado post implementation resource utilization reports generated for each model. The power results have been obtained by running Vivado power report for a

Table 4.3: Experimental Results on ZSoC for 350 KHz bandwidth signal

Parameter	Waveform	V1	V2	V3	V4	V5
No. of DSP48 units	OFDM	NA	594 (66%)	610 (67.783%)	610 (67.783%)%)	610 (67.783%)
	F-OFDM	NA	898 (99.78%)	900 (100%)	900 (100%)	900 (100%)
	WOLA-OFDM	NA	594 (66%)	610 (67.783%)	610 (67.783%)%)	610 (67.783%)
No. of slices	OFDM	NA	5995 (10.97%)	7779 (14.23%)	8095 (14.81%)	10837 (19.83%)
	F-OFDM	NA	8023 (14.68%)	11414 (20.89%)	11167 (20.43%)	11371 (20.52%)
	WOLA-OFDM	NA	5995 (10.97%)	10300 (18.85%)	11197 (20.49%)	11268 (20.62%)
No. of LUTs	OFDM	NA	13664 (6.25%)	18851 (8.62%)	19032 (8.71%)	27918 (12.77%)
	F-OFDM	NA	15979 (7.31%)	24066 (11.01%)	23421 (10.71%)	23421 (10.71%)
	WOLA-OFDM	NA	13664 (6.25%)	26058 (11.92%)	26972 (12.15%)	26979 (12.16%)
No. of Multiplexers	OFDM	NA	35	683	1071	1071
	F-OFDM	NA	35	683	1071	1071
	WOLA-OFDM	NA	35	1131	1524	1524
No. of Registers	OFDM	NA	1130	2050	2382	2382
	F-OFDM	NA	1736	2656	2988	2988
	WOLA-OFDM	NA	1130	2184	2476	2476
Dynamic Power (in Watts)	OFDM	NA	2.02	1.94	1.95	1.97
	F-OFDM	NA	2.37	2.27	2.31	2.71
	WOLA-OFDM	NA	2.02	1.94	1.95	1.97

fixed environment.

4.3 Timing Information

Table. 4.4 and table. 4.5 depict the total time taken per frame by the processor to implement the function targeted on it. It can be seen that the PS operation time decreased as we move the system components to PL. Due to the addition of filter to the archtecture, FOFDM has the highest operation time among the three waveforms. We can see that the data packing and unpacking operations required in V2-V5 are highly time consuming, thus leading to an increase in PS frame time in V2 as compared to V1 in case of OFDM and WOLA-OFDM. However, in case of F-OFDM, moving the filtering operation to PL compensates for this increase. These results have been generated with the help of embedded coder's real-time code execution profiling

Table 4.4: PS frame time for 650 KHz bandwidth signal in ms

Waveform	**V1**	**V2**	**V3**	**V4**	**V5**
OFDM	5.06	7.51	7.42	7.37	7.35
FOFDM	10.21	7.55	7.44	7.37	7.35
WOLA	5.47	7.59	7.35	7.35	7.31

Table 4.5: PS frame time for 350 KHz bandwidth signal in ms

Waveform	**V1**	**V2**	**V3**	**V4**	**V5**
OFDM	4.39	7.50	7.42	7.39	7.31
FOFDM	10.2	7.48	7.39	7.39	7.31
WOLA	4.40	7.55	7.39	7.35	7.31

tool.

4.4 Summary

In this chapter, detailed performance and complexity analysis of various candidate waveforms for wireless transceiver system on Zynq System on Chip (ZSoC) platform, consisting of programmable logic (PL) such as FPGA and processing system (PS) such as ARM, is presented. Detailed experimental results demonstrate the trade-off between these waveforms with respect to parameters such as out-of-band attenuation, area, delay and power requirements. It is also observed that the FPGA resource utilization will increase as we move block to PL from V1-V5, while the PS step time will decrease. Also, among the different co-designs, we see maximum increase in resource utilization when IFFT/FFT operations are implemented on PL and correspondingly maximum decrease in the PS frame time is also observed. Analyzing the results, we could conclude that WOLA-OFDM, has slighter higher area requirement compared to OFDM and F-OFDM, but has better spectral containment compared to OFDM with no increase in power requirement and delay. F-OFDM offers significantly high OOB attenuation as compared to OFDM and WOLA-OFDM but at the cost of higher resource utilization, power consumption and delay. However, in all the configurations, the utilization of the FPGA slices and LUTs for the PL section is less than 27% and 35% respectively leaving enough resources for higher layers, thus making F-OFDM a promising candidate for upcoming wireless communication systems.

Chapter 5

Conclusion and Future Works

This chapter presents a brief summary of work done in this thesis. Some future works linked to the same field of research are also identified.

5.1 Conclusion

In this thesis work, detailed performance and complexity analysis of various candidate waveforms for wireless transceiver system on Zynq System on Chip (ZSoC) platform, consisting of programmable logic (PL) such as FPGA and processing system (PS) such as ARM, is presented. Various configurations of the architecture are realized by dividing it into two sections, one for PL and other for PS. We demonstrate the flexibility offered by co-design approach to decide which part of the transceiver to implement on programmable logic (PL) such as FPGA and which on processing system (PS) such as ARM to meet the given area, delay and power constraints. Detailed experimental results demonstrate the trade-off between these waveforms with respect to parameters such as out-of-band attenuation, area, delay and power requirements.

5.2 Future Work

Some future works to purse research in this area identified are:

1. **Implementation of FBMC based transceiver system on ZSoC**
   The proposed architecture will be extended to filter bank multi-carrier (FBMC) based transceiver system. FBMC is another proposed waveform the the upcoming 5G technology and is expected to perform better than the existing waveforms.

2. **Implementation of GFDM based transceiver system on ZSoC**
   Generalized Frequency Division Multiplexing (GFDM) is another candidate waveform being proposed for high speed wireless communication. Our study can be extended to implementation and performance analysis of GFDM on ZSoC.

3. **Over-the-air transmission using F-OFDM and WOLA-OFDM based RF transceivers** Future work includes realization of over-the-air transmission using RF transceivers to analyze the effect on performance in real radio environment. We would also like to analyze the effect of serialization of the architecture in PL on the area, delay and power complexity.

Appendix A

Steps for implementation of QPSK HW-SW cosimulation example

A.1 Software Installation

1. Install MATLAB 2016a or higher versions.
2. Install Xilinx Vivado 2015.2. Upgrade the software to Vivado 2015.2.1. Remember to install the SDK.
3. Install the following Hardware Support Packages in MATLAB
   Add Ons $->$ Get Hardware Support Package $->$ Install from Internet $->$
   Install the following packages:
   3.1. ARM Cortex A Embedded Coder
   3.2. HDL Coder
   3.3. HDL Verifier (Required for FPGA-In-Loop implementation)
   3.4. Communication Systems Package
   3.5. Zynq SDR 7000 package
4. Follow the steps of the target-updater as asked.
   4.1. Set the IP address of the board to be 192.168.3.2 (The host IP should be set to 192.168.3.1 or any other on the network)
   4.2. Load the SD Card.
   4.3. The zynq hardware needs to be configured. Check that the ethernet, UART and JTAG connections have been made. Run the following commands on MATLAB command window:
   >>h=zynq;
   >>h.setupZynqHardware();

A.2 Open the example and Configure MATLAB

1. Set up the path to Vivado. This is required for HDL Code generation.
   >>hdlsetuptoolpath('ToolName','Xilinx Vivado','ToolPath','C:/Xilinx/Vivado/2015.2/bin/vivado.bat');
   >>setupzynqradioipcoregen

2. Open Example
   >>zynqRadioHWSWQPSKAAD9361AD9364SL

A.3 Generate the Hardware-Software Model

A.3.1 HDL Workflow Advisor

1. Right click on the HDL_QPSK block $->$ HDL Code $->$ HDL Workflow Advisor

2. In step 1.1 set the following parameters:

   2.1. Target Workflow: IP Core Generation

   2.2. Target Platform: ZC706 & FMCOMMS2/3/4

   2.3. Synthesis tool: Xilinx Vivado

   2.4. Specify the folder name

3. Now, pres **Run This Task**

4. You will now see that, the parameter values are automatically loaded as per your hardware specifications

5. In 3.1.1, set the options as per your requirement.

6. Right click on 4.3 and press **Run to this task**. This will generate the bit file to be loaded on the FPGA of the zynq board. Select the option for run the process externally to free the MATLAB.

7. An interface model will be generated along with a library model containing the AXI interface blocks for the Tx and RX, and a DUT black box(for the HDL part).

A.3.2 Hardware/Software interface model

1. To open the interface model, type the following command in command window:
   >>open_system('zynqRadioHWSWQPSKAD9361SL_interface')

2. Copy the contents of this model to the newly generated interface model. Use the appropriate interface block according to your target hardware and comment out the original contents of the model.

3. The following modifications are required to be done in the generated interface model:

   3.1. File − > Model Properties − > copy the init, preload and pstload functions from the golden model to your model.

   3.2. Model Configuration parameters − > Code generation − > Interface − > select the 'Variable Size Signals' − > press OK.

   3.3. Also check that the solver type is set to 'Fixed step' and solver to 'Discrete (no continuous state)'

   3.4. in the interface model, open 'For Iterator Subsystem/MATLAB function' and change line 16 to *endIdx=startIdx+MessageLength-1;*

   3.5. Save the model and reopen for the changes to take effect.

A.4 Run the Model

1. Once the systen_wrapper.bit file is generated,then

2. Connect the board (JTAG and Ethernet) to the host computer and power it on.

3. Setup the target board and test the connections using following commands:
   >> *hdev=sdrdev('ZC706 and FMCOMMS2/3/4');*
   >> *hdev.info*
   >> *testConnection(hdev)*
   >> *downloadImage(hdev,'FPGAImage',hdl_prj_test3\vivado_prj.runs\impl_1\system_wrapper.bit');*

4. Open the interface model.

5. Change the simulation mode to **External**.

6. **RUN** the simulation.

7. A command prompt will open showing the status of the model and output.

8. Try changing the input mode through the switch in the model and notice the change in output.

9. Since the simulation time is set to infinity, you will have manually stop the simulation in order to exit.

Bibliography

[1] N. Agrawal, S.J. Darak and F. Bader, "Reconfigurable Filtered OFDM Waveform for Next Generation Air-to-Ground Communications", in *36th AIAA/IEEE Digital Avionics Systems Conference (DASC)*, St. Petersburg, FL, USA, Sept. 2017.

[2] L. Ma and C. Zhang, "5G wareforms design for aeronautical communications ", in *35th IEEE/AIAA Digital Avionics Systems Conference (DASC)*, pp. 1–7, Sacramento, CA, USA, Sept. 2016.

[3] N. Neji, R. de Lacerda, A. Azoulay, T. Letertre and O. Outtier, "Survey on the Future Aeronautical Communication System and Its Development for Continental Communications", in *IEEE Transactions on Vehicular Technology*, vol. 62, no. 1, pp. 182–191, Jan. 2013.

[4] J. Abdoli, M. Jia and J. Ma, "Filtered OFDM: A new waveform for future wireless systems", in *16th IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 66–70, Stockholm, Sweden, June. 2015.

[5] R. Zayani, Y. Medjahdi, H. Shaiek and D. Roviras, "WOLA-OFDM: A Potential Candidate for Asynchronous 5G", in *IEEE Globecom Workshops*, pp. 1–5, Washington, DC, Dec. 2016.

[6] "Zynq-7000 All Programmable SoC Overview", *Application Processor Unit*, 2012.

[7] B. Drozdenko, M. Zimmermann, Tuan Dao, M. Leeser and K. Chowdhury, "High level hardware software codesign of an 802.11a transceiver system using Zynq SoC ", in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 682-683, San Francisco, CA, Apr. 2016.

[8] B. Drozdenko, M. Zimmermann, Tuan Dao, M. Leeser and K. Chowdhury, "Hardware Software Codesign of Wireless Transceivers on Zynq Heterogeneous Systems ", in *IEEE Transactions on Emerging Topics in Computing*, vol. PP, pp. 99, May. 2017.

[9] R. Dobai and L. Sekanina, "Towards evolvable systems based on the Xilinx Zynq platform", in *IEEE International Conference on Evolvable Systems (ICES)*, Singapore, pp. 682-683, Apr. 2013.

[10] L.H. Crockett, R.A. Elliot, M.A. Enderwitz and R.W. Stewart, "The Zynq Book: Embedded Processing with the Arm Cortex-A9 on the Xilinx Zynq-7000 All Programmable Soc", in *Strathclyde Academic Media,* 2014.

[11] UG761, "Design Reference Guide," *AXI Xilinx,* Mar. 2011.

[12] R. Nee and R. Prasad, "OFDM for wireless multimedia communications",in *Artech House, Inc.,* 2000.

[13] C. Langton, "Orthogonal frequency division multiplex (OFDM) tutorial ",in *Intuitive guide to principles of communications,* 2004.

[14] F. Wu and Mosa Ali Abu-Rgheff, "Time and frequency synchronization techniques for OFDM systems operating in gaussian and fading channels: A tutorial", in *Proc. of the 8th Annual Postgraduate Symposium on The Convergence of Telecommunications, Networking and Broadcasting (PGNET)*, Liverpool, UK, June. 2007.

[15] X. Cheng, Y. He, B. Ge and C. He, "A Filtered OFDM Using FIR Filter Based on Window Function Method,"in *83rd IEEE Vehicular Technology Conference (VTC Spring),* pp. 1–5, Nanjing, China, May. 2016.

[16] A. Farhang, M. Molavi Kakhki and B. Farhang-Boroujeny, "Wavelet-OFDM versus filtered-OFDM in power line communication systems,"in *5th IEEE International Symposium on Telecommunications (IST),* pp. 691–694, Tehran, Iran, Dec. 2010.

[17] S. J. Darak, A. P. Vinod, E. M-K. Lai, Honggang Zhang and Jacques Palicot, "Linear Phase VDF Design with Unabridged Bandwidth Control over the Nyquist Band,"in *IEEE Transactions on Circuits and Systems - II (TCAS-II)*, vol. 61, no. 6, pp. 428-432, Apr. 2014.

[18] S. J. Darak, A. P. Vinod, and E. M-K. Lai, "A Low Complexity Recongurable Non-uniform Filter Bank for Channelization in Multi-standard Wireless Communication Receivers,"in *Journal of Signal Processing Systems (Springer)*, vol. 68, no. 1, pp.95-111, July 2012.