A Filtered Multitone (FMT) Overlay Implementation with Custom Instructions on an Altera FPGA

Xin Xin

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Electrical Engineering

Peter M. Athanas, Chair
Jeffrey H. Reed
Carl B. Dietrich

April 24, 2013
Blacksburg, Virginia

Keywords: Multicarrier Communication, FMT, Overlay, Cognitive Radio, Altera FPGA

A Filtered Multitone (FMT) Implementation with Custom Instructions on an Altera FPGA

Xin Xin

(ABSTRACT)

There is a belief that radio frequencies are running out. However, according to a report from the Federal Communications Commission (FCC) in 2002, a different story was told : At any given time and location, much of the prized spectrum lies idle. At the same time, FCC revealed the fact that, in many bands, spectrum access is a more significant problem than physical scarcity of spectrum, in large part due to legacy command-and-control regulation that limits the ability of potential spectrum users to obtain such access. Hence, as opposed to static spectrum access, dynamic spectrum access (DSA) was proposed to solve the predicament. One such DSA model propose the existence of Primary users (licensed users) and Secondary users (unlicensed users). Multicarrier communication technology is adopted to enable the coexistence of PU and SU. Orthogonal Frequency Division Multiplexing (OFDM) technology has been popular for multicarrier communications. A disadvantage for OFDM in the Cognitive Radio environment is its large side lobes in the frequency domain, which is a result of single-symbol pulse duration. Filter Bank Multicarrier (FBMC) uses filters that have small side lobes to synthesize/analyze the sub-carriers so as to greatly alleviate the previous mentioned disadvantage. FMT is one FBMC technique. Although many hardware implementations have been explored during last few decades on OFDM, few FMT hardware implementation results, especially Hardware/Software Co-design, have been presented. This paper presents a HW/SW Co-design implementation result of FMT transceiver on the Altera DE4 board.

Acknowledgements

Looking back to the endeavor to successfully complete this thesis as a journey, I received inspiration, guidance and support from many people. It is with immense gratitude that I acknowledge the support and help of those who helped me along this journey directly and indirectly.

First and foremost, I wish to thank Dr.Peter M.Athanas. Without his support, it would had been impossible to complete the research work and thesis writing. I would also like to thank Dr. Jeffrey H. Reed and Dr. Carl B. Dietrich for serving as members of my committee. An honor, I considered as, to work with all the members in Configurable Computing Machine Lab (CCM Lab).

In the end, I owe my deepest gratitude and am indebted to my parents. Without their self-giving support in known and unknown situations, it would be out of the question to even conceive the slightest idea of accomplishing this thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction to Cognitive Radio

1.1 Background

Radio spectrum is a very precious resource. In 2002, the Federal Communications Commission (FCC) published a report [4], which revealed an interesting fact: "In many bands, spectrum access is a more significant problem than physical scarcity of spectrum, in large part due to legacy command-and-control regulation that limits the ability of potential spectrum users to obtain such access." In [5], Haykin points out if scans were made to spectrum usage in urban area, it can be found that: 1) some frequency bands in the spectrum are largely unoccupied most of the time; 2) some other frequency bands are only partially occupied; 3) the remaining frequency bands are heavily used. This further implies if *spectrum holes* [5] can be utilized, spectrum utilization efficiency can be enhanced. Moreover, in [5], the author provided a definition of *spectrum holes* as below: "a spectrum hole is a band of frequencies assigned to a primary user, but, at a particular time and specific geographic location, the band is not being utilized by that user." Making use of these available spectrum holes indicates the use of dynamic spectrum access, as opposed to the previous static spec-

trum access. Under this background, Cognitive Radio (CR) was coined by J.Mitola [6, 7]. While there has not been a universally acknowledged definition of cognitive radio, by nature, it usually refers to a radio that can automatically detect its environment, and make changes to its transmission spectrum or other parameters accordingly and rapidly.

1.2 Motivation

The users who paid for the expensive band license are called Primary Users (PU). Standing on the opposite side, Secondary Users (SU) are unlicensed users. As a matter of fact, PU do not utilize their expensive bands at all times or locations, which makes it possible for the SU to step in and utilizes the bands. The only premise is SU cannot interfere with PU while PU are active. Consequently, SU needs to sense the available spectrum all the time and adapt itself accordingly and rapidly. It is this rapid self-adaptation that ushers us to the adoption of Cognitive Radio. It is beneficial not only that it generates profit for PU from leasing the bands, but also SU are able to find available bands to transmit.

1.3 Cognitive Radio Cycle

Before discussing the basics of CR, let us probe into the term Software Radio or Software Defined Radio (SDR) since Cognitive Radio actually evolved from the idea of SDR. The term SDR was also first coined by J.Mitola in [8], where it refers to the class of re-programmable or reconfigurable radios. SDR is a radio system where components, e.g. mixers, filters, modulators/demodulators, that have been typically implemented in hardware are instead implemented by means of software on a personal computer or embedded systems. Thus, the same piece of hardware can perform different functions at different times. On top of SDR,

CR emphasizes the ability to sense the outside world and adapt itself to it accordingly and pro-actively. A cognition cycle is used to describe the functions of CR. Several editions of cognition cycle are published, e.g. [9]. The Mitola's seven-stage cognitive cycle is depicted in Fig. 1.1.

1. **Sensing.** The capability to sense not only the spectrum, but location, temperature and any other environmental parameter.
2. **Perception.** The capability to judge the information collected by sensors.
3. **Orienting.** Looking for the of operation parameters required under the perceived conditions.
4. **Planning.** Offering a wide range of alternatives in temporal calculus, constraint-based scheduling, task planning, etc.
5. **Making decisions.** Choosing the best action to perform.
6. **Taking action.** Producing changes on the environment, such as transmitting on a specific frequency band or asking other machines for additional information or commands.
7. **Learning.** A function of observations and decisions.

1.4 Dynamic Spectrum Management

As we discussed in the previous sections, a large part of the difficulty of obtaining an available bands are due to the spectrum management, rather than the spectrum scarcity. Dynamic spectrum management (DSM), also referred to as dynamic spectrum access (DSA), is a set

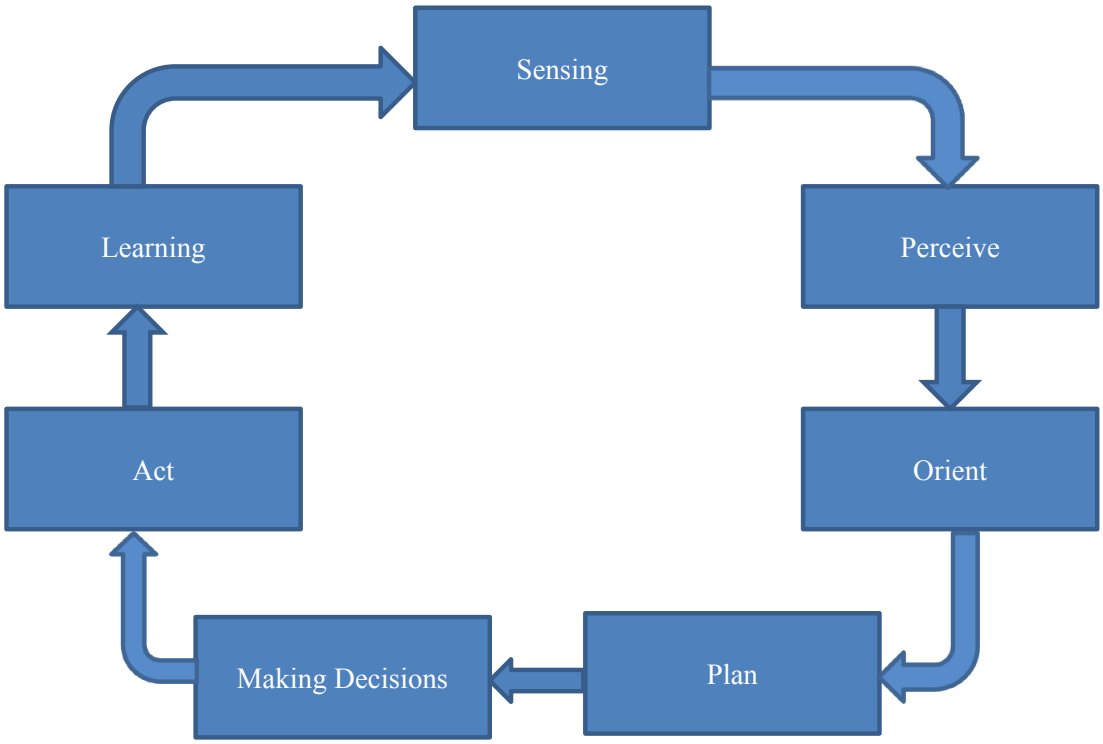

Figure 1.1: Mitola's Seven-stage cognitive cycle

of techniques based on theoretical concepts in network information theory and game theory that is being researched and developed to improve the performance of a communication network as a whole [10]. In [1], a taxonomy of DSA is presented, which is illustrated in Fig. 1.2. Detailed explanations of each item in Fig. 1.2 are listed below:

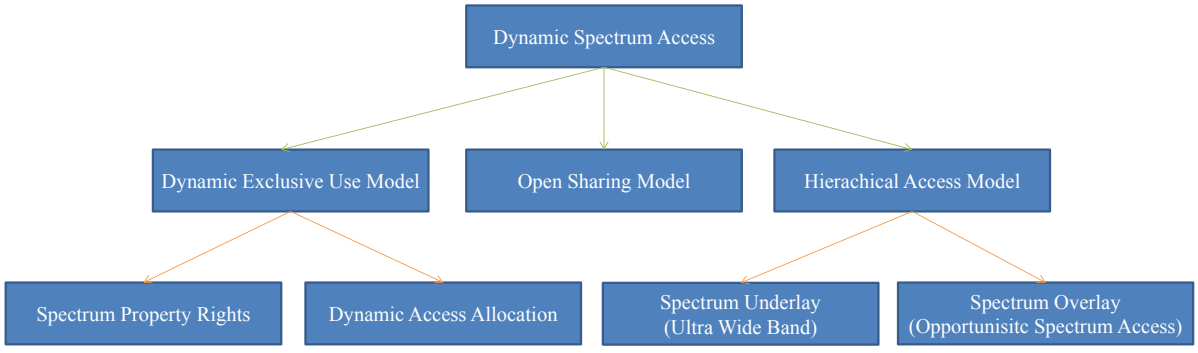

Figure 1.2: A taxonomy of dynamic spectrum access. [1]

Dynamic Exclusive Use. Spectrum bands are licensed to services for exclusive use. This model maintains the basic structure of the current spectrum regulation policy. Two approaches are included under this model:

1. **Spectrum Property Rights [11].** This approach allows licensees to sell and trade spectrum and to freely choose technology.
2. **Dynamic Spectrum allocation [12].** Dynamic spectrum assignment by exploiting the spatial and temporal traffic statistics of different services.

Open Sharing Model [13, 14]. This model employs open sharing among peer users as the basis for managing a spectral region. Wireless services operating in the unlicensed industrial, scientific bands support this model.

Hierarchical Access Model. This model proposes the existence of licensed Primary users (PU) with the highest priority for the use of the spectrum, and the unlicensed Secondary users (SU). Under this model, two spectrum sharing between PU and SU approaches are proposed: spectrum underlay and spectrum overlay. In Fig. 1.3 Part a, a spectrum underlay is displayed, where SU take advantage of the interference margin allowed by the PU and use the transmission modes that spread the signal over a wide band with low spectrum density. Since SU operate at the same frequency band as PU, this method practically lifts up the noise temperature for PU. In Fig. 1.3 Parts b and c, overlay systems are illustrated, where SU only use the frequency band where PU are not utilizing. The difference between OFDM overlay and FBMC overlay lies in the power leakage from SU to PU. As one may observe, the leakage from SU in OFDM overlay is higher than SU in FBMC overlay. This is due to an improvement method that FBMC adopted. We will discuss more on FBMC in the rest part of this thesis.

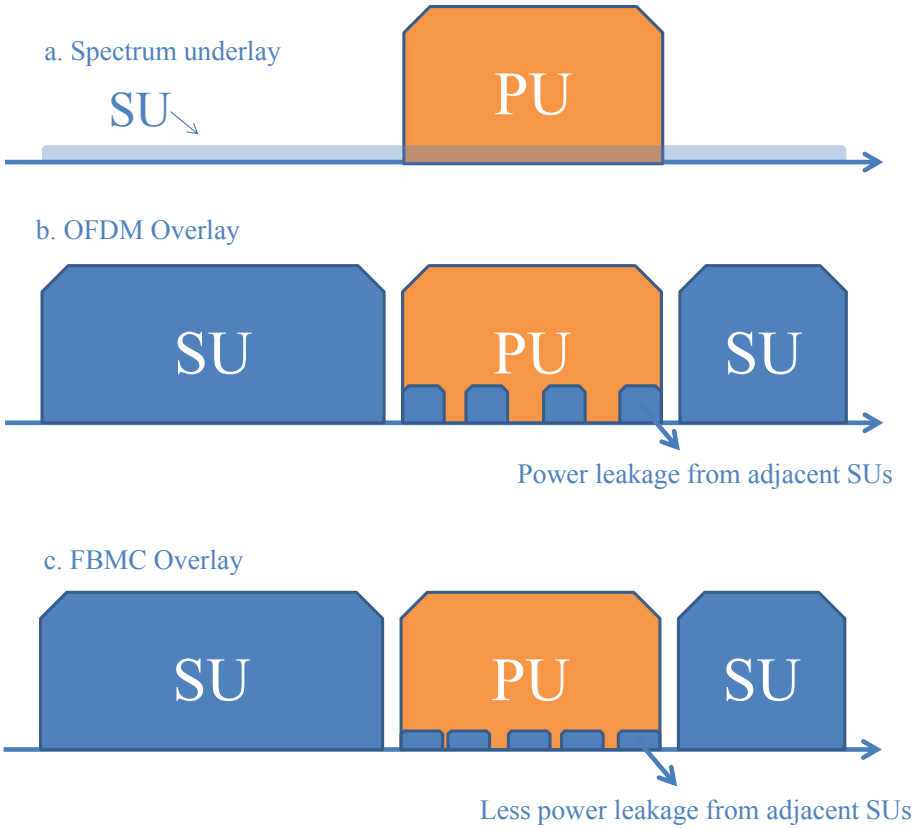

Figure 1.3: Overlay Versus Underlay

1.5 Why implementing FMT on an Altera FPGA

Augmenting an Altera NIOS II with custom instruction enables users to accelerate time-critical software algorithm by implementing it on hardware, on any design that has a NIOS II processor on it. Besides, custom instructions can be viewed as a method to integrate IPs into NIOS II. One merit of connecting IPs through custom instruction is to avoid excessive bus traffic. Plus, few FMT overlay implementations on Altera Devices were reported. For example, OFDM CR implementations on Xilinx FPGAs were reported in [15, 16], FMT underlay implementation on Xilinx FPGAs is available in [17], as well as FMT overlay implementation on Xilinx FPGAs from [3]. Moreover, in terms of easy implementation,

FMT favors us best as a FBMC technique. Hence, FMT overlay transceiver with custom instruction is implemented on an Altera FPGA.

1.6 Thesis Organization

The rest of this thesis is organized as below. In the next Chapter, OFDM basics will be reviewed. The large side lobe weakness of OFDM is brought up. Hence, in Chapter 3, OFDM evolution, including Filtered OFDM and FBMC, is illustrated. Filtered OFDM and FBMC both alleviated from suffering large side lobe leakage in frequency domain. Chapter 4 discusses Filter Multitone more in depth as a FBMC method. FMT hardware implementation is presented in the chapter after that. In the end, the integration of FMT engine and NIOS II is shown. Results and conclusions are stated in the final chapter.

Chapter 2

OFDM Basics

The nature of CR, which includes some degree of spectrum sensing and being able to adapt itself to make use of different spectrum holes, leads us to multicarrier communication technology. This is to say, as a CR transmitter, it has to be able to confine the spectral content of the transmitter within the selected bands, i.e. spectrum holes. Orthogonal frequency-division multiplexing (OFDM) is such an technology. Actually, OFDM was the first multicarrier technology for CR, and as of today, OFDM is still the dominant technology for broadband multicarrier communications. In this chapter we will review the basics of OFDM.

2.1 OFDM transceiver structure

A basic OFDM transceiver is shown in Fig. 2.1. Serial data is first mapped to serial symbols based on any linear constellation, e.g. BPSK, QPSK or 16-QAM. These symbols are considered as in frequency domain. Afterwards, they are converted to parallel and modulated onto each sub-carrier by IFFT operator. Cyclic prefix (CP) is added to avoid inter-symbol interference (ISI). This modulated signal is then converted back to serial format

again and transmitted through channel. On the receiver end, basically reverse operations are performed. The received signal is first converted to parallel format, and then remove CP, perform FFT operation and sent through a de-constellation mapper which demodulates out the serial data. The use of CP results in a loss of bandwidth efficiency. The basic concept of OFDM overlay is to assign a zero amplitude to the sub-carriers that are in the same frequency as the primary user.

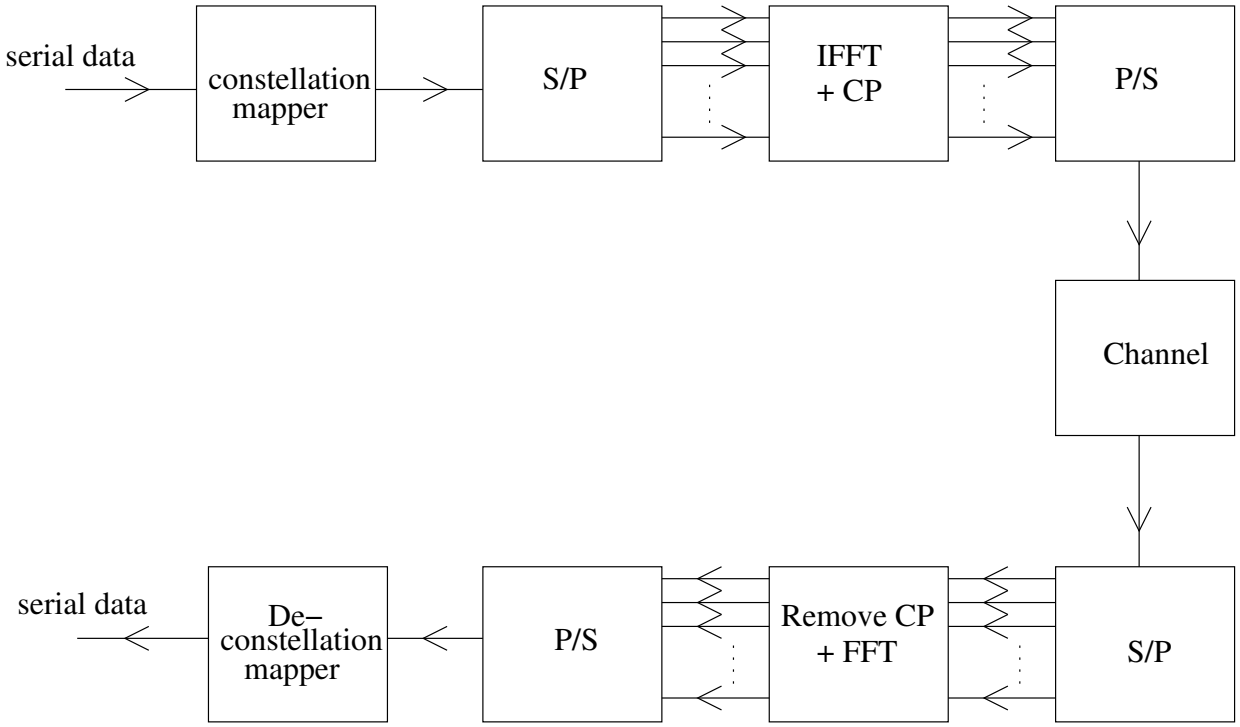

Figure 2.1: OFDM transceiver structure

2.2 Merits and weakness of OFDM

OFDM is widely adopted because of a number of advantages it offers, [2, 15]:

1. Orthogonality of sub-carrier signals, trivial generation of transmit signals and trivial equalization.

2. Closely spaced orthogonal sub-carriers partition the available bandwidth into a maximum collection of narrow sub-bands.

3. Adaptive modulation schemes can be applied to sub-carrier bands to maximize bandwidth efficiency.

4. Arbitrary sets of sub-carriers can be turned on and off to match specific spectrum holes.

5. Simplified tasks of carrier and symbol synchronization.

Though OFDM possesses the above mentioned merits, it does come with some unwanted weakness. These weakness are listed briefly as follows:

1. High Peak-to-mean-power ratio (PAPR)

2. Synchronization is sensitive to frequency offset, e.g. oscillators difference between receiver and transmitter.

3. Use of Cyclic Prefix (CP) results in a loss of bandwidth efficiency.

4. Large side lobe results in power leakage

To the interest of the topic of this thesis, large side lobe draws more attention of the author. The use of FFT/IFFT operator indicates a rectangular pulse shape for each sub-carrier. Rectangular pulse shape in time domain leads to a sinc waveform in the frequency domain. Sinc waveforms further ushers us to the undesirable large side lobes; the peak of the first side lobe is only 13dB below the peak of its main lobe. This large side lobe between sub-carriers results in power leakage among different sub-carriers, especially between adjacent sub-carriers. This phenomenon is called mutual interference [2]. The problem is even worse

in CR settings, where PU and SU are transmitting independently and may be using different standards. In such settings, the only way to fight against this power leakage which comes from the large side lobe is to use a filter mechanism, which will be discussed in the next chapter. In following subsection, general techniques that help to alleviate OFDM power leakage problem are investigated.

2.3 PAPR in SC-FDMA and FBMC

High PAPR is usually considered as a major OFDM drawback. To fight against this drawback, several PAPR reduction schemes for OFDM were proposed, including clipping, Selected Mapping (SLM), Partial Transit Sequences (PTS), Tone Reservation (TR), Tone Injection (TI) and Active Constellation Extension (ACE) [18]. The quantitive analysis of these PAPR schemes for OFDM signals can be found in [19].

Other than the PAPR reduction schemes mentioned above, , a special form of OFDM, which is called SC-FDMA, can also greatly relieve the PAPR problem. It used a precoding to each user data set in each OFDM symbol to control its PAPR [20]. It has an additional DFT processing block preceding the IFFT engine in the conventional OFDMA, as shown in Fig.2.2. The quantitive analysis regarding how much SC-FDMA can improve in terms of PAPR than OFDMA can be found in [21]. Besides, in [21], it is also pointed out that pulse shaping increases PAPR.

FBMC is a OFDM evolvement that fight against another drawback in the conventional OFDM, which is large side lobe results in power leakage. In regards to the comparison of PAPR between FBMC and OFDM, although FBMC also suffers from a PAPR problem, it is found that FBMC is better in PAPR [22].

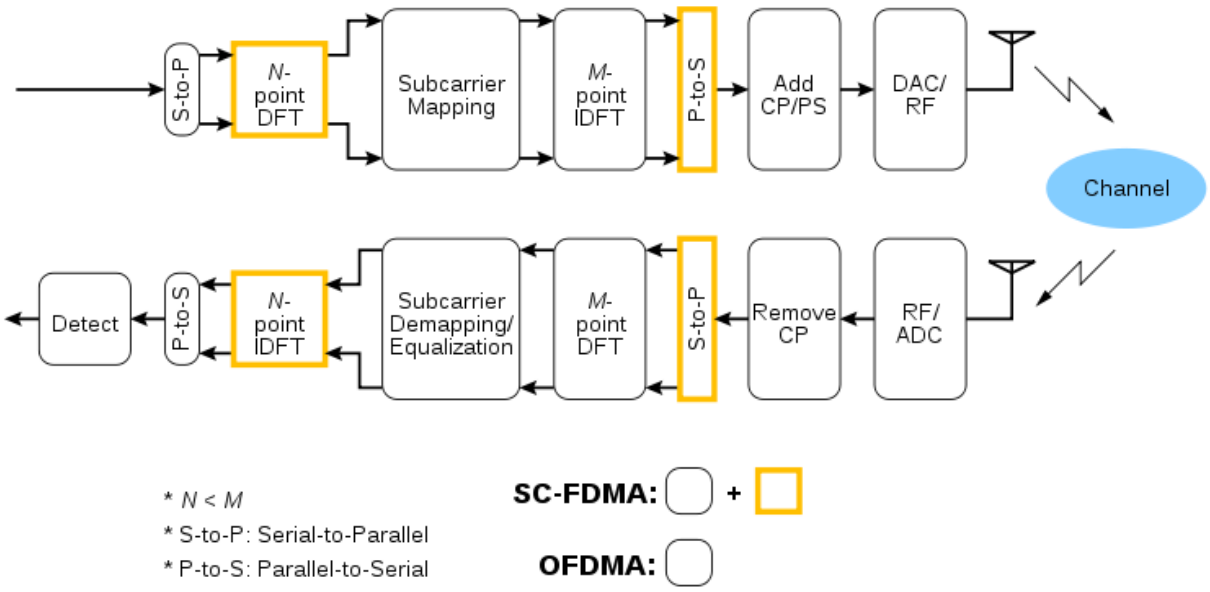

Figure 2.2: SC-FDMA transceiver structure from WikiPedia

2.4 Techniques to mitigate mutual interference

Several techniques that helps to mitigate mutual interference are listed as below:

1. Use of synthesis/analysis filters that have small side lobes.
2. Guard sub-carriers at the boundaries between PU and SU [23].
3. Canceling sub-carriers that combine destructively [24].
4. Mapping of the transmitted symbol sequence to a set of sequences and choosing a sequence out of this set with the lowest side lobe [25, 26].

All the techniques mentioned above reduce the OFDM transmission efficiency, due to the addition of extra data, e.g. canceling sub-carriers, or assigning zero to the sub-carriers in the boundaries.

2.5 Summary

This chapter presented a brief review of OFDM transceiver. The use of CP reduced OFDM bandwidth efficiency. However, many desirable and important features that OFDM offers made it the most important and popular multicarrier communication technology. The rest part of this chapter drew attention to the large side lobe problem in OFDM, which could prevent it from being the best solution in many future communication systems. In the end, several mutual interference mitigation techniques were summarized. One of these techniques was to use filter mechanism, which is the most effective way. In the coming chapter, this evolution is discussed.

Chapter 3

OFDM Evolution

As mentioned in the previous chapter, OFDM has a large side lobe in for each sub-carrier in frequency, which leads to power leakage among sub-carriers. The use of synthesis/analysis filters that have small side lobes is the most effective method to restrict OFDM power leakage problem. This modification can be considered as an evolution of OFDM, which leads to filtered OFDM, and filter bank multicarrier (FBMC). In order to unify OFDM, filtered OFDM and FBMC, a unified formulation for OFDM and FBMC is illustrated in the coming section. In fact, OFDM can be generalized as a special FBMC method with a rectangular filter shape of duration equal to one symbol [3].

3.1 Unified OFDM/FBMC Architecture

In [2], a unified OFDM/FBMC architecture is presented in continuous time domain as it serves best for discussion, and is shown in Fig. 3.1. The input is defined as $s_k(t) = \sum s_k[n]\delta(t - nT)$. $s_k[n]$ is the sub-carrier data symbol, k refers to the index of sub-carrier and T is the symbol time spacing. The difference between OFDM and FBMC lies in the

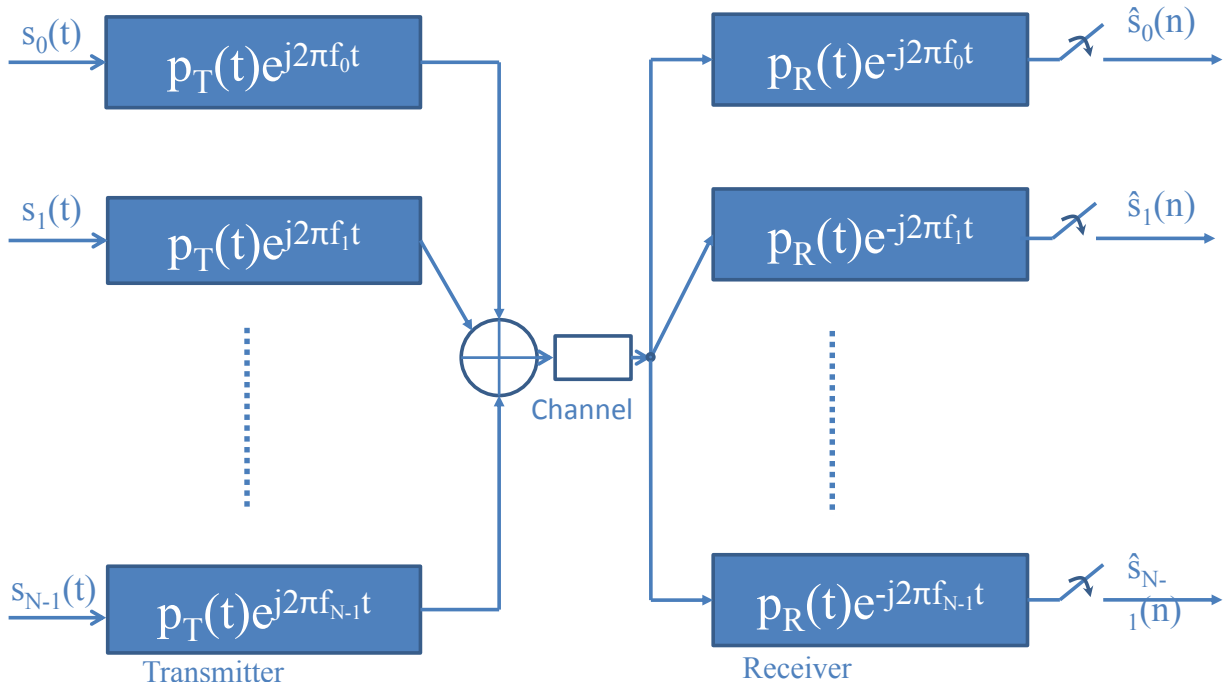

Figure 3.1: Unified OFDM/FBMC Architecture [2].

selection of the synthesis/analysis prototype filters, $p_T(t)$ and $p_R(t)$. In this sense, OFDM can be viewed as a special FBMC method with $p_T(t)$ and $p_R(t)$ as rectangular pulses. Again from this unified architecture, we can see that rectangular pulses in time domain leads to sinc function in frequency domain, which is well known of large side lobes. The peak of the first side lobe is only 13dB below its main peak, and other side lobes are also relative large. This limitation will finally lead to power leakage to its adjacent frequency bands.

In OFDM, even though $p_T(t)$ and $p_R(t)$ are both rectangular, their width are not the same. For $p_T(t)$, its width is the symbol time spacing T, whereas for $p_R(t)$, its width is T_{FFT}. T_{FFT} is the FFT/IFFT duration in time. This further indicates that the sub-carrier frequency spacing is $1/T_{FFT}$. In conventional OFDM, the difference between T_{FFT} and T is the width of Cyclic Prefix (CP). In FBMC, no CP is required to allow the orthogonality among sub-carriers. Hence T_{FFT} and T are of the same width, which means in FBMC CP is not used. Since the use of CP leads to a loss of bandwidth efficiency, FBMC is considered to be able to

reach higher bandwidth efficiency than OFDM in general. However, this might not always hold true, and we will discuss more on the bandwidth efficiency comparison between OFDM and FBMC later.

3.2 Filtered OFDM

Filtered OFDM replaces the conventional rectangular pulse with a pulse that has soft transition at the beginning and end. In Fig. 3.2, a detailed pair of prototype filters in filtered OFDM is presented [2].

The improvement comes from a pulse that has a soft beginning and end, which maps to a lower sidelobe in frequency domain. Adjacent filtered OFDM symbols overlap in a period of T_0. $T - T_0$ should be greater than $T_{FFT} + T_1$, or equivalently, $T > T_{FFT} + T_0 + T_1$. The introduction of T_0 and T_1 will further reduce the bandwidth efficiency of OFDM, which means T_0 and T_1 should be small to achieve better bandwidth efficiency. However, on the other hand, in [2], it is pointed out, in order to achieve low side lobes, T_0 has to be comparable with T_{FFT}. Thus, in general, filtered OFDM suffers greatly from spectral efficiency loss [27].

3.3 FBMC

There are three FBMC techniques: Filtered Multitone (FMT) [28], Staggered Multitone (SMT) [29] and Cosine-modulated Multitone (CMT) [30]. SMT is also know as OFDM-OQAM, and OQAM stands for offset QAM. In this section, the basics of FMT, SMT and CMT are reviewed.

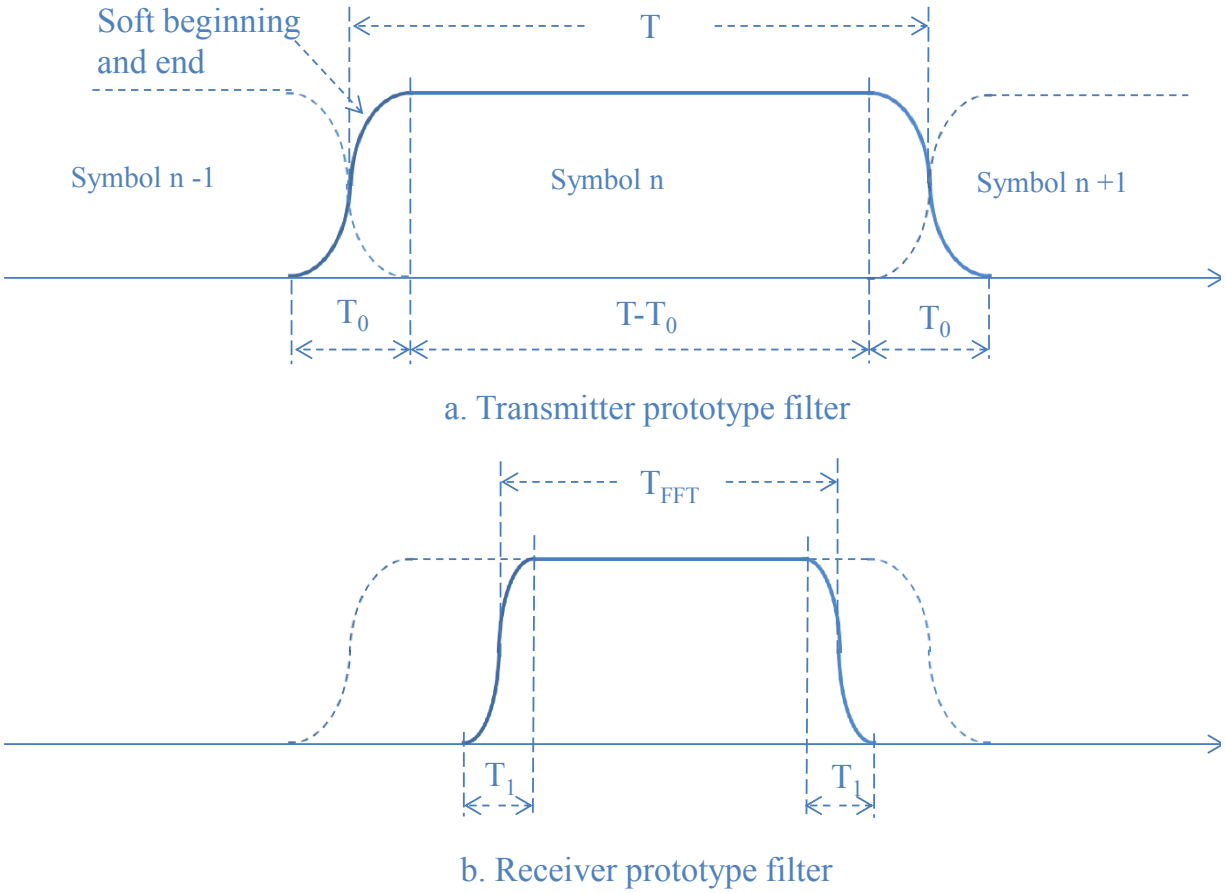

Figure 3.2: Filtered OFDM [2].

3.3.1 Filtered Multitone

Sub-carriers in FMT does not overlap and guard bands are used to separate sub-carriers, thus resulting in a loss of bandwidth efficiency compared with SMT and CMT. However, in [31], it is pointed out FMT favors us in terms of hardware implementation simplicity and requires the least DSP resources.

In Fig. 3.3, an FMT spectra is illustrated. Sub-carriers do not overlap as expected. Sub-carrier frequency spacing F equals to $(1+\alpha)/T$, where α is the roll-off factor in designing

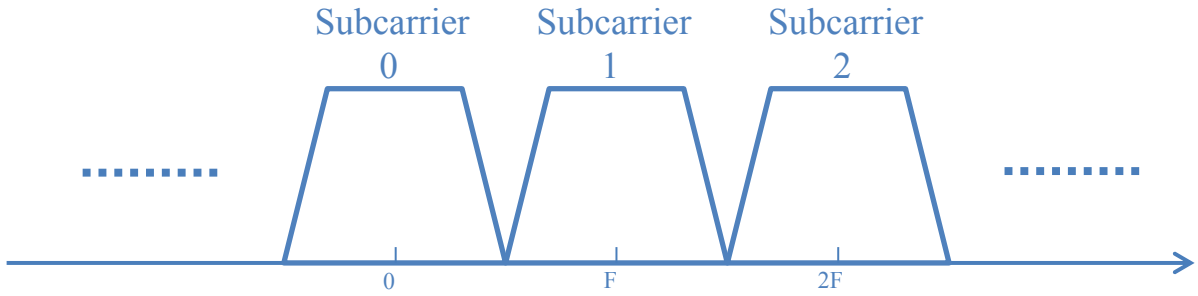

Figure 3.3: FMT spectra

the FMT square-root prototype filters, and T is the FFT time as well as symbol time spacing. Before discussing about the bandwidth efficiency in quantity, the author would like to introduce the term of symbol density. It refers to the amount of data that has been transmitted in every time T along the frequency axis at spacing $F = 1/T_{FFT}$. Therefore, in case of FMT, every time T, data is transmitted with subcarrier frequency spacing of $(1+\alpha)/T$. As a result, the symbol density for FMT is $1/(1+\alpha)$, which is less than 1 since α usually is between 0 to 1.

3.3.2 Staggered Multitone

Staggered Multitone is also known as OFDM-OQAM. However, Staggered Multitone serves more concisely and acurately as we will see the SMT spectra staggers back and forth as time progresses.

In Fig. 3.4, an SMT transmitter from [2], is presented. The time offset $T/2$ is introduced through the prototype filter $p(t - T/2)$.

In Fig. 3.5, an example of SMT spectra is shown. Blue spectra stands for the subcarriers that have a phase shift of an even factor of $\pi/2$, while orange spectra corresponds to those subcarriers with a phase shift of an odd factor of $\pi/2$. As we can see, for every $T/2$, a phase shift of $\pi/2$ is applied to adjacent subcarriers, as if staggering in time. Moreover, for every

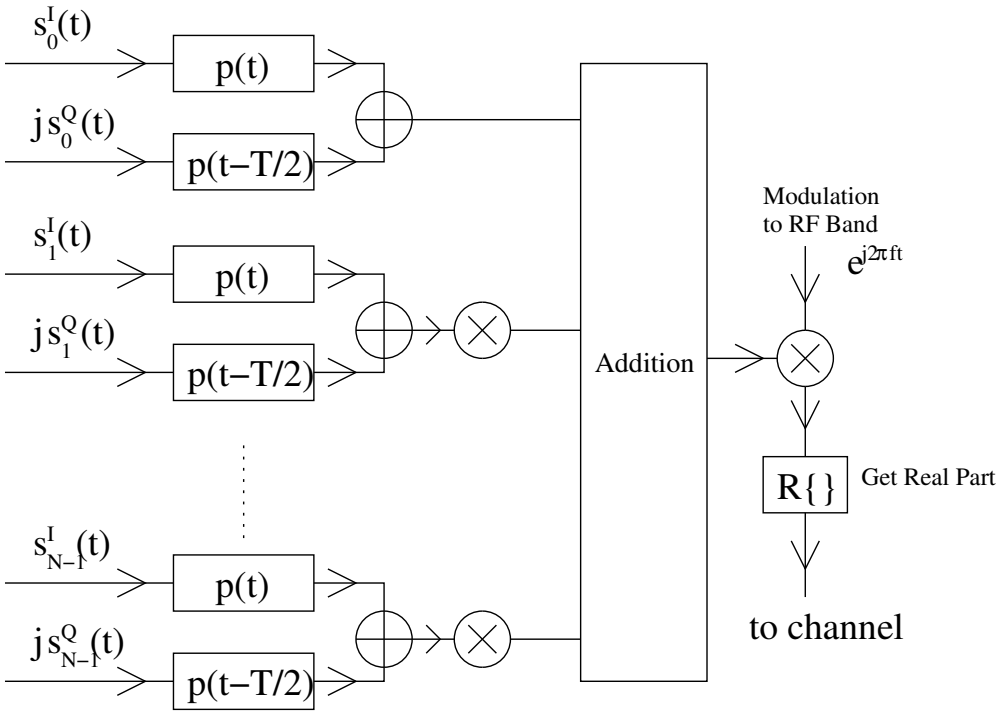

Figure 3.4: SMT transmitter

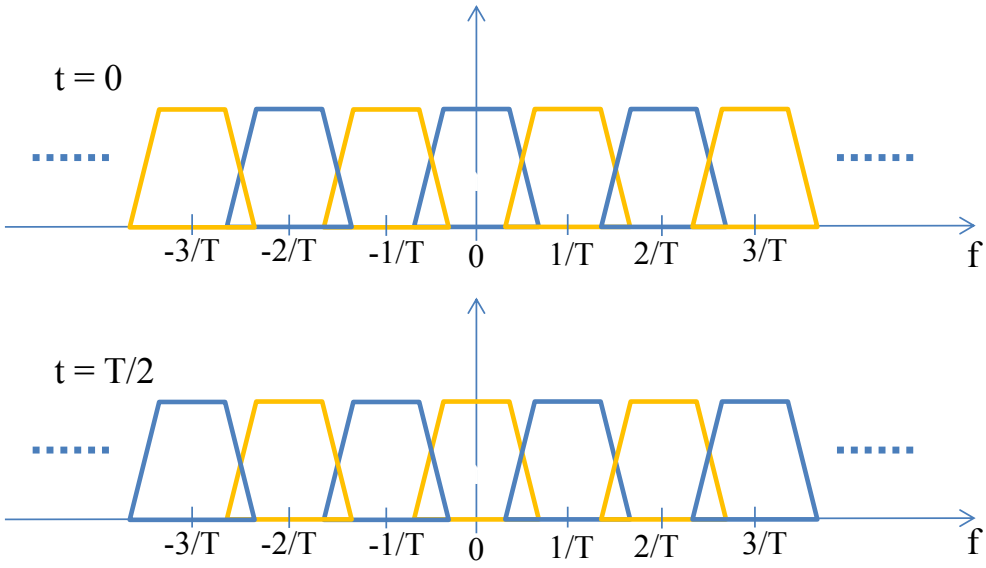

Figure 3.5: SMT Spectra

time interval $T/2$, data is transmitted with subcarrier spacing of $1/T$. Hence, bandwidth efficiency can be calculated as $1/((T/2) \times (1/T))$, which equals to 2. In fact, as one may observe, as opposed to FMT, the subcarriers in SMT overlap with each other, and SMT does reach the maximum bandwidth efficiency.

3.3.3 Cosine-modulated Multitone

A CMT transmitter is described in Fig. 3.6.

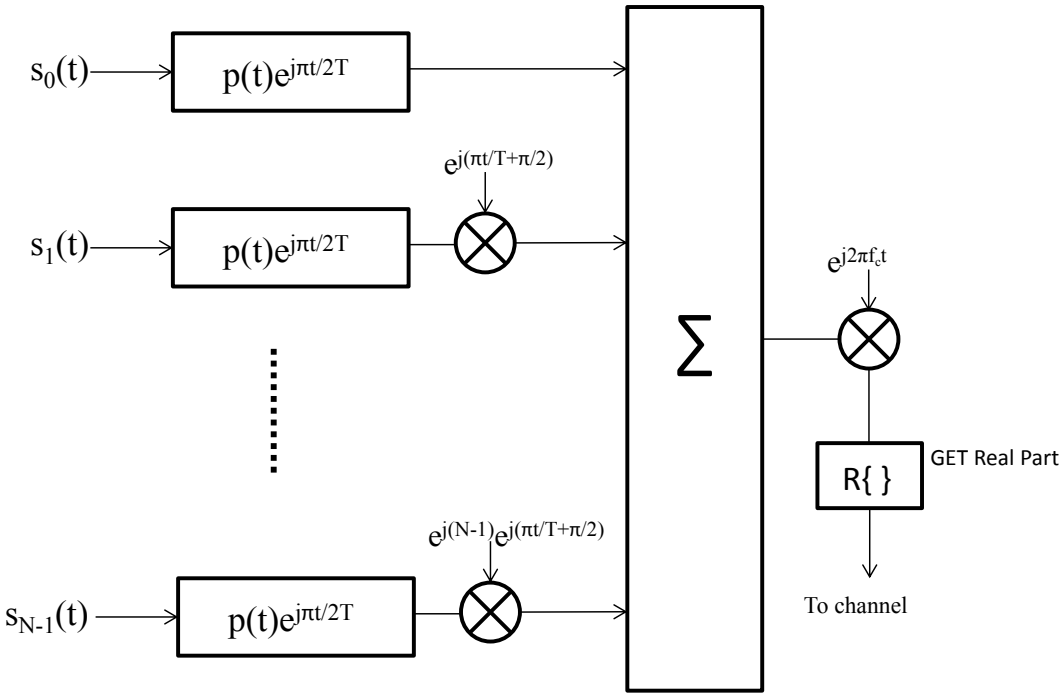

Figure 3.6: CMT transmitter

CMT is essentially based on Vestigial Side-Band (VSB). Compared to FMT, CMT also reaches the maximum bandwidth efficiency. CMT sends data in every time T, with a sub-carriers frequency spacing of $1/2T$, hence CMT bandwidth efficiency equals to $1/((T) \times (1/2T)) = 2$. In fact, sub-carriers spectra in CMT also overlap with each other, and sub-

carrier frequency spacing is half of SMT sub-carrier frequency spacing. However, since it sends data with a cycle of T, which is double of SMT, hence CMT and SMT, in the end, has the same bandwidth efficiency. Both CMT and SMT reaches maximum bandwidth efficiency, rather than FMT sacrifices bandwidth efficiency for hardware implementation simplicity.

3.4 Summary

In this chapter, evolution from the conventional OFDM was described. All these variants were proposed for the sake of alleviating conventional OFDM large side lobes. Filtered OFDM used a pulse shaping filter that has a soft beginning and end to mitigate large side lobes. This method in general suffered greatly from a loss of bandwidth efficiency. Filter Bank Multicarrier technique was discussed after that. It included three modulation methods, Filtered Multitone, Staggered Multitone and Cosine-modulated Multitone. An basic review of the design ideas was presented for these three FBMC techniques. As pointed out previously, FMT used a guard band to separate sub-carriers; thus resulting a loss of bandwidth efficiency, while sub-carriers in CMT and SMT overlap, and both of them reached the maximum bandwidth efficiency. Next, FMT will be discussed more in details.

Chapter 4

FMT with a Polyphase Filter Bank

Modern FPGAs provide large amount of programmable logic as well as DSP resources, memories in a single die, which makes FMT System-on-Chip (SOC) design possible. An Altera NIOS II processor augmented with custom instructions enables users to accelerate a time-critical software algorithm by implementing it on hardware. It is an effective interface to attach DSP IPs to NIOS II. Plus, few FMT overlay implementation on Altera Devices are reported. For example [15, 16] are about OFDM CR on Xilinx FPGAs, [17] is FMT underlay on Xilinx FPGAs, [3] is FMT overlay on Xilinx FPGAs. Hence, we chose to implement FMT overlay transceiver with custom instructions on Altera FPGAs. Before implementing FMT on hardware, a more efficient model that uses polyphase filter bank from [3] will be reviewed.

4.1 Brief Review of FMT Transmitters

A generic FMT transmitter is presented in Fig. 4.1. N is the maximum number of sub-carriers. The sub-carrier separation is $f_s I/N$, where I is an integer interpolation rate and f_s is the symbol rate. The relation between I and N is that $I \geq (1+\alpha)N$. α is the roll-

off factor in the square-root Nyquist pulse filter. Given the previous mentioned sub-carrier separation $f_s I/N$, this infers that the minimum FMT sub-carrier separation is $f_s(1+\alpha)$, and this separation increases if a greater I is used.

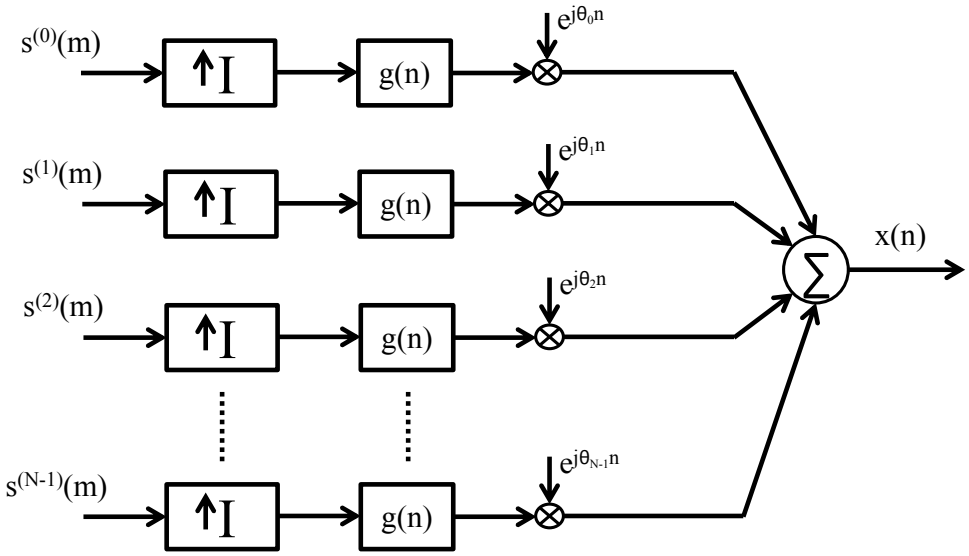

Figure 4.1: Generic FMT transmitter

Efficient realization of Fig. 4.1 which uses polyphase filter bank is shown in Fig. 4.2.

4.2 Brief Review of FMT Receivers

Fig. 4.3 illustrates the FMT receiver that corresponds to Fig. 4.1, where $h_k(n)$ is a passband filter based on analysis filter $h(n)$ by the relation of $h_k(n) = h(n)e^{j\theta_k n}$, and $h(n)$ is matched to synthesis filter $g(n)$. $r(n)$ is obtained after the signal $x(n)$ is affected by the channel response and thermal noise.

Also, an efficient realization of an FMT receiver which uses polyphase filter bank is presented in Fig. 4.4.

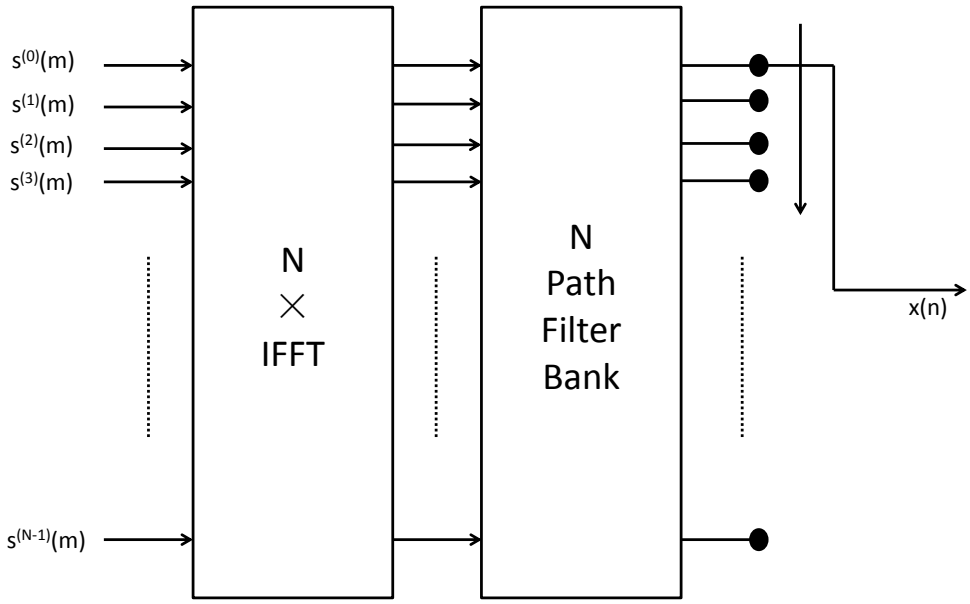

Figure 4.2: Efficient FMT transmitter with the use of filterbank

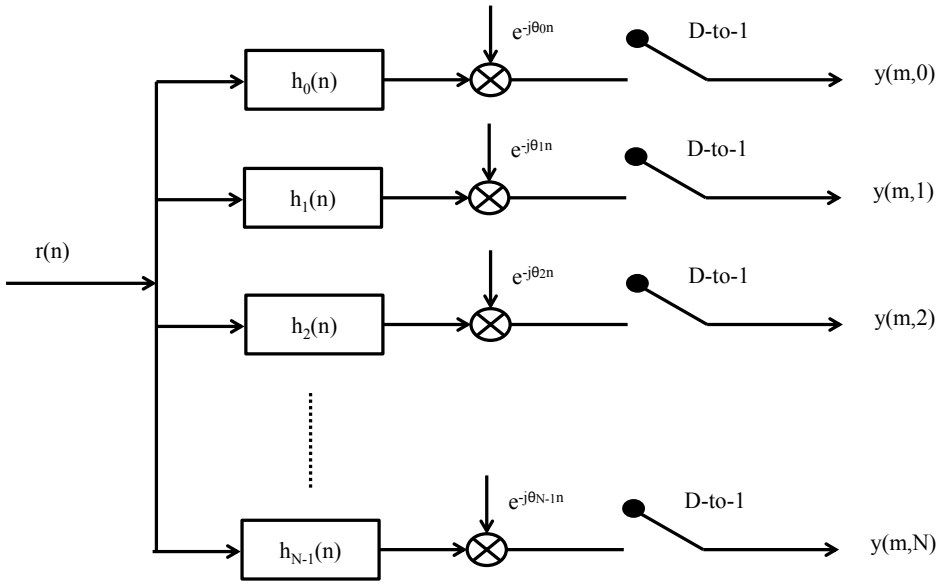

Figure 4.3: FMT receiver

4.3 CAZAC sequence

A Constant Amplitude Zero Auto Correlation waveform (CAZAC) is a periodic complex-valued signal with modulus one and out-of-phase periodic (cyclic) autocorrelation equal to

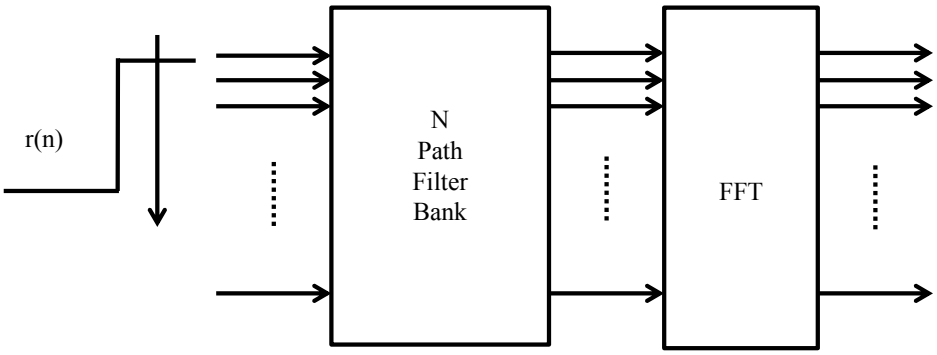

Figure 4.4: Efficient FMT receiver with filter bank

zero [32]. It is also known as Zadoff-Chu sequences [33]. CAZAC sequences are often used to in synchronization algorithms in communication systems. CAZAC sequences are defined as below:

$$C_r(k) = e^{-j\pi rk^2/N}$$

, where N is an even number. The following properties are important CAZAC features [3]:

1. The cyclic autocorrelation equals to zero for lags different than zero [34].

$$\sum_{k=0}^{N-1} C_r(k)C_r^*(k+m) = \begin{cases} N & \text{for } m = 0 \mod N \\ 0 & \text{for } m \neq 0 \mod N \end{cases} \tag{4.1}$$

2. The DFT of a CAZAC sequence is also a CAZAC sequence [35, 36].

3. CAZAC sequences have a low peak-to-average power (PAPR).

4. Modulation property: when $r = N - 1$, a cyclic time shift is equivalent to modulating the sequence.

5. Fractional time offsets produce spectral leakage effects.

4.4 Summary

In this chapter, the original mathematical model of FMT was presented. The use of polyphase filter banks improved the original model into a more efficient FMT model. Efficient transmitters and receivers were described in Fig. 4.2. and Fig. 4.4. CAZAC sequences were often used in the synchronization algorithms, due to the possession of many desirable properties. The cyclic autocorrelation property made it a candidate for frame detection.

Chapter 5

FMT Hardware Implementation

5.1 Hardware Architecture of FMT Transmitter

In this subsection, datapath of the FMT transmitter is presented and the data flow is illustrated in Fig. 5.1.

The cad bus interface is used to communicate to the outside world. It converts the bus writing events on different address to FIFO writing events or allocation vector writing events. It is a 16-bit width interface. The cad bus address mapping can be found in Table.5.1.

Table 5.1: CAD bus address mapping

Address	Write mapping
0x00	Start to write allocation vector
0x01	write allocation vector
0x02	End allocation vector
0x03	Data write
0x04	End Data Write

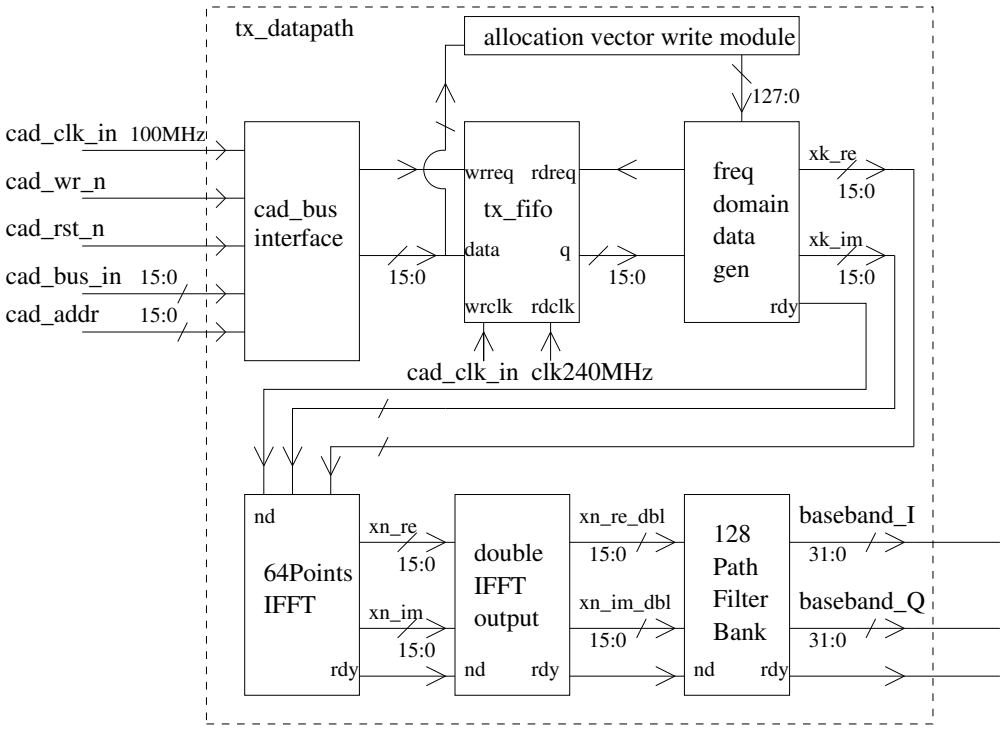

Figure 5.1: Transmitter Hardware Datapath

Allocation vector controls the sub-carrier type. The sub-carrier type includes null, positive pilot, negative pilot and data; hence, two bits are needed for each sub-carrier. Since there are 64 sub-carriers in this design, a 128 bits allocation vector is in the datapath. Tx_FIFO is a dual port FIFO, since cad bus works at a relative lower frequency and the rest parts of the datapath works at 240 MHz. Between the processing blocks, a pair of simple handshake signals, `nd` and `rdy`, are used to synchronize each parts. `Nd` means "new data", `rdy` stands for "ready".

Each FMT symbol corresponds to 64 input points to the IFFT engine. This IFFT input is data in the frequency domain. Double IFFT output block simply copies every 64 points input set to a 128 points output. For example, the real part of `xn` is doubled as in Fig. 5.2. This doubling IFFT output process results in an up-sampling rate of 2. Up-sampling rate of

2 offers the benefits of increasing of precision of estimating the sampling time offset precision on the receiver side.

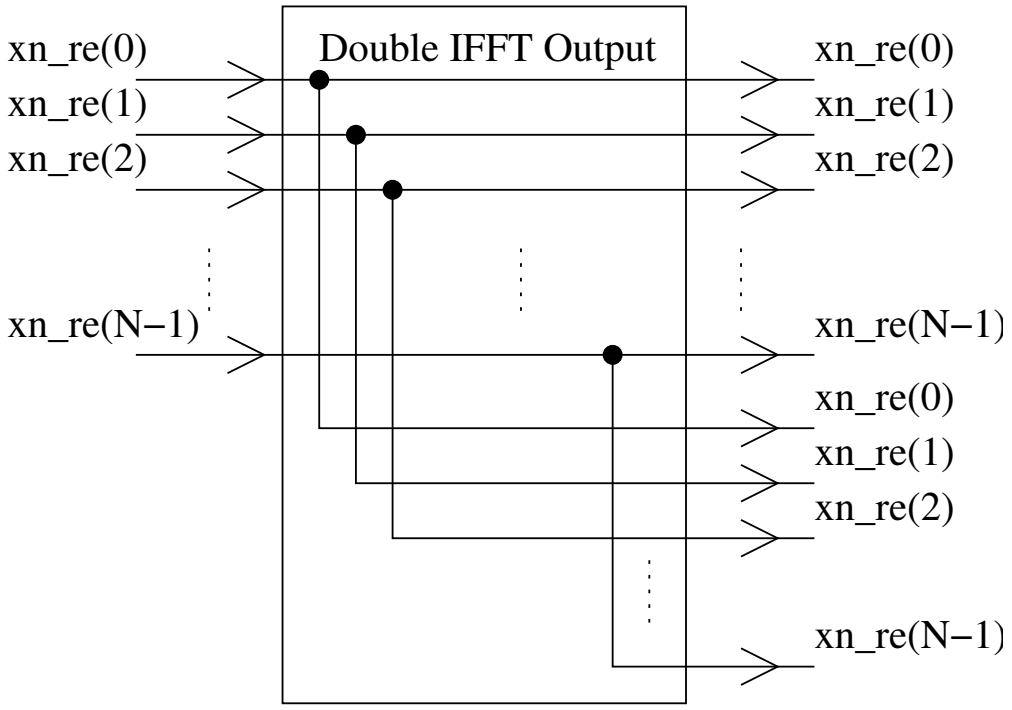

Figure 5.2: Double IFFT real part output

Therefore, one FMT symbol has 128 points after the process of "`double IFFT output`". Given this, the pulse shaping filter that spans over two symbols should have 256 taps. This filter is subject to a polyphase decomposition in 128 paths; hence, a 2-tap FIR filter should be applied for each path. An efficient way of implementing such a polyphase filter bank is to reuse a 2-tap FIR processing block and substitute in the correct coefficients for every path. Fig. 5.3 shows the idea of the transmitter polyphase filter bank hardware design. For every input, a controller, which is not shown in the diagram, selects the correct corresponding coefficients, for example h0 and h128, and the previous stored input to calculate the output and then stores the current input in preparation of the next round computation. The frequency response of the transmitter filter bank is shown in Fig. 5.4. In the time domain, the

transmitter filter bank is designed to offer a root of raised cosine prototype filter response with a roll off factor $\alpha = 0.99$.

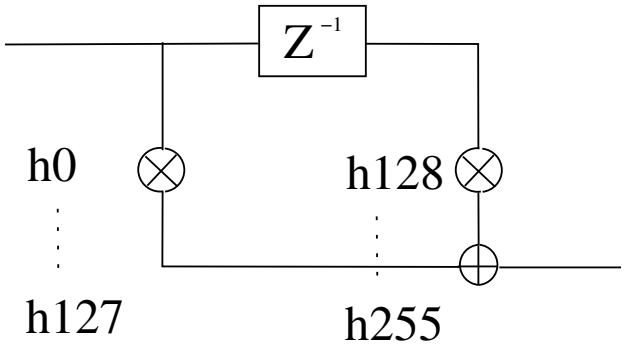

Figure 5.3: Transmitter filter bank reuses a 2-tap FIR

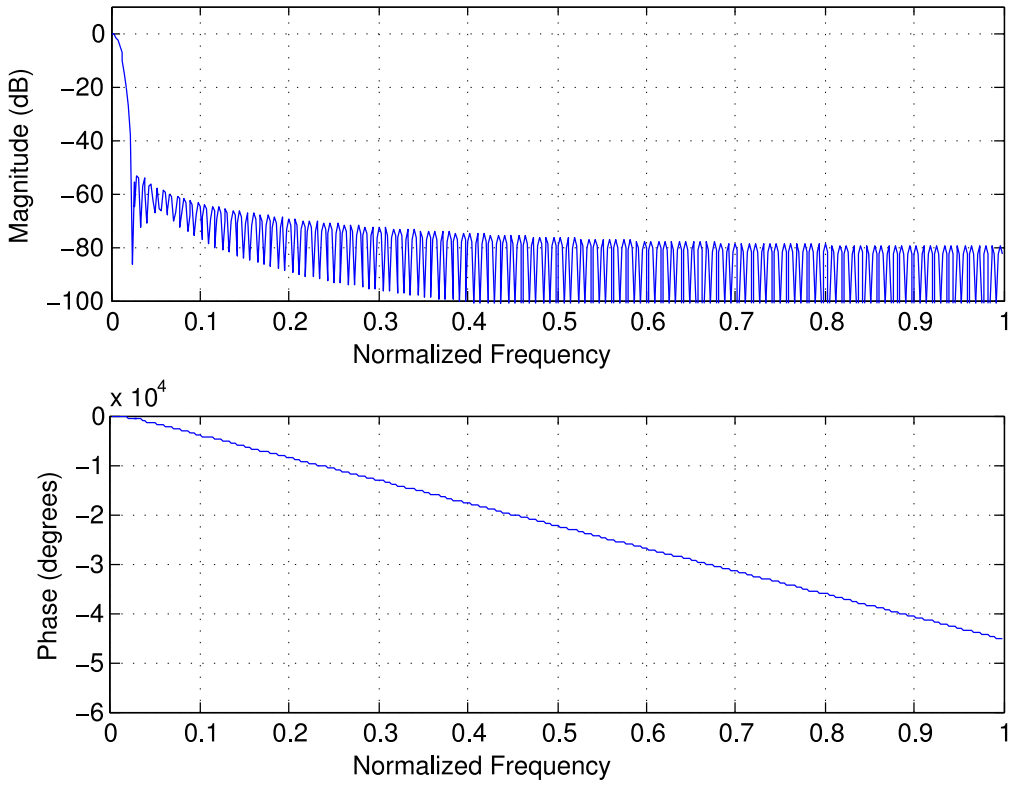

Figure 5.4: Frequency response of transmitter filter bank

`Baseband_I`, and `baseband_Q` represent the FMT baseband complex quadrature signal. In this project, `baseband_I`, and `baseband_Q` has a rate of 20 MHz. This would lead to

a symbol rate f_s of 20MHz/128, which equals to 156.25 kHz. On top of that, the subcarrier separation of $f_s(1+\alpha)$ can be calculated, which is 312.5 kHz. Assuming the currently adopted maximum number of active sub-carriers is limited to 52 (4 pilots and 48 data, inspired from IEEE 802.11, however these number can be changed at anytime under the environment of CR), and taking into the consideration of edge sub-carriers, the bandwidth of baseband would be $((52+2) \times 312.5 \text{ kHz})) = 16.875$ MHz. Besides as mentioned previously, adjacent FMT symbols overlap. In this case, each symbol occupies two times of $1/f_s$.

5.2 Hardware Architecture of FMT Receiver

Since we adopted the same architecture as in [3], the final block diagram of the FMT receiver core datapath is presented in Fig. 5.5. Detailed mathematical derivation of this FMT receiver can be found in [3]. Basic synchronization idea is to perform a cyclic cross-correlation between the received preambles, which are based on CAZAC sequence, and local copy of CAZAC sequence. A peak point of the cyclic cross-correlation calculation is searched in order to locate the timing offset. The channel equalizer uses one tap for each sub-carrier and pilot signals are used to track phase changes.

The algorithm used for frame detection and timing offset correction is shown in Algorithm.1, [3]. The mathematical symbols used in Algorithm.1 inherit the same symbols as in Fig. 5.6. Next, a brief explanation of Algorithm.1 is presented. Having skimmed all the mathematical derivations for simplicity of understanding the top picture, the thing that the author would like to point out is, in order to perform a cyclic cross-correlation, all the calculations in Fig. 5.6 must be performed in order. The cyclic cross-correlation result is the output of IFFT in Fig. 5.6. $R^{(p)}(l)$ stands for the p^{th} lag of the cross cyclic-correlation. The peak magnitude value of $R^{(p)}(l)$ is searched. The p^{th} lag which generates the peak cyclic cross-correlation

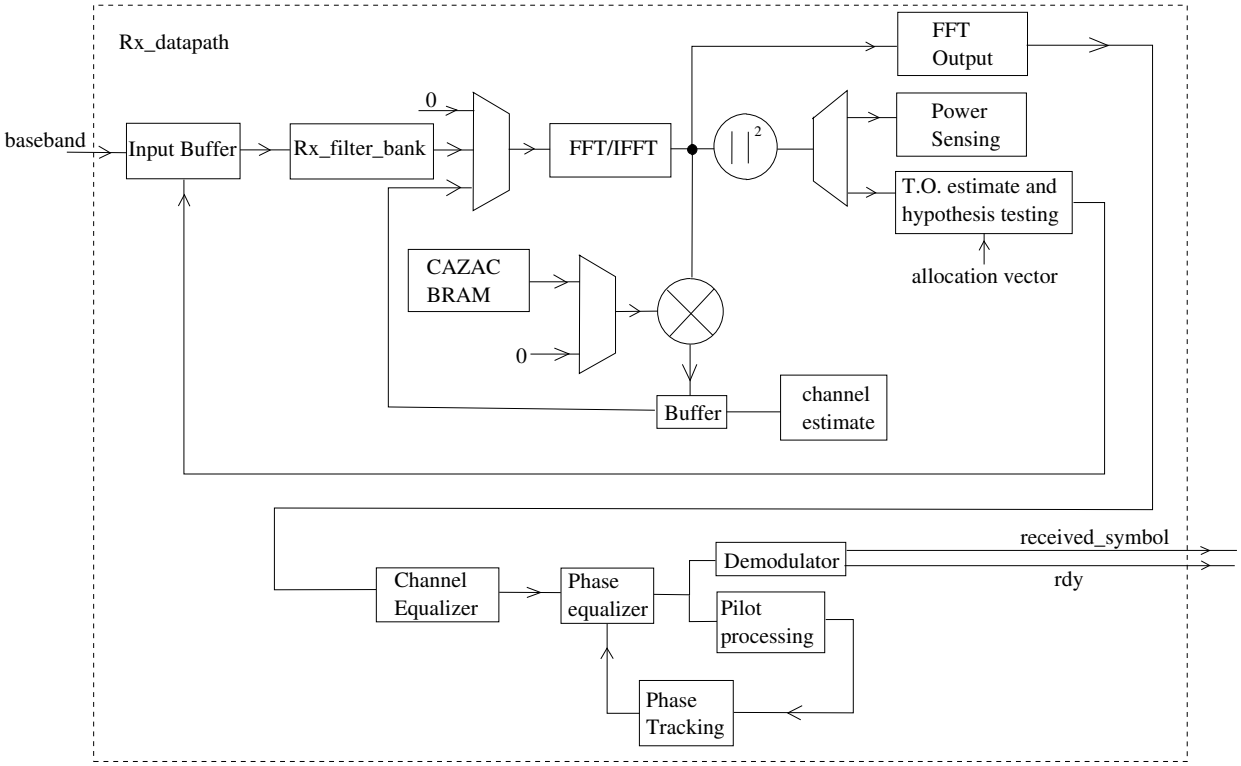

Figure 5.5: FMT receiver core datapath

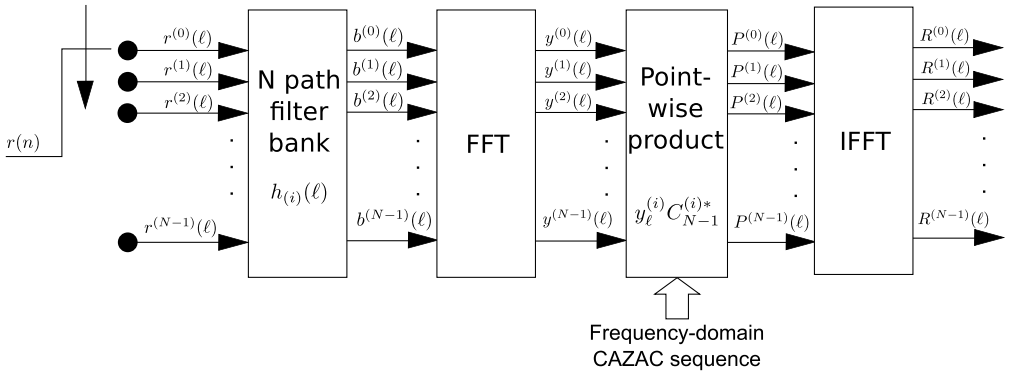

Figure 5.6: Data flow of cyclic cross-correlation in hardware blocks [3].

magnitude is recorded in $\hat{\theta}_\ell$. A hypothesis test is performed as below to make sure the peak detected is a legal result from CAZAC sequences cross-correlation.

$$H_\ell = |R(\ell, \hat{\theta}_\ell)|^2 > \gamma \frac{\rho}{N} \sum_{p=0}^{N-1} |R(\ell, p)|^2 + \epsilon \tag{5.1}$$

$\ell \leftarrow 0$
$peak \leftarrow 0$
$Preamble1Det \leftarrow 0$
while $Preamble1Det \neq 1$ **do**
  $\bar{y}(\ell) \leftarrow \mathsf{FFT}\left\{\bar{b}(\ell)\right\}$
  $\overline{W}(\ell) \leftarrow \overline{V}\bar{C}^{*}_{N-1}\bar{y}(\ell)$ {$\overline{V}$ performs interference rejection}
  $\overline{R}(\ell) \leftarrow \mathsf{IFFT}\left\{\overline{W}(\ell)\right\}$ {Calculation of the cyclic cross-correlation}
  $\hat{\theta}_\ell \leftarrow \arg\max_p \{|R(\ell,p)|\}$
  $H(\ell) \leftarrow |R(\ell,\hat{\theta}_\ell)|^2 > \gamma\frac{\rho}{N}\sum_{p=0}^{N-1}|R(\ell,p)|^2 + \epsilon$ {Hypothesis test}
  if $peak < |R(\ell,\hat{\theta}_\ell)|^2$ **then**
    $peak \leftarrow |R(\ell,\hat{\theta}_\ell)|^2$
  end if
  if $\left(|R(\ell,\hat{\theta}_\ell)|^2 < peak\right)$ and $H(\ell-1)=1$ **then**
    $Preamble1Det \leftarrow 1$ {Peak detected while hypothesis tested positive}
    Align input buffer by $\hat{\theta}_{\ell-1}$
  end if
  $\ell \leftarrow \ell + 1$
end while

Algorithm 1: Algorithm for frame detection and timing offset correction

In conclusion, for every frame the receiver obtains, it calculates the cyclic cross-correlation and find the maximum magnitude cross-correlation with the lag recorded $\hat{\theta}_\ell$. Afterwards, a hypothesis test is performed. Among all the received frames, the peak of all the maximum cross-correlation magnitudes is searched, and input buffer is aligned based on the this peak, which is $\hat{\theta}_{\ell-1}$ in Algorithm.1. In the end, all the related calculations can be done by the receiver core datapath in Fig. 5.5. FFT and IFFT operations use a same engine as long as the latency allows.

5.3 Digital Intermediate Frequency Processing

Until now, the previously discussed FMT transmitter and receiver are still operating at baseband. In this section, the digital intermediate frequency related design are discussed,

including Digital Up Converter (DUC), Digital Down Converter (DDC). A DUC provides the link between the digital baseband and analog RF front end and is required on the transmitter of a generic transceiver [37]. The sampling frequency of the baseband data stream is usually increased before it is modulated onto a high frequency carrier. A DDC, on the other hand, provides the link between analog RF front end to digital baseband of receiver. The maximum DAC sampling rate is 250 MHz, and the maximum ADC sampling rate is 150 MHz for the daughter board of "AD/DA Data Conversion Card" from Terasic, which is the AD/DA devices we used in this project. We select the DAC sampling rate as 240 MHz and Digital Intermediate Frequency as 60 MHz in this project. This selection will benefit us in terms of generating cos(60 MHz) and sin(60 MHz), which we will see in the following discussion.

The digital up converter used in this project is shown in Fig. 5.7. Since the gap of sampling rate between DAC (240 MHz) and baseband (20 MHz) is 12 times, 12-times-up interpolater is required. Breaking this large interpolater into two stages helps in reducing the total number of filter taps. In this project, two interpolaters, implemented with the Altera FIR IP, with factor of 3 and 4 respectively are used. After the baseband reached the sampling rate of 240 MHz, complex quadrature modulation with carrier frequency at 60 MHz can be performed. The modulation adopted the following equation $I * \cos(60\ MHz) - Q * \sin(60\ MHz)$. Since I, Q has a rate of 240 MHz, which is 4 times of carrier frequency 60 MHz, only four constants (0,1,0,-1) are needed for multiplication with sin(60 MHz), and similarly, (1,0,-1,0) are for cos(60 MHz) case. As a result, no real multiplier is needed to perform the complex quadrature modulation. Furthermore, as the DAC only has a precision of 14 bits, a truncation operation block is added before sending data to DAC.

Fig. 5.8 shows the theoretical frequency response of the two previously mentioned interpolaters. The spectrum in the broken lines is expected to be rejected. The idea frequency

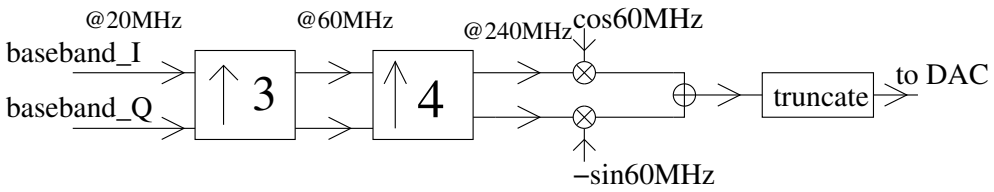

Figure 5.7: Digital Up Converter

response of each interpolater filter is shown in the bold black line. The actually frequency response for these two interpolaters are shown in Fig. 5.9.

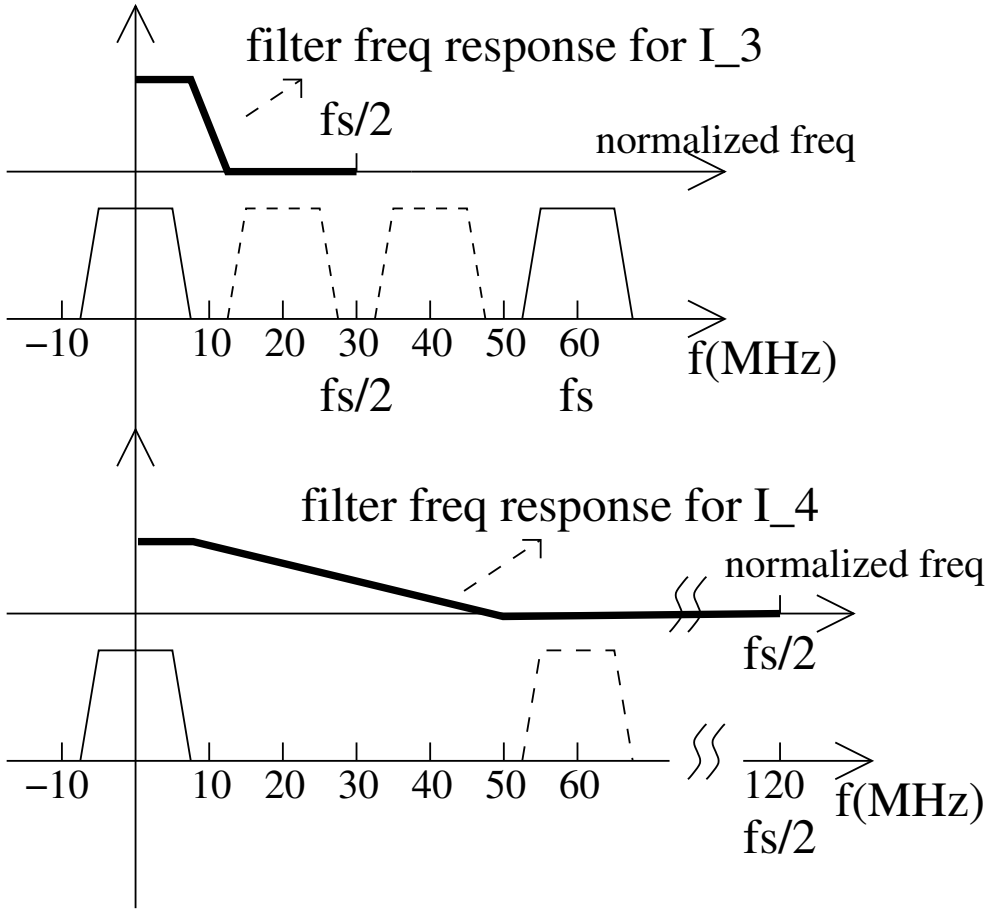

Figure 5.8: Theoretical Interpolater Frequency Response

The sampling frequency of ADC is chosen to produce an aliased copy of the received carrier at one quarter of the sampling frequency [38]. To calculate the aliased frequencies, the

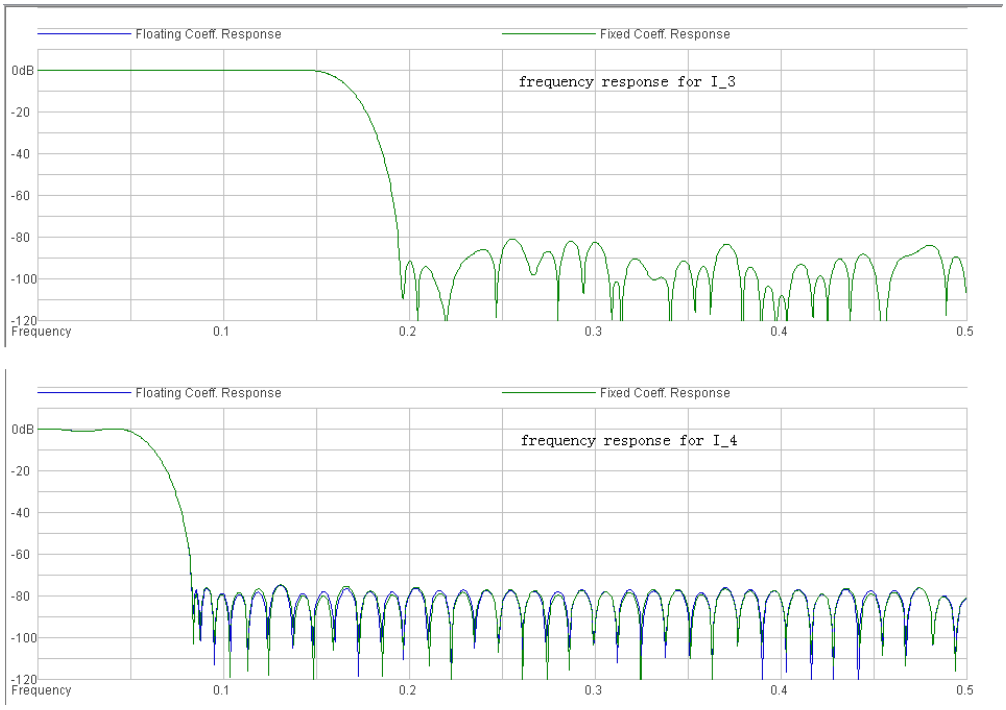

Figure 5.9: Actual Interpolater Frequency Response

following equation, $f_c = (k \pm 1/4)F_s$, must be fulfilled for integer values of k, under the constraint that the signal bandwidth B plus the transition band B_T must be less than half of the sampling frequency F_s of ADC. In our case, f_c is 60 MHz, and given the maximum ADC sampling rate is 150 MHz, we choose the ADC sampling rate F_s as 80 MHz. This will create an alias at 20 MHz. Given this result, a conceptual down converter is shown in Fig. 5.10. It simply takes the signal from ADC and multiply with a carrier of 20 MHz to move it back to baseband and uses a low pass filter $h(n)$ to obtain baseband signal. Afterwards, a decimator converts the baseband sampling rate back to 20 MHz. Given the ADC sampling rate 80 MHz is 4 times of the alias frequency 20 MHz, the multiplication with $\cos(20\ MHz)$ and $\sin(20\ MHz)$ can be again simplified with multiplication with constants of (1,0,-1,0) and (0,1,0,-1). Further reductions can be obtained by considering multiplying with zero as a down sampling operation, and push the signs into $h(n)$, which would lead to

$h^{(0)}(n) = h(2n)(-1)^n$ and $h^{(1)}(n) = h(2n+1)(-1)^n$, and detailed deduction can be found in [3]. An efficient DDC is shown in Fig. 5.11.

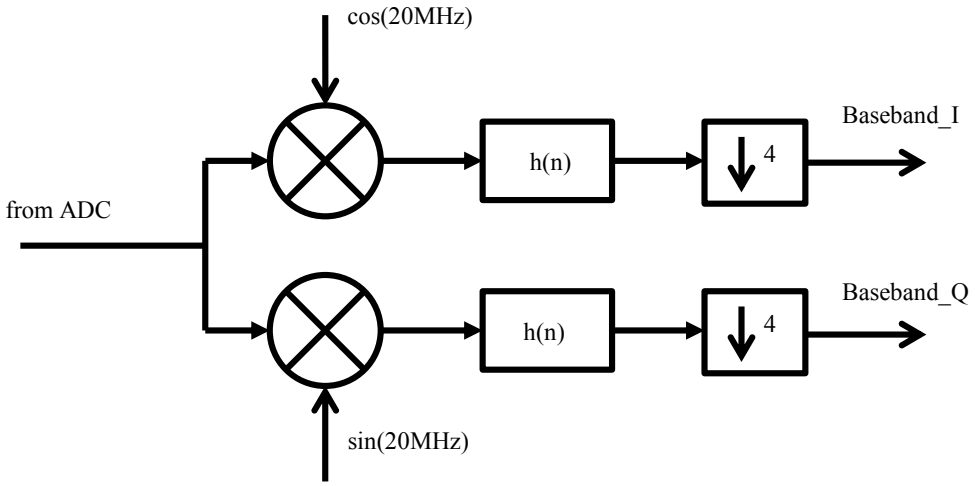

Figure 5.10: Conceptual DDC

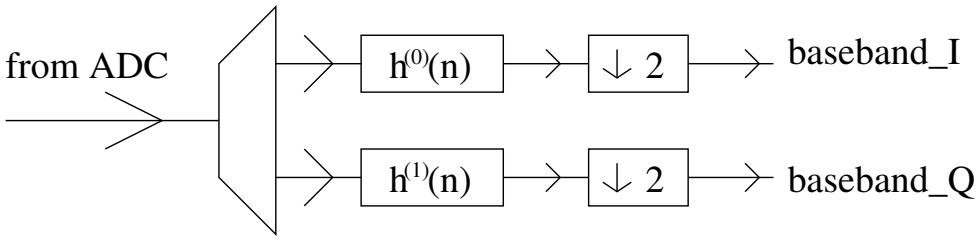

Figure 5.11: Efficient DDC

A complete FMT transceiver has now been presented in the previous discussion, a summary of the FMT used in this project is listed in Table.5.2.

5.4 Summary

In this chapter, the hardware implementation of FMT transceiver was presented. Since the structure used in this article was the same as the structure in [3], only conclusive graphs were described. For receivers, only the core processing part of datapath was shown in Fig.

Table 5.2: FMT Transceiver Specification

FFT Size (Subcarrier Number)	64
Baseband Sampling Rate	20 MHz
Roll-off Factor in tx fb	0.99
Symbol rate	156.25 kHz
Subcarrier separation	312.5 kHz
DAC Sampling Frequency	240 MHz
ADC Sampling Frequency	80 MHz
Digital Intermediate Freq	60 MHz
Modulation Type	BPSK, QPSK, 16QAM

5.5 and the frame detection and timing offset compensation algorithm from [3] was included after that. The last part in the hardware implementation, which was the digital intermediate frequency processing part, was described in the end. In the end, digital up-converter and digital down-converter was illustrated.

Chapter 6

FMT HW/SW Co-design

6.1 NIOS II Custom Instruction Overview

Altera's NIOS II processor is a powerful, versatile soft CPU. NIOS II Custom instructions are custom logic blocks adjacent to ALU in the processor datapath. Custom instruction offers the user the ability to tailor the NIOS II processor core to meet the needs of a particular application. Fig. 6.1 presents how custom instruction connects to NIOS II ALU.

Custom instruction logic can take one or more clock cycles. In the case of only one clock cycle, the custom logic is considered as purely combinational. This combinational logic finishes its logic function within one clock cycle.

If multiply clock cycles are required for the custom logic, `start` and `done` handshake signals must be applied as control signals to communicate with NIOS II. In this project, the custom instruction occupies multi-cycles. NIOS II CPU sends `dataa, datab, n` to the custom instruction wrapper, and reads `result` back. The content of `dataa, datab` and `n` is under the control of a C instruction that is running on top of NIOS II CPU. Therefore,

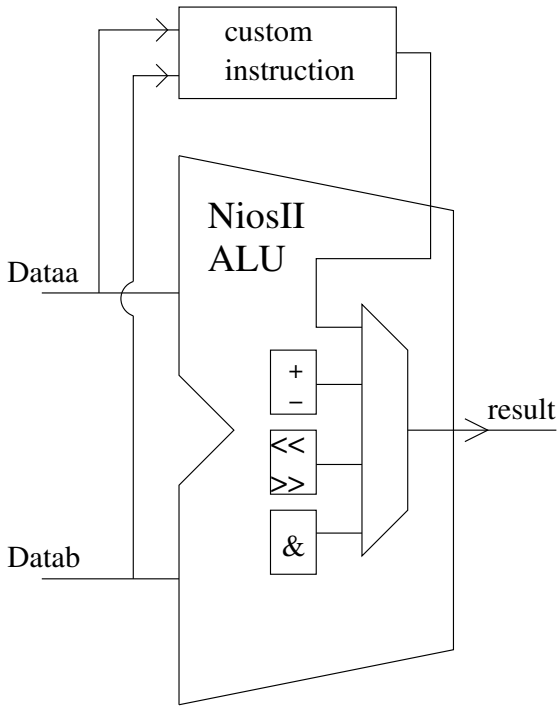

Figure 6.1: Altera Custom Instruction connects to NIOS II ALU

it is called "custom instruction". The NIOS II software interface to C language is simple and abstract the details of the custom instruction from the software developer. For each custom instruction, the NIOS II design tool generates a macro in the system header file, `system.h`. Usually the macro for a certain custom instruction has the following format in the C program, *ALT_CI_DE4_CUSTOM_INSTRUCTION_0(n,A,B)*. This enables the user, from software aspect, to manipulate in hardware the value of `n, dataa` and `datab` and read `result` back from the return value.

6.2 FMT Transceiver Connection NIOS II

The final connection between the FMT transceiver and NIOS II is shown in Fig. 6.2. A custom instruction wrapper is a self-defined block that generates `cad_bus` write events or

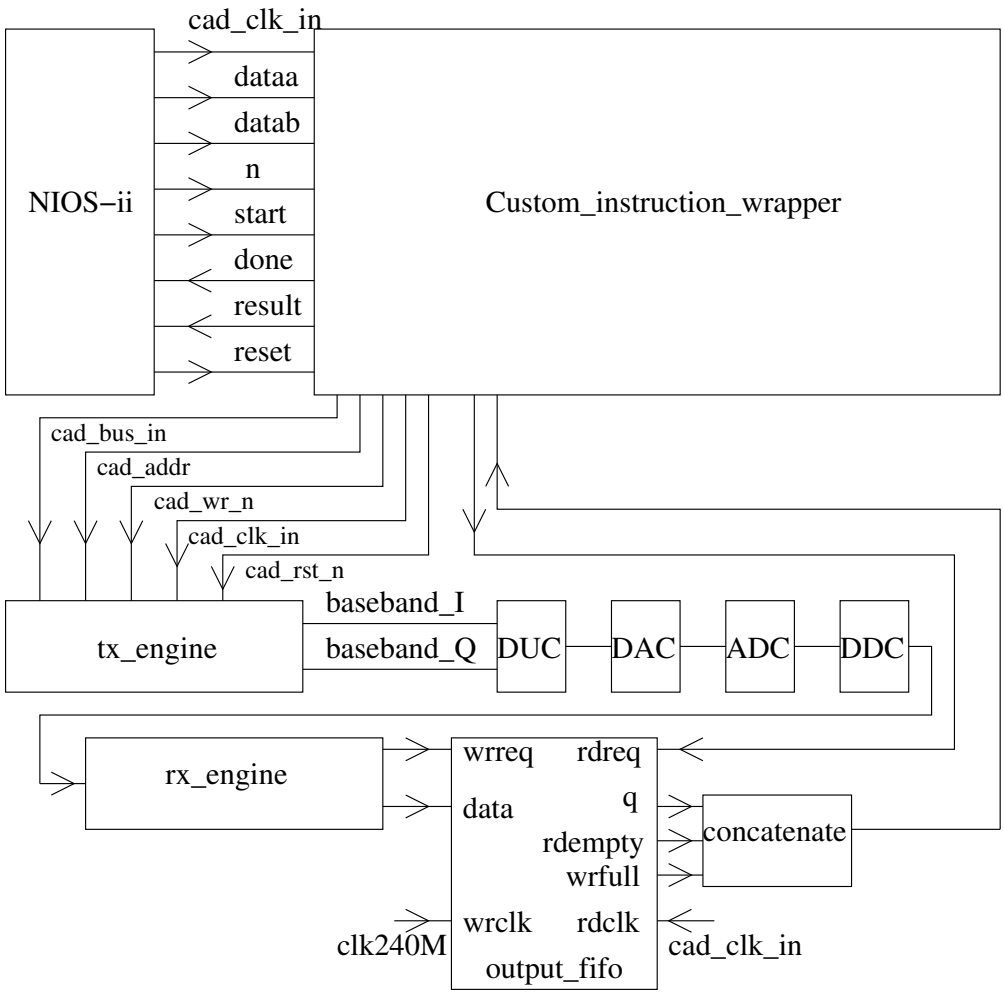

Figure 6.2: FMT transceiver connects to NIOS II through custom instruction

read result back based on `n, dataa and datab` it received. In this design, when n equals to 0, this custom instruction wrapper generates `cad_bus` writing events, and when n equals to 1, it reads in the status of the output FIFO and send it back through result, when n equals to 2, it generates a rdreq signal to the output FIFO and read the result back. A piece of C program is listed below to illustrate how to send data to transmitter.

```
temp = 39885;
for(i=0;i<973;i++)
{
```

ALT_CI_DE4_CUSTOM_INSTRUCTION_0(0,temp,3);

temp++;

}

ALT_CI_DE4_CUSTOM_INSTRUCTION_0(0,0,4);

This piece of C program basically sends 973 consecutive numbers beginning from 39885 to the transmitter. The parameter `n` equal to 0 makes the custom instruction wrapper to generate `cad_bus` writing events. With `n` equals to 0, `datab` will be assigned to `cad_addr` and `dataa` will be assigned to `cad_bus_in`. `Cad_wr_n` will be low for one clock cycle for each `cad_bus` writing event. In the end of this piece of C program, `cad_addr` is assigned as 4, which notifies the transmitter the end of data writing. Similar instructions, shown as below, are used to read the output FIFO status or output FIFO content back. Since these two instructions do not require `dataa` and `datab`, zero is assigned.

ALT_CI_DE4_CUSTOM_INSTRUCTION_0(1,0,0);

ALT_CI_DE4_CUSTOM_INSTRUCTION_0(2,0,0);

C program that is running on top of NIOS II in this project is always checking if the output FIFO is not empty. If it is not empty, it will generate a `rdreq` signal to read the received data until nothing is in the FIFO again.

Chapter 7

Results and Conclusion

7.1 Hardware implementation results

7.1.1 Hardware Platform

The FMT in this thesis is implemented on Altera DE4 Development and Education board. It features an Altera the Stratix IV 4SGX230 FPGA. Stratix IV FPGAs are based on the Taiwan Semiconductor Manufacturing Company (TSMC) 40-ns process technology. A daughter card of a "AD/DA Data Conversion Card" from Terasic is used to carry out digital/analog conversion. The daughter card communicates with the Stratix IV through the HSMC interface.

7.1.2 Used Altera IP

In the core processing part of both the transmitter and receiver, some Altera IP cores are used, which includes FFT/IFFT core, FIR filter core, the FIFO, Divider and PLL. Other

important processing units, e.g. CORDIC translate, CORDIC rotate, transmitter filter bank and receiver filter bank are implemented in a customized way by our team. The implementation of CORDIC and filter bank through pure Verilog coding instead of using existing IPs is one of the major technical challenges in this thesis.

7.1.3 FPGA Utilization

The FPGA utilization for the whole HW/SW co-design as well as transmitter, receiver is listed in Table.7.1. These data are post place and route data. It can be found that FMT transmitters used about 20 % combinational logic than receivers. Besides, FMT receivers used much more DSP resources than FMT transmitters. The integration of NIOS II does not cost much.

An prototype OFDM model in [15] is also built and compiled in Quartus to obtain a general number in terms of the FPGAs resources used. In this thesis, this original OFDM is only built for comparison purpose with FMT engine in terms of resources used, thus, it does not work in terms of functionality, but all the processing blocks are included to generate a aproximate number for resources used. The number of resources used for both OFDM engine and FMT engine on DE4 board is listed in Table.7.2.

In Table.7.2, overhead refers to the percentage of extra resources of an FMT engine consumed than an OFDM engine. Basically, the overhead is calculated by the value under FMT column minus the value under OFDM column and divide over the value of OFDM column. If this is a positive number, for example, the first row in Table.7.2, this means FMT consumes 3.7 % more combinational ALUTs than OFDM. From this table, approximately, FMT consumes about 10 % more in terms of combinational logic than OFDM. This is reasonable, since the major difference is the addition of the polyphase filter bank in FMT, which does not use

Table 7.1: FPGAs Utilization

Total FPGA Utilization	
Combinational ALUTs	29,716 / 182,400 (16 %)
Memory ALUTs	246 / 91,200 (< 1 %)
Dedicated logic registers	41,270 / 182,400 (23 %)
Total pins	305 / 888 (34 %)
Total block memory bits	7,583,248 / 14,625,792 (52 %)
DSP block 18-bit elements	101 / 1,288 (8 %)
Total PLLs	1 / 8 (13 %)
TX FPGA Utilization	
Combinational ALUTs	15054
Memory ALUTs	126
Dedicated logic registers	21272
Total block memory bits	183824
DSP block 18-bit elements	20
RX FPGA Utilization	
Combinational ALUTs	12023
Memory ALUTs	120
Dedicated logic registers	16682
Total block memory bits	192256
DSP block 18-bit elements	77

Table 7.2: OFDM and FMT FPGAs Utilization Comparison

	OFDM	FMT	Overhead
Combinational ALUTs	25743	26721	3.7 %
Memory ALUTs	208	258	24 %
Dedicated logic registers	33650	37640	11.8 %
Total block memory bits	688636	375696	-45 %
DSP block 18-bit elements	65	91	40 %
DSP 9×9	1	1	0 %
DSP 18×18	36	53	47.2 %
DSP 36×36	4	2	50 %

much combinational units. FMT engine uses more memory in this case, due to the different size of fifo used in these two engines. FIFO size or depth is a flexible parameter in the radio

design, hence the block memory bits overhead may not indicate a true overhead cost. As expected, DSP resources are used more in FMT engine, thanks to the addition of polyphase filter bank. It turns out that an FMT engine consumes about 40 % more DSP resources than an OFDM engine.

Table 7.3: FPGAs Utilization Comparison for transmitters

	OFDM(TX)	FMT(TX)	Overhead
Combinational ALUTs	10341	15023	45.2 %
Memory ALUTs	64	126	96.8 %
Dedicated logic registers	13780	21246	54.1 %
Total block memory bits	407056	183824	-54.1 %
DSP block 18-bit elements	12	20	66.6 %
DSP 9×9	0	0	0 %
DSP 18×18	12	16	33.3 %
DSP 36×36	0	0	0 %

Table 7.4: FPGAs Utilization Comparison for Receivers

	OFDM(RX)	FMT(RX)	Overhead
Combinational ALUTs	15375	11636	-24 %
Memory ALUTs	144	132	-8.3 %
Dedicated logic registers	19977	16334	-18.2 %
Total block memory bits	281580	191872	-31.8 %
DSP block 18-bit elements	53	71	33.9 %
DSP 9×9	1	1	0 %
DSP 18×18	24	37	54.1 %
DSP 36×36	4	2	50 %

FPGAs resources usage comparisons for transmitters and receivers respectively are listed in Table.7.3 and Table.7.4. Both FMT transmitters and receivers use more DSP resources than OFDM. However, surprisingly, it can be seen that FMT receiver even uses about 20 % less combinational logic than OFDM receivers. On the other hand, FMT transmitters does use approximately 40 % more combinational logics.

7.1.4 FMT Spectrum

The main difference between FMT and OFDM is the use of filter bank, which restricted the large side lobes. Physical data on hardware are obtained by SignalTap Logic Analyzer. Basically, it is a tool provided by Altera which enables user to examine the behavior of actual internal signals while the design is running at full speed on an FPGA. After obtaining the data from the FPGA, it is analyzed in Matlab for spectrum diagram.

The baseband power spectrum density (PSD) before filtering, can be seen in Fig. 7.1. As seen, the power level difference between data sub-carriers and null sub-carriers is about 20 dB and the first side lobe is at about -3 dB, which results in approximate 13 dB difference from the main lobe.

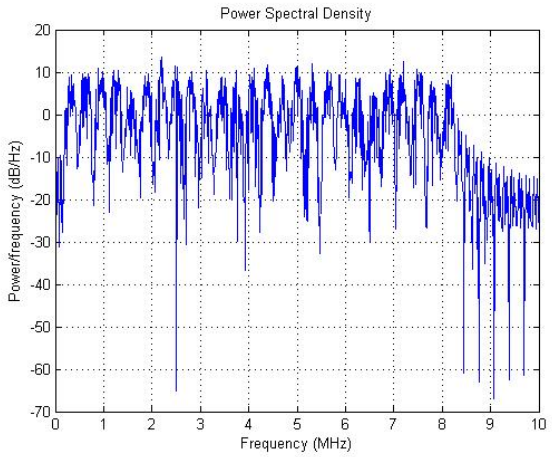

Figure 7.1: FMT baseband power spectrum density before filtering

The PSD of filtered baseband is supposed to have a much smaller side band. Data obtained through SignalTap from DE4 boards are analyzed for the filtered baseband spectrum in Fig. 7.2. As expected, it has much smaller side lobes. Even the first side lobe is about 55 dB below the main lobe. The modulated digital intermediate frequency PSD can be found in as in Fig. 7.3.

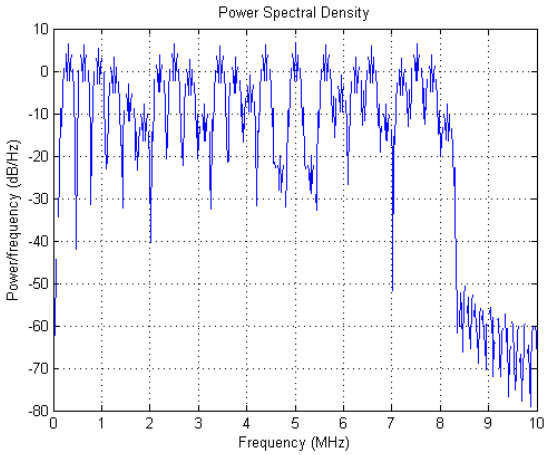

Figure 7.2: FMT Baseband power spectrum density

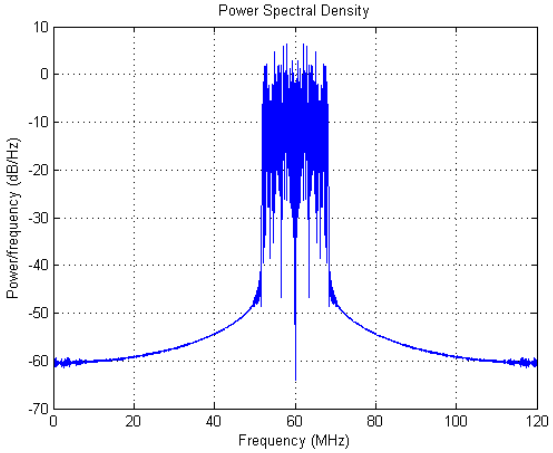

Figure 7.3: FMT IF power spectrum density

7.2 Conclusion

In this thesis, a HW/SW Co-design for a FMT transceiver with NIOS II custom instruction was implemented. Numbers of used FPGAs resources were shown for FMT. The comparison between OFDM and FMT in terms of used FPGAs resources was also illustrated. The results revealed that about 10 % more combinational units were used for FMT and 40 % more DSP 18-bits elements were used for FMT as well. Surprisingly, for FMT receivers, it even used less combinational logic resources. On the other hand, both FMT receivers and

transmitters do use more DSP resources. In general, this overhead is not significant; hence, given the improvement that FMT offers, this technique is a good evolvement from OFDM. The custom instruction feature allowed the user to easily manipulate the FMT radio through pure software interface, e.g. change the data to be sent, activate different sets of sub-carriers, change modulation type for each sub-carrier, on the fly. Since few FMT transceiver hardware implementations are reported, we believe our results to be a good reference example. Meanwhile, the overhead, based on the evolvement from OFDM to FMT in terms of FPGAs resources, is slim. The author believes the overhead found in this thesis is a good hardware usage reference example. Future work includes researching different synthesis/analysis filter design, for example isotropic orthogonal transform algorithm prototype [39].

Bibliography

[1] Q. Zhao and B. M. Sadler, "A survey of dynamic spectrum access," *Signal Processing Magazine, IEEE*, vol. 24, no. 3, pp. 79–89, 2007.

[2] B. Farhang-Boroujeny, "OFDM versus filter bank multicarrier," *Signal Processing Magazine, IEEE Journal on*, pp. 92–112, 2011.

[3] A. Recio, *Spectrum-Aware Orthogonal Frequency Division Multiplexing*. PhD thesis, Virginia Tech, USA, 2010.

[4] F. spectrum Policy Task Force, "Report of the spectrum efficiency working group," Nov 2002.

[5] S. Haykin, "Cognitive radio: Brain-empowered wireless communications," *Selected Areas in Communications, IEEE Journal on*, pp. 201–220, 2005.

[6] J. Mitola and G. Maguire, "Cognitive radio: Making software radios more personal," *Personal Communications, IEEE Journal on*, pp. 13–18, 1999.

[7] J. Mitola, *Cognitive radio: An integrated agent architecture for software defined radio*. PhD thesis, Royal Inst. Technol. (KTH), Stockholm, Sweden, 2000.

[8] J. Mitola, *Software Radio Architecture: Object Oriented Approaches to Wireless System Engineering*. John Wiley and Sons, 2000.

[9] A. B. MacKenzie, J. H. Reed, P. Athanas, C. W. Bostian, R. M. Buehrer, L. A. DaSilva, S. W. Ellingson, Y. T. Hou, M. Hsiao, J.-M. Park, *et al.*, "Cognitive radio and networking research at virginia tech," *Proceedings of the IEEE*, vol. 97, no. 4, pp. 660–688, 2009.

[10] A. Garhwal and P. P. Bhattacharya, "Dynamic spectrum access in cognitive radio: a brief review," *International Journal of Computer Application in Engineering Sciences, Special Issue on Computer Networks & Security*, vol. 1, pp. 149–153, 2011.

[11] D. N. Hatfield and P. J. Weiser, "Property rights in spectrum: taking the next step," in *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on*, pp. 43–55, IEEE, 2005.

[12] L. Xu, R. Tonjes, T. Paila, W. Hansmann, M. Frank, and M. Albrecht, "Drive-ing to the internet: Dynamic radio for ip services in vehicular environments," in *Local Computer Networks, 2000. LCN 2000. Proceedings. 25th Annual IEEE Conference on*, pp. 281–289, IEEE, 2000.

[13] Y. Benkler, "Overcoming agoraphobia: building the commons of the digitally networked environment," *Harv. JL & Tech.*, vol. 11, p. 287, 1997.

[14] W. Lehr and J. Crowcroft, "Managing shared access to a spectrum commons," in *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on*, pp. 420–444, IEEE, 2005.

[15] A. Recio and P. Athanas, "Physical layer for spectrum-aware reconfigurable OFDM on an FPGA," in *Digital System Design: Architectures, Methods and Tools (DSD), 2010 13th Euromicro Conference on*, 2010.

[16] C. Vennila, K. Palaniappan, K.-V. Krishina, and K. Seok-bum, "Dynamic partial reconfigurable FFT/IFFT pruning for OFDM based cognitive radio," in *Circuits and Systems (ISCAS), 2012 IEEE International Symposium on*, pp. 33–36, May 2012.

[17] D. Wasden, H. Moradi, and B. Farhang-Boroujeny, "Design and implementation of an underlay control channel for cognitive radios," *Selected Areas in Communications, IEEE Journal on*, pp. 1875–1889, 2012.

[18] D.-W. Lim, S.-J. Heo, and J.-S. No, "An overview of peak-to-average power ratio reduction schemes for ofdm signals," *Communications and Networks, Journal of*, vol. 11, no. 3, pp. 229–239, 2009.

[19] T. Jiang and Y. Wu, "An overview: peak-to-average power ratio reduction techniques for ofdm signals," *Broadcasting, IEEE Transactions on*, vol. 54, no. 2, pp. 257–268, 2008.

[20] C. H. Yuen, P. Amini, and B. Farhang-Boroujeny, "Single carrier frequency division multiple access (sc-fdma) for filter bank multicarrier communication systems," in *Cognitive Radio Oriented Wireless Networks & Communications (CROWNCOM), 2010 Proceedings of the Fifth International Conference on*, pp. 1–5, IEEE, 2010.

[21] H. G. Myung, J. Lim, and D. J. Goodman, "Peak-to-average power ratio of single carrier fdma signals with pulse shaping," in *Personal, Indoor and Mobile Radio Communications, 2006 IEEE 17th International Symposium on*, pp. 1–5, IEEE, 2006.

[22] D. S. Waldhauser, L. G. Baltar, and J. A. Nossek, "Comparison of filter bank based multicarrier systems with ofdm," in *Circuits and Systems, 2006. APCCAS 2006. IEEE Asia Pacific Conference on*, pp. 976–979, IEEE, 2006.

[23] T. Weiss, J. Hillenbrand, A. Krohn, and F. K. Jondral, "Mutual interference in ofdm-based spectrum pooling systems," in *Vehicular Technology Conference, 2004. VTC 2004-Spring. 2004 IEEE 59th*, vol. 4, pp. 1873–1877, IEEE, 2004.

[24] S. Brandes, I. Cosovic, and M. Schnell, "Sidelobe suppression in ofdm systems by insertion of cancellation carriers," in *Vehicular Technology Conference, 2005. VTC-2005-Fall. 2005 IEEE 62nd*, vol. 1, pp. 152–156, IEEE, 2005.

[25] I. Cosovic and V. Janardhanam, "Sidelobe suppression in ofdm systems," *Multi-Carrier Spread-Spectrum*, pp. 473–482, 2006.

[26] I. Cosovic and T. Mazzoni, "Suppression of sidelobes in ofdm systems by multiple-choice sequences," *European transactions on telecommunications*, vol. 17, no. 6, pp. 623–630, 2006.

[27] M. Morelli, C.-C. Kuo, and M.-O. Pun, "Synchronization techniques for orthogonal frequency division multiple access (ofdma): A tutorial review," *Proceedings of the IEEE*, vol. 95, no. 7, pp. 1394–1427, 2007.

[28] G. Cherubini, E. Eleftheriou, and S. Olcer, "Filtered multitone modulation for vdsl," in *Global Telecommunications Conference, 1999. GLOBECOM'99*, vol. 2, pp. 1139–1144, IEEE, 1999.

[29] B. Saltzberg, "Performance of an efficient parallel data transmission system," *Communication Technology, IEEE Transactions on*, vol. 15, no. 6, pp. 805–811, 1967.

[30] B. Farhang-Boroujeny, "Multicarrier modulation with blind detection capability using cosine modulated filter banks," *Communications, IEEE Transactions on*, vol. 51, no. 12, pp. 2057–2070, 2003.

[31] P. Amini, R. Kempter, R. Chen, L. Lin, and B. Farhang-Boroujeny, "Filter bank multitone: A physical layer candidate for cognitive radios," in *Software Defined Radio Technical Conference*, November 2005.

[32] U. Rohrs and L. Linde, "Some unique properties and applications of perfect squares minimum phase cazac sequences," in *Communications and Signal Processing, 1992. COMSIG'92., Proceedings of the 1992 South African Symposium on*, pp. 155–160, IEEE, 1992.

[33] P. Zhou, H. Hu, H. Wang, and H.-h. Chen, "An efficient random access scheme for ofdma systems with implicit message transmission," *Wireless Communications, IEEE Transactions on*, vol. 7, no. 7, pp. 2790–2797, 2008.

[34] C. Cordeiro, K. Challapali, D. Birru, and N. Sai Shankar, "Ieee 802.22: the first worldwide wireless standard based on cognitive radios," in *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on*, pp. 328–337, IEEE, 2005.

[35] K. Youngok and K. Jaekwon, "Low complexity fft schemes for multicarrier demodulation in ofdma systems," *IEICE transactions on communications*, vol. 90, no. 11, pp. 3290–3293, 2007.

[36] S. He and M. Torkelson, "Computing partial dft for comb spectrum evaluation," *Signal Processing Letters, IEEE*, vol. 3, no. 6, pp. 173–175, 1996.

[37] AlteraCorp, "Accelerating duc and ddc system designs for wimax," May 2007.

[38] F. J.Harris, C. Dick, and M. Rice, "Digital receivers and transmitters using polyphase filter banks for wireless communications," *Microwave Theory and Techniques, IEEE Transactions on*, pp. 1395–1412, 2003.

[39] B. Le Floch, M. Alard, and C. Berrou, "Coded orthogonal frequency division multiplex [tv broadcasting]," *Proceedings of the IEEE*, vol. 83, no. 6, pp. 982–996, 1995.

Appendix A: NIOS II Software for CCM FMT

```
#include <stdio.h>
#include <system.h>

int main()
{
  int temp, i, flag =0, rx_data, rx_data_cnt=0;

  printf("This is CCMLAB FMT Radio!\n");

  temp = 39885;    // temp = 9BCD in hex
  // Writing Data to TX
  printf("Beginning to write data to transmitter\n");

  for(i=0;i<973;i++)
  {
```

```
    ALT_CI_DE4_CUSTOM_INSTRUCTION_0(0,temp,3); // n =0 ---->
        writing data to TX, dataa = cad_bus_in, datab= cad_addr
      temp++;
      if (  (i%100) == 0) printf("..");
}

ALT_CI_DE4_CUSTOM_INSTRUCTION_0(0,51966,4); //Writing CAFEh to
    TX at addr of 4-->means end writing


//waiting for the Output FIFO to be NOT empty
for(;;)
{
   temp = ALT_CI_DE4_CUSTOM_INSTRUCTION_0(1,0,0);
   temp = temp / 65536;
   if (temp == 1 && flag ==0) {printf("\nWaiting
       Receiver.....");  flag = 1;}
      if (temp == 0) {printf("\nData found in
          RX_output_fifo\n");break;}
}


//reading data from RX
for(;;)
{
```

```
    temp = ALT_CI_DE4_CUSTOM_INSTRUCTION_0(1,0,0);
    temp = temp / 65536;
    if (temp == 0) {rx_data =
       ALT_CI_DE4_CUSTOM_INSTRUCTION_0(2,0,0);
                    printf("rx_data[%d] = %d \n", rx_data_cnt,
                       rx_data);
                    rx_data_cnt++;
                   }
      if( rx_data_cnt == 973) break;
  }

  printf("Testing CCMlab FMT is OVER!\n");

  return 0;
}
```

Listing 1: NIOS II software for testing FMT engine