CHAPTER 4

WCDMA AND OFDM IMPLEMENTATION

4.1 INTRODUCTION

The reconfigurable architecture suitable to WCDMA and OFDM based WiFi technology has been discussed in Chapter 3. The receiver algorithms needed in the baseband sections of WCDMA and OFDM receivers are presented in this chapter. This chapter addresses an efficient algorithmic implementation of RAKE receiver of WCDMA and FFT processing of OFDM technology. These algorithms belong to the class of algorithmic strength reduction, where the number of complex multiplication operations is reduced in lieu of an increase in the number of simple addition operations.

4.1 WCDMA RECEIVER IMPLEMENTATION

4.1.1 RAKE Receiver Implementation

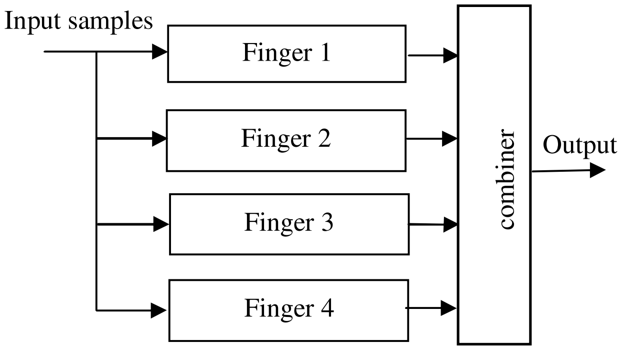

Figure 4.1 Components of RAKE receiver

Figure 4.1 shows the components of RAKE receiver, to be implemented using the architecture presented in Chapter 3. It contains 4 fingers and a combiner block. Each RAKE finger is an independent receiver for the signal from a specific cell on a specific propagation path. Multipath propagation channels and soft handover situations are handled by multiple fingers, one for each propagation path. The output symbols of all the fingers are then combined coherently and synchronously by the RAKE receiver's combiner block to yield the received data symbols. The functions of RAKE fingers will be discussed in the following section.

4.1.1.1 RAKE finger

Each RAKE finger independently detects the signal contribution from each of the multipath in a given cell. Detection is done by correlating the received signal sample stream over the length of Spreading Factor (SF) samples with a time aligned copy of the scrambling and spreading codes. At its input, the RAKE finger takes input samples at the chip rate, and it produces the correlation results at the symbol rate.

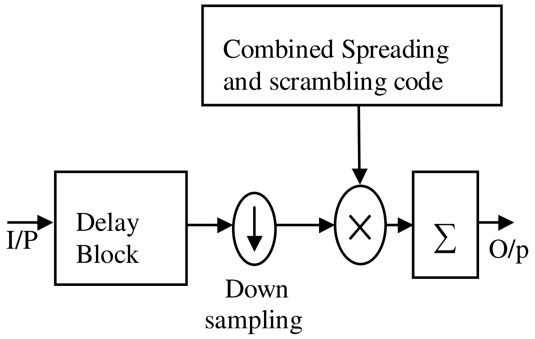

Figure 4.2 Schematic diagram of RAKE finger

Figure 4.2 shows the schematic diagram of a RAKE finger. Each RAKE finger consists of delay block, down-sampler which reduces the rate of the input signal into the data, multiplier for multiplying the combined spreading and scrambling codes with the down-sampled input signal and an accumulator of length spreading factor.

4.1.1.2 Optimized RAKE finger implementation

In WCDMA receivers, demodulation is performed in the RAKE fingers by correlating the received signal with a spreading code over a period corresponding to the spreading factor. The set of RAKE finger correlators are the most computationally expensive part of a WCDMA receiver, since they are normally implemented in specialized hardware blocks. The output Y of i^{th} Rake finger for the sample 'n' can be expressed as

$$Y_i(n) = \sum_{k=0}^{L_{sf}-1} C_s(k + nL_{sf})R(k + nL_{sf}) \tag{4.1}$$

where L_{sf} is the length of spreading factor which is usually 4 to 16 chips , C_s is the combined spreading and scrambling code and R is the received spread spectrum signal. Both C_s and R are complex numbers. Since the scrambling and spreading codes are always of ±1, the multiplication and addition of each correlation stage is simplified. So Equation (4.1) is simplified to

$$(R_r + jR_i)(C_{sr} - jC_{si}) = (R_rC_{sr} + R_iC_{si}) + j(R_iC_{sr} - R_rC_{si}) \tag{4.2}$$

Multiplication is an important fundamental operation in arithmetic operations of communication systems. In fact, multiplication-based operations such as Multiply and Accumulate and inner product are among some of the frequently used computation-intensive arithmetic functions currently implemented in many DSP applications (such as convolution, FFT, filtering and others). They usually contribute significantly to the time delay and take up a great deal of silicon area in the DSP system. Since multiplication dominates the execution time of most DSP algorithm, using a high-speed multiplier is very desirable (Kiat-Seng Yeo and Kaushik Roy 2005). With an ever increasing quest for greater computing power on battery-operated mobile devices, design emphasis has shifted to optimizing area, conventional delay time and minimizing power dissipation while maintaining high performance. Compared with previous designs, optimizing area and delay time and low power consumption design allow portable devices to operate longer with the same amount of battery charge.

So area efficient and fast multipliers are the essential blocks for high performance computing. Therefore multiplier reduction is needed in the architecture design so that large number of other circuits may be integrated on a single chip. The reconfigurable architecture presented in Chapter 3 uses area efficient and fast multiplier type called Dadda as discussed in Section 3.3.4.3. The number of multipliers is further reduced in the RAKE receiver architecture as follows.

If the value +1 is represented as logic ‘0’ and the value -1 is represented as logic ‘1’, Equation (4.2) is simplified as follows

$$(R_r+jR_i)(C_{sr}-jC_{si}) = \begin{cases} R_r+R_i+j(R_i-R_r), \text{ when } C_{sr}=0, C_{si}=0 \\ R_r-R_i+j(R_i+R_r), \text{ when } C_{sr}=0, C_{si}=1 \\ -(R_r-R_i)-j(R_i-R_r), \text{ when } C_{sr}=1, C_{si}=0 \\ -(R_r+R_i)-j(R_i-R_r), \text{when } C_{sr}=1, C_{si}=1 \end{cases} \quad (4.3)$$

Since the code input is binary valued, a complex multiplication in the correlations is simplified to one real addition/subtraction and one imaginary addition/subtraction. Selection of addition or subtraction is done with the use of multiplexers. So the total resources required to implement a RAKE receiver using Equation (4.3) are two adders and two subtractors only. The code input is applied to a multiplexer to select the corresponding addition and subtraction as given in Equation (4.3). PE and its resources required for the proposed implementation of RAKE finger in WCDMA is shown in Figure 4.3.

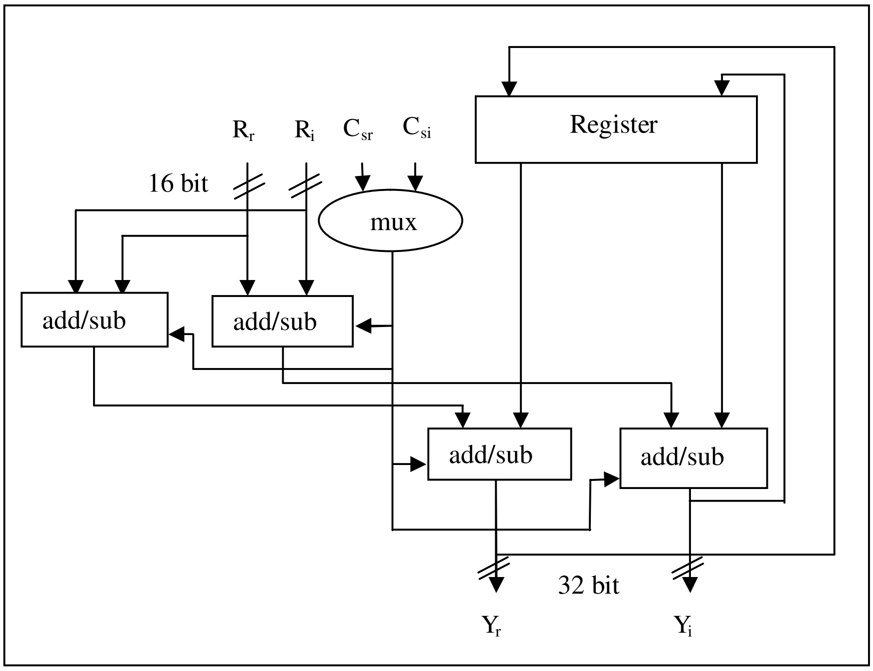

Figure 4.3 PE and its resources of RAKE finger

4.3 OFDM RECEIVER IMPLEMENTATION

4.3.1 IFFT/FFT

The key kernel in an OFDM receiver is FFT processor. In WiFi standards it works with 64 carriers at a sampling rate of 20 MHz, so a 64-point IFFT/FFT processor is required. The FFT is derived from the main function which is called DFT. The idea of using FFT instead of direct computation of DFT is that the computation can be made faster where this is the main criterion for implementation. In direct computation of DFT the computation for N-point DFT is calculated one by one for each point. But for FFT, computation is done in parallel, and this algorithm helps save a lot of time.

The derivation starts from the fundamental DFT equation for an N point FFT. The equation of IDFT is given in Equation (4.4) and the equation of DFT is given in Equation (4.5).

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)W_N^{-nk}, \qquad n = 0,1,.......N-1 \tag{4.4}$$

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{nk}, \qquad k = 0,1,.......N-1 \tag{4.5}$$

where W is called twiddle factor, which is defined as

$$W_N^{nk} = e^{-j2\pi nk/N} \tag{4.6}$$

This factor is stored as a look up table in order to reduce computation and this factor does not need to be recalculated. The twiddle factor table depends on the number of points used. This thesis uses Radix-2 algorithm for implementing FFT computation.

There are two different radix-2 algorithms available, namely

a) Decimation-in-Time (DIT) Algorithm and

b) Decimation-in-Frequency (DIF) Algorithm.

FFTs can be decomposed using DFTs of even and odd points, which are called a DIT-FFT, or they can be decomposed using a first-half/second-half approach, which is called DIF-FFT. In this thesis, radix-2 DIF algorithm is employed for implementing the design.

4.3.2 Optimized FFT Implementation

The 2-point Butterfly structure of the DIF-FFT algorithm (Paul Heysters et al 2003) is shown in Figure 4.4 where multiplication is performed with the twiddle factor after subtraction.

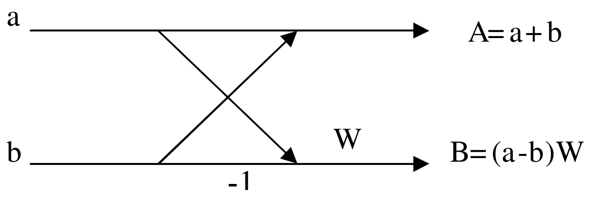

Figure 4.4 2-Point butterfly structure of DIF-FFT

Multiplication is certainly the most vital operation in signal processing in a communication system, and its implementation in an integrated circuit requires large hardware resources and significantly affects the size, performance, and power consumption of a system (Heiko Hinkelmann et al 2009). So an efficient way of multiplier reduction in FFT computation is done as follows. Consider the problem of computing the product of two complex numbers R and W.

$$\begin{aligned} B = RW &= (R_r+jR_i)(W_r+jW_i) \\ &= (R_rW_r-R_iW_i)+j(R_rW_i+R_iW_r) \qquad (4.7) \end{aligned}$$

From Equation (4.7), direct architectural implementation requires a total of four multiplications and one real subtraction and one imaginary addition to compute the complex product. However, by applying algorithmic strength reduction transformation technique (Chandrakasan et al 1995) Equation (4.7) is reformulated as,

$$B_r = (R_r-R_i)Wi+R_r(W_r-W_i) \qquad (4.8a)$$

$$B_i = (R_r - R_i)W_i + R_i(W_r + W_i) \qquad (4.8b)$$

As can be seen from Equations (4.8a) and (4.8b), by using algorithmic strength reduction transformation technique the total number of multiplications is reduced to only three. This however is at the expense of having two additional subtractors and one adder. This type of multiplier reduction technique is exploited to design FFT architecture of OFDM receiver that requires less hardware resources. The reduction of multipliers leads to a reduction in hardware complexity while implementing FFT algorithm in the proposed architecture. This transformation can lead to reduction in silicon area and power consumption in a VLSI implementation or iteration period for 64-point FFT processor implementation.

In order to implement the above optimized 2-point butterfly DIF-FFT operation, it makes use of the two types of PEs to produce the output. Figures 4.5(a) and 4.5(b) show PE Type 1 and PE Type 2 and their resources required for the implementation of FFT computation. PE Type 1 is mapped on PEAS that is used for real and imaginary addition and subtraction. PE Type 2 is mapped on PEM that is used for real and imaginary multiplication with the complex twiddle factor coefficients after subtraction.

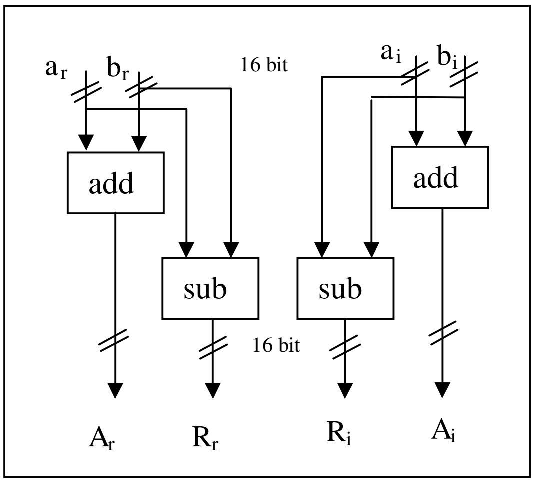

Figure 4.5(a) PE type 1 and its resources

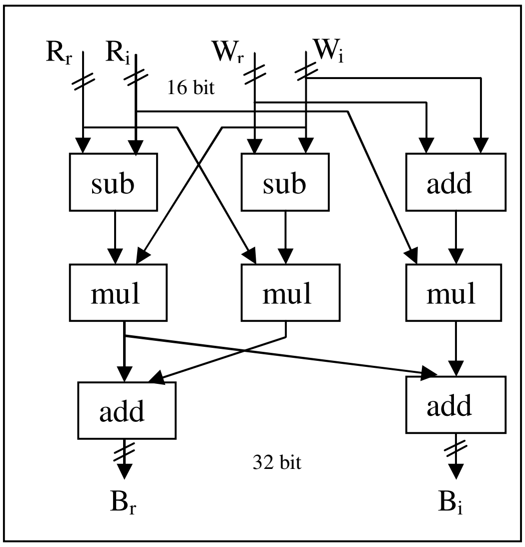

Figure 4.5(b) PE type 2 and its resources

4.4 SUMMARY

In this chapter, efficient implementations of RAKE receiver which is the main component of WCDMA receiver and FFT which is the key kernel of OFDM receiver have been proposed. In order to reduce computational resources, multiplier-less receiver implementation and algorithmic strength reduction transformation technique have been exploited in hardware implementation. More detailed information about the results of these implementations and comparisons with the existing architecture implementation are provided in Chapter 6.