Design and Simulation of 64-Point FFT Using Radix-4 Algorithm

Ramesh Kumar.B[1], K.Satyanarayana Raju[2]

[1,2]*Department of ECE, MVGR College of Engineering, Vizianagaram, Andhrapradesh*

***Abstract*: A parallel and pipelined Fast Fourier Transform (FFT) processor for use in the Orthogonal Frequency division Multiplexer (OFDM) . Unlike being stored in the traditional ROM, the twiddle factors in our pipelined FFT processor can be accessed directly. A novel simple address mapping scheme and the modified radix 4 FFT also proposed. Finally, the pipelined 64-point FFT processor can be completely implemented.**

Keywords : OFDM, radix- 4,DIT-FFT,DIF-FFT

1. INTRODUCTION

Active radar systems are commonly used in military and commercial applications for the detection and tracking of objects through a medium such as air. Most active radar systems work by transmitting a signal pulse through the medium and receiving the reflected energy wave field as it scatters off objects in the medium. By processing the received wave field, an active radar system can determine an object's distance, velocity, and other features. A passive radar system can consist solely of an array of receiving antennas as opposed to an active radar system with a co-located transmitter and receiver.

Without a transmitter the passive radar system relies on other sources of electromagnetic waves, such as AM or FM radio waves, TV broadcast, nearby radar, cell towers, or a wideband waveform to “illuminate” objects in the medium. Various signal processing algorithms can be applied to the data from the antenna array. One algorithm of interest is the Direction of Arrival (DOA) of the incoming scattered waveform and therefore the object. A computationally efficient version of the DFT is known as the Fast Fourier Transform (FFT) and is commonly used to reduce the computational burden.

In the communication world of today more and more OFDM systems are brought on-line with an ever increasing number of standards, applications, and services, all with different requirements on the physical layer of the transceiver. The physical layer is most often, due to speed and power constraints, implemented as an ASIC and thus locked to one specific case. Hence, the number of required implementations will grow fast. Due to this fact, there will be an interest from the industry’s point of view to search for a flexible architecture that can be configured to function with several standards, applications, or services. A flexible architecture will lead to reusage of design and implementation and thus reduced cost. This project addresses both OFDM in theory and the implementation aspects of a flexible hardware solution for digital baseband OFDM.

In this project a flexible FFT processor and a flexible
OFDM transmitter are presented. The FFT processor is a central part of an OFDM transceiver, and has been fabricated both as a stand alone chip and as part of an OFDM transmitter chip. The transmitter chip includes in addition to the FFT, a signal mapper and a cyclic prefix inserter. It is shown that high flexibility is obtained in the transmitter with a reasonable amount of extra hardware. Comprehensive measurements on power consumption are presented for both chips.

A scheme to reduce hardware and delay in an OFDM transceiver is proposed in this project. The scheme shows that the data buffers in an OFDM transceiver, due to a cyclic prefix and a bit reversing pipelined FFT processor, can be significantly reduced with a cyclic suffix and a bidirectional FFT processor.

2. BASIC THEORY OF OFDM

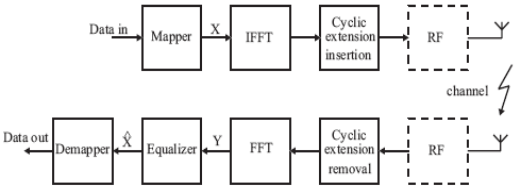


Figure 2.1: The digital baseband parts of an OFDM transceiver.

The digital baseband parts of an OFDM transceiver [12], is shown in Figure 1. The basic idea of OFDM is to divide the available spectrum into N orthogonal subchannels. In the mapper, data is converted to signals located in the frequency domain where each subchannel is assigned one signal. The signals are then transformed to the time domain with the inverse fast Fourier transform (IFFT). The FFT is an efficient method to implement the DFT algorithm, based on a divide and conquer approach [13]. The implementation of an FFT is discussed in Chapter 3 and from now on, only the FFT notation will be used. The last digital part of the transmitter inserts a cyclic extension to remove the effects of intersymbol interference (ISI) and interchannel interference (ICI).

The FFT and the IFFT of length N, is defined by the equations

$$X_k = \sum_{i=0}^{N-1} x_i W_N^{ki}, \quad 0 \le k < N \qquad (2.1)$$

$$x_i = \frac{1}{N}\sum_{k=0}^{N-1} X_k W_N^{-ki}, \quad 0 \le k < N, \qquad (2.2)$$

Where $W_N = e^{-j\frac{2\pi}{N}}$ is called twiddle factor,
i is the sample index, and
k is the subcarrier index.

In the receiver the cyclic extension is discarded and the transmitted signals are transformed back to the frequency domain with an FFT. The N received signals points Y, can be written as

$$Y = FFT(IFFT(X)\circ g + \beta)$$

where '∘' is a circular convolution, X the N transmitted signal points, g the channel impulse response, and β is the channel noise. Due to the fact that the cyclic extension removes both ISI and ICI, and that the signals are located in the frequency domain, the effect of the channel impulse is easily removed in the equalizer, compared to single tone systems. An estimate of the transmitted signals is written as

$$\hat{\mathbf{X}} = \frac{\mathbf{Y}}{\mathbf{G}}$$

where G is the FFT of g. To estimate g known data are transmitted over the channel, these data are called pilots. In the demapper the estimated signals ˆX are converted back to data.

3. IFFT AND FFT

The Inverse Fast Fourier Transform (IFFT) transforms signals from frequency domain to time domain and the FFT performs the reverse operation. To keep signals in the frequency domain simplifies some signal processing operations, e.g. convolution in the time domain becomes multiplication in the frequency domain. The transformation into the time domain is done in order to reduce the number of backend RF-oscillators and demodulators [5].

The available bandwidth (B) in frequency is split into N sub channels, one for each subcarrier, where each has a power spectrum shape of a squared sine pulse. The individual power spectra, after the IFFT in the transmitter, of a number of subcarriers. Since the IFFT is a linear operation the sub-carriers can be separated again with the FFT, even though there spectrum overlap. This is also true after the signals have passed a multipath channel, due to the cyclic extension. The property that each subcarrier is unaffected by the other subcarriers is called orthogonality, hence the name OFDM.

FFT

The first designed chip is an FFT processor. The FFT processor has a central position both in the OFDM transmitter and receiver. The FFT is a computationally demanding operation that requires an ASIC implementation to reach high performance, i.e. high throughput combined with low energy consumption.

FFT Architecture

The FFT and IFFT Equation 2.1 and 2.2 has the property that, if

FFT(Re(xi)+ jIm(xi)) = Re(Xi)+ jIm(Xi)

and

IFFT(Re(Xi)+ jIm(Xi)) = Re(xi)+ jIm(xi),

where xi and Xi are N words long sequences of complex valued, samples and sub-carriers respectively, then

1/N * FFT(Im(Xi)+ jRe(Xi)) = Im(xi)+ jRe(xi).

Thus, it is only necessary to discuss and implement the FFT equation. To calculate the inverse transform, the real and imaginary part of the input and output are swapped. Since N is a power of two, scaling with 1/N is the same as right shift the binary word Log2(N) bits. Even simpler, is to just remember that the binary point has moved log2(N) bits to the left. Not performing the bit shift until, if ever, it is necessary, which depends on how the output from the IFFT will be used.

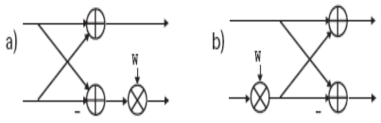


Figure 3.1: (a)A radix-2 DIF butterfly and (b) a radix-2 DIT butterfly

where W is the twiddle factor.

The FFT algorithm can be realized with a butterfly operation as the basic building block [23]. There are two types of butterfly operations, decimation in time (DIT) and decimation in frequency (DIF), both are shown in Figure 3.1. The difference between DIT and DIF lies in the position of the twiddle factor multiplication, which is either performed before or after the subtraction and addition. Since the FFT is based on divide and conquer, it will be most efficient if the input sequence is of length N = rp, where N is called point, r is called radix, and p is a positive integer. To compute an N-point FFT, p stages of butterflies are connected. An example of a hardware mapped N = 16-point radix-2 DIF FFT is shown in Figure 3.2. Observe that the input data, x(n), occurs in consecutive order, whereas the output data, X(n), is rearranged. The order of the rearranged data is called bit reversed order and is explained in Section 3. It is possible to reconfigure the FFT in Figure 3.2 so that the input occurs in bit reversed order and the output occurs in natural order. The reconfiguration can be thought of as vertically moving the horizontal data paths in Figure 3.2, while the cross connections remain fixed to respective path, until the output is in natural order. Also notice that, if all arrows in Figure 3.2 is turned around a DIT FFT is performed instead of a DIF FFT.

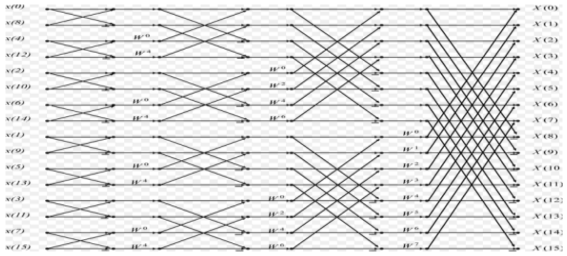


Figure 3.2: A hardware mapped N = 16-point radix-2 DITFFT Algorithm.

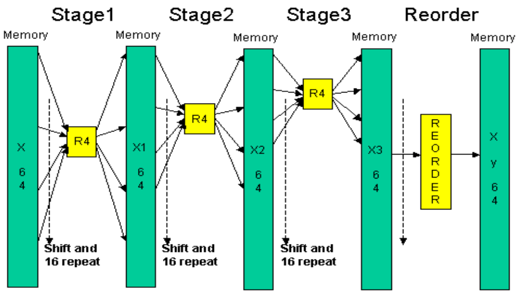


Figure:3.3 RADIX-4 64 points FFT architecture

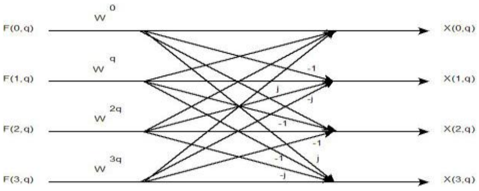


Figure 3.4: A Simple Radix 4 DIF FFT algorithm.

When N is a power of 4, i.e. N =4p, a radix-4 FFT can be used instead of a radix-2 FFT. With a radix-4 the computational complexity is reduced, i.e. the number of complex multiplications is reduced compared to a radix-2 FFT. The drawback with a radix-4 is that the butterfly structure is more complicated. Another, better type of radix-4 is the radix-22, there the butterflies still look like radix-2 butterflies but the number of complex multipliers are reduced as if it were a radix-4 [24]. Figure 3.2 shows an N = 16-point radix-22 DIF FFT. Each stage in a radix-22 FFT consists of two butterflies, one trivial multiplier, i.e. multiplication with −j, and one complex multiplier that multiplies data with the twiddle factor.

4. SIMULATIONRESULTS

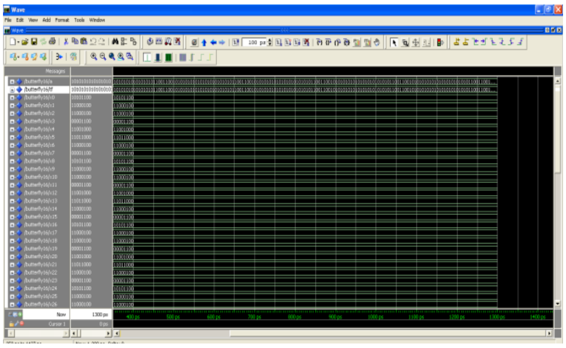
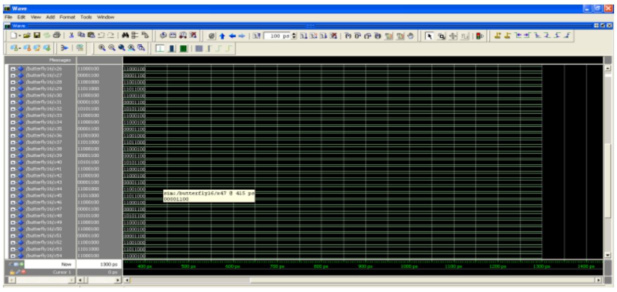
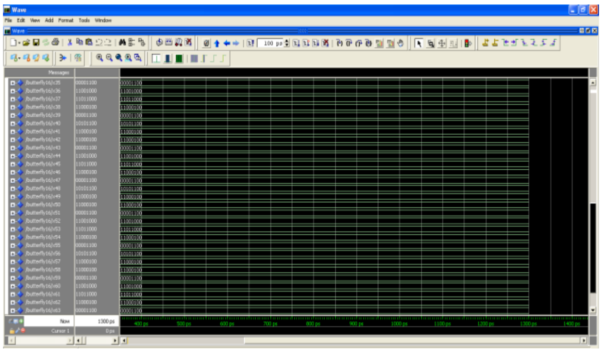


Figure: 4.1



```
/butterfly16/x26  11000100
/butterfly16/x27  00001100
/butterfly16/x28  11001000
/butterfly16/x29  11011000
/butterfly16/x30  11000100
/butterfly16/x31  00001100
/butterfly16/x32  10101100
/butterfly16/x33  11000100
/butterfly16/x34  11000100
/butterfly16/x35  00001100
/butterfly16/x36  11001000
/butterfly16/x37  11011000
/butterfly16/x38  11000100
/butterfly16/x39  00001100
/butterfly16/x40  10101100
/butterfly16/x41  11000100
/butterfly16/x42  11000100
/butterfly16/x43  00001100
/butterfly16/x44  11001000
/butterfly16/x45  11011000
/butterfly16/x46  11000100
/butterfly16/x47  00001100
/butterfly16/x48  10101100
/butterfly16/x49  11000100
/butterfly16/x50  11000100
/butterfly16/x51  00001100
/butterfly16/x52  11001000
/butterfly16/x53  11011000
/butterfly16/x54  11000100
Now  1300 ps
Cursor 1  0 ps
sim:/butterfly16/x47 @ 415 ps
00001100
```

Figure: 4.2



```
/butterfly16/x35  00001100
/butterfly16/x36  11001000
/butterfly16/x37  11011000
/butterfly16/x38  11000100
/butterfly16/x39  00001100
/butterfly16/x40  10101100
/butterfly16/x41  11000100
/butterfly16/x42  11000100
/butterfly16/x43  00001100
/butterfly16/x44  11001000
/butterfly16/x45  11011000
/butterfly16/x46  11000100
/butterfly16/x47  00001100
/butterfly16/x48  10101100
/butterfly16/x49  11000100
/butterfly16/x50  11000100
/butterfly16/x51  00001100
/butterfly16/x52  11001000
/butterfly16/x53  11011000
/butterfly16/x54  11000100
/butterfly16/x55  00001100
/butterfly16/x56  10101100
/butterfly16/x57  11000100
/butterfly16/x58  11000100
/butterfly16/x59  00001100
/butterfly16/x60  11001000
/butterfly16/x61  11011000
/butterfly16/x62  11000100
/butterfly16/x63  00001100
Now  1300 ps
Cursor 1  0 ps
```

Figure: 4.3

Figure: 4.1, 4.2 & 4.3 simulation results of the 64-point FFT

5. CONCLUSION

In this project it is shown that a baseband ASIC can be fast and at the same time flexible with a low power consumption. The term fast do not refer to extreme clock frequencies but to the fact that no part of the designs needs more than one clock cycle to process a sample once the pipe is filled. Hence, the design does not need to be clocked any faster than the requested bandwidth and compared to modern CMOS technology this is a low number, in the order of 5-100 MHz. There are several advantages with a low clock frequency, firsit is possible use a low power/low speed cell library with low static leakage current and secondly, it is easier to create a clock tree. In addition, there will be more margins to reduce supply voltage if the required clock frequency is low. To have a low leakage power is important in these designs, since flexibility is achieved with independent modules, where the operation mode decides if a module should be used or not. The unused modules are not clocked and hence only consume static leakage power.

REFERENCES

[1] R. W. Chang, “Synthesis of Band-Limited Orthogonal Signals for Multichannel Data Transmission,” Bell System Tech. J., vol. 45, pp. 1775–1796, Dec. 1966.

[2] ETS 300401, ETSI, “Digital Audio Broadcasting (DAB);DAB to mobile, portable and fixed receivers,” 1995.

[3] ETSI EN 300 744, “Digital video broadcasting (DVB);framing structure, channel coding, and modulation for digital terrestrial television,” 2001.

[4]European IST Project, “Power Aware Communications for Wireless OptiMised personal Area Networks(PACWOMAN),” http://www.imec.be/pacwoman/.

[5] S.B. Weinstein and P. M. Ebert, “Data transmission by frequency division multiplexing using the discrete Fourier transform,” IEEE Transactions on Communications, vol. 19, pp. 628–634, Oct. 1971.

[6] A.Peled and A. Ruiz, “Frequency domain data transmission using reduced computational complexity algorithms,” in Int. Conf. Acoustic, Speech, Signal Processing, Denver, CO, 1980, pp. 964–967.

[7] L.J. Cimini, “Analysis and Simulation of a Digital Mobile Channel Using Orthogonal Frequency Division Multiplexing ,” IEEE Transactions on Communications, vol. 33, pp. 665–675, July 1985.

[8] ETSI TS 101 475, “Broadband Radio Access Networks (BRAN); HIPERLAN Type 2 Physical (PHY) layer, v1.1.1,” 2000, http://portal.etsi.org/bran/.

[9] IEEEstd 802.11a, “High-speed Physical Layer in 5 GHz Band,” 1999, http://ieee802.org/.

[10] IEEEstd 802.11g, “High-speed Physical Layer in 2.4 GHz Band,” 2003, http://ieee802.org/.

[11] J. Bingham, “Multicarrier Modulation for Data Transmission: An Idea Whose Time Has Come,” IEEE Communications Magazine, vol. 8, pp. 5–14, May 1990.

[12] O. Edfors, M. Sandell, J. van de Beek, D. Landström and F. Sjöberg, “An introduction to orthogonal frequency-division multiplexing,” TULEA 1996:16, Div. of Signal Processing, Luleå, Tech. Rep., a University of Technology, Luleå 1996.

[13] J. W. Cooley and J. W. Tukey, “An Algorithm for Machine Calculation of Complex Fourier Series,” Math. Comput., vol. 19, pp. 297–301, Apr. 1965.

[14] R. Grunheid, E. Bolinth, and H. Rohling, “A blockwise loading algorithm for the adaptive modulation technique in OFDM systems,” in Proc. of Vehicular Technology Conference, VTC 2001 Fall, Atlantic City, NJ, USA, Oct. 7-11 2001, pp. 948–951.

[15] J. G. Proakis, Digital Communications. McGraw-Hill, 2001.

[16] M. Russel and G. Stuber,“Interchannel interference analysis of OFDM in a mobile environment,” in Proc. IEEE Vehic. Technol. Conf., vol. 2, Chicago, IL, 1995, pp. 820–824.

[17] N. Petersson, “Peak and power reduction in multicarrier systems,” 2002, licentiate thesis, Lund University, Sweden.

[18] S.Johansson, “ASIC Implementation of an OFDM Synchronization Algorithm,” 2000, Licentiate Thesis, Lund University, Sweden.

[19] R. Morrison, L. J. Cimini, and S. K. Wilson, “On the Use of a Cyclic Extension in OFDM,” in Proc. of Vehicular Technology Conference, VTC 2001 Fall, vol. 2, Atlantic City, NJ, USA, Oct. 7-11 2001, pp. 664–668.

[20] J. Rabaey,A. Chandrakasan, and B. Nikolic, Digital Integrated Circuits, A Design Perspective. Prentice-Hall, 2003.

[21] K. Parhi, VLSI Digital Signal Processing Systems. New York, NY, USA: John Wiley & Sons, 1999.

[22] Transmeta Corporation, “Transmeta Long Run Power Management,” http://www.transmeta.com.

[23] J. G. Proakis and D. G. Manolakis, Digital Signal Processing. Prentice-Hall, 1996.

[24] S. He, “Concurrent VLSI Architecture for DFT Computing and Algorithms for Multi-output Logic Decomposition,” Ph.D. dissertation, Lund University, 1995.

[25] K. Nazifiand G. Hasson, “Industry’s First RTL Power Optimization Feature Significantly Improves Power Compiler’s Quality of Results,” 1998, www.synopsys.com/news/pubs/rsvp/spr98/rsvp spr98 6.html.

[26] W. Li and L. Wanhammar, “A Pipelined FFT Processor,” in IEEE Workshop on Signal Processing Systems, 1999, pp. 654–662.

[27] S. Johansson, S. He, and P. Nilsson, “Wordlength Optimization of a Pipelined FFT Processor,” in Proc. of 42nd Midwest Symposium on Circuits and Systems, Las Cruces, NM, USA, Aug. 8-11 1999.