DSP Implementation Aspects of an OFDM Based Wireless MAN Modem

Behrouz Maham
Department of ECE, University of Tehran, Tehran, Iran
b.maham@ece.ut.ac.ir

Reza Ali-Hemmati
Iran Telecommunication Research Center (ITRC), Tehran, Iran
rhemmati@itrc.ac.ir

Abstract—For future broadband telecommunication systems and networks based on OFDM technology, efficient implementation of an OFDM modem is a major requirement. In this paper, we are going to present some fundamental aspects of implementation of an OFDM based wireless MAN modem on TI Digital Signal Processors (DSP) C6000 family. To this purpose, we model a typical MMDS system based on IEEE802.16a standard, and simulate it in TI Code Composer Studio™ (CCS) environment with C programming language and compare the floating and fixed point implementation aspects of the system. This is useful and enhances rapid prototyping of a DSP based wireless MAN modems. The wireless MAN system will be demonstrated a high speed wireless communication system. One basic component of common OFDM systems is the computation of the DFT and IDFT which was our main focus in this paper via considering some efficient FFT algorithms like the Radix-2 and the Radix-4. In the next part, we compare various methods of implementing of a simple channel estimation and equalization for compensation of the signal affected by frequency selective fading channel model which were employed in our simulation.

Keywords- DSP, OFDM, DFT, channel estimation

I. INTRODUCTION

Orthogonal Frequency Division Multiplexing (OFDM) has received considerable interest for transmission of high bit rate data steams over frequency-selective fading channels, recently. It has been used in wide variety of standards such as Digital Audio Broadcasting (DAB), Digital Video Broadcasting (DVB) and Wireless LAN standards such as HIPERLAN/2 and IEEE802.11a. Fixed Broadband Wireless Access (FBWA) networks such as MMDS based on IEEE802.16a standard is another application of this technique. This technology is designed to provide wireless last-mile broadband access in Metropolitan Area Networks (MAN), delivering performance comparable to traditional cable, DSL or T1 offerings.

The IEEE802.16a Wireless MAN- PHY is based on OFDM modulation as a mandatory technique and designed for NLOS (near line of sight) operation in the 2-11 GHz frequency bands. Many blocks and parameters for physical layer are specified in the standard which we implement and simulate them in the Simulink™. However in our case here, we have assumed non-coded situation which is the coded bits generated randomly and used for mapping and modulation.

We implement this structure for the OFDM transceiver in CCS. We construct our model with TMS320C6000 simulator in two cases of C64x and floating-point C67x.

In this paper at first the OFDM system and parameters are described in section II, then FFT algorithms implemented and simulated in the two mentioned cases are described and analyzed in section III. In section IV channel estimation in OFDM system is described. Some results and observations will come in section V. Finally in section VI the conclusion is drawn with consideration on both performance and precision.

II. OFDM SYSTEM STRUCTURE

The proposed Physical Layer (PHY) OFDM transceiver has been specified in the IEEE802.16a standard supports an extensive range of options. The basic values of these parameters we customize them for our typical MMDS system for licensed bands are shown in Table 1. Figure 1 shows the block diagram of the transmitter. When source bits are coded and interleaved, they are converted from serial to parallel and gray mapped to complex symbols. QPSK, 16-QAM and 64-QAM constellations are specified in the standard; however we choose 16-QAM modulation here. These mapped symbols constitute the 192 data subcarriers of OFDM symbol. After adding 8 pilot subcarriers, DC and some zero (null) subcarriers as guard bands the resulting total 256 subcarrier OFDM symbol is modulated by IFFT to convert symbols to time domain waveform. The equally spaced orthogonal subcarriers with distance of 23.44 KHz leads to useful symbol duration of 42.67 μs. The last process that resides in transmitter’s DSP is adding a copy of the last 64 samples of the useful symbol, termed cyclic prefix (CP), which is for avoiding ISI in the multipath channel, while maintaining orthogonality of the tones. This final process leads to total number of 320 samples per OFDM symbol with duration of 45.7 μs. This is equivalent to sampling frequency of 7MHz at input of the channel.

Due to burst by burst data transmission, we have to add a preamble for channel estimation. In the uplink mode, a short training sequence is used as preamble, but in the downlink mode, a long training which consists of the same short preamble is used. So as a general case we use the short training sequence for channel estimation.

Figure 2 illustrates the most important parts of the OFDM
Table 1: OFDM System Parameters [1]

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Subcarriers</td>
<td>256</td>
</tr>
<tr>
<td>N-used</td>
<td>200</td>
</tr>
<tr>
<td>Channel Bandwidth</td>
<td>6MHz</td>
</tr>
<tr>
<td>fs/BW</td>
<td>7/6</td>
</tr>
<tr>
<td>Symbol Duration $T_s$</td>
<td>45.716 µsec</td>
</tr>
<tr>
<td>Guard Time $T_g$</td>
<td>$\frac{3}{4}$ of $T_s$</td>
</tr>
<tr>
<td>Modulation</td>
<td>16QAM</td>
</tr>
<tr>
<td>Frequency offset indices of guard carriers</td>
<td>-128, -127…, -101, +101, +102…, +127</td>
</tr>
<tr>
<td>Frequency offset indices of Pilot carriers</td>
<td>-84, -60, -36, -12, 12, 36, 60, 84</td>
</tr>
</tbody>
</table>

receiver. Here we assume ideal synchronization. In our study, we simulated our receiver in two platforms. We first consider the receiver which resides on the C64x DSP. After receiving the Q27 scaled signal, the CP is discarded. The remaining OFDM symbol is passed to FFT block to demodulate received waveform. The channel estimation and the Equalizer are expressed in the coming sections. As we use the C64x DSP, fixed-point computation must be used. So output values are scaled. The scaled factor for our transmitter is Q27 ($2^{27}$) due to our FFT algorithm needs.

For wireless MAN network a suitable channel model which is proper for this application should be used to achieve a near-real receiver input signal for simulation and analysis. Here we use the SUI channel models [2] and choose SUI-3 for this purpose. The SUI-3 channel model specifies statistical parameters of small-scale delay spread and fading gains which can be realized by a three non-zero tap FIR filter structure in our model. Since our purpose is to simulate and implement the OFDM transceiver, here we implement the channel in the Simulink model. So, after recording the scaled values of transmitter’s output in a file, we are passing the data through the described multipath channel and AWGN noise added. Then, values are normalized and the distorted signal is again written in the file after scaling up. Also we emphasis on frequency-selectivity property of the channel; that is, it represents the power of the received signal has some deeply fading amplitudes.

III. FFT ALGORITHMS

As we have seen in section 2, the FFT algorithm (Fast Fourier Transform) to compute the DFT (Discrete Fourier Transform) and IDFT is a central part of an OFDM modem. The implementation of a FFT algorithm is very expensive, because it needs many operations. Assuming a total OFDM symbol duration of 36µs, the DFT has to be computed $2.8 \times 10^4$ times per second. Several FFT algorithms are disposed to compute the DFT. Here we discuss only 2 very efficient structures. In [3] and [4], detailed derivation of the Radix-2 and Radix-4 are given. The solution for the FFT will entail three main issues: Achieving the various precision requirements, implementing the algorithms as efficiently as possible and dealing with the inevitable radix reversal that occurs in most FFT algorithms.

$$X[k] = \sum_{i=0}^{N-1} W^{ki} \cdot x[i] \quad ; W = e^{-\frac{2j\pi}{N}}$$  

(1)

In the Fast Fourier Transform “Equation 1” is recursively decomposed into series of passes. For the Radix-2 case, the function is decomposed into two subintervals $2k$ and $2k+1$.

$$X[2k] = \sum_{i=0}^{N/2-1} (x[i] + x[i + \frac{N}{2}]) \cdot W^{ki}$$  

(2)

$$X[2k + 1] = \sum_{i=0}^{N/2-1} (x[i] - x[i + \frac{N}{2}]) \cdot W^{i}W^{ki}$$  

(3)

For the Radix-4 case, the function is decomposed into four subintervals $4k$, $4k+1$, $4k+2$ & $4k+3$

$$X[4k] = \sum_{i=0}^{N/4-1} \left( x[i] + x[i + \frac{N}{4}] + x[i + \frac{N}{2}] + x[i + \frac{3N}{4}] \right) W^{ki}$$  

(4)

$$X[4k + 1] = \sum_{i=0}^{N/4-1} \left( x[i] - x[i + \frac{N}{2}] - x[i + \frac{N}{4}] + x[i + \frac{3N}{4}] \right) W^{i}W^{ki}$$  

(5)

$$X[4k] = \sum_{i=0}^{N/4-1} \left( x[i] + x[i + \frac{N}{4}] + x[i + \frac{N}{2}] + x[i + \frac{3N}{4}] \right) W^{ki}$$  

(6)

$$X[4k + 1] = \left( \sum_{i=0}^{N/4-1} x[i] - x[i + \frac{N}{2}] - x[i + \frac{N}{4}] + x[i + \frac{3N}{4}] \right) W^{i}W^{ki}$$  

(7)

The issue of radix is related to both performance and precision. The Radix-2 and Radix-4 FFT have different performance and precision effects in a fixed-point processor. In Radix-4 transformation, more computation is performed in each stage before the fixed point is returned to its starting position. Truncation occurs roughly only half as often as in the Radix-2 case. However, as more computation is performed in each stage, the chances of numerical overflow are increased over the Radix-2 case. In processing core with high computational ability the Radix-4 will be roughly twice as fast as a Radix-2 transform. Twiddle Factors and bit reversal indices are static and cannot be changed during system operation. So we computed them in advance and set them in the memory.

IV. CHANNEL ESTIMATION

The preamble transmission required by 802.16a, grant to implement a channel equalization algorithm. This sequence is built transmitting 50 complex values, equal to $\pm 1 \pm j$ in both short and long preamble, but latter one contains extra symbol having 100 real values, equal to $\pm 1$. At the receiver you have complex values that represent the effect of the channel. In this
manner one can use these values to calculate appropriate correction coefficients to minimize the fading destructive effects. Because of the extremely high bit rates supported, such algorithm should be very fast and accurate. Trade off between this two constrains allow to use a zero forcing equalization algorithm. The process is composed in two phases: the first one estimates the effect of the channel and evaluates the correction coefficients, the second one equalizes the processed data.

In the estimation phase the complete 201 complex values (number of data subcarriers = 192, number of pilot subcarriers = 8, a null DC carrier), should be constructed from 50 received complex values. So we use interpolation function in the C code to interpolate between these values, before entering corrupted preamble symbols into FFT. The division process is easily computed by shifting values to right because we have $1/(\pm 1 \pm j).2^0$ equals to 1/8 that means to shift values to right by three. The key problem is the evaluation of the division needed in performing $1/H$ (H stands for estimated channel). The fastest and extremely precise algorithm chosen is based upon the Newton’s method in searching function zeros. If you could create a function that has a zero in value we wish, we could use an iterative, and extremely fast converging, algorithm. Such a function could be of the form $f(x) = 1/x-k$, in which k is the value we wish to evaluate the reciprocal. Then we have the iteration formula: $x_{i+1} = 2x_i - kx_i^2$. Reference [5] also examined this algorithm for an OFDM transceiver under IEEE802.11 standard. The rapidity of convergence of the algorithm depends on the choice of the seed $x_0$. This value is provided by a look-up table properly designed. The C67x floating-point DSP has intrinsic function of _rcpsp that computes the floating-point reciprocal with Newton’s method within single-cycle.

Once the estimation coefficients are ready, we can easily equalize the data by means of one complex multiplication per carrier. Let’s suppose that the transmitted data are in the form $(y_i + jy_2)$ and the received corrupted one is $(x_i + jx_2)$. If the channel coefficient is the $(h_1 + jh_2)$, we have $(x_i + jx_2) = (y_i + jy_2) / (h_1 + jh_2)$. The estimation phase gives us the value $1 / (h_1 + jh_2)$ or, better, two values $R = h_1 / (h_1^2 + h_2^2)$ and $I = h_2 / (h_1^2 + h_2^2)$, so that we have to do is:

$$ (y_1 + jy_2) = (x_1.R + x_2.I) + j(x_2.R - x_1.I) \quad (8) $$

V. Results

As a criterion for comparison, we use 32 bits for both data and twiddle factor coefficients to minimize quantization effects and the C64x platform. The results for the signal to quantization noise (SQNR) for some FFT length are shown in Table 2. The pure DFT is computed using double precision floating-point arithmetic and compared with the output from each algorithm. Signal to noise ratio is defined as follows:

$$ \text{SQNR} = 10 \log_{10} \left[ \sum_{n=0}^{N-1} \left( \text{Re}(X[i]) \right)^2 \right]^{1/2} \left[ \sum_{n=0}^{N-1} \left( \text{Im}(X[i]) \right)^2 \right]^{1/2} $$

where X[i] is the fixed-point frequency and $X'$ is the double precision frequency. The results of this quantity are as Table 2. Observe that the Radix-4 FFT outperforms the Radix-2
FFT. The superiorities of the Radix-4 over the raised cosine pulsed-shaped system can be clearly seen, especially in lower performed in each stage the probability of numerical overflow in the Radix-4 is increased over the Radix-2 case. The best way to solve this is to detect if the overflow occurs at the output of each stage, then scale down the result based on how many bits have grown before feeding the result into the next stage. The effect of this process is clear in Table 2, as FFT length becomes larger, the chances of bit growth detection and scaling down are increased. So Radix-4 has a higher SQNR loss as FFT length increases in respect of the Radix-2 case.

<table>
<thead>
<tr>
<th>FFT length</th>
<th>SQNR for Radix2 [dB]</th>
<th>SQNR for Radix4 [dB]</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>53.9</td>
<td>70.0</td>
</tr>
<tr>
<td>16</td>
<td>53.0</td>
<td>67.7</td>
</tr>
<tr>
<td>64</td>
<td>51.9</td>
<td>64.7</td>
</tr>
<tr>
<td>256</td>
<td>51.1</td>
<td>58.8</td>
</tr>
</tbody>
</table>

Table 2: Internal quantization effect

Table 3: FFT performance for two used algorithms. (A: real additions; M: real multiplications)

<table>
<thead>
<tr>
<th>FFT length</th>
<th>A</th>
<th>M</th>
<th>Cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radix-2</td>
<td>256</td>
<td>5634</td>
<td>3576</td>
</tr>
<tr>
<td>Radix-4</td>
<td>256</td>
<td>5008</td>
<td>1824</td>
</tr>
</tbody>
</table>

According to Table 3, it is obvious that with the C64x simulator, we have found performance improvement with Radix-4 for FFT calculation, in terms of clock cycles and number of multiplications. In this case, we observe that Radix-4 has roughly half number of cycles in respect of Radix-2. We could expect this from the fact that the clock cycles are mainly affected by the number of multiplications. Since channel equalization performs a lot of multiplications and divisions, the optimization methods for division calculation such as Newton’s method should be utilized. As we mentioned in section 4, the C67x floating-point DSP has intrinsic function of _rcpsp that computes the floating-point reciprocal in an efficient manner. Therefore in our study, we changed our simulated platform to C67x to compare our equalizer with C64x platform. Observing Table 4, it is clear that even if the C67x receiver FFT length. As we mentioned before, as more computation is using _rcpsp, consumes less clock cycles, C64x has better performance due to increasing clock rate roughly twice as the floating-point C67x.

In this work, we also observe that in addition to limited dynamic range of the fixed-point computation, using of the interpolation between the received preambles in the channel estimation process causes up to 10dB SNR loss in recovered signal due to ignoring high frequency effects of channel.

VI. CONCLUSION

In this paper an OFDM system, compatible to wireless MAN communication systems based on IEEE802.16a, has been analyzed regarding implementation aspects. Since OFDM requires efficient algorithm, two known FFT algorithms have been analyzed. With the C64x we have found a performance improvement, for FFT calculation, in term of number of clock cycles required around 50% with respect to that obtained with Radix-2 in 32 bit format on the C64x platform.

In addition, we compare their quantization effects and solve overflow problem of the Radix-4 by scaling down bit contents, compromising to SQNR loss. Also, here we have considered the implementation issues of the system’s equalizer and compared them in terms of quantization error that depends on fixed-point calculations and the interpolation effect, and performance of the channel estimation on different platforms. Finally we can say that, although the C64x has better performance but due to some advantages of the floating-point computations have been shown, C67x may be used for such purposes, too.

REFERENCES