Analog FFT Interface for Ultra-Low Power Analog Receiver Architectures

Nima Sadeghi
iCORE High Capacity Digital Communication Lab
Hessam M. Nik
Department of Electrical & Computer Engineering
University of Alberta
Edmonton, Alberta Canada T6G 2V4
(nima, hessam, schlegel, vgauDET)ece.ualberta.ca

Abstract—Our project is to design and implement an analog receiver including an analog decoder and a low power analog FFT processor. The FFT can be represented by a graph that is similar to the underlying graph in iterative decoders, and could potentially be implemented using comparable analog circuits with simple structures. Our system uses an OFDM with differential BPSK modulation. We simulated the system, modeled the transistors mismatch and simplified the circuit for the 256-bit FFT. Our goal is eventually to design the FFT processor in CMOS 0.18µm technology in low power subthreshold regime.

I. INTRODUCTION

For ultra-low power applications High Capacity Digital Communication Lab has designed an analog receiver consisting of a (256,121) Turbo Product Decoder [1] and we want to build an FFT interface to extend this project. We propose to use the OFDM transmission format due to its versatility and process simplicity [2]. This requires that the receiver transforms received sampled signals by a Fast Fourier Transform (FFT) to generate the LLR values required by the analog decoder.

A 256-bit FFT designed and simulated in Matlab. In our simulations we considered the transistors mismatch due to the process variations and some circuit simplifications to implement it eventually using CMOS technology. Our figure of merit to design such a low-power analog FFT is not to degrade the decoder performance defined by its SNR. In this work we use the system level design and its mathematical representation. Then we explain the FFT structure and some circuit considerations. At the end we discuss the system performance and the simulation results.

II. SYSTEM OVERVIEW

At the system level we look at a communication system model which consists of a transmitter, communication channel, and receiver as shown in Fig.1. We explain each individual part in this model now and at the end we look at the bit error rate performance results.

At the transmitter binary information bits are encoded by an error correcting code such as a Turbo or LDPC code in our case. A serial-to-parallel data converter gives $M*N$ coded bits to a symbol mapping block to generate $N$ 2$M$-ary symbols. We will concentrate on BPSK, i.e., $M=1$. The mapping is differential to avoid phase recovery. The Inverse Fast Fourier Transform (IFFT) creates an orthogonal frequency division multiplexed (OFDM) transmission signal. The IFFT creates both in-phase and quadrature channels. After RF up-conversion, the complex equivalent baseband signal $S(t)$ with a symbol period [0,T], containing $S_I(t)$ and $S_Q(t)$, is transmitted.

At the receiver, the RF signal is down-converted. The signal is sampled at times $nT_s$ producing $S(nT_s) = S_n$ where $T_s = \frac{T}{N}$. We assume that the timing information is available at this point. After sampling, we can model the channel at the receiver. We add additive white Gaussian noise (AWGN), and consider frequency offsets, $\epsilon e^{j2\pi f_k t}$, multiplying each sample. The $n$ noisy samples are demodulated by an $N$ point FFT processor. For symbol detection, we use a circular differential demodulator in which we multiply adjacent samples to cancel out the unknown common phase offset. The estimated samples are delivered to the error control decoder to recover the original transmitted bits.

A. Mathematical View of the System

1) OFDM Transmitter: The data symbols $d_n$ for the different frequency channels are in general complex, i.e., $d_n = a_n + jb_n$. Using the IFFT generates the I/Q baseband waveform $S_n(t) = S_I(t) + jS_Q(t)$ where
by the RF carrier frequency spacing to generate the different baseband frequency channels.

**2) Receiver Architecture:** We discuss the receiver architecture and the recovery technique.

We rewrite (6) by substituting

\[ S(t) = \text{Re} \left( \sum_{k=0}^{N-1} d_k e^{j\omega_k t} \right) \quad 0 \leq t \leq T \]

and zero elsewhere. To discuss the receiver architecture and the recovery technique in the beginning, we assume that the channel is ideal and we showed that data recovery can be accomplished via an FFT transformation.

3) Phase Offset Consideration: Now consider the receiver samples when there is a phase offset \( \theta \). The new samples \( S_n \) are now given from (7) by

\[ y_k = \hat{d}_k = \frac{1}{N} \sum_{l=0}^{N-1} d_l \sum_{n=0}^{N-1} e^{j2\pi n(l-k)} \]

which is a rotation of the original data, \( a_k \) and \( b_k \), by \( \theta \). Here we assume that the phase offset is constant for all \( n \) samples. If it is not known we may use differential modulation.

Consider the following differential QPSK modulation as example, in which the data are modulated differentially as

\[ d_k = e^{j\Phi_k} = e^{j(\Phi_{k-1}+\Delta\Phi_k)}; \quad \Delta\Phi_k = 0, \frac{\pi}{2}, \frac{3\pi}{2} \]

where \( \Phi_{k-1} \) is a phase reference for symbol \( d_k \), and \( \Delta\Phi_k \) is our original coded information bit, \( c_k \). If we conjugate each incoming complex symbol at the receiver and multiply it by the next symbol, we obtain the original data regardless the phase offset.

\[ r_k = (y_{k-1}^*)y_k = (d_{k-1}^*)e^{-j\theta}d_ke^{j\theta} = e^{-j(\Phi_{k-1})}e^{j(\Phi_{k-1}+\Delta\Phi_k)} = e^{j(\Delta\Phi_k)} = c_k \]

To generate the phase reference for the first symbol \( c_1 \) for differential modulation we use a tail-biting method; we consider \( d_N \) from the previous block is the same as \( d_0 \) for the new block. At circular differential modulator we add the phase of the adjacent coded information bits \( c_k \), using complex multiplication

\[ d_k = c_k \cdot d_{k-1}; \quad 1 \leq k \leq N \]

and at the demodulator we subtract the phase of the adjacent noisy symbols \( y_k \), using complex multiplication

\[ y_{k-1}^* \cdot y_k = c_k; \quad 1 \leq k \leq N \]

After differential demodulation, \( N \) estimated symbols are delivered to the decoder to extract information bits as depicted in Fig.1.

The phase offset is taken care of by using this differential scheme at the cost of \( N=256 \) number of complex multipliers at the receiver front end between the FFT processor and the decoder.
The butterfly diagram of an 8-bit FFT processor is depicted in Fig. 2 [3]. Looking at this diagram, we observe that additions and multiplications by constant complex values, $W_8^1, W_8^2$ and $W_8^3$, are the only two operations required to create an FFT processor. It is well suited for implementation using analog CMOS circuits based on the fact that adding the currents is just tying two wires together and weighting can be achieved by transistor sizing. Therefore the FFT block can be simply implemented in CMOS technology using only current mirrors as basic blocks.

Each input in the butterfly graph is a complex differential value

$$x_k = (x_{ki+} - x_{ki-}) + j(x_{kq+} - x_{kq-})$$  (17)

and the Ws are complex values on the unity circle

$$W_N^k = e^{j2\pi k/N} = \cos \frac{2\pi k}{N} + j\sin \frac{2\pi k}{N} = WF_i + jWF_q$$  (18)

where $N$ is the number of points in the FFT and the $WF_i$ and the $WF_q$ are Weighting Factors, WFs, to operate with the real and imaginary parts of its input signals respectively.

In Fig. 3 we have extended the 8-bit FFT diagram to obtain the 256-bit FFT structure. One can see that the sub-block 4 FFT and 8 FFT in this graph are exactly the same as those depicted in Fig. 2. It is straightforward to follow the algorithm shown in Fig. 3 to generate any larger FFT.

A. Circuit consideration

As mentioned above, the entire 256 FFT processor can be built in CMOS technology using only current mirrors. On Fig. 2 all white dots are current mirrors without scaling, i.e., having the same $W/L$ ratio. The dark crossed circles are complex multipliers which contain 4 current mirrors with different scaling factors to generate complex multiplication of 17 and 18. The black dots are summation nodes realized by tying wires together.

The weighting factors on the last stage of the 256 FFT shown in Fig. 3 contain all other WFs in all other lower stages. Therefore the number of different weighting factors is 128. Due to the circular distribution of WFs on the unity circle and our complex differential signal’s model, we actually do not need to generate all different values of the 256 FFT diagram. We can interchange the real-imaginary and/or positive-negative signals properly to reduce the number of different values for WFs by a factor of four, i.e., 32 different values for all WFs existing on 256 FFT. This idea has been illustrated graphically in Fig. 4.

The basic current mirror circuit [4] is shown in Fig. 5. To use this circuit just for mirroring the scaling factor, $WF$, should be 1. We modeled the mismatch between transistors in each current mirror as an additive white Gaussian random variable.
and we checked how sensitive the FFT is to the different values of the variance of this random variable as a model for common non-ideal technology variations. We explain its impact in the simulation result section.

\[
\begin{align*}
I_n & \downarrow \\
\left[ \frac{W}{L} \right]_1 & \downarrow \\
I_{\text{out}} & = WF \cdot I_n (1 + \varepsilon) \\
\left[ \frac{W}{L} \right]_2 & = WF \cdot \left[ \frac{W}{L} \right]_1
\end{align*}
\]

Fig. 5. Current mirror: basic circuit to implement FFT. \( \varepsilon \) is a white Gaussian random variable to model the transistors mismatch.

IV. SYSTEM PERFORMANCE AND SIMULATIONS

A. System Performance

Considering the BER vs. SNR performance, for large constellations the power of noise is doubled when using differentially coherent detection versus a coherent technique. The power performance of differential QPSK compared to common coherent PSK is about 2.3 \( \text{dB} \) worse at BER of \( 10^{-4} \); however for differential BPSK it is less than 1 \( \text{dB} \) [5].

In a spread spectrum scenario the limiting factor is interference from different users. Therefore by using higher order modulation designers try to get higher spectral efficiency, i.e., the differential QPSK in general is more spectrally efficient than differential BPSK. However in spread spectrum communication with multiuser channel the spectral efficiency of an N-QPSK system is the same as a 2N-BPSK.

The advantage of using differential BPSK in this standard is to gain more power efficiency, as explained above, while the spectral efficiency remains the same by doubling the users compared to differential QPSK modulation. Therefore we have chosen differential BPSK modulation for our system.

B. Simulation results

We simulated the system shown in Fig. 1 first without an encoder/decoder. We used an ideal 256-FFT to get the exact performance of uncoded differential BPSK transmission. We measured the sensitivity of the 256-FFT both to the transistor mismatch and values of weighting factors.

Fig. 6 shows the bit error rate, (BER), performance curve of the system versus signal to noise ratio, \( \text{SNR} = \frac{E_b}{N_0} \). On this plot the star-dot solid line is the ideal 256-FFT. The circle-dot solid line represents the results for considering 5 \% tolerance, i.e. the variance of the random variable is \((0.05)^2\), on every current mirror node in the entire 256-FFT processor. The performance loss due to this mismatch is about 2 \( \text{dB} \) at BER \( 10^{-3} \).

Also in another simulation we used only 3 different values of weighting factor rather than 32, shown in Fig. 4, to simplify the circuit. The results are dashed lines on the plot. The loss is less than 1 \( \text{dB} \). Consequently the FFT is more sensitive to mismatch between the transistors than the exact value of different weighting factors.

As a next step we are now simulating the entire system including encoder/decoder. The decoder decreased the loss of 2 \( \text{dB} \) at BER \( 10^{-3} \) due to mismatch by more than 1 \( \text{dB} \).

Based on the system simulation we have plan to design a 256-FFT circuit operating on subthreshold regime in CMOS 0.18\( \mu \text{m} \) technology using Cadence and eventually fabricate it by the support of the Canadian Microelectronics Corporation.

V. CONCLUSION

The fundamental goal of our project is to move towards system-level integration of analog decoders with other basic communications receiver components, while maintaining the power consumption advantages of analog decoders to make them suitable for use with energy scavenging methods. We modeled the actual 256-FFT circuit, considering mismatch and circuit simplification of weighting factors on our communication system to find the performance of the system. The decoder decreased the performance loss due to the circuit mismatch. Therefore the analog 256-FFT could be used as an input interface for our analog decoder to satisfy the low power consumption constraint.

REFERENCES