CHAPTER 3

MODIFICATION OF RADIX-2 FFT

Orthogonal Frequency Division Multiplexing transmitter and receiver consist of the Inverse Fast Fourier Transform and Fast Fourier Transform blocks. The effective functioning of these blocks is paramount to increasing throughput and reducing latency in both the transmitter and receiver. These blocks perform both multiplexing/ de-multiplexing and modulation/ demodulation, which is the core for any OFDM transceiver. The IFFT block multiplexes low rate data streams that modulate sub carriers which are spaced apart by $\Delta f = \frac{n}{T}$ (where $n$ is an integer and $T$ is the symbol duration), while FFT block de-multiplexes the OFDM signal into several low data rate streams carried by periodically spaced subcarriers. The IFFT and FFT blocks are the central blocks of the any OFDM transceiver.

The IFFT/FFT blocks perform both multiplexing and modulation on a set of inputs and if the inputs arrive serially, all the inputs that constitute the set should arrive before the IFFT/FFT evaluation can commence. This process of conversion of serial to parallel data which is a prerequisite for IFFT and FFT will cause latency. Latency is a measure of time delay experienced in a system and it limits the maximum rate at which information can be transmitted. Whereas throughput can be defined as the quantity of raw material or information processed or communicated in a given period by a system or can be seen as the number of processes completed per unit time. Hence latency is measured as the time that elapses between the arrival of the
first input to the generation of the first output out of this block and it is given in clock cycles or in seconds. Latency can also be measured in terms of the time that elapses between the arrival of the first input and the generation of the last output of the FFT/IFFT block. In real time applications the time that elapses between the arrival of the inputs and generation of outputs in the transmitter/receiver will have to be minimised to increase throughput.

3.1 IFFT/FFT in OFDM Transceivers

The use of FFT/IFFT in an OFDM transmitter and receiver has been explored extensively and the latency of an OFDM transceiver is primarily due to the IFFT/FFT blocks. The baseband OFDM system as shown in Figure 3.1 will involve a serial to parallel convertor which converts a high speed (complex) data into parallel streams of low speed (complex) data, which is fed into the IFFT block. The data which arrives serially can be converted into parallel streams of data after scrambling or interleaving. In case the system used is not a baseband OFDM system after the inputs are scrambled or interleaved, OFDM symbols are generated by using a relevant modulation scheme. This block is called mapper, since it maps a digital sequence into a corresponding modulation symbol, for example QAM symbols as shown earlier in Figure 1.1

![Figure 3.1 Baseband OFDM Transmitter](image-url)
Baseband OFDM receiver shown in Figure 3.2 takes in an OFDM signal through a Low Pass Filter and converted into a digital equivalent by a analog to digital convertor (ADC). The resulting signal is converted into parallel streams of low data rate streams. These streams are fed into the FFT block which demultiplexes and demodulates the signal to provide parallel streams of low data rate signals which are arranged serially to form a high data rate signal. Irrespective of whether a baseband OFDM system or a different OFDM system is used, the incoming high speed data has to be converted into parallel streams of low speed data.

3.2 Latency in conventional OFDM systems

The IFFT/FFT algorithm which is used for the evaluation of the DFT/IDFT needs all the inputs to evaluate every output. Every input contributes to every output and hence all the inputs are needed to evaluate every output. The inputs are from a high data rate serial sequence and hence arrive sequentially. These are then interleaved or scrambled into a new set of low data rate parallel sequences. Even if interleaving or scrambling is not done but a simple serial to parallel conversion takes place as in the case of a baseband OFDM transmitter or receiver, it will introduce latency.
This latency is primarily due to the inherent characteristic of the FFT algorithm which performs in-place DFT. In order to keep the intermediate and final results in place the processing is done in a parallel fashion after all the inputs have arrived. The latency that arises here is due to the time lost to waiting for all the inputs to arrive. In case the size of the FFT is large the latency which is proportional to the product of the reciprocal of the data rate (R) and the size of the FFT (N), that is \( T_L \propto \frac{N}{R} \), can be significant.

The process of FFT calculation starts after the arrival of all the inputs. After this initial waiting period, an N point FFT would need the time to perform a specific number of complex multiplications and complex additions the latter dependant on the FFT algorithm used.

The latency induced by the use of FFT/IFFT functionality can be reduced if time lost in waiting for the arrival of all the inputs can be reduced. This can be achieved by converting the large sized FFT into smaller sized FFTs; thereby the time lost to waiting for all the inputs to arrive is reduced. Using smaller sized FFTs to form large sized FFTs will allow the processing to take place even as the inputs arrive. The smaller sized FFTs will introduce a latency of a smaller order, but as they combine together to form a large sized FFT the latencies will accrue to contribute a larger latency.

If the latency in smaller sized FFTs is reduced it will result in reduced overall latency. When the size of FFTs is small, in order to further reduce latency, evaluation of the FFT has to start as the input arrives. As an input arrives the contribution that outputs receive from the inputs need to be identified. As every input arrives the contribution by them to every one of the outputs are noted. As inputs arrive the contributions due to various inputs are accumulated for every output.
As the inputs arrive serially and consecutively it is preferable that this approach evaluates the contribution that every input makes to every one of the final outputs. This approach is not computationally complex and allows one to one mapping between an input and every output. The popularity of radix-2 Fast Fourier transforms revolves around the fact that the number of complex additions and multiplications is less and the computational complexity is relatively low. In order to formulate this approach that allows one to one mapping between inputs and outputs to start processing as input arrives, radix-2 FFT algorithms is modified for the above mentioned reasons.

The computational complexity of radix 2 FFT is less while also involving less number multiplications. When all the FFT algorithms are compared, the radix-2 FFT/IFFT algorithm involves less number of multiplications along with reduced complexity. Of course there are other FFT algorithms like Split radix FFT which use lesser number of multiplications and additions when compared to radix-2, radix-4 or radix-8. But the computational complexity of the Split radix FFT is relatively much higher that it is not preferred even though the number of multiplications is less.

The key factor in this effort to formulate a new approach, is to take the Radix-2 FFT/IFFT algorithm which is computationally less complex and modify it to obtain the contribution that every input makes to every output. This approach is applied to both a 4 point sequence and 8 point sequence; the reason for the choice of such small length sequences is to exploit the symmetry of the twiddle factors. The Radix-2 FFT algorithm can be Decimation-In-Time or Decimation-In-Frequency but the former is preferred for the subsequent discussions.
3.3 Modified approach for small values of $n$

The Modified (Part-by-part-evaluation-on-arrival - PPEA) approach is based on the premise that the inputs arrive one at a time and their contributions to different outputs are identified and are accumulated as they arrive. Hence this approach is centred on evaluating part by part every input’s contribution to the outputs and accumulating them. But since this would increase the number of multiplications and additions it is preferred to use radix-2 FFT/IFFT algorithms for small values name $N=4$ and $N=8$.

By using this approach the latency involved in waiting for a set of inputs to arrive is reduced, as inputs are processed as soon as they arrive but increases the arithmetic operations. In order to reduce the multiplications FFTs of smaller sizes are chosen and hence the contributions of an input to all the outputs are calculated without much latency.

3.3.1 PPEA approach – 4 point FFT

When radix-2 FFT/IFFT algorithms are used for small values such as mentioned above, the complex values of the twiddle factors are mostly unity, which can be exploited to reduce the number of multiplications. It is often observed that when number of multiplications is reduced, the computational complexity and latency are considerably reduced. When $N = 4$, at the arrival of every input, 2 negations & 4 additions are performed as shown in Figure 3.3
A decision has to be made after obtaining the input’s real and imaginary parts and their negative versions, about which should reach the appropriate output accumulators. This decision is based on the sample count, which is modulo 3. Hence the individual input’s contribution to the outputs will mostly be the input itself or its negative. In 4 point PPEA FFT, no multiplications are involved and hence after the arrival of the last input, it takes only one clock pulse duration to generate the first, in fact all the outputs.

The contributions that every input makes towards every output are accumulated in separate real and imaginary outputs accumulators and hence after the arrival of the last input, only the contributions of the last input are identified. The same is routed and accumulated in the respective output real and imaginary accumulators. That is the reason it takes only one clock duration after the arrival of the last input to generate all the outputs of the 4 point FFT.
3.3.2  PPEA approach – 8 point FFT

When the 8 point PPEA FFT is considered, the samples numbered 0, 2, 4 & 6 undergo 2 negations & 16 additions while samples numbered 1, 3, 5 & 7 undergo 2 negations, 18 additions & 2 multiplications as they pass through the symbol combiner and they are accumulated in the respective real and imaginary accumulators. The last input that arrives is numbered 7 and hence undergoes 2 multiplications. The outputs of the 8 point PPEA FFT are available 1 clock period after the arrival of the last input. Very few inputs will need to be scaled by the factor $1/\sqrt{2}$. Every input’s contribution towards every output is calculated and accumulated. Hence at the arrival of even numbered input sample 2 negations & 16 additions are performed, while at the arrival of odd numbered input sample 4 negations, 20 additions & 2 multiplications are performed on the real and imaginary parts of the input.
3.4 PPEA approach for large values of N

The OFDM transceivers need to perform IFFT/FFT for values much larger than 4 & 8 and hence there arises a need to extend this to large values of N. PPEA for obtaining FFT/IFFT for a given value of N is an approach which involves a method to obtain the contribution of every input that arrives towards every output, routing the contributions to the relevant outputs and accumulating them as real and imaginary components. When a small value of say N = 4 or 8 is considered, as already stated most if not all of the twiddle factors will have unity real or imaginary parts while N is large to obtain the contributions of every input to every output is laborious and time consuming.

The use of PPEA for small values of N results in decrease in latency but if the size of FFT is increased the latency caused by the waiting time for the arrival of a set of inputs will reduce but the number of arithmetic operations will increase manifold hence increasing the latency. When N is a very large the number of arithmetic operations to calculate the individual contribution of the present input to every output is large. The time taken to calculate the contribution and accumulate the contribution for every output is large. The latency exceeds symbol duration given that the twiddle factors are no longer unity but take on a range of values. Hence the FFT for large values of N can be constructed from small sized FFTs.

For a large sized FFT, when N= 4096, ‘divide and conquer’ approach is used and it employs a 4 point PPEA FFT. For performing 4096 point FFT/IFFT the input sequence which will have 4096 symbols, will arrive serially. Since N = 4096 = 4^6, six 4 point PPEA FFT stages are used,
sandwiching a twiddle factor multiplication between 2 successive FFT stages. Even when these successive 4 point FFTs are evaluated, they are overlapped to reduce latency.

In a 4096 point FFT, initially four point FFTs are performed between four inputs that are \(\frac{4096}{4^1}\) input samples/symbols apart. After 3 * \(\left(\frac{4096}{4^1}\right)\) inputs have arrived the 4 outputs of the 4 point FFT is available for twiddle factor multiplication. Since 4 point PPEA FFTs are used, as the last \(\frac{4096}{4^2}\) start to arrive, one by one 4 point FFTs get completed. When the FFTs are completed, accumulated outputs are multiplied by twiddle factors before the next FFT is completed. After the twiddle factor multiplication the result is immediately available for the next 4 point FFT operation which involves samples that are \(\frac{4096}{4^2}\) symbols apart. As PPEA approach is used to perform 4 point FFT every output out of the multiplier can be processed and sent to the corresponding accumulator to perform the next four point FFT. The above mentioned steps are iteratively repeated over six consecutive stages and the pipelined stages are overlapped to reduce latency.

For performing a 512 point FFT, 8 point FFT PPEA is used. When compared with a 4 point PPEA FFT, an 8 point PPEA FFT involves a few multiplications to calculate the contribution that an input makes towards every output. When \(N = 512 = 8^3\), the FFT is performed by using 8 point PPEA FFT, the process is similar to that explained the 4096 point FFT using 4 point PPEA FFT. The input samples/symbols that are \(\frac{512}{8^1}\) apart are used to perform 8 point FFT consecutively and is repeated \(\frac{512}{8^1}\) times. Since PPEA is used, after 7 * \(\left(\frac{512}{8^1}\right)\) inputs have arrived the 8 FFT outputs are available. These
outputs are multiplied by twiddle factors after which they are available for the next stage of 8 point PPEA FFT which are performed between points that are apart by \( \frac{512}{8^2} \). This is followed by a corresponding twiddle factor multiplication and finally by a PPEA FFT between 8 adjacent points.

### 3.5 Performance analysis

A 4 point DFT if calculated using the conventional radix-2 Decimation in time Fast Fourier Transform algorithm will need 2 negations, 4 real multiplications and 8 real additions per input sample while the same if calculated using PPEA FFT will require 8 real additions, 2 negations and no multiplications. This is evident in Figure 3.5, which shows the total number of arithmetic operations that need to be performed by radix 2 FFT and PPEA FFT for N=4.

![Figure 3.5 Arithmetic operations (real) for 4 point FFT using PPEA FFT vs. radix 2 FFT](image)
When the two methods are compared, it appears as though the arithmetic operations do not differ much except in the number of multiplications, but the given values are calculated per input sample. In PPEA the evaluation is undertaken as the inputs arrive and hence to compare the latency performance of it with that of radix 2 FFT the number of operations after the arrival of the last input shown in Figure 3.6 has to be studied. It is found that in PPEA FFT only 8 real additions and 2 negations needs to be performed while the radix 2 FFT needed all the arithmetic operations to be executed after the arrival of the last input.

Similarly when 8 point DFT is calculated using radix 2 FFT and PPEA FFT and compared, it is found that the former needs 3 negations, 9 additions and 6 multiplications per input sample while the latter needs 3 negations, 20 additions and 1 multiplications per input sample.
In the Figure 3.7, it is clear that the total number of negations are 24 & 16, additions 72 & 144 and multiplications are 48 & 32 for radix 2 FFT.
and PPEA FFT for N= 8 respectively. This does not appear to decrease latency but appears to increase as the number of additions is increasing, but as Figure 3.8 shows PPEA FFT is better due to the fact that 2 negations, 20 additions and 2 multiplications only needs to be performed after the arrival of the last input while in the case of radix 2 FFT all the arithmetic operations, that is 24 negations, 72 additions and 48 multiplications need to be performed after the arrival of the last input.

3.5.1 Inference from results

After the arrival of the last input PPEA FFT, the modified radix 2 FFT takes time only for 1 negation, 1 addition & 1 multiplication when N=8 and 1 negation and 1 addition in the case of N=4. This is because the symbol combiner for N=8 has 4 negators, 4 adders and 2 multipliers apart from 16 output accumulators. Also the symbol combiner for N=4 uses 2 negators and 8 output accumulators. Hence in both cases the FFT outputs are available within one clock period after the arrival of the last input.

3.6 Summary

In this chapter the radix 2 FFT algorithm for N = 4 & 8 was considered for modification. The algorithm was modified to evaluate the contribution of every input, as it arrives, towards every output’s real and imaginary parts. When the modification was done, emphasis was on reducing the number of arithmetic operations, especially the number of multiplications. The number of arithmetic operations that need to be performed after the arrival of the last input is reduced thereby reducing latency.