IMPLEMENTATION OF FFT AND IFFT ALGORITHMS IN FPGA

Ilgaz Az1 Suhap Şahin2 Cihan Karakuzu3 Mehmet Ali Çavuşlu4

1,3,4 Department of Electronic and Communication Engineering, Kocaeli University 41040, İzmit, Kocaeli
2 Department of Computer Engineering, Kocaeli University 41040, İzmit, Kocaeli

1e-mail : ehm_Ilgaz@hotmail.com 2e-mail : suhapsahin@kou.edu.tr 3e-mail : cihankk@kou.edu.tr 4e-mail : alicavuslu@yahoo.com

ABSTRACT:
This article explains implementing of Fast Fourier (FFT) and Inverse Fast Fourier Transform algorithms (IFFT) in FPGA. The reason of designing the study on FPGA base is that FPGAs are able to rearrange of logical blocks and moreover, mathematical algorithms can confirm faster by means of parallel data processing. For operating these algorithms, it is used the family of Xilinx Virtex2P xc2vp30fg676-7 FPGA device as a hardware. In programming the hardware and writing codes, VHDL is used. The results show that FFT and IFFT algorithms result in 0.6 µs and 0.72 µs cycle time respectively.

Keywords: FFT and IFFT, FPGA, VHDL

I. INTRODUCTION
Fast Fourier Transform (FFT), using pattern dilution in time and frequency to take patterns from a signal is that a mathematical operations which allow to calculate Discrete Fourier Transform (DFT) quickly.

Inverse Fast Fourier Transform (IFFT), an array which is obtained its results in time and frequency domain allows us to obtain datas which are taken by pattern dilution in time and frequency method. Owing to this conversion, it was passed from results to first datas. Because of this algorithms, the patterns which are taken in time or frequency domain, are converted to time and frequency domain.

The first knowing FFT algorithm was suggested by Gauss in 1805, but the algorithm which is invented by James Cooley & John Tuckey is become the common knowing one [1] and this algorithm made it possible to live enormous developments in DSP field. Because, in directly calculating of DFT, calculation load which is directly proportional with square of patterns number (N) reduced to proportional level with N*logN by means of FFT that based on the principle of pattern dilution in time. By reducing this mathematical load, it allows to process real time implementations such an algorithm that is wanted to implement as hardware [2].

Designers always effort to provide the balance between speed and generality. As multipurpose chips that can realize multiple function but relatively slow, can be produced, it can be produced special purpose chips which can realize a few functions but very speed. A new kind of processor can provide us the speed and also functionality together. The most important feature of FPGAs chips which base on reconfigurable and parallel data processing, is their unattainable speed level that they rised. Because FPGA has configurable logic blocks which act as logic gates. (Figure 1). Logic blocks, at first can be configured to realize A function then this blocks can be reconfigured to realize B function and for another C function this chip can be reconfigured later because of their programmable devices. [3]

In this study, FFT and IFFT algorithms are provided to implement as hardware thus it is composed an alternative source for using this algorithms in field of speed and applicable subjects (DSP, OFDM). In addition, because of using FFT and IFFT algorithms in WiMAX technology. It is aimed at combine this technology and FPGA processor under the same field.

II. THEORICAL INFORMATION
FFT and IFFT algorithm based on a specific mathematical equations array. Certain number of datas that obtained from a signal are replaced in this equations to count DFTs and owing to this equations, processes are counted very fast than normal DFT equations.
Mathematical expression of DFT are given in equations (1) and (2) as follow;

\[ X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn} + W_N^{kn} \sum_{n=0}^{N/2-1} x[n] W_N^{kn} \]  
(1)

\[ X[k] = X_1[k] + W_N^{k/2} X_2[k] , \quad k=0,1,2,\ldots,N-1 \]  
(2)

The \( X_1[k] \) shows \( x_1[n] \) array’s N/2-point-DFT and \( X_2[k] \) shows \( x_2[n] \) array’s N/2-point-DFT. Because of DFT’s specility, \( X_1[k] \) ve \( X_2[k] \) are N/2-periodic. So that the expression in (3) and (4) equations are obtained.

\[ X_1[k + N/2] = X_1[k] \quad (3) \]

\[ X_2[k + N/2] = X_2[k] \quad (4) \]

The relation \( W_N^{k+N/2} = - W_N^k \) in phase factor gives symmetric feature and the relation \( W_N^{k+N} = W_N^k \) gives periodic feature. Utilizing from symmetric and periodic features of phase factor, made it possible for us to calculate DFT very fast. If symmetric and periodic features of phase factor are applied on DFT equations the first N/2 value of \( X[k] \) and the next N/2 value of \( X[k] \) can obtained in equation (5) and in equation (6) respectively.

\[ X[k] = X_1[k] + W_N^{k} X_2[k] , \quad k=0,1,2,\ldots,(N/2)-1 \]  
(5)

\[ X[k + N/2] = X_1[k] - W_N^{k} X_2[k] , \quad k=0,1,2,\ldots,(N/2)-1 \]  
(6)

An N-point \( x[n] \) array is decomposed in odd and even indexes of N/2-point of patterns and this process will be continued till to obtain at last 2-point pattern array. The coefficients of 2-point patterns are replaced in equations (5) and (6) to calculate DFTs and this process will be continued till to obtain N-point pattern array.

Mathematical expression of IFFT are given in equation (7) as follow;

\[ X[n] = \frac{1}{N} X[k] e^{-j2\pi nk/N} \]  
(7)

Owing to IFFT equation above, the data which their FFT results are obtained in time and frequency domain, we can turn back to the first data array which obtainet by pattern dilution in time and frequency method

### III. FFT and IFFT Implementation in FPGA with VHDL

The architecture of FFT and IFFT must be specified with schematic or algorithmic at first step of FPGAs based system design. When FFT and IFFT based FPGAs system design specify the architecture of FFT and IFFT from a symbolic level, this level allows us using VHDL which stands for VHSIC (Very High Speed Integrated Circuit) Hardware Programming Language [5]. VHDL allows many levels of abstractions and permits accurate description of electronic components ranging from simple logic gates to microprocessors. VHDL have tools needed for description and simulation which leads to a lower production cost.

This section presents VHDL code and hardware implementation for the FFT and IFFT previously described.

The FPGA implementations of FFT and IFFT architecture was developed using VHDL with 32 bit floating point arithmetic. Because floating points have greatest amount of dynamic range for any applications. Unfortunately, there is currently no clear support for floating-point arithmetic in VHDL[4]. As a result, VHDL library was designed for using FFT and IFFT algorithm on FPGAs.
The library supports to the IEEE-754 standards for single-precision (32-bit) floating point arithmetic. The variables needed from library are appointed using parameters. Figure 2 is explaining flow chart for how to our code works. The flow chart coded with VHDL in the below. First step, the value we want to calculate appointed the variables (r_giris1, r_giris2, im_giris1, im_giris2). Phase values as global variables are appointed values with IEEE-754 standards. These variables which have sequences are saved in a counter, appointed and stuffed a function respectively. Table 1 describes code as a sample for the appointment.

Table 1: Appointment code from FFT algorithm

```vhdl
elsif sy>25 and sy<29 then
  r_giris1 <= x_et0_re;
  r_giris2 <= x_ece0_re;
  im_giris1 <= x_et0_im;
  im_giris2 <= x_ece0_im;
  aci_re <= bir;
aci_im <= sifir;
if sy=28 then
  x_cift0_re <= fonk_re;
  x_cift0_im <= fonk_im;
end if;
```

The local functions to be spine for the whole algorithm have operations which are designed special task. Local functions supply appointment of global and angle variables are processed immediately and multiple, add and subtract the variables with flow chart using functions have the VHDL library. This kid of local functions have latch variables which are descrit in function.

This variables save operation results to use another part of the function. Angle values multiplied input values which have imaginary and real part. Imaginary input is subtracted to adder which is result of real input and real product result addition. Actually performing process is multiply which has imaginary input and its real result has saved as a intermediate output.
Table 2: Local function code from FFT algorithm

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Virtex2P</th>
<th>xc2vp30fg676-7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Slices</td>
<td>4559 of 13696 %33</td>
<td>3453 of 13696 %25</td>
</tr>
<tr>
<td>Number of Flip-Flops</td>
<td>1908 of 27392 %6</td>
<td>1354 of 27392 %4</td>
</tr>
<tr>
<td>Number of LUTs</td>
<td>8353 of 27392 %30</td>
<td>6370 of 27392 %23</td>
</tr>
<tr>
<td>Number of IOBs</td>
<td>65 of 416 %15</td>
<td>33 of 416 %7</td>
</tr>
<tr>
<td>Number of MULT18X18</td>
<td>8 of 136 %5</td>
<td>8 of 136 %5</td>
</tr>
<tr>
<td>Number of GCLK</td>
<td>1 of 16 %6</td>
<td>1 of 16 %6</td>
</tr>
</tbody>
</table>

IV. CONCLUSION:

The implementations are performed on Xilinx’s Virtex2P FPGAs and VHDL libraries were designed for using FFT and IFFT algorithm on FPGAs. The implementation result show that we need 0.6 µs to obtain FFT implementation result and 0.72µs for the IFFT implementation result. The times of FFT and IFFT implementations, which not exceed 1 ms is fairly important for real time applications. So this result show that FPGA supplies great speed for digital communication algorithms. This study shows that FPGAs are versatile devices for implementing many different applications. As FPGAs allow the hardware design via its configuration on software control, the improvement of circuitry design is just a matter of modifying, debugging and downloading the new configuration code in a short time. FFT and IFFT algorithms require computational density and need a lot of time in real time application, implementations of FPGA is convenient solution.

V. REFERENCES


VI. ABBREVIATION AND SYMBOLS

WiMAX: Worldwide Interoperability Microwave Access
OFDM: Orthogonal Frequency Division Multiplexing
VHDL: Very High Speed Integrated Circuit Description Language
DSP: Digital Signal Processing

\[ W_n^k = e^{-j2\pi k/N} \] : Phase factor
k : Index of pattern in time and frequency domain
N : Number of pattern in time and frequency domain
\( x[n] \) : Array of pattern dilution in time domain
\( X[k] \) : \( x[n] \) array of taken DFT
t : Odd number index of array
\( \varsigma \) : Even number index of array