ISSN: 2278 – 7798
International Journal of Science, Engineering and Technology Research (IJSETR)
Volume 5, Issue 5, May 2016

FPGA Implementation of FFT based Digital Instantaneous Frequency Measurement (DIFM) technique

Chinmaya Mishra, J. Lingaiah, G.Kalyan

***Abstract*— Instantaneous Frequency measurement (IFM) is an important parameter for signal characterization and became very critical process in modern signal intelligence applications. Channelization is a preferred approach for IFM, since it can report the instantaneous frequency of a signal(s), even when signal changing its frequency very rapidly.**

In this project FFT based DIFM modules are implemented targeting for FPGA platform. In the first version 8 point radix-2 pipelined FFT based Digital Channelizer is implemented in VHDL for DIFM applications. The implementation is based on Decimation in Frequency architecture. The FFT core implementation is based on 16 bit Q8 format (8 bit fractional part) sign-magnitude representation. The arithmetic units for implementation of complex number multipliers and adders are implemented in VHDL and tested with appropriate test benches. The top level entity is implemented using structural modeling, and sub-modules are implemented with a combination of structural and behavioral modeling. The fundamental block of FFT processor consists of butterfly processor followed by a complex multiplier (for twiddle factor multiplication). Pipeline stages are introduced at each butterfly stage, to improve the time domain resolution of DIFM.

ModelSim simulator tool is used for functional verification of DIFM core. Xilinx synthesis tool (XST) is used to synthesize the designs. Xilinx Spartan 3E FPGA board is used to test the results using Chipscope tool.

***Keywords*— Xilinx Spartan 3E FPGA, FFT, Modelsim, Chipscope Analyzer.**

I. INTRODUCTION

Digital channelization can be considered as a digital filter bank. An input signal will appear at a certain output according to its frequency. By measuring the outputs from the filter bank, the frequency of the input signal can be determined. The only practical approach to building a wideband digital receiver for signal intelligence applications with today's technology is through channelization. An efficient method of performing channelization is by employing the fast Fourier transform (FFT).

An FFT algorithm uses a divide-and-conquer approach to reduce the computation complexity for calculation of DFT, using symmetry and periodicity properties [4]. The most popular algorithms are Cooley and Tukey radix-r algorithms. In which the transform length N is a power of a basis r (i.e., N = rv) [5]. The two approaches for calculation of FFT are decimation in time (DIT) and decimation in frequency (DIF) [3]. Channelization is used for finding the instantaneous parameters of the signals in signal intelligence applications. This approach allows reporting the instantaneous frequency of a signal, even in the presence of multiple emitters.

Channelization is one of the most important operations in building digital receivers. The equivalent analog operation is the filter bank. Therefore, digital channelization can be considered as a digital filter bank. It can also be considered as an N-port network with one input and *N*-1 outputs. An input signal will appear at a certain output according to its frequency. By measuring the outputs from the filter bank, the frequency of the input signal can be determined. The only practical approach to building a wideband digital receiver for signal intelligence applications with today's technology is through channelization. An efficient method of performing channelization is by employing the fast Fourier transform (FFT).

To build a signal intelligence receiver using FFT, the length and the overlap of the FFT are very important parameters. These parameters are related to the minimum pulse width and the frequency resolution, which determines the sensitivity of the receiver. The frequency information can be obtained from the outputs of the digital filters. In order to obtain the input frequency, the filter outputs must be further processed. The main objectives of a receiver are to determine the number of input signals and their frequencies. The circuit used to accomplish these goals is referred to as the encoder. The encoding circuit is the most difficult subsystem to design a receiver.

Most research effort is spent on the encoder design. This is true for both digital and analog receivers. The main problems are to avoid the generation of false signals and the detection of weak signals. In an analog filter bank, the shape of the filter is difficult to control, and it is difficult to build filters with uniform performance such as the bandwidth and the ripple factor; therefore, the encoder must accommodate this problem. The shape of each individual filter in a digital filter bank can be better controlled. As a result, the encoder should be slightly easier to design because it does not need to compensate for the filter differences.

In this paper an architecture addressing above challenges is proposed and same is verified on Spartan 3E FPGA board. The remaining part of this paper is divided into 5 sections. The section II has implementation of FFT based DIFM. Section III has VHDL simulation results. The section IV presents the hardware verification of FFT based DIFM. The last section concludes the work.

II. IMPLEMENTATION

This section describes the implementation wise aspects of the FFT based DIFM architecture.

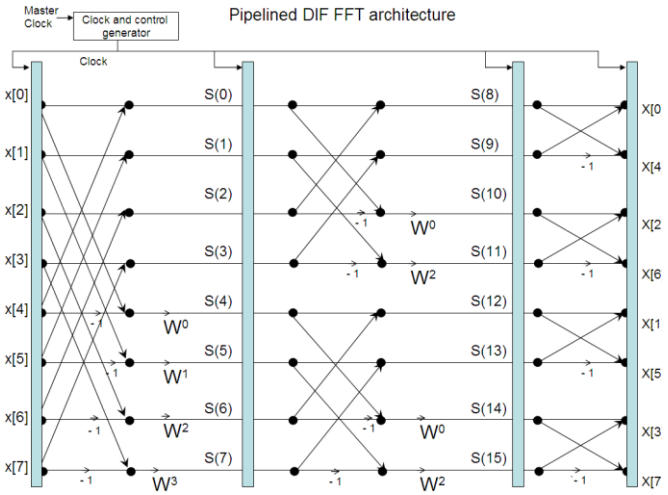

Fig.1 Pipelined DIF FFT architecture

The architecture presented above is implemented in VHDL [6]. The basic blocks binary adders, binary multipliers, butterfly and twiddle factor multiplier modules are used for implementing the above architecture. The first and second stages consist of four butterflies followed by twiddle factor multipliers. The third stage consists of four butterflies. In between each stage, at the input and at the output pipeline stage is used to increase the throughput.

The straightforward approach for building a filter bank is to build individual filters, each one with a specific center frequency and bandwidth. Each digital filter can be either a finite impulse response (FIR) or infinite impulse response (IIR) type.

Theoretically, each filter can be designed independently with a different bandwidth or shape. In this arrangement, if the input data are real (as opposed to being complex) the output data are also real. The output is obtained through convolving the input signal *x(n)* and the impulse response of the filter *h(n)*. One of the disadvantages of this approach is that the operation of the filter bank is computationally complex.

It is desirable to build a receiver with uniform frequency resolution; that is, the filters have the same shape and bandwidth. It is easier to build such a filter bank through FFT techniques than by using individual filter design because there is less computation. In the previous section, it is stated that the outputs of a filter bank can be obtained from convolution and also from the FFT operation [1]. In this section the similarity between the FFT and convolution operation will be illustrated. In the FFT operation, one set of data in the time domain can be used to find one set of data in the frequency domain. In order to process the input data in a continuous manner, the FFT must also operate continuously [2].

The first input applied is given below (notice that only real input is given, imaginary part is zero).

X = [1 2 3 4 5 6 7 8]

The FFT is performed for the above input and results are listed in the below Table. The outputs observed are in sign-magnitude Q8 format (1 – sign bit, 7 – integer bits, 8 – fractional bits). The following table shows output values in HEX and corresponding Decimal values in the above mentioned format.

FFT index	Output (in HEX)		Corresponding real value (assuming Q8 sign-mag format)
	Real	Imag	
0	2400	0000	36 + j 0
1	8400	09A0	- 4 + j 9.625
2	8400	0400	- 4 + j 4
3	8400	01A0	- 4 + j 1.625
4	8400	0000	- 4 + j 0
5	8400	81A0	- 4 – j 1.625
6	8400	8400	- 4 – j 4
7	8400	89A0	- 4 – j 9.625

Table.1 FFT computed output observations

The above results can be verified using MATLAB. The same input which is applied in VHDL implementation is given to MATLAB for validation. The MATLAB produced FFT results for the same input are same as results observed in simulation.

FFT computation in MATLAB:

First command is the input to FFT. The applied input is:

>> X= [1 2 3 4 5 6 7 8]

Second command is FFT computation for above input. FFT is applied with only real input. Imaginary inputs are applied as zeros.

>> fft(X)

```
>> X=[ 1 2 3 4 5 6 7 8]

X =

   1   2   3   4   5   6   7   8

>> fft(X)

ans =

  Columns 1 through 4

  36.0000          -4.0000 + 9.6569i  -4.0000 + 4.0000i  -4.0000 + 1.6569i

  Columns 5 through 8

  -4.0000          -4.0000 - 1.6569i  -4.0000 - 4.0000i  -4.0000 - 9.6569i
```

Fig.2 FFT results provided by MATLAB

The above figure is the screen shot of MATLAB command window showing the output. It can be observed that the results obtained from MATLAB command window and Modelsim are same. Hence the FFT core functionality is verified. In the same way, FFT results are observed for complex inputs also. DIFM, Digital Instantaneous Frequency Measurement is calculated from the computed magnitude. The highest magnitude peak and the corresponding index value are observed. The maximum peak index value is used to compute frequency of the input signal. The magnitude computation values for an example can be observed in the simulation results.

III. SIMULATION RESULTS

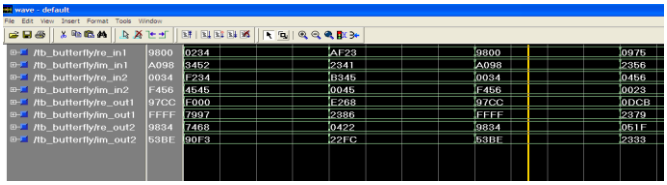

Fig.3 simulation results of Butterfly multiplier

The above figure shows simulation results of butterfly multiplier. Signals *re_in1*, *re_in2*, *im_in1* and *im_in2* are the inputs to the butterfly multiplier. The signals re_out1, *re_out2*, *im_out1* and *im_out2* are the outputs of the butterfly multiplier.

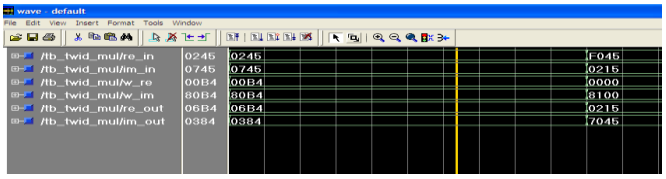

Fig.4 simulation results of twiddle factor multiplier

The above figure shows simulation results of twiddle factor multiplier. Signals *re_in*, *im_in*, *w_re* and *w_im* are the inputs to the twiddle factor multiplier. The signals *re_out* and *im_out* are the outputs of the twiddle factor multiplier.

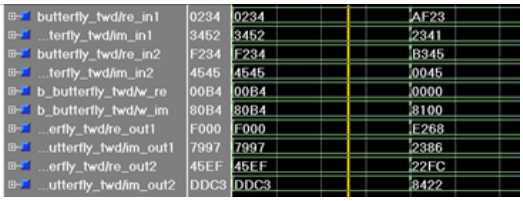

Fig.5 simulation results of butterfly followed by Twiddle factor multiplier

The above figure shows simulation results of butterfly followed by Twiddle factor multiplier.

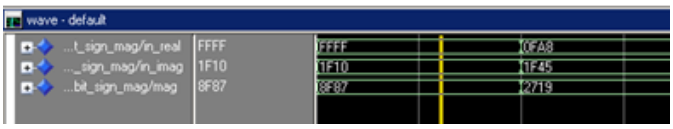

Fig.6 simulation results of Magnitude Computation

Magnitude computation simulation results are shown in the above figure. The signals *in_real* and *in_imag* are the inputs of the magnitude computation. The computed magnitude is available at the signal mag.

In the present section VHDL simulation of the implemented FFT based DIFM are shown.

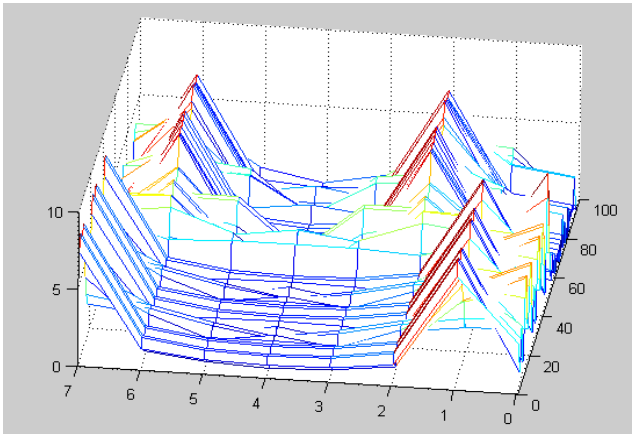

Fig.7 DIFM output MATLAB results

IV. HARDWARE VERIFICATION

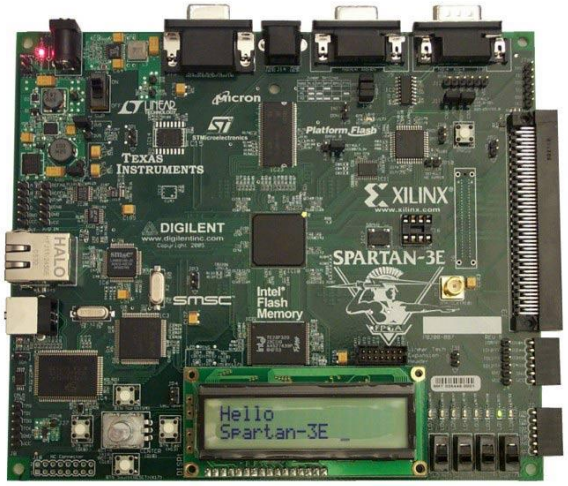

Fig.8 Spartan 3E FPGA board

The above figure shows FPGA hardware which is used for hardware validation of FFT based DIFM. The hardware Xilinx FPPGA Spartan 3E, Device type is XC3S500E, Package is FG320.

Device utilization summary:

Selected Device: 3s500efg320-5

Number of Slices:	4594	out of	4656	98%
Number of Slice Flip Flops:	3013	out of	9312	32%
Number of 4 input LUTs:	8156	out of	9312	87%

Number of bonded IOBs:	66	out of	232	28%
Number of GCLKs:	8	out of	24	33%

Timing Summary:

Speed Grade: -5

Minimum period:	52.472ns
Maximum Frequency:	19.058MHz
Minimum input arrival time before clock:	2.055ns
Maximum output required time after clock:	4.105ns

V. CONCLUSION

In this paper, a Fast Fourier Transform based Digital Instantaneous Frequency Measurement was implemented. First, Fast Fourier Transform is performed and Magnitude computation is done. The index value where highest magnitude is detected is noted and used for frequency calculation. The simulation results are validated with help of Modelsim 6.2c and MATLAB. The concept of FFT based DIFM is developed on hardware reconfigurable device Xilinx Spartan 3E.

REFERENCES

[1] Bergland, G. D. "A Guided Tour of the Fast Fourier Transform." IEEE Spectrum 6, 41-52, July 1969.

[2] Ahmed Saeed, M. Elbably, G. Abdelfadeel, and M. I. Eladawy, "Efficient FPGA implementation of FFT/IFFT Processor", INTERNATIONAL JOURNAL OF CIRCUITS, SYSTEMS AND SIGNAL PROCESSING, Issue 3, Volume 3, 2009

[3] Alan V. Oppenheim, Ronald W. Schafer with John R. Buck, Discrete Time Signal Processing, Second Edition

[4] B. Parhami, Computer Arithmetic, Algorithms and Hardware Designs, 1999

[5] James W. Cooley and John W. Tukey, An Algorithm for the Machine Calculation of Complex Fourier Series

[6] Peter J. Ashenden, The Designer's Guide to VHDL, Second Edition.

[7] Ali Saidi ," DECIMATION-IN-TIME-FREQUENCY FFT ALGORITHM" Motorola Applied Research, Paging and Wireless Data Group Boynton Beach.

Authors:

Chinmaya Mishra, pursuing M. Tech in VLSI System Design at Arjun College of Technology and Sciences, Batasingaram village, Hayathnagar, R.R. Dist, Hyderabad, Telangana, India.

J. LINGAIAH, presently working as Head of Department & Associate Professor in the department of Electronics and Communication Engineering at Arjun College of Technology and Sciences, Batasingaram Village, Hayathnagar, R.R. Dist, Telangana, India.

GUDIPALLI KALYAN, working as an Application Engineer at Unistring Tech Solutions Pvt. Ltd, Hyderabad, Telangana, India.