

# FPGA Implementation of 64 Point FFT Processor

Vasantha Sudheer N, Venu Gopal B

**Abstract-** A power efficient Fast Fourier Transform (FFT) processor for use in the Direction of Arrival (DOA) estimation of a wideband waveform is presented. The target device for implementation is a Xilinx Spartan-3 Xc3s200 Field Programmable Gate Array (FPGA). The FFT processor was developed using the Xilinx ISE in Verilog code. Although the parallel and pipelined architecture uses a large portion of the available FPGA resources, the architecture does yield a high throughput. The total power consumed by the design is 0.081W.

**Keywords-** Direction of Arrival (DOA), Fast fourier transform (FFT),FPGA.

## I. INTRODUCTION

Active and passive radar systems are two commonly used radar systems in different military and non military applications respectively. Active radar systems are commonly used in military and commercial applications for the detection and tracking of objects through a medium such as air. Most active radar systems work by transmitting a signal pulse through the medium scatters off objects in the medium. by processing the received wave field, an active radar system can determine an object's distance, velocity, and other features. The passive detection of objects has become of particular interest to the military in scenarios where the medium needs to be monitored covertly. A passive radar system can consist solely of an array of receiving antennas as opposed to an active radar system with a co-located transmitter and receiver. Without a transmitter the passive radar system relies on other sources of electromagnetic waves, such as am or fm radiowaves, tv broadcast, nearby radar, cell towers, or a wideband waveform to "illuminate" objects in the medium [1]. An active radar system has the benefit of a known transmit waveform. A passive radar system does not have knowledge of the transmitted signal and thus has to rely on digital signal processing techniques to extract information from an array of sensors over a period of time.

Wideband sources can have different bandwidths between 500Mhz to 5 GHz. The bandwidth of the signal in this paper is considered to be 3 GHz. The wideband passive radar system is shown below in which the object is illuminated by the wideband source while the scattered waveform along with the reference waveform is received by an array of antennas, digitized, and processed.

A well established algorithm such as the Multiple Signal Classification (MUSIC) algorithm [4] can be applied to calculate Direction of Arrival(DOA). The best method of determining the Direction of Arrival of wideband signal is to first decompose the signal into narrowband frequency components by converting the time domain signal into frequency domain through Discrete Fourier Transform,

which will produce further frequency components which can be processed independently. A computationally efficient version of the DFT is known as the Fast Fourier Transform (FFT) and is commonly used increase the computational speed and reduces the computational time.

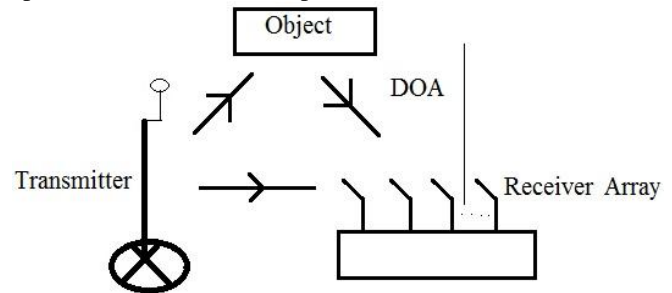


Fig:1 Radar System.

The coherent signal subspace method is the process under which the wideband DOA algorithm separated the frequency components via the Fast fourier transform. A low resolution estimate of the DOA is calculated using any beamforming algorithm. The low resolution estimate is used to identify areas where a high resolution DOA estimate is necessary. The calculation of the high resolution CCSM DOA estimate uses a set of 4096 ( $64^2$ ) samples from each antenna. Each set of 4096 samples is divided into 64 segments of 64 samples and the 64-point FFT is computed for each segment. The covariance matrix for each frequency component is obtained by collecting the corresponding frequency components for each of the antennas in the array and estimating the covariance over the 64 segment record length. These covariance matrices are then combined into a single focusing matrix. After the focusing matrix is formed, a high resolution DOA can be estimated using the narrowband MUSIC algorithm. The CSSM algorithm is summarized in Figure 2 for a general array of n antennas. For this example, a phase shift and scale block was chosen for beamforming in the frequency domain, however other beamforming methods apply.

Using the CSSM algorithm to estimate the DOA of a signal with a bandwidth of 4 GHz will require high speed digital signal processing. Per Nyquist's sampling theorem, the array's ADCs must sample at a frequency of at least 8 GHz. In order to process the data continuously, a throughput of 8 giga-samples per second must be maintained throughout the system.

In this paper the design and simulation of a 64-point FFT processor capable of 6 GS/s throughput. This FFT processor will be a building block for the CSSM wideband DOA estimation algorithm.

Revised Manuscript Received on September, 2012.

Vasantha Sudheer N, M.Tech ECE Department,JNTU Kakinada University/ Kaushik College of Engineering/Visakhapatnam, India.

Venu Gopal B, Assoc .Professor, Dept. of ECE, Kaushik College of Engineering /visakhapatnam, India.

II. FFT DERIVATION

The DFT for N samples is defined as

$$X(K) = \sum_{n=0}^{N-1} x(n) W_N^{nK} \tag{1}$$

where  $k=0,1,2,\dots,N-1$ , and  $X(k)$  is the frequency domain representation of  $x[n]$ . The twiddle factor  $W$

$N$  is defined as:

$$W_N = e^{-j2\pi K/N} \tag{2}$$

The FFT is a set of algorithms which are more computationally efficient than the DFT [6]. Table 1 summarizes the computational requirements of several FFT algorithms as derived in [7]. The radix-4 algorithm was selected based on the reduced number of operations compared to the radix-2 algorithm. Other algorithms such as the radix-8 and split-radix algorithms offer even more reductions; however, for a length 64 FFT these reductions are not significant. The Winograd algorithm minimizes the number of multiplications, but also adds a degree of complexity and significantly increases the total number of operations.

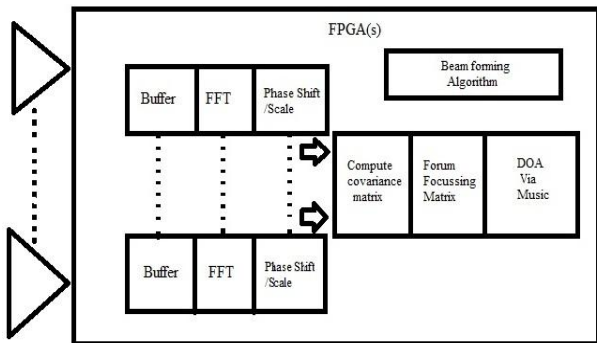


Fig 2: CSSM Algorithm

Table 1: Operations required for 64-point FFT

Operation	Radix-2	Radix-4	Radix-8	Split Radix	Winograd
Real Additions	1032	976	972	964	1394
Real Multiples	264	208	204	196	198
Total	1296	1174	1176	1160	1592

From [2], the expression for the radix-4 FFT is

$$X(p,q) = \sum_{l=0}^3 [W_N^{lq} F(l,q)] W_4^{lp} \text{ for } p = 0,1,2,3 \tag{3}$$

Where

$$F(l,q) = \sum_{m=0}^{N/4-1} x(l,m) W_{N/4}^{mq} \tag{4}$$

For  $l=0,1,2,3$  and  $q=0,1,\dots,N/4-1$  and

$$x(l,m) = x(4m+l) \tag{5}$$

$$X(p,q) = X(N/4p + q) \tag{6}$$

Note that (4) is a  $N/4$  length DFT. Equation (3) in matrix form, as shown in [2], is

$$\begin{matrix} X(0,q) & 1 & 1 & 1 & 1 & W_N^0 F(0,q) \\ X(1,q) & 1 & -j & -1 & j & W_N^q F(1,q) \\ X(2,q) & 1 & -1 & 1 & -1 & W_N^{2q} F(2,q) \\ X(3,q) & 1 & j & -1 & j & W_N^{3q} F(3,q) \end{matrix} \tag{7}$$

for  $q=0,1,\dots,N/4-1$ .

Equation (7) can be expressed in signal flow form as shown in Figure 3. This form reveals the radix-4 butterfly which is the basic building block of the FFT processor.

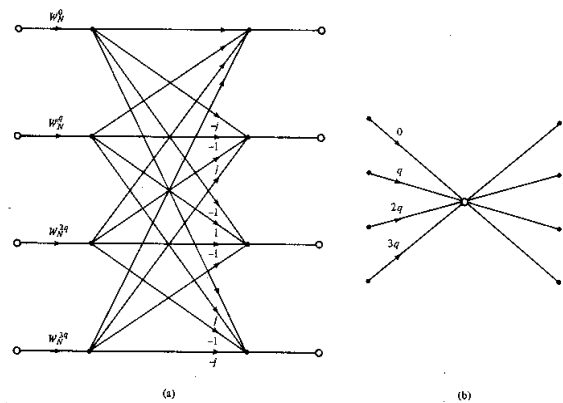


Fig 3: Radix 4 butterfly

Equation (7) can be further decomposed by taking each of the  $F(l,q)$  DFTs and applying the same process of splitting the DFT into four DFTs. The result will be a network of radix-4 butterflies.

III. FFT ARCHITECTURE

A literature survey was conducted to find a suitable architecture for implementing the radix-4 FFT algorithm. Many FFT architectures exist, ranging from power efficient architectures [9-12] to area efficient architectures [13-14]. There are also pipelined [11-12,14-16] architectures. These architectures were not optimal for a high throughput solution since they did not utilize parallel processing. In [17], a parallel FFT architecture was developed for FPGA. A quad-pipeline architecture was implemented using a front-end 4-point FFT butterfly circuit to separate the sequence into groups of four. These groups form four pipelines which are processed independently in parallel. The authors mention the possibility of extending the concept to “k” groups of parallel pipelines, but only implement the case  $k=4$ . In [18], a high speed FFT was implemented on a distributed array of custom designed DSPs. Each of the processors in the array was assigned a specific function, for example, in the case of a eight-processor 64-point complex FFT, three DSPs are reserved for memory operations, three are reserved for FFT butterfly operations, and two are reserved for shuffling the input and output sequences. This architecture would not be efficiently mapped to a single FPGA. Over the course of the survey, it was determined that the architecture for maximum throughput is the fully pipelined fully parallel architecture. In the literature, several authors mention this method [17] but this architecture is rarely implemented due to the large computational requirements. For the 64-point FFT, this corresponds to 3 ranks of 16 radix-4 butterfly processors in parallel. This architecture will use the maximum amount of resources, but will yield the highest throughput. Because of the parallel pipelined architecture each one of the 208 multiplications will require a dedicated multiplier circuit on the FPGA.

For the input / output (I/O) interface, multiplexers and de-multiplexers can be used to interface the FPGA with outside resources.

Figure 4 shows a graphical description of an example I/O interface. The example uses two clock frequencies inside the FPGA, 500 MHz for the I/O interface, and 125 MHz for the actual FFT processor. A throughput of 8 GS/s is maintained across the entire system.

#### IV. IMPLEMENTATION

The FFT processor was designed in System Generator, which is a model-based design tool using the MATLAB Simulink environment. System Generator is a bit-true, cycle-true simulation and design verification program [13].

The FFT processor was implemented using the radix-4 butterfly as the basic component. After the initial radix-4 butterfly network was laid out, various hand-optimizations were made. Initially, the FFT processor used 264 multiply operations. However, certain twiddle factor multiplications resulted in simplified logic, reducing the number of multiplications to 208. This concurs with Table 1. The device for implementation was the Xilinx Spartan 3. With clock frequencies up to 550 MHz and up to 1200 I/O pins, the Spartan 3 is ideal for high-throughput applications. With regards to high speed digital signal processing, the Spartan 3 contains up to 640 dedicated arithmetic units known as DSP48E slices.

#### V. RESULTS

Two FFT processors were developed, one with 8-bit samples, and another with 10-bit samples from the DC. Based on ISE's timing reports, both the 8-bit and 10-bit FFT processors will meet the timing constraints and yield throughputs of 8.03 GS/s and 8.05 GS/s respectively. However, the proposed I/O interface did not meet timing. The 8-bit I/O interface achieved a throughput of 4.7 GS/s while the 10-bit I/O interface achieved a throughput of 4.1 GS/s. The slow path consists primarily of routing delays which account for 87% of the total delay. These routing delay problems may be mitigated though an intelligent selection of pins for the input and output resources, or adding delay to the input/output paths. With regards to FPGA utilization, the following table summarizes the results for the 10-bit FFT processor. The chosen Spartan 3 platform for implementation was the xc3s200-4ft256.

Table 2: Device Utilization Summary.

Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	2,063	9,312	22%
Number of 4 input LUTs	3,073	9,312	33%
Number of occupied Slices	2,155	4,656	46%
Number of Slices containing only related logic	2,155	2,155	100%
Number of Slices containing unrelated logic	0	2,155	0%
Total Number of 4 input LUTs	3,096	9,312	33%
Number used as logic	3,005		
Number used as a route-thru	23		
Number used as Shift registers	68		
Number of bonded IOBs	87	190	45%

Number of RAMB16s	4	20	20%
Number of BUFGMUXs	1	24	4%
Number of MULT18X18SIOs	4	20	20%

#### Power Utilization Summary:

Device	
Family	Spartan3e
Part	xc3s500e
Package	ft256
Grade	Commercial
Process	Typical
Speed Grade	-4
Environment	
Ambient Temp (C)	25.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	0
Characterization	
PRODUCTION v1.2.06-23-09	

On-Chip	Power (W)	Used	Available	Utilization (%)
Clocks	0.000	1	---	---
Logic	0.000	3083	9312	33.1
Signals	0.000	4703	---	---
IOs	0.000	87	190	45.8
BRAMs	0.000	4	20	20.0
MULTs	0.000	4	20	20.0
Leakage	0.081			
Total	0.081			

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	31.1	82.5	27.5

Supply Summary		Total	Dynamic	Quiescent
Source	Voltage	Current (A)	Current (A)	Current (A)
Vccint	1.200	0.026	0.000	0.026
Vccaux	2.500	0.018	0.000	0.018
Vcco25	2.500	0.002	0.000	0.002

Supply Power (W)	Total	Dynamic	Quiescent
	0.081	0.000	0.081

#### VI. CONCLUSIONS

A highly parallel and pipelined architecture for a FPGA-based, high speed, FFT processor for wideband DOA applications has been presented. This 64-point FFT was simulated in System Generator and then implemented on a Spartan 3 FPGA using ISE. The FFT processor uses much of the FPGA's resources but yields a high throughput. The designed FFT processor met the throughput constraint of 8GS/s and higher throughput rates may be possible. The FPGA I/O interface could only process 4.2 GS/s due to an insufficient I/O interface. Future work would include designing an I/O interface which meets the throughput requirements for the CSSM DOA algorithm



REFERENCES

- [1] M.A. Ringer, G.J. Frazer and S.J. Anderson “Waveform Analysis of Transmitters of Opportunity for Passive Radar” DSTO Electronics and Surveillance Research Laboratory, 1999
- [2] Pollard, Robert. “How Passive Radar Sensors can Support Air Traffic Control” BAE Systems, Advanced Technology Centre, 2006
- [3] Hsio-Feng, Chiang,”Waveform Generation for Ultra-Wideband Radar System”, Naval Postgraduate School, Montrey, California, 1993
- [4] R. O. Schmidt, “Multiple emitter location and signal parameter estimation” IEEE Transactions on Antennas and Propagation, vol. AP-34, No. 3, pp. 276-280, March 1986.
- [4] Hong, Wang, “Multiple-target direction finding”, Graduate School of University of Minnesota, PhD Thesis, July 1985.
- [5] Proakis, J.G. “Digital Signal Processing: Principles, Algorithms, and Applications”, Prentice Hall Inc., 1996.
- [6] J.W. Cooley and J.W. Tukey, “An Algorithm for the Machine Computation of the Complex Fourier Series,” Math. Computation, Vol. 19, pp. 297-301, April 1965.
- [7] Chi-hau Chen, “Signal Processing Handbook”, CRC Press, 1988.
- [8] Chidambaram, Ramesh. “A Scalable and High-Performance FFT Processor, Optimized for UWB-OFDM”. July 2005. Technische Universiteit Delft, Department of Electrical Engineering
- [9] T. Pitkanen, T. Partanen, J. Takala. “Low-Power Twiddle Factor Unit for FFT Computation” SAMOS 2007
- [10] C. Chang, C. Wang “Efficient VLSI Architectures for Fast Computation of the Discrete Fourier Transform and Its Inverse. IEEE Transactions on Signal Proc essing, vol. 48 No. 11 Nov 2000
- [11] M. Hasan, T. Arslan, J.S. Thompson “A Novel Coefficient based Lower Power Pipelined Radix-4 FFT Processor for Wireless LAN Applications”
- [12] M. Hasan, T. Arslan and J.S. Thompson, “A Novel Coefficient Ordering based Low Power Pipelined Radix-4 FFT Processor for Wireless LAN Applications”, IEEE Transactions on Consumer Electronics, Vol. 49, No. 1 February 2003.
- [13] www.xilinx.com
- [14] C. Wang and Y. Lin, “An Efficient FFT Processor for DAB Receiver Using Circuit-Sharing Pipeline Design”, IEEE Transactions on Broadcasting, Vol. 53, No. 3, September 2007
- [15] J. J. Fuster and K. S. Gugel, “Pipelined 64-Point Fast Fourier Transform For Programmable Logic Devices”, Dept. of Electrical and Computer Engineering, University of Florida, Gainesville, FL
- [16] Dillon Engineering, www.dilloneng.com.
- [17] J. Palmer and B. Nelson, “A Parallel FFT Architecture for FPGAs”, Dept. of Electrical and Computer Engineering, Brigham Young University, Provo, UT 84602, FPL 2004, LNCS 3203, pp. 948-953, 2004.
- [18] Omar Sattari, “Fast Fourier Transforms on a Distributed Digital Signal Processor”, University of California-Davis, 2004.
- [19] J. Takala and K. Punkka, “Scalable FFT Processors and Pipelined Butterfly Units”, Tampere University of Technology, Tampere, Finland, SAMOS 2004, LNCS 3133, pp. 373-382, 2004.
- [20] W. Li and L.Wanhammar , “Efficient Radix-4 and Radix-8 Butterfly.