# Application Specific Integrated Circuit Implementation of Discrete Fractional Fourier Transform

K. K. Sharma and Abdul Samad

*Abstract*—**Over the last two decades, discrete fractional Fourier transform (DFrFT), which is a generalization of the discrete Fourier transform (DFT), has attracted a lot of attention of the researchers. Several methods for computation of the DFrFT have been discussed in the literature but very little effort has been made towards realizing a dedicated hardware for its real time computation. In this paper we propose an application specific integrated circuit (ASIC) design for 16-point DFrFT computation of a 1D signal, which has been synthesized using Very High Speed Integrated Circuit hardware description language (VHDL). The same design can be easily extended for any number of signal samples. The simulation results of the proposed ASIC are very close to the theoretical DFrFT results obtained using the MATLAB software.**

*Index Terms*—**Discrete fractional fourier transform, ASIC, VHDL, eigen vector decomposition.**

## I. Introduction

The fractional Fourier transform (FrFT) [1]−[3], which is a generalization of the conventional Fourier transform, has become a powerful tool for many signal processing applications over the last two decades [4]−[5].

Many discrete versions of the FrFT, i.e., discrete fractional Fourier transform (DFrFT), have also been proposed in the literature, but very little effort has been made towards realizing a dedicated hardware for its real time computation for 1D signals [6]−[7].

Reference [8] shows FPGA implementation of 4−point DFrFT computation using verilog hardware description language (HDL). As far as known to the authors, no application specific integrated circuit (ASIC) based design is yet available for real time implementation of the DFrFT. ASIC rules out FPGA in terms of speed. As ASICs are designed for a specific application, they can be optimized for that specific application. Hence, we can have high speed of operation in ASIC designs.

In this paper we propose an ASIC design for 16−point DFrFT computation which has been synthesized using Very High Speed Integrated Circuit hardware description language (VHDL). The definition of DFrFT used in this paper for ASIC implementation is based on eigen vector decomposition based definition presented in [9]. The tools used to design the complete ASIC of the proposed architecture are ModelSim from Modeltech Inc., Design Compiler & Formality from Synopsys Inc. and SoC encounter & virtuoso from Cadence Inc. The same design can be easily extended for computation of the DFrFT for any number of signal samples. The simulation results of the proposed ASIC are also presented. The results are in close agreement with the theoretical DFrFT results obtained using the MATLAB software.

Rest of the paper is organized as follows. In section II we present a brief review of the DFrFT. In section III we present the proposed methodology for computation of the DFrFT. The ASIC implementation of the proposed design is presented in section IV and the simulation results are presented in section V. The paper is concluded in section VI.

Manuscript received November 28, 2012; revised January 29, 2013.

The authors are with the Department of Electronics and Communication, National Institute of Technology, Jaipur, India (e-mail: kksharma_mrec@yahoo.com, abdulsamad.jaipur@gmail.com).

## II. Review of DFrFT

The $N$ point DFrFT matrix with angle parameter $\alpha$ is defined in terms of the discrete Hermite-Gaussian functions as given by (1) [8].

$$F^{\alpha}[m,n] = \sum_{k=0}^{(N-1)or\ N} u_k[m] e^{-j\alpha k} u_k^T[n], \tag{1}$$

where $k \neq N$ for $N$ odd and $k \neq N-1$ for $N$ even, $U$ is a matrix consisting of eigenvectors of DFT (also called as discrete Hermite-Gaussian functions) arranged in columns as given by (2) [8].

$$U = \begin{pmatrix} u_0[1] & u_1[1]\ldots & u_{N-2}[1] & u_M[1] \\ u_0[2] & u_1[2] & u_{N-2}[2] & u_M[2] \\ \vdots & \vdots & \vdots & \vdots \\ u_0[N] & u_1[N] & u_{N-2}[N] & u_M[N] \end{pmatrix}, \tag{2}$$

where $M = N-1$ for $N$ odd and $M = N$ for $N$ even. The matrix $E$ is a diagonal matrix which contains the eigenvalues $e^{-j0\alpha}, e^{-j1\alpha}, e^{-j2\alpha}, \ldots, e^{-j(N-2)\alpha}, e^{-jM\alpha}$ of DFrFT matrix $F^a$ as diagonal elements. Moreover, the superscript $T$ in (1) stands for transposition operation of a matrix. Reference [9] shows that as the DFT matrix commute with matrix $S$, given below in (3), its eigenvectors will be same as that of matrix $S$.

$$S = \begin{bmatrix} -2 & 1 & 0 & \cdots & 0 & 1 \\ 1 & 2\cos(\frac{2\pi}{N})-4 & 1 & \cdots & 0 & 0 \\ 0 & 1 & (2\cos(\frac{2\pi}{N}2)-4) & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 0 & 0 & \cdots & 1 & 2\cos(\frac{2\pi}{N}(N-1))-4 \end{bmatrix} \tag{3}$$

As we can see matrix $S$ is a tridigonal matrix except for

corner elements and its eigenvectors will give us entries of columns of matrix $U$ .

The $N$ _point DFrFT with angle $\alpha$ of the input signal $f[n]$ , denoted by $f^{\alpha}[n]$ , can be calculated by

$$f^{\alpha}[n] = F^{\alpha} f[n] \tag{4}$$

Or equivalently,

$$f^{\alpha}_{N\times 1} = U_{N\times N} \times (E_{N\times N} \times U^{T}{}_{N\times N} \times f_{N\times 1})) \tag{5}$$

where $X$ indicates matrix multiplication operation. For the proposed architecture the matrix $E$ is replaced with a column matrix $C$ that contains the eigenvalues of DFrFT for given input angle $\alpha$ and middle matrix multiplication is replaced by an array multiplication. The resulting expression is

$$f^{\alpha}_{N\times 1} = U_{N\times N} \times (E_{N\times N} \times U^{T}{}_{N\times N} \times f_{N\times 1})) \tag{6}$$

where $X$ indicates the array multiplication operation. If we define two vector matrices A and B such that,

$$\left.\begin{aligned} A_{N\times 1} &= U^{T}_{N\times N} \times f_{N\times 1} \\ B_{N\times 1} &= C_{N\times 1} \times A_{N\times 1} \end{aligned}\right\} \tag{7}$$

Then (6) becomes,

$$f^{\alpha}_{N\times 1} = U_{N\times N} \times B_{N\times 1} \tag{8}$$

Thus using (6) and (8), one can compute the DFrFT of the signal.

## III. Proposed Methodology of DFrFT Computation

In this section we present the proposed methodology for computation of DFrFT based on (6). The entire computation is divided in three levels as shown in process flow diagram in Fig.1 where $f_{N\times 1}$ represents the input signal vector whose DFrFT is to be computed. In the level I computation, the signal vector $f_{N\times 1}$ will be multiplied by matrix $U^{T}$ and the result of matrix multiplication is stored into $N\times 1$ matrix $A$ . An array multiplication between matrix $A$ and eigenvalue matrix $C$ is carried out in level II and it's result is stored into a matrix $B_{N\times 1}$ . In the level III computation the matrix $B$ is multiplied (through matrix multiplication) with eigen vector matrix $U$ and the final DFrFT result is stored into $F^{\alpha}_{N\times 1}$ .

### A. Level I Implementation

To implement the required computation of this level, the $N$ columns of matrix $U^{T}$ are stored in $N$ number of ROMs as shown in Fig. 2. First each element of the input vector $f_{N\times 1}$ , which is stored in a RAM, is multiplied with the corresponding entries of each ROM and the result of such multiplications are added in the respective accumulators $A_1$ to $A_N$ , associated with each ROM.

### B. Level II Implementation

To implement the computation of this level, for a specific parameter of the DFrFT $\alpha$ , the array multiplication of the eigenvalues and the results of level I, stored in accumulators $A_1$ to $A_N$ , is performed.

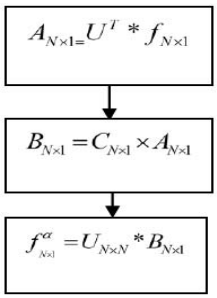


Fig. 1. Process flow.

It may be seen from Fig. 3 that eigenvalues are stored in two separate ROMs storing the values of cosine and sine for angle $\alpha$ . After the array multiplication of $A$ and $E$ the result is stored in a matrix $B$ .

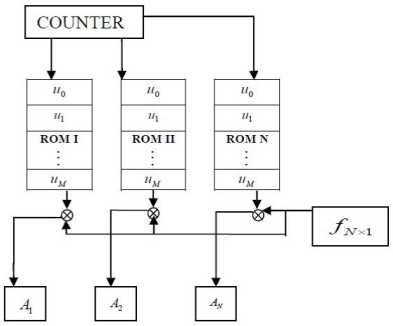


Fig. 2. Level I Implementation

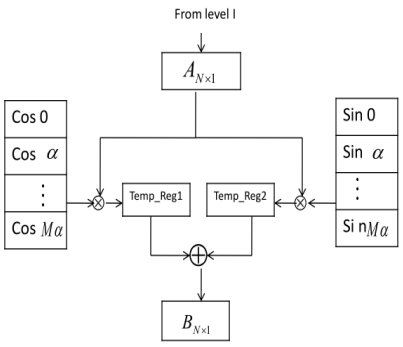


Fig. 3. Level II Implementation

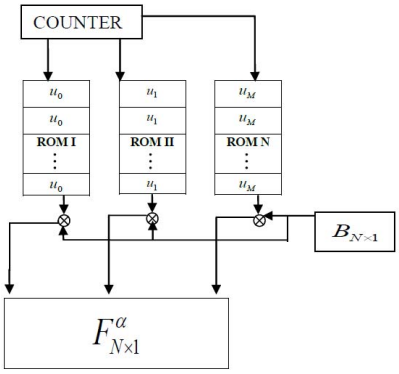


Fig. 4. Level III Implementation

### C. Level III Implementation

To implement the computation of this level the eigenvector matrix $U_{N\times N}$ is multiplied with the matrix $B_{N\times 1}$ calculated in level III. The columns of matrix $U$ are stored in $N$ number of ROMs. The data stored in the ROMs is fetched and multiplied with the values stored in matrix $B$ as shown in Fig. 4.

## IV. ASIC Implementation of Design

After doing the datapath design described above, we follow the steps shown in Fig 5 to develop the ASIC design as given in [10].

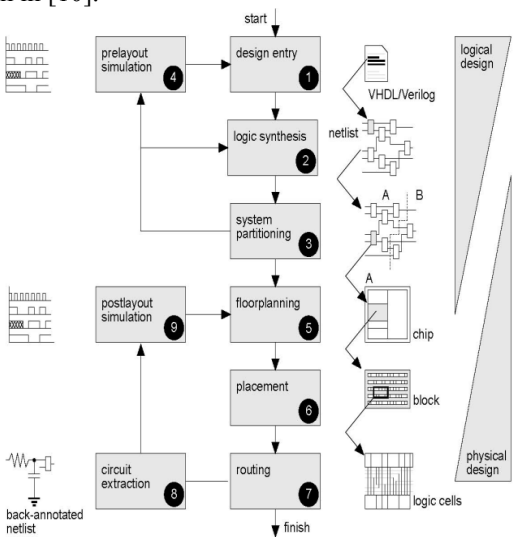


Fig. 5. ASIC design flow

In the initial phase the design is realized primarily as a technology-independent Hardware Description Language (HDL), a format very similar to a programming language, to describe the design‘s functionality. The design is realized as a technology-dependent netlist that consists of a series of instances of circuits from the ASIC vendor’s library, interconnected in a manner to implement the functionality described in the previous view.

In the second phase the design is realized as a physical view, in which the logic circuits described in the previous view are physically placed on a piece of silicon, called a die, and interconnected by various layers of wiring.

Detailed design flow of an ASIC is shown in Fig.5 [10]. In the initial design phase (HDL design capture) we construct resistor–transfer level (RTL) description from the behavioural description of the specified design described in level I. The RTL design is then functionally verified with verification vectors. After the successful verification of RTL description we convert it into logic level description in the second phase HDL design synthesis.

We perform logic optimization step to remove any redundant logic in the design. In logic to technology step, we use the synthesized tool to implement the data structure of logic in the form of gates by using the cells provided in the technology library. This representation is known as netlist. The process of building a netlist and its partitioning is automatically carried out in synthesis tool. After performing Timing/area optimization, we functionally verify the netlist. In functional verification identical stimulus is run with the original RTL and the synthesized gate level description of the design. The output is compared to find any mismatches. In timing verification gate-level netlist is checked for timing by use of timing simulation or by static timing verifier.

Floor planning, placing and routing are efficiently done by the SoC encounter tool. The Graphic Database System (GDS) file is obtained after floorplanning, placing and routing which is then imported into a tool for developing its physical layout. Layout verification and its implementation are also carried out by the available tools. In the proposed ASIC design we have certain limitations that the layout verification step has not completed, but it can also be done. Once layout verification is over, we can implement it.

Table I describes the various tools used for the ASIC implementation.

TABLE I: Tools Used for the Simulations

| | |
|---|---|
| Logic Simulation | ModelSim(modeltech) |
| Logicverification | Formality (Synopsys) |
| Logic Synthesis | Design Compiler (Synopsys) |
| Physical Design | SoC Encounter (Synopsys) |
| Layout | Virtuoso(Cadence) |

## V. Simulation Results

In this section we present the simulation results of the proposed hardware for the case $N=16$.

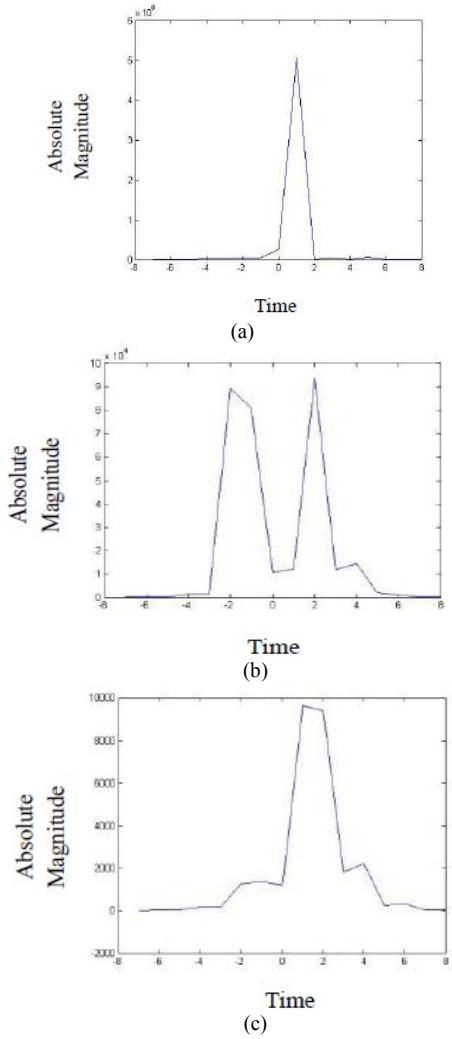


Fig. 6. Plot between the absolute value of DFrFT of the signal & time for angle parameters (a) $\alpha=\pi/6$ (b) $\alpha=\pi/4$ (c) $\alpha=\pi/2$

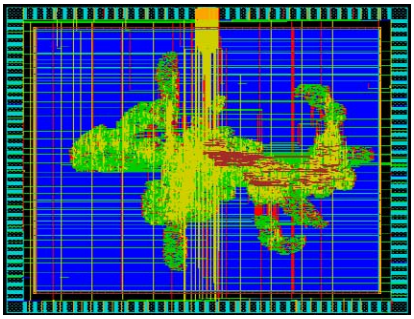

Fig. 7. ASIC Design of DFrFT

TABLE II: PARAMETERS OBTAINED AFTER LOGIC SYNTHESIS OF PROPOSED DESIGN

| AREA | POWER | TIME |
|---|---|---|
| Number of ports 267 | Global Operating Voltage =1.8v | data arrival time 0.21ns |
| Number of nets 523 | Dynamic Power units = 1mW | |
| Number of cells 2 | Leakage Power Units = 1pW | |
| Number of references 2 | Cell Internal Power = 138.4598 mW | |
| Combinational Area 1487649.0 | Net Switching Power = 114.721 mW | |
| Non combinational Area 86007.703125 | Total Dynamic Power = 253.181 mW | |
| Total Cell Area 1573527.50 | Cell Leakage Powe = 510.258 nW | |

In other words, there will be 16 complex numbers or real numbers in signal $f_{N\times 1}$. The proposed architecture discussed in the previous section is designed using VHDL. The design has been simulated using ModelSim simulator with random input samples $f[n]$ as a test vector for specific values of the angle parameter $\alpha$ of the DFrFT equal to $\pi/2$, $\pi/4$ and $\pi/6$. As we can see from Fig.6(c) that when $\alpha$ equals to $\pi/2$, the output is approximately a sinc pulse because at this angle DFrFT reduces to DFT (Discrete Fourier Transform). The area covered by the cells in the chip, the time delay in computation and the overall power consumed by the components are also calculated and shown in the TABLE II below.The area is given in *library units*. Library unit is some measurable unit (for example μm², number of equivalent AND-gates, or something else) that the library vendor has decided to use. This unit is library specific unit and needs to be verified from the library documentation if not known. In the UMC library we are using the unit for area is μm². The data arrival time is the time required for signal to travel from path start point to a path end point. Table II provides the summary of the results after logic synthesis of the design. Final ASIC design is given in Fig. 7.

## VI. CONCLUSION

In this paper we propose an ASIC design for 16-point DFrFT computation of a 1D signal, which has been synthesized using Very High Speed Integrated Circuit hardware description language (VHDL). The same design can be easily extended for any number of signal samples. The simulation results of the proposed ASIC are very close to the theoretical DFrFT results obtained using the MATLAB software. The implementation results shows that the proposed design is suitable for most of the signal and image processing applications**.**

Since the computation is based on 16-points DFrFT the results diverge from exact results of DFrFT , but the results for large number of points would be in precision to the results of actual DFrFT.

## REFERENCES


[1] L. B. Almeida, "The fractional Fourier transform and time-frequency representation," *IEEE Trans. Signal Processing*, vol. 42, pp. 3084–3091, Nov. 1994.

[2] A. C. McBride and F. H. Kerr, "On Namia's fractional Fourier transforms," *IMA J. Appl. Math*., vol. 39, pp. 159–175, August 1987.

[3] H. M. Ozaktas, N. Erkaya, and M. A. Kutay, "Effect of fractional Fourier transformation on time-frequency distributions belonging to the Cohen class," *IEEE Signal Processing Lett*., vol. 3, pp. 40–41, Feb. 1996.

[4] V. Namias, "The fractional order Fourier transform and its application to quantum mechanics*," J. Inst. Math. Applicat*., vol. 25, pp. 241–265, 1980.

[5] H. M. Ozaktas, B. Barshan, D. Mendlovic, and L. Onural, "Convolution, filtering and multiplexing in fractional Fourier domains and their relation to chirp and wavelet transform," *J. Opt. Soc. Am. A*, vol. 11, pp. 547-559, Februray 1994.

[6] S. C. Pei, M. H. Yeh, and C. C. Tseng, "Discrete fractional Fourier transform based on orthogonal projections," *IEEE Trans. Signal Process.*, vol. 54, pp. 3815-3828, May 1999.

[7] S. C. Pei, W. L. Hsue, and J. J. Ding, "Discrete fractional Fourier transform based on new nearly tridiagonal commuting matrices," *IEEE Trans. Sig. Process*., vol. 54, pp. 3815-3828, Oct. 2006.

[8] M. V. N. V. Prasad , K. C. Ray, and A. S. Dhar, "FPGA implementation of Discrete Fractional Fourier Transform," in *Proc. IEEE Int. Conf. Signal Processing and Communcation (IEEE-SPCOM),* Bangalore, 2010, pp. 1-5.

[9] C. Candan, M. A. Kutay, and H. M. Ozaktas, "The Discrete Fractional Fourier Transform," *IEEE Trans. Sig. Process*., vol. 48, pp.1329-1337, May 2000.

[10] M. J. S. Smith, *Application Specific Integrated Circuits*, Boston, Toronto: Addison-Wesley, 1997, pp. 30.



**K. K. Sharma** is an Associate Professor in the Department of Electronics and Communication, National Institute of Technology, Jaipur, INDIA. He has received Ph. D. from IIT Delhi, INDIA.



**Abdul Samad** is an Assistant Professor in the Department of Electronics and Communication, GLA University Mathura, U.P., INDIA. He has received the M.Tech degree in VLSI Design from Malviya National Institute of Technology, Jaipur, INDIA.