Θ_I-SELECTION MODULE IMPLEMENTATION USING ADAPTIVE CORDIC FOR FFTARCHITECTURES

[1]SumathiRampuri *Student of M.techVLSI Department ofECE,Research Center for VLSI and EmbeddedSystems,SreeVidhyanikethan Engineering College*Tirupathi, India rsumathi53@gmail.com

[2]M Bharathi *Assistant Professor, Department of ECE,ResearchCenter for VLSIand EmbeddedSystems,SreeVidhyanikethan EngineeringCollege*Tirupathi,India bharathi891@gmail.com

***Abstract*—This electronic document is a "live" template and already defines The overview of this paper is to design a micro rotation selection block for TW based FFT. This design is mainly based on Adaptive CORDIC algorithm. CORDIC stands for COordinate Rotation DIgital Computer. Two ways to implement the CORDIC algorithm are 1.Vector mode and 2. Rotation mode. The main difference between Vector and rotation are, vector mode is used to compute an angles at given point where as Rotation mode is used to compute the sine and cosine terms at given point. This paper mainly deals with vector method by using micro rotation for complex multiplications in FFT architectures. Ɵi-selection (Ɵi- sel) block for CORDIC based Twiddle Factor (TW) architecture is a computation approach which does an angle selection. CORDIC algorithm is easy to implement trigonometric, hyperbolic and exponential functions based on micro rotation for VLSI Signal processing. The simulation results are examined using Xilinx ISE 14.5Tool.**

Keywords—Fast Fourier Transform (FFT), Adaptive CORDIC Algorithm, Ɵi- selection (Ɵi- sel).

I. INTRODUCTION

In digital signal processing the Fast Fourier Transform (FFT)andInverseFastFourierTransform(IFFT)areplaying an important role. The fast computation method of Discrete Fourier Transform (DFT) algorithm is Fast Fourier Transform(FFT).

In 1965 [1], Cooley and Tukey was first introduced the FFT signal flow graph. So that, FFT became most utilized circuits in many of signal processing and communication applications such as CDMA [5], OFDM [6], WiMAX [2], MIMO [4], WLAN [7], and 3GPP-LTE [3]. In addition to this FFT requires a computational technique in other applicationsliketheimageprocessingapplicationofFourier- Domain Optical Coherence Tomography (FD-OCT) [9], Synthetic Aperture Radar (SAR) [10], and the multimedia application of Digital Video Broadcasting-Terrestrial or DigitalvideoBroadcasting-Handheld(DVB-T/DVB-H)[8].

FFT architectures are of two types. They are Fixed-point and Floating-point architecture. The problem with fixed point architecture is Fixed-point Arithmetic Error (FAE) [11]. In systems fixed point or floating point FFT implementations are used for high speed with low resource cost or High Precision with wide data range.

In Twiddle Factor (TF) implementation, there are two categories. Namely, Look-Up Table (LUT) [14]-[17] and Direct computation based on COordinate Rotation DIgital Computer (CORDIC) algorithm [12],[13],[18]. Where the LUT 's uses a floating point multiplication and it stores thetrigonometric constants where as the CORDIC is used to make a direct computation.

The advantages of LUT based design is high precision, low latency, and reasonable cost. But the problem with LUT based technique suites for only a limited number of FFT- point than a large number of FFT-points. Here by increasing the number of FFT-points the number of stored elements in LUT also increases. Therefore, there is a memory requirement problem. This problem can be overcome by many other techniques such as binary tree decomposition algorithm [21], memory access optimization [22], memory minimization method [19], and Read-After-Write latency optimization[20].Inthesetechniquesalsothereisatrade-off between memory reduction, performance of accuracy, and the throughputrate.

For large-point FFT systems and Application Specific Integrated Chip (ASIC) implementation, a direct computation method is selected i.e., CORDIC. CORDIC algorithm provides a memory free solution for Twiddle Factor (TF). But the main disadvantage of CORDIC algorithm is high latency because of more number of iterations. To overcome this disadvantagewe can move on with Adaptive CORDIC (ACor) method and it was introduced by Y. H. Hu et al. in 1993 [23]. The improvements in the method was introduced by Hong Thu Nguyen et al. in 2015 [24]. The main idea of the ACor method is to reduce the number of iterations and giving the equivalent or better accuracy performance. By reducing the numberofiterations,theAcormethodgivesalow-resources, low-latency, andhigh-precision.

II. BASIC CORDICTECHNIQUES

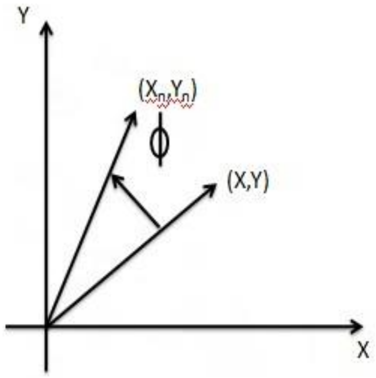
Lets us consider, the two vectors [X,Y] of the same magnitude. And the second vector is obtained by rotatingthe first vector with an angle of(∅).

Now we can calculate Vn = [Xn,Yn] based on the input vector and the rotational angle (∅). The required equations are given below:

$$Xn = X0 \cos(\emptyset) - Y0 \sin(\emptyset)$$

$$Yn = Y0 \cos(\emptyset) + X0 \sin(\emptyset)$$

(1) Where ∅ is a rotationangle.



which indicates the direction of next rotation is used to reduce the angular error. The difference equation is based on logic as

Zi+1 = Zi – di .tan(∅i) (6)

Where Zi (i = 0, 1, …….., N-1). Zi indicates the remaining rotation before performing the rotation by ∅i.And di is the direction of angle of rotation, which indicates the direction of rotationis

$$d_i = \begin{cases} -1, & \text{if } z_i < 0 \\ +1, & \text{otherwise} \end{cases} \quad (7)$$

Fig. 1: Rotation of vector [X,Y] by ∅ degrees.

To concatenate angles ∅, (i=0,1,2, ..., N-1) andtoNow the equation (5), can be represented as below: Xi+1 = cos (∅i) (Xi – di . Yi . 2-i)

evaluate the angle ∅ = $\sum_{i=1}^{n} \theta_i$. The input and output to theYi+1 = cos (∅i) (Yi + di .Xi . 2-i)

Zi+1 = Zi – di.2-i (8)

rotation by $\emptyset_i$with [Xi,Yi] and [Xi+1, Yi+1].

Xi+1 = Xi cos (∅i) – Yi sin (∅i)

Yi+1 = Yi cos (∅i) + Xisin(∅i) (2)

In the above equation, common term cos (∅i), and the equations can be rewritten as,

Xi+1 = cos (∅i) (Xi – Yi sin (∅i)/ cos (∅i)) Yi+1 = cos (∅i) (Yi + Xi sin (∅i)/ cos (∅i))(3)

B. Gain Compensation

To concatenate several rotations, it requires lots of multiplications because equation (5) contains cos(∅i) term. To solve this problem we have to consider that

∅i = arctan (2-i), i € {0, 1, 2,…….}

And

We know that, sin/cos = tan, so we can place tan in the above equation as given below:

Xi+1 = cos (∅i) (Xi – Yi tan(∅i))

Yi+1 = cos (∅i) (Yi + Xitan(∅i)) (4) Wheretan(∅i))=±2-i,(i=0,1,2,…….,N-1).

Under this condition, the above equation becomescos(∅i) = cos (arctan (2-i)) = $\frac{1}{\sqrt{1+2^{-2i}}}$

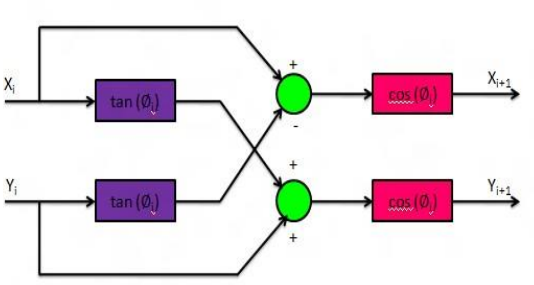
because, we know that

$$\cos(x) = \pm \frac{1}{\sqrt{1+\tan^2 x}}$$

Xi+1 = cos (Øi) (Xi – Yi . 2-i)

Yi+1 = cos (Øi) (Yi + Xi.2-i)

which gives, $1 + 2^{-2i}$



(5)Now, to divide equation (8) by cos(Øi) =

Xi+1 . ai = Xi – di .Yi .2-i Yi+1 .

ai = Yi + di .Xi .2-i

Where ai = 1+2-2i. At the right side the computation is performed as shift-and-addition parts. So the amplified gain ai is obtained xi+1 and yi+1.

The gain is compensate by multiply a constant Ai on both sides. Let Ai+1 =ai Ai. The recursive equations is obtainedas

Fig. 2: Organizing computations of the transform. For some specific angle Øi, multiplication by tan (Øi) can be replaced by an arithmetic shift and some sign multiplication.

A. Determining Rotation Directions

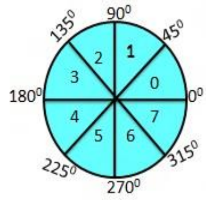
Define For the composite rotation of an angle Ø is uniquely defined by the elementary rotation direction is sequence (d0, d1, ………, dN-1).

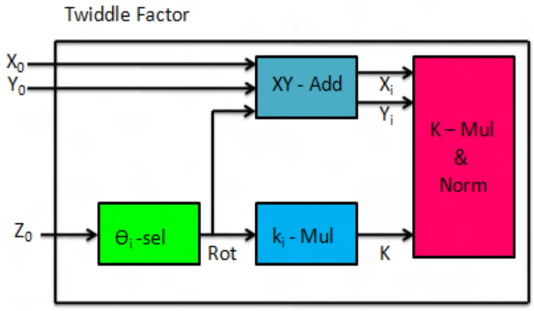
An angle accumulator is used to determine the sequence. Theaccumulatoraccumulatestheelementaryrotationangles,Xi+1 . Ai_+1 = XiAi – di .YiAi . 2-i Yi+1 . Ai_+1 = YiAi – di .YiAi . 2-i

By equating, the above steps in N iterations. The Gain can be calculated by using the equation given below.

$$AN=\prod_{i=0}^{N-1} \sqrt{1 + 2^{-2i}}$$

In this method, the scale factors are considered for computations during iterations. So that, the process is complicate and occupies more memory. The whole process will be stopped, when the Z value is larger thanthe threshold value.





segment	correction
0	$\sin\Theta_0$ $\cos\Theta_0$
1	$\sin\Theta_1 = \cos\Theta_0$ $\cos\Theta_1 = \sin\Theta_0$
2	$\sin\Theta_2 = \cos\Theta_0$ $\cos\Theta_2 = -\sin\Theta_0$
3	$\sin\Theta_3 = \sin\Theta_0$ $\cos\Theta_3 = -\cos\Theta_0$

segment	correction
4	$\sin\Theta_4 = -\sin\Theta_0$ $\cos\Theta_4 = -\cos\Theta_0$
5	$\sin\Theta_5 = -\cos\Theta_0$ $\cos\Theta_5 = -\sin\Theta_0$
6	$\sin\Theta_6 = -\cos\Theta_0$ $\cos\Theta_6 = \sin\Theta_0$
7	$\sin\Theta_7 = -\sin\Theta_0$ $\cos\Theta_7 = \cos\Theta_0$

Fig. 3: 8-segment normalized angles for the ACor method.

The input angle has range of [00 to 450] for the applied ACor algorithm is noted. If the range is extended then it shouldbenormalized.Fromthefigure//explainsaboutthe8- segment trigonometric circle and by using simple trigonometrictransformationsthechangeofanglefromother segment tozeroth-segment. Table I. The first 16 iteration values of tan(ø), and ø.

i	Tan(ø) = 2^{-i}	ø = arctan(2^{-i})
0	1	45.000000000
1	1/2	26.565051177
2	1/4	14.036243468
3	1/8	7.125016349
4	1/16	3.576334375
5	1/32	1.789910608
6	1/64	0.895173710
7	1/128	0.447614171
8	1/256	0.223810500
9	1/512	0.111905677
10	1/1024	0.0559552892
11	1/2048	0.027976453
12	1/4096	0.013988227
13	1/8192	0.006994114
14	1/16384	0.003497057
15	1/32768	0.001748528

III. FFT TWIDDLE FACTORARCHITECTURE

The total processing time is reduced by reducing thetotal number of iterations so that the values in the Look-Up-Table and the accuracy performance will be less. By limiting the number of iterations, the trade-off between the accuracy and timingresources.Fig. 4: Twiddle Factor Architecture overview.

The above figure consists of 4 important modules that is Ɵi – selection (Ɵisel), X-Y iteration addition (XY-Add), micro-lengthfactorkiiterationmultiplication(ki–Mul),and length–factorkmultiplicationandoutputnormalization(K– Mul andNorm).

The process starts with Ɵi – sel module, which receives Z0 as new input value. The micro rotation series is selected by the Ɵi – sel module and XY – Add and ki – Mul are used to transmit the Rot values. The XY – Add and ki – Mul modules receives a single value at each clock operation and then the iterative computation process Xi – Yi and ki respectively. The XY - Add modules also requires the initial values X0 and Y0 as shown in the figure. In the next clock operation Ɵi – sel module as used to transmit the Rot value. By using the two modules the final values of Xi – Yi and the length factor K is obtained.

Table II. 8-Segment information for data recovery

Segment	Angle Norm	X-Y swap		reversed	
		Input	Output	X	Y
0	z0	No	No	No	No
1	π/2-z0	Yes	No	Yes	No
2	z0-π/2	No	Yes	Yes	No
3	π-z0	Yes	Yes	Yes	Yes
4	z0-π	No	No	Yes	Yes
5	3π/2-z0	Yes	No	No	Yes
6	z0-3π/2	No	Yes	No	No
7	2π-z0	Yes	Yes	No	No

As the ACor range is [0-45], other than the ranges values mustbenormalizedasshownintable2.Z0isusedtomodify theinputvaluesX0andY0.Ifnecessarytheinputandoutput values can be swapped and sine can bereversed.

A. Θ_i–Sel module Block

The Ɵi – sel modules are the Z-Normalized, Arthimetic Logic Unit (ALU), Absolute Value Modifier (ABS), Set rotation (SetRot), Ɵi look-up table (Ɵi-LUT), and the input angle Z0 can be normalized by using the controller. The results of ALU can be positive toby Add/sub the current Zi with the Ɵi value.

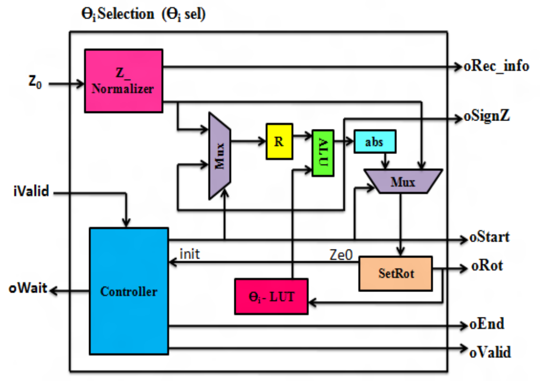


Fig. 5. Block diagram for Θi–Sel module.

If necessary the input angle of z0 is normalized by Z_Normalizer to the zeroth –segment from the table-II. The normalized angle is the NormZ signal and the oRec_info signal will carry recovery information.

The next zi value is evaluated by adding/subtracting the current value zi at the ALU and the register with Θi-LUT forms appropriate Θi value. By using abs module the ALU output can be changed to positive value and at the outside oSignZ signal is transferred as signed outside.

IV. Results

The simulation is carried out in Xilinx ISIM Tool andthe synthesisisperformedinXilinxISE14.5ToolforSpartan3E FPGA Family with Device XC3S500E, Package of FG320 and Speed Grade of -5. The design utilizes 81 slices out of 4656 slices, 0 LUTs out of 9312 LUTs and has a combinational path delay of 12.047ns as shown in Figure 6. The corresponding RTL and Technology Schematics are shown in figures 7 and 8 respectively. The simulation result is shown in figure 9, where the input given is 3F49, for which as per the design, the output obtained is3F49.

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	81	63288	0%
Number of fully used LUT-FF pairs	0	81	0%
Number of bonded IOBs	75	498	15%
Number of BUFG/BUFGCTRL/BUFHCEs	1	16	6%

Fig.6.DeviceUtilizationSummaryΘi–Selmodule.

RTL and Technology Schematics as Separate Figures (7&8).

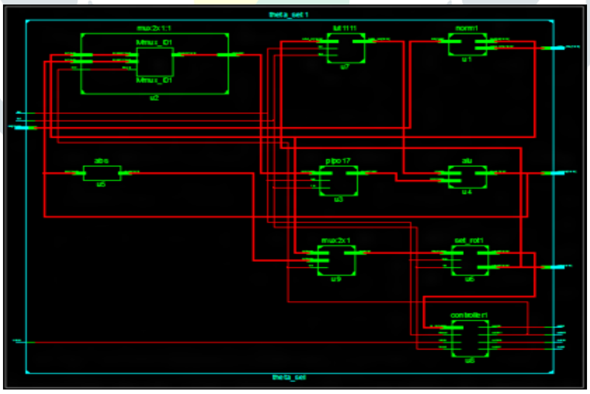


Fig. 7. RTL Schematic for Θi–Sel module.

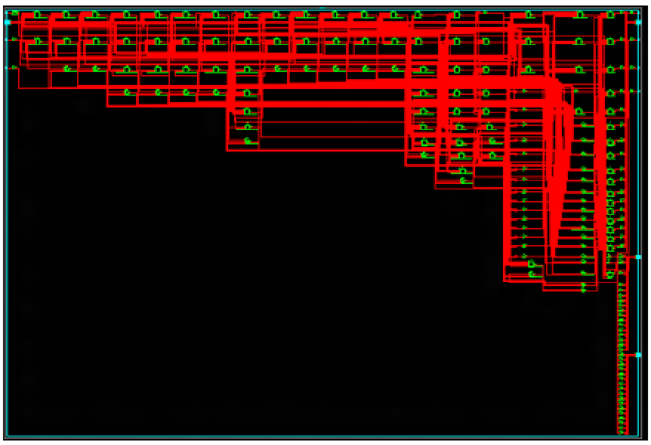


Fig. 8. Technology Schematic for Θi–Sel module.

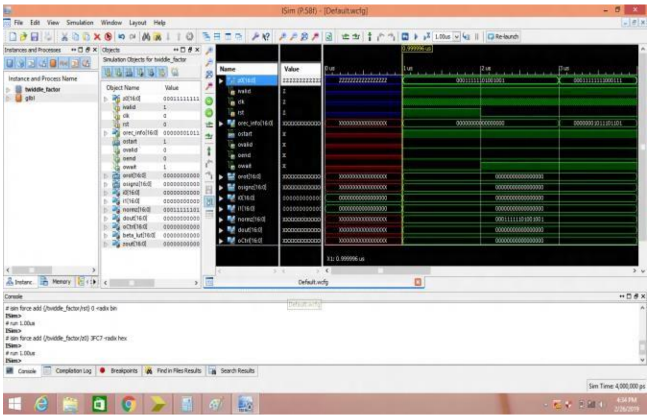


Fig. 9. Simulation Results for Ɵi–Sel module.

V. CONCLUSION

The design of theta I sel block was implemented in this paper. Adaptive CORDIC (ACor) algorithm is a decision making algorithm so that, it reduces the number of iterations which in turns reduces the latency compare to the conventional CORDIC algorithm.

VI. REFERENCES

[1] J. W. Cooley and J. W. Tukey, "An Algorithm for The Machine Calculation of Complex Fourier Series," Mathematics of Computation, vol. 19, no. 90, pp. 297–297, May1965.

[2] J. G. Andrews, A. Ghosh, and R. Muhamed, Fundamentals of WiMAX: Understanding Broadband Wireless Networking. Prentice Hall, 2007.

[3] E. Dahlman, 3G Evolution: HSPA and LTE for Mobile Broadband. Academic,2008.

[4] A. F. Molisch and X. Zhang, "FFT-Based Hybrid Antenna Selection Schemes for Spatially Correlated MIMO Channels," IEEE Comm. Letters, vol. 8, no. 1, pp. 36–38, Jan.2004.

[5] Y. Tang and B. Vucetic, "The FFT-based Multiuser Detection for DSCDMA Ultra-Wideband Communication Systems," in Int. Workshop on Ultra Wideband Syst. Joint with Conf. on Ultra Wideband Syst. and Tech. (Joint UWBST & IWUWBS 2004). IEEE, 2004, pp.111–115.

[6] V. Arunachalam and Alex N. J. Raj, "Efficient VLSI Implementation of FFT for Orthogonal Frequency Division Multiplexing Applications," IET Circ., Devices & Syst., vol. 8, no. 6, pp. 526–531, Nov. 2014.

[7] C.-T. Lin, Y.-C. Yu, and L.-D. Van, "A Low-Power 64-Point FFT/IFFT Design for IEEE 802.11a WLAN Application," in IEEE Int. Symp. On Circ. and Syst. IEEE, 2006, pp.4523–4526.

[8] A. Cortes, J. F. Sevillano, I. Velez, and A. Irizar, "An FFT Core for DVB-T/DVB-H Receivers," in 13th IEEE Int. Conf. on Electronics, Circ. and Syst. IEEE, Dec. 2006, pp.102–105.

[9] J. Li, M. V. Sarunic, and L. Shannon, "Scalable, High Performance Fourier Domain Optical Coherence Tomography: Why FPGAs and Not GPGPUs," in IEEE 19th Annual Int. Symp. on Field- Programmable Custom Computing Machines. IEEE, May 2011, pp. 49–56.

[10] C. Le, S. Chan, F. Cheng, W. Fang, M. Fischman, S. Hensley, R. Johnson, M. Jourdan, M. Marina, B. Parham, F. Rogez, P. Rosen, B. Shah, and S. Taft, "Onboard FPGA-based SAR Processing for Future Spaceborne Systems," in Proc. of IEEE Radar Conf. IEEE, 2004, pp. 15–20.

[11] C.F.GeraldandP.O.Wheatley,AppliedNumericalAnalysis,7thed. Pearson Higher Ed USA,2003.

[12] J. Zhou, Y. Dong, Y. Dou, and Y. Lei, "Dynamic Configurable Floating-Point FFT Pipelines and Hybrid-Mode CORDIC on FPGA," in Int. Conf. on Embedded Software and Syst. Sichuan, China: IEEE, 2008, pp.616–620.

[13] X. Xiao, E. Oruklu, and J. Saniie, "Reduced Memory Architecture for CORDIC-based FFT," in Proc. of IEEE Int. Symp. on Circ. and Syst. IEEE, May 2010, pp.2690–2693.

[14] Earl E. Swartzlander and H. H. M. Saleh, "FFT Implementation with Fused Floating-Point Operations," IEEE Trans. on Computers, vol. 61, no. 2, pp. 284–288, Feb.2012.

[15] J. H. Min, S.-W. Kim, and Earl E. Swartzlander, "A Floating-point Fused FFT Butterfly Arithmetic Unit with Merged Multiple-Constant Multipliers," in Conf. Record of the 45th Asilomar Conf. on Signals, Syst. and Computers (ASILOMAR). Pacific Grove, USA: IEEE, Nov. 2011, pp.520–524.

[16] A. Kaivani and S. Ko, "Floating-Point Butterfly Architecture Based onBinarySigned-DigitRepresentation,"IEEETrans.onVeryLarge

[17] Scale Integration (VLSI) Syst., vol. 24, no. 3, pp. 1208–1211, Mar. 2016.

[18] J. Sohn and E. E. Swartzlander, "Improved Architectures for a Floating-Point Fused Dot Product Unit," in IEEE 21st Symp. on Computer Arithmetic. Austin, USA: IEEE, Apr. 2013, pp.41–48.

[19] Phuong-Thao Vo-Thi, Trong-Thuc Hoang, Cong-Kha Pham, Duc- Hung Le, "A Floating-point FFT Twiddle Factor Implementation Based on Adaptive Angle Recoding CORDIC," in Int. Conf. on Recent Advances in Signal Processing, Telecomm. & Computing (SigTelCom). Danang, Vietnam: IEEE, Jan. 2017, pp.21–26.

[20] R. Radhouane, P. Liu, and C. Modlin, “Minimizing the Memory Requirement for Continuous Flow FFT Implementation: Continuous Flow Mixed Mode FFT (CFMM-FFT),” in Proc. of IEEE Int. Symp. on Circ. and Syst. Emerging Tech. for the 21st Century, vol. 1. Presses Polytech. Univ. Romandes, 2000, pp.116–119.

[21] S. Mou and X. Yang, “Research on the RAW Dependency in Floating-point FFT Processors,” in 8th ACIS Int. Conf. on Software Engi., Artificial Intel., Networking, and Parallel/Distributed Computing (SNPD 2007). IEEE, Jul. 2007, pp.88–92.

[22] H.-Y. Lee and I.-C. Park, “Balanced Binary-Tree Decomposition for Area-Efficient Pipelined FFT Processing,” IEEE Trans. on Circ. And Syst. I: Regular Papers, vol. 54, no. 4, pp. 889–900, Apr.2007.

[23] X. Xiao, E. Oruklu, and J. Saniie, “An Efficient FFT Engine With Reduced Addressing Logic,” IEEE Trans. on Circ. and Syst. II: Express Briefs, vol. 55, no. 11, pp. 1149–1153, Nov.2008.

[24] Y.H. Hu and S. Naganathan, “AnAngle Recoding Method for CORDIC Algorithm Implementation,” IEEE Trans. on Computers, vol. 42, no. 1, pp. 99–102, Jan.1993.

[25] Hong-Thu Nguyen, Xuan-Thuan Nguyen, Trong-Thuc Hoang, DucHung Le, and Cong-Kha Pham, “Low-resource Low-latency Hybrid Adaptive CORDIC with Floating-point Precision,” IEICE Electronics Express (ELEX), vol. 12, no. 9, pp. 20 150 258–20 150 258,2015.