Design and Implementation of Low Power and Highly Scaled DCT Architecture with CORDIC Algorithm

Bhaskar S.V
Assistant Professor
Department of E&C Engg
CBIT Kolar, India

Easwara. M
Assistant Professor
Department of E&C Engg
CBIT Kolar, India

ABSTRACT
This project deals with the hardware implementation of the DCT and IDCT algorithm in a more efficient way by the use of CORDIC algorithm. DCT and IDCT are the most widely used transform technique in Digital Image Processing and Digital Signal Processing. This project presents an efficient approach for multiplier less implementation for N-point DCT approximation, which based on coordinate rotation digital computer (CORDIC) algorithm which makes use of shift and add operation for computation. The proposed algorithm is the most popular because of its computational efficiency and structural simplicity. It has advantages such as regular data flow, uniform post scaling factor, arithmetic sequence rotation angels. In this project an N- point DCT is deduced using two N/2-point DCTs by using orthogonal properties of DCT and IDCT,and also adders are replaced by carry skip adder [1]. Signal flow of 8-point DCT and IDCT CORDIC algorithm are coded and functionality of the design will check using ModelSim simulator. The design will synthesize using Cadence and Xilinx ISE Synthesis tool and the bit file will dumped to a Spartan 3 FPGA kit.

Keywords
Cordinate Rotation Digital computer (CORDIC), discrete cosine transform (DCT), fastradix – 2 algorithm.

1. INTRODUCTION
Recently, much research has been done on low-power DCT designs [2]. In consideration of VLSI-implementation, Flow-Graph Algorithm (FGA) is the most popular way to realize the fast DCT (FDCT) [3]. In 1989, Loeffler et al. [4] proposed a low-complexity FDCT/IDCT algorithm based on FGA that requires only 11 multiply and 29 addoperations. However, the multiplications consume about 40% of the power and almost 45% of the total area . Thus, Tran [5] proposed the binDCT which approximates multiplications with add and shift operations. Although the binDCTreduces the computational complexity significantly, it also loses about 2 dB in PSNR compared to the Loeffler DCT [6].Jeong et al. [7] have proposed a Cordic based implementation of the DCT. COordinate Rotation DIgital Computer (Cordic) is an algorithm which can be used to evaluate various functions in signal processing [8], [9]. In addition, the Cordicalgorithm is highly suited for VLSI-implementation.This paper proposes a Design of less Area and Low power DCT architecture with CORDIC Algorithm. Based on the proposed algorithm, signal flows of DCTs and inverse DCTs (IDCTs) are developed and de-duced using their orthogonal properties, respectively. Similar to the Cooley-Tukey fast Fourier transformation (FFT) algorithm, the proposed algorithm can generate the next higher-order DCT from two identical lower-orderDCTs. Furthermore, it has some distinguish advantages, such as FFT-like regular data flow, uniform post-scaling factor, in-place computation and arithmetic-sequence rotation angles. By using the unfolding CORDIC technique, this algorithm can overcome the problem of difficult to realize pipeline that in conventional CORDIC algorithms. This results in a pipeline and high-speed VLSI implementation. Compared to existing DCTs, the proposed algorithm has low computational complexity, and is highly scalable, modular, regular, and able to admit efficient pipelined implementation. In addition, this letter also provides an easy way to implement the reconfigurable or unified architecture for DCTs and IDCTs using the orthogonal property.

2. CORDIC ALGORITHM
CORDIC, an acronym for COordinate Rotation DIgital Computer. What makes CORDIC fast are two facts: (1) When we take a binary number and multiply it by 2*n*, we shift the binary point *n*places to the right and when we divide by 2*n* we shift the binary point *n*places to the left. (2) The operations on a computer that are cheapest and fastest to perform are (A) addition and subtraction, (B) comparing numbers to see which is larger or smaller, (C) storing and retrieving numbers from memory, and (D) shifting the binary point. Addition and subtraction are very fast, but not multiplication or division. The machine does compute multiplications and divisions by powers of 2 very quickly however, by just shifting the binary point. CORDIC exploits this, and thus uses only operations (A)–(D) to evaluate sines and cosines. The algorithm credited to volder[8] is derived from general rotation transform:

$$x' = x\cos\phi - y\sin\phi$$

$$y' = y\cos\phi + x\sin\phi$$

Which rotates vector in Cartesian plane by an angle ϕ. These can be rearranged as:

$$x' = \cos\phi[x - y\tan\phi]$$

$$y' = \cos\phi[y + x\tan\phi]$$

Rotation angles are restricted so thattan$\phi = \pm 2^{-i}$,the multiplication by tangent term is reduced to simple shift and add operations. Arbitary angles of rotation are obtainable by performing series of successive smaller elementary rotations.If the decision at each iteration is to which direction to rotate, then term $\cos\delta_i$ is constant because $\cos\delta_i$=cos$-\delta_i$The iterative rotation now expressed as:

$$x_{i+1} = k_i(x_i - d_i y_i 2^{-i})$$

$$y_{i+1} = k_i(y_i + d_i x_i 2^{-i})$$

$$z_{i+1} = z_i + d_i \emptyset_i$$

Where, i denotes the number of rotation required to reach the required angle of the required vector, the product of $k_i =$

$(\cos(\tan^{-1} 2^{-i}))$ and d_i= ±1 indicates the direction of rotation.

K=$\prod_{i=0}^{n-1} k_i$

Removing scaling constant from iterative equation yields a shift-add algorithm for vector rotation. The product of k_i's can be applied elsewhere in the system or it can be treated as systems processing gain. The product approaches to 0.6073 as number of iterations go to infinity. Therefore rotation algorithm has gain A_n,of approximately 1.647. The exact gain depends on number of iterations and obeys the relation

$$A_n = \prod_n \sqrt{1+2^{-2i}}$$

The angle of composite rotation is uniquely defined by sequence of directions of elementary rotations. That sequence can be represented by decision vector. The set of all decision vectors is an angular measurement system based on binary arctangents. Conversion between this angular system and any other can be accomplished using table look up. Better conversion methods uses additional adder sub tractor that accumulates elementary angles of rotation at each iteration. The angle accumulator adds third difference equation to the CORDIC algorithm:

$z_{i+1} = z_i - d_i \tan^{-1} 2^{-i}$.

3. CORDIC DCT

Jeong et al. [3] have proposed an 8-point Cordicbased DCT with six Cordic Rotations to realize multiplierless approximation which requires 104 add and 84 shift operations. In addition, it has a very regular structure which is suitable for VLSI design.

In order to realize a vector rotation for the Cordic algorithm, that is rotating a vector (*x*,*y*) by an angle ϕ, the circular rotation angle is described as

$$\emptyset = \sum_i d_i \tan^{-1} 2^{-i}$$

With d_i=1,-1

Then, the vector rotation can be performed using the iterative equation given by

$$x_{i+1}=x_i-d_i y_i 2^{-i}$$

$$y_{i+1}=y_i+d_i x_i 2^{-i}$$

4. CORDIC BASED FAST RADIX-2 DCT ALGORITHM

For N-point signal x[n], the DCT is defined as

$$C[k] = \alpha[k] \sum_{n=0}^{N-1} x[n] \cos\left[\frac{(2N+1)k\pi}{2N}\right], k = 0,1 \dots N-1$$

…(4.1)

Where $\alpha[k] = \begin{cases} \frac{1}{\sqrt{N}} & if\ k = 0 \\ \sqrt{2/N} & otherwise \end{cases}$

Neglecting the scaling factor, rewriting the equation 4.1

$$\tilde{C}[k] = \sum_{n=0}^{N-1} x[n] \cos\left[\frac{(2N+1)k\pi}{2N}\right], k = 0,.N-1$$

……. (4.2)

A length-N input sequence x[n], with N is power of two can be decomposed into $x_L[n]$ and $x_H[n]$,

$$\left.\begin{aligned} x_L[n] &= \frac{1}{2}\{x[2n] + x[2n+1]\} \\ x_H[n] &= \frac{1}{2}\{x[2n] - x[2n+1]\} \end{aligned}\right\}$$

……(4.3)

Where n=0, 1, 2….(N/2)-1

So the original signal x[n] can be obtained from $x_L[n]$ and $x_H[n]$, as follows

$$\left.\begin{aligned} x[2n] &= x_L[n] + x_H[n] \\ x[2n+1] &= x_L[n] - x_H[n] \end{aligned}\right\}$$……..…(4.4)

Substituting eqn3.4 in eqn4.2, eqn4.2 can be written as

$\tilde{C}[k] = 2\cos\left(\frac{\pi k}{2N}\right)\sum_{n=0}^{N/2-1} x_L[n]\cos\left[\frac{(2n+1)k\pi}{2N}\right] +$
*2*sin*πk2Nn=0N2−1xH[n]*sin*(2n+1)2N*……(4.5)

Where k=0, 1…N-1

Since

$$\cos\left[\frac{(2n+1)(\frac{N}{2}-k)\pi}{N}\right] = \sin\left(\frac{2n+1}{2}\pi\right)\sin\left(\frac{2n+1}{N}k\pi\right)$$

……..(4.6)

So equation4.5 becomes

$$\begin{aligned} &\tilde{C}[k] \\ &= 2\cos\left(\frac{\pi k}{2N}\right)\sum_{n=0}^{\frac{N}{2}-1} x_L[n]\cos\left[\frac{(2n+1)k\pi}{2N}\right] \\ &+ 2\ \sin\left(\frac{\pi k}{2N}\right)\sum_{n=0}^{\frac{N}{2}-1}(-1)^n x_H[n]\cos\left[\frac{(2n+1)\left(\frac{N}{2}-k\right)\pi}{N}\right] \end{aligned}$$

……..(4.7)

$$\begin{aligned} &\tilde{C}[N-k] \\ &= -2\sin\left(\frac{\pi k}{2N}\right)\sum_{n=0}^{\frac{N}{2}-1} x_L[n]\cos\left[\frac{(2n+1)k\pi}{2N}\right] \\ &+ 2\cos\left(\frac{\pi k}{2N}\right)\sum_{n=0}^{\frac{N}{2}-1}(-1)^n x_H[n]\cos\left[\frac{(2n+1)\left(\frac{N}{2}-k\right)\pi}{N}\right] \end{aligned}$$

…….. (4.8)

Where k=0, 1… (N/2)-1

From eqn4.7 and eqn4.8, we find that each equation has two (N/2)-point $\widetilde{DCT}$ with two different coefficients, and the four coefficients just make one CORDIC. Hence we combine the two equations to realize a CORDIC based fast DCT algorithm.

Let

$\tilde{x}_H[n] = (-1)^n x_H[n]$ …… (4.9)

Combining the constant values 2 and $\sqrt{2}$ in recursively decomposing stages with the post scaling factor, the DCT can be written as

$$C(k) = \frac{1}{\sqrt{N}}\tilde{C}\mathrm{L}[0] = \frac{1}{\sqrt{N}}\sum_{n=0}^{\frac{N}{2}-1} x_L[n]\ , k = 0$$

$$C(k) = \frac{1}{\sqrt{N}}\tilde{C}\hat{H}[0] = \frac{1}{\sqrt{N}}\sum_{n=0}^{\frac{N}{2}-1}(-1)^n x_H[n]\ , \mathrm{k} = \mathrm{N}/2$$

$$C(k) = \frac{1}{\sqrt{N}}\begin{bmatrix}\tilde{C}(k)\\ \tilde{C}(N-k)\end{bmatrix} = \begin{bmatrix}\cos\left(\frac{\pi k}{2N}\right) & \sin\left(\frac{\pi k}{2N}\right)\\ -\sin\left(\frac{\pi k}{2N}\right) & \cos\left(\frac{\pi k}{2N}\right)\end{bmatrix}\begin{bmatrix}\tilde{C}L(k)\\ \tilde{C}\hat{H}\left(\frac{N}{2}-k\right)\end{bmatrix},$$

k=1…(N/2)-1 eqn...4.10

Where $\tilde{C}L$and $\tilde{C}\hat{H}$denote the (N/2)-1 point $\widetilde{DCT}$ of the $x_L[n]$ and$x\hat{H}$(n) respectively.

According to eqn4.10, N-point DCT can be decomposed into two N/2 –point DCTs based on CORDIC algorithm. Since the basic operation of the algorithm is 2-point DCT, this algorithm is called fast radix-2 DCT.

Rotation angles of the CORDICs are fixed and they are the arithmetic sequences with common difference$\left(\frac{-\pi}{2N}\right)$.Another important aspect of the algorithm is that all the outputs,$C(k)$, k=0,1…(N/2)-1, have uniform post scaling factor.

5. DCT SIGNAL FLOW BASED ON PROPOSED ALGORITHM

In this project a novel coordinate rotation digital computer (CORDIC)-based fast radix-2 algorithm for low power and less area is proposed for computation of discrete cosine transformation (DCT). Among radix algorithms, the radix-2 algorithm is the most popular because of its computational efficiency and structural simplicity. The proposed algorithm can generate next higher order DCT from two identical lower orders DCTs. Compared to existing DCT algorithms, this proposed algorithm has lower computational complexity.And consume less power and area Furthermore; the proposed algorithm is highly scalable, regular, modular, and suitable for pipelined VLSI implementation. The general signal-flow graph for the proposed fast DCT algorithm is given in the Fig 5.1. To compute N-point DCT this algorithm needs two N/2-point DCTs and one CORDIC array. CORDIC array has N-1 CORDICs. Rotation angles of the CORDICs are arithmetic sequences with common difference of $^{-\pi}/_{2N}$. CORDICs uses only additions/subtractions and shifting operations.In the proposed algorithm the adders and sub tractors are replaced by one single adder or sub tractor model it will design by using carry skip adder, this adder reduces the area and power. Compare to the existing system in the proposed system require less area and power.For special applications, a double-angle formula can be used to reduce CORDIC types. Hence, the architecture based on the signal flow is highly modular. Furthermore, the modified unfolded CORDIC [9] can be used to speed up computations and overcome recursive problems in conventional CORDICs. Similarly, the fast algorithm for the

point IDCT can be deduced like the fast DCT algorithm. Alternatively, it can be obtained more easily using their orthogonal property. As is known, the DCT and IDCT are orthogonal transformations, and the signal flow of the N-point IDCT can be easily obtained by inverting the transfer function of each building block and reversing the signal flow direction i.e. change the CORDIC from clockwise rotation to anti-clockwise rotation with same angle and change all the adders to sub tractors and all sub tractors to adder.

Fig.5.2 shows the proposed less area low power DCT architecture with CORDIC algorithm. This architecture reduces the disadvantages of existing Radix-2 DCT algorithm in the proposed architecture the adders are replaced by using carry skip adder reduces the area and power.Fig5.3 shows the proposed less area low power DCT architecture with CORDIC algorithm.Regular and pure feed-forward data paths of the signal flow make them suitable for pipelined VLSI implementation. For special applications, a double-angle formula can be used to reduce CORDIC types. Hence, the architecture based on the signal flow is highly modular.

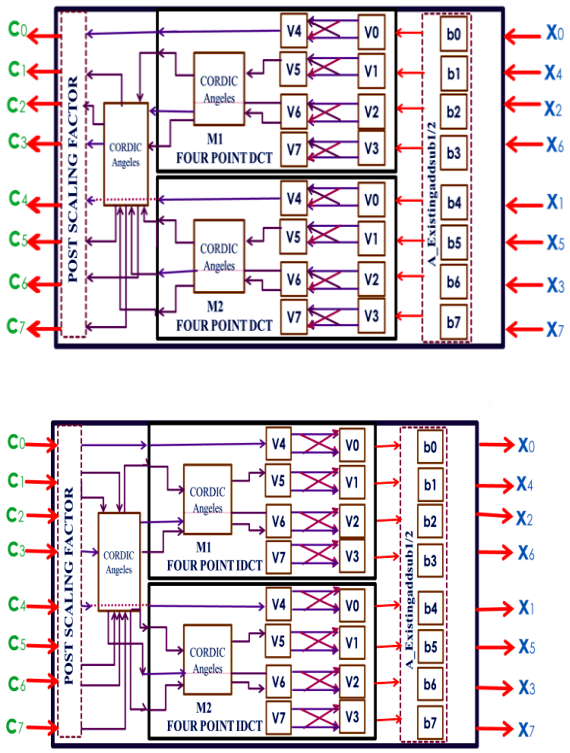

Fig 2 Signal flow of proposed 8-point DCT

Similarly, the fast algorithm for the N-point IDCT can be deduced like the fast DCT algorithm. Alternatively, it can be obtained more easily usingtheir orthogonal property.

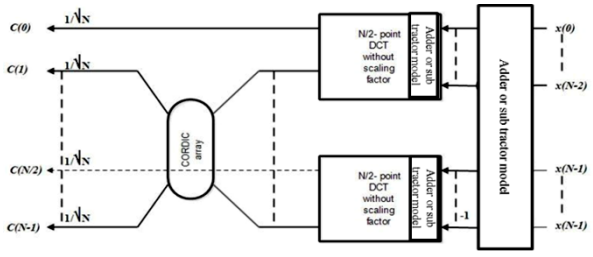

Fig3Signal flow of proposed 8-point IDCT

6. EXPERIMENTAL RESULTS

Cordic based proposed algorithm has been simulated using XILINX software and results are compared with existing DCT algorithms. Table 6.1below has the summary of the power and area ofthe proposed and existing DCT algorithm. From the table it is clear that proposed CORDIC based DCT algorithm

has lower computational complexity compared to the existing algorithms and also this proposed algorithm is highly scalable, low power modular, regular and suitable for pipelined VLSI implementation**.**

AREA						
		EXISTING		PROPOSED		
	MAIN MODULE	M2	M1	MAIN MODEL	M2	M1
CELL	842	256	255	488	135	135
CELL AREA(μm²)	5586	2101	2100	4231	1348	1348

POWER						
		EXISTING		PROPOSED		
	MAIN MODULE	M2	M1	MAIN MODEL	M2	M1
LEAKAGE POWER(nW)	22848.885	9454.12	9465.008	19564.023	6529.753	6524.39
DYNAMIC POWER(nW)	614444.807	275432.12	268087.776	351125.246	137020.459	135743.519
TOTAL POWER(nW)	637293.693	2484887.83	277552.284	370689.269	143550.212	142267.968

Table 6.1comparison of area and power of DCT

The existing system model consist of 842 cells it require area of 5586 μm². Existing system contain 488 cells and area of 4231 μm². By comparing these two the proposed system require less area.

Similarly when come to the power the existing system require total power of 637293.693nW (0.6mW) and proposed system require 370689.269nW (0.3mW) by comparing these two proposed require low power compare to existing.

Computational complexity of the proposed algorithm can be summarized by following two equations

Number of CORDIC's is

$$C(N) = \sum_{n=1}^{\log\left(\frac{N}{2}\right)-1} 2^{\log\left(\frac{N}{2}\right)-1} \quad 2^{n-1}$$

And number of additions is

$$A(N) = N \log_2 N$$

The functionality of proposed CORDIC based DCT algorithm is carried out through ModelSim simulator. Results for 8-point DCT and IDCT are shown below and these results can be verified using matlab.

Fig.6.1 shows the ModelSim simulator output for 8-point DCT. For 8-point input sequence x(n)={8,16,24,32,40,48,56,64}, DCT obtained is given by C(k)={101.808,-51.5332,0,-5.39544,0 ,-1.59959,0,-0.406306}.

Fig.6.2 shows the ModelSim output for 8-point IDCT. Input is 8-point DCT sequence C(k)={101.808,-51.5332,0,-5.39544,0 ,-1.59959,0,-0.406306} and the output is the reconstructed sequence

x(n)= {8,16,24,32,40,48,56,64}.

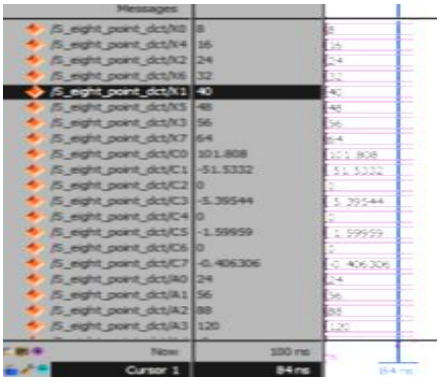

Fig.6.1 Device utilization summary of existing method

From the ModelSim simulator outputs, it can be inferred that the functionality of the proposed algorithm is working fine.

Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Number of 4 input LUTs	262	7,168	3%	
Logic Distribution				
Number of occupied Slices	133	3,584	3%	
Number of Slices containing only related logic	133	133	100%	
Number of Slices containing unrelated logic	0	133	0%	
Total Number of 4 input LUTs	263	7,168	3%	
Number used as logic	262			
Number used as a route-thru	1			
Number of bonded IOBs	128	141	90%	
Total equivalent gate count for design	2,964			
Additional JTAG gate count for IOBs	6,144			

Fig 6.2 ModelSim simulator output for 8-point DCT

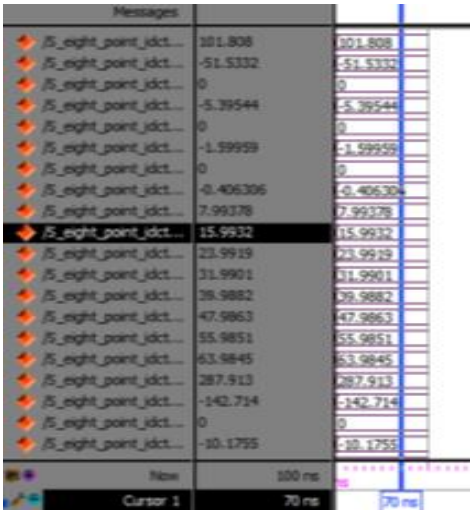

Fig 6.3 ModelSim simulator output for 8-point IDCT

Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Number of 4 input LUTs	424	7,168	5%	
Logic Distribution				
Number of occupied Slices	225	3,584	6%	
Number of Slices containing only related logic	225	225	100%	
Number of Slices containing unrelated logic	0	225	0%	
Total Number of 4 input LUTs	424	7,168	5%	
Number of bonded IOBs	128	141	90%	
Total equivalent gate count for design	2,805			
Additional JTAG gate count for IOBs	6,144			

Fig.6.4 Device utilization summary of proposed method

Fig 6.5 CORDIC based DCT hard ware implementation

7. CONCLUSION AND FUTURE SCOPE

This paper presents a novel CORDIC-based radix-2 fast DCT algorithm. This algorithm can generate the next higher order DCT from two identical lower-orders DCTs. Compared to existing DCT algorithms, proposed algorithm has several distinct advantages, such as low computational complexity, and being highly scalable, modular, regular, and able to admit efficient pipelined implementation.

Furthermore, the proposed algorithm also provides an easy way to implement a reconfigurable or unified architecture for DCTs and IDCTs, which will be researched in our future work. In addition, other types (types I and IV) of CORDIC-based fast DCT algorithms will also beexamined in future work.

8. REFERENCES

[1] Haihuangand Liyi Xiao "Cordic Based Fast Radix-2 DCT Algorithm, IEEE Signal Processing Letters, Vol. 20, No. 5, May 2013

[2] N. J. August and D. S. Ha, "Low power design of DCT and IDCT for low bit rate video codecs," *IEEE Transactions on Multimedia*, vol. 6, no. 3, pp. 414–422, June 2004.

[3] W. H. Chen, C. Smith, and S. Fralick, "A Fast Computational Algorithm for the Discrete Cosine Transform," *IEEE Trans. Commun.*, vol. 25, no. 9, pp. 1004–1009, Sept. 1977.

[4] C. Loeffler, A. Lightenberg, and G. S. Moschytz, "Practical fast 1-D DCT algorithms with 11-multiplications," in *Proc. ICASSP*, vol. 2, May 1989, pp. 988–991

[5] J. Liang and T. D. Trac, "Fast multiplierless approximations of the DCT with the lifting scheme," *IEEE Trans. Acoust., Speech, SignalProcessing*, vol. 49, no. 12, pp. 3032–3044, Dec. 2001.

[6] C. C. Sung, S. J. Ruan, B. Y. Lin, and M. C. Shie, "Quality andPower Effcient Architecture for the Discrete Cosine Transform," *IEICE Transactions on Fundamentals Special Section on VLSI Design and CAD Algorithms*, to appear in Dec. 2005.

[7] H. Jeong, J. Kim, and W. K. Cho, "Low-power multiplierless DCT architecture using image correlation," *IEEE Trans. Consumer Electron.*, vol. 50, no. 1, pp. 262–267, Feb. 2004.

[8] J. Volder, "The CORDIC trigonometric computing technique," *IRE Trans. Electron. Comput.*, vol. EC-8, pp. 330–334, 1959.

[9] J. Walther, "A unified algorithm for elementary functions," in *Proc. Spring Joint Comput. Conf.*, vol. 38, 1971, pp. 379–385

[10] C.-C. Sun, S.-J.Ruan, B. Heyne, and J. Goetze, "Low-power and high quality CORDIC-based loeffler DCT for signal processing," *IET Proc.-Circuits, Devices Syst.*, vol. 1, no. 6, pp. 453–461, 200