

# Linköping University Post Print

## Alternatives for Low-Complexity Complex Rotators

Fahad Qureshi, Mario Garrido and Oscar Gustafsson

N.B.: When citing this work, cite the original article.

©2010 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Fahad Qureshi, Mario Garrido and Oscar Gustafsson, Alternatives for Low-Complexity Complex Rotators, 2010, The 17th IEEE International Conference on Electronics, Circuits, and Systems, (ICECS 2010), Athens, Dec-12-15, 2010.

Postprint available at: Linköping University Electronic Press

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-65909>

# Alternatives for Low-Complexity Complex Rotators

Fahad Qureshi, Mario Garrido, and Oscar Gustafsson  
 Department of Electrical Engineering, Linköping University  
 SE-581 83 Linköping, Sweden  
 E-mail: {fahadq, mariog, oscarg}@isy.liu.se

**Abstract**—Complex rotations find use in common transforms such as the Discrete Cosine Transform (DCT) and the Discrete Fourier Transform (DFT). In this work we consider low-complexity realization of constant angle rotators based on shifts, adders, and subtracters. The results show that redundant CORDIC and scaled constant multiplication are providing the best results, depending on which angle is considered. It is also shown that the precision can vary several bits using the same number of adders and subtracters, and, hence, the correct choice of rotator architecture is crucial for a low-complexity realization.

## I. INTRODUCTION

A rotation is a complex multiplication where the magnitude of the complex coefficient is equal to one, i.e., only the phase of the data is affected. Rotations happen, among other DSP algorithms, in many transforms, e.g., the Discrete Fourier Transform (DFT) [1] and the Discrete Cosine Transform (DCT) [2]. Also, many algorithms implementing the DFT, such as Fast Fourier Transform (FFT) [3] algorithms and the Goertzel algorithms [1], and the DCT, such as the fast DCT [4], will be based on rotations.

A general rotation of  $\alpha$  rad is written as:

$$\begin{aligned} x' &= x \cdot \cos \alpha - y \cdot \sin \alpha \\ y' &= y \cdot \cos \alpha + x \cdot \sin \alpha \end{aligned} \quad (1)$$

where  $x$  and  $y$  are the real and imaginary parts of the data, respectively. The result is represented by the real and imaginary components  $x'$  and  $y'$ , respectively.

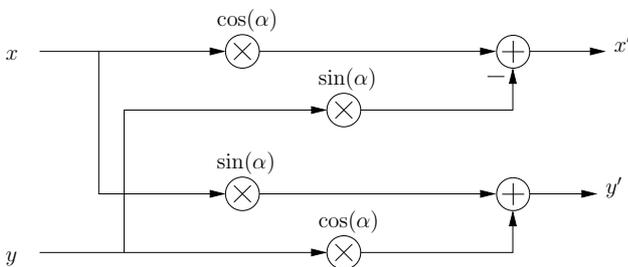


Fig. 1. General complex rotation.

The rotation in (1) is shown in Fig. 1. This rotation can be computed in several different ways, including a general complex multiplication [5] and the CORDIC algorithm [6]. When the rotation is known in advance it is possible to simplify the computation leading to an optimized shift-and-add

realization. This holds for both the multiplication approach and the CORDIC one.

It is sometimes advantageous from an implementation point of view to introduce a scaling factor in (1). This means that the coefficient does not have unit gain. This is inherent in the CORDIC algorithm as each sub-rotation introduces a gain. For multiplication-based approaches it can also be advantageous, as it can lead to the case that one of the coefficients, or even both of them, becomes very simple. For DCT algorithms this scaling factor can be compensated in later stages of the image coding process [7]. For DFT algorithms it is often required that the scaling is the same for several different rotators. Hence, here we will primarily consider rotations for the DCT, even though the results, with additional constraints, can be applied to DFT as well.

In this work, we consider the realization of constant rotations with a focus on low-complexity realizations based on shifts, adders, and subtracters. As the complexity of adders and subtracters are about the same, we will refer to both as adders. Also, as shifts can be hard-wired in bit-parallel arithmetic, we will focus on the number of adders as the cost to minimize.

The rest of the paper is arranged as follows. In the next section, different alternatives for rotators are presented. Then, in Section III the errors and complexity are presented for the different alternatives, and the obtained results are discussed. Finally, some conclusions are given in Section IV.

## II. ROTATOR ALTERNATIVES

### A. CORDIC

CORDIC (COordinate Rotation DIgital Computer) [6] is one popular algorithm for the implementation of multiplier-less rotations. It realizes rotation by means of a series of shifts and additions, which reduces the amount of hardware.

The CORDIC algorithm decomposes the angle that has to be rotated,  $\theta$ , into a sum of  $M$  predefined angles,  $\alpha_i$ , according to:

$$\theta = \sum_{i=0}^{M-1} \delta_i \alpha_i + \epsilon \quad (2)$$

where  $\epsilon$  is the error of the approximation,  $\delta_i$  indicates the direction of the so called micro-rotation and:

$$\alpha_i = \tan^{-1}(2^{-i}) \quad (3)$$

These angles that define the micro-rotations have the property that they can be rotated by shifts and additions, which reduces

significantly the hardware resource. These micro-rotations are carried out as follows:

$$\begin{aligned} x_{i+1} &= x_i - y_i \delta_i 2^{-i} \\ y_{i+1} &= y_i + x_i \delta_i 2^{-i} \end{aligned} \quad (4)$$

The hardware circuit for calculating the case of  $\delta_i = 1$  is depicted in Fig. 2. In Fig. 2, the angle  $\alpha_i$  that the input datum is rotate is chosen by setting the number of bits that are shifted before the additions and subtractions are carried out.

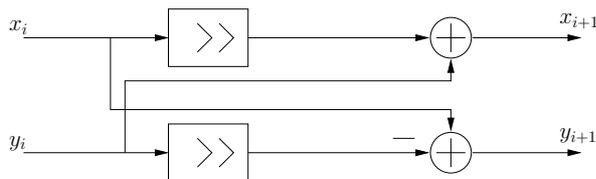


Fig. 2. CORDIC micro-rotation.

Usually  $\delta \in \{-1, 1\}$ . This forces to calculate all the micro-rotations either clockwise or counterclockwise and assures a constant gain of the CORDIC, which can be compensated by multiplying the outputs by:

$$K = \prod_{i=0}^M \cos(\alpha_i) = \prod_{i=0}^M \cos(\tan^{-1}(2^{-i})) \quad (5)$$

This option is preferable when the circuit is used for rotating several different angles, and a constant gain for all of them is required, as happens in the rotators for the FFT [8]. However, in a constant rotator only a single angle  $\theta$  must be rotated. In this case it is better to consider  $\delta_i \in \{-1, 0, 1\}$ . This approach is called redundant CORDIC [9] and allows to remove certain micro-rotations, reducing the number of adders.

### B. Constant multiplication

For constant multiplication, it possible to replace the general multiplier by shifts, adders, and subtractors. Adders and subtractors have same complexity so we refer to both as adders. When an input signal is multiplied by more than one constant, a simple method is to realize each multiplier individually, which can be done optimally for up to 19-bits coefficients [10]. However, it is also possible to utilize redundancies between the constants in order to reduce the complexity of the hardware. In terms of complexity, the shift operations are free, only reduce the number of adders in multiple constant multiplications (MCM) to implement the constant multiplications. In particular, for complex multiplications each input is multiplied by two constant coefficients. A dedicated algorithm for realizing MCM with two constants has been proposed in [11] and is used in this work.

In fact that, any rotation angle  $\alpha$  can be realized with constant multiplications so it is possible to implement the complex rotations. General rotator transform is defined as

$$\begin{aligned} x' &= x \cdot \cos \alpha - y \cdot \sin \alpha \\ y' &= y \cdot \cos \alpha + x \cdot \sin \alpha \end{aligned} \quad (6)$$

The constant multiplication algorithm for equation (6) is based on the implementation on constant value of the sine and cosine functions. The complexity is depending upon the precision requirement of rotations.

### C. Scaled constant multiplication

Scaling is a method to readjust the internal parameters of the system without changing the transfer function [5]. This procedure can be applied to a complex rotation in order to make one of the coefficients be equal to 1. Thus, by extracting the term  $\sin \alpha$  from equation (6), the following equations are obtained:

$$\begin{aligned} x' &= \sin \alpha \left( x \cdot \frac{\cos \alpha}{\sin \alpha} - y \right) \\ y' &= \sin \alpha \left( y \cdot \frac{\cos \alpha}{\sin \alpha} + x \right) \end{aligned} \quad (7)$$

This scaling allows to reduce the internal constant multiplications to two, in contrast to the four ones required in equation (6). Then, the outputs are scaled by a constant factor. In the case under study, this scaling can be incorporated to the corresponding constant of the quantizer, leading to savings in the number of adders. The resulting constant multiplications can be straightforwardly realized using the optimal approach in [10].

The analogous case consist in taking the common factor  $\cos \alpha$  according to:

$$\begin{aligned} x' &= \cos \alpha \left( x - y \cdot \frac{\sin \alpha}{\cos \alpha} \right) \\ y' &= \cos \alpha \left( y + x \cdot \frac{\sin \alpha}{\cos \alpha} \right) \end{aligned} \quad (8)$$

### D. General scaled constant multiplication

The scaling explained in previous section can be further generalized. Thus, a scaling factor  $R$  can be considered, which transform equation (6) into:

$$\begin{aligned} x' &= 1/R (x \cdot R \cos \alpha - y \cdot R \sin \alpha) \\ y' &= 1/R (y \cdot R \cos \alpha + x \cdot R \sin \alpha) \end{aligned} \quad (9)$$

This general scaled constant multiplication allows to look for the value of  $R$  that leads to the lowest rotation error. However, finding the best value of  $R$  requires an exhaustive search on a very fine grid. Hence, here we only note that this possibility would lead to better or as good results as the two previous methods based on constant multiplication. However, due to the computational complexity involved in performing this search, no results are presented in the current work.

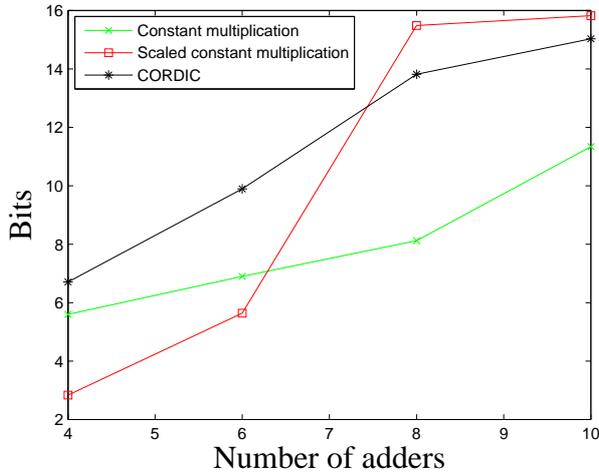


Fig. 3. Number of bits against adders for rotation  $\pi/16$ .

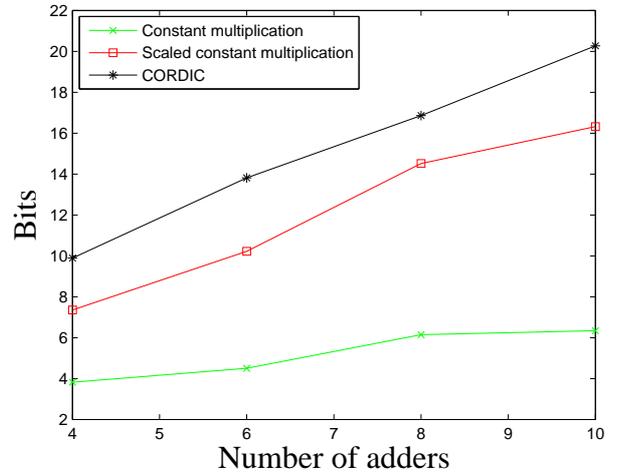


Fig. 5. Number of bits against adders for rotation  $3\pi/16$ .

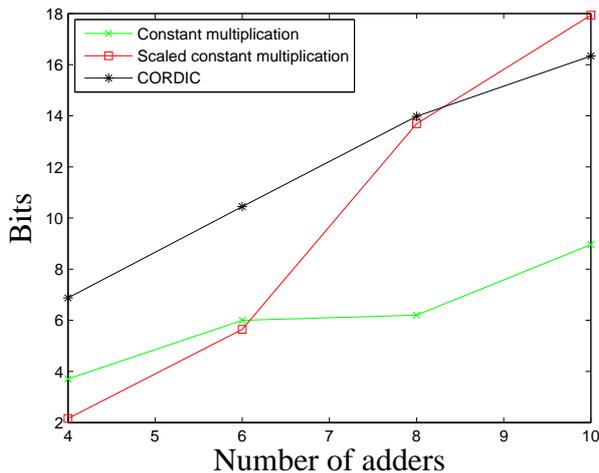


Fig. 4. Number of bits against adders for rotation  $\pi/8$ .

### E. Addition aware quantization

For the constant multiplication cases it is often possible to optimize the error with the same complexity compared with rounding by addition aware quantization [12]. In [12],  $E$  additional fractional bits are used to realize that there are exactly  $2^E$  different representable coefficients for which  $\epsilon \leq 2^{-(N+1)}$ , including the one obtained by rounding to  $N$  fractional bits.

These  $2^E$  combinations are searched for the best solution. For each precision requirement, the solution with smallest maximum quantization error among those solutions with the smallest addition count is selected.

## III. RESULTS

Figure 3 analyzes different alternatives for calculating a rotation by  $\pi/16$  radians. It shows the precision measured in terms of the number of bits as a function of the number of

adders. This precision in terms of correct fractional bits is defined as:

$$N_{\text{CFB}} = -\log_2 \epsilon, \quad (10)$$

where  $\epsilon$  is error of the rotation with respect to the ideal rotation. This error is caused by the finite arithmetic used in the hardware circuits. Specifically, in case of the constant multiplication and scaled constant multiplication it is due to the quantization of the sine and cosine coefficient of the angles. On the other hand, for the CORDIC algorithm the error is based on the fact that the angle cannot be exactly approximated with a finite sequence of micro-rotation.

The results for constant multiplication and scaled constant multiplication have been obtained by applying the addition aware quantization methodology [12]. For the case of the CORDIC, all the possible combinations of micro-rotations have been calculated considering  $\delta_i \in \{-1, 0, 1\}$ , and the sequence of micro-rotations that best approximate the angle has been chosen. From Fig. 3 it can be observed that the algorithm that best approximates the angle depends on the number of adders used for the rotation. In case of 4 or 6 adders are utilized, the best result is obtained by the CORDIC algorithm. However, if 8 or 10 adders are available, a better approximation can be carried out using the scaled constant multiplication method. This shows that none of the algorithms is better than the other ones in all circumstances.

Besides, Figs. 4 and 5 show the same analysis, for the angles  $\pi/8$  and  $3\pi/16$  respectively. In the first case, the CORDIC algorithm obtains the best results for 4, 6 and 8 adders, whereas the precision is higher for the scaled constant multiplication in case 10 adders are used. On the other hand, in the case of the angle  $3\pi/16$  depicted in Fig. 5 it can be observed that the CORDIC provides the most accurate results independently of the number of adders.

By comparing all three graphs it can be noted that the

TABLE I  
BEST CASES FOR VARYING NUMBER OF ADDERS

Rotation Angle	Number of Adders	Best Algorithm				
		Algorithm	Coefficients	Scaling factor	Error	Correct bits
$\pi/16$	4	CORDIC	00011	0.9903	$9.58 \cdot 10^{-3}$	6.705
	6	CORDIC	1101	0.6276	$1.05 \cdot 10^{-3}$	9.895
	8	SCM	1287, 256	0.1951	$0.021 \cdot 10^{-3}$	15.483
	10	SCM	658943, 131072	0.1951	$0.017 \cdot 10^{-3}$	15.820
$\pi/8$	4	CORDIC	01001	0.8927	$8.53 \cdot 10^{-3}$	6.873
	6	CORDIC	01001001	0.8927	$7.17 \cdot 10^{-4}$	10.445
	8	CORDIC	00110101	0.9622	$6.20 \cdot 10^{-5}$	13.977
	10	SCM	79109, 32768	0.3827	$3.98 \cdot 10^{-6}$	17.936
$3\pi/16$	4	CORDIC	0101	0.8875	$1.05 \cdot 10^{-3}$	9.895
	6	CORDIC	01010000001	0.8875	$6.95 \cdot 10^{-5}$	13.812
	8	CORDIC	010100000010001	0.8875	$8.42 \cdot 10^{-6}$	16.857
	10	CORDIC	010100000010001001	0.8875	$7.92 \cdot 10^{-7}$	20.267

optimum results are very dependent on the angle that must be rotated. Thus, for each single angle the algorithms should be evaluated in order to obtain the optimum case.

Table I summarizes the best results for each angle and each number of adders according to the graphs. In the Table it is indicated the best algorithm for each case, and how to calculate the rotation according to it. For the CORDIC algorithm the sequence of values  $\delta_i$  is provided for  $i = 0 \dots M$ , being  $\alpha_i = \tan^{-1}(2^{-i})$ . Note that  $\bar{1}$  is used for indicating  $\delta_i = -1$ . For the cases of scaled constant multiplication (SCM) the coefficient indicates how the constant value  $\cos \alpha / \sin \alpha$  is quantized. The hardware architecture that uses the indicated number of adders can be obtained from these values [10].

Moreover, the table shows the error of the approximation that leads to the precision bits, as well as the scaling factor. This scaling factor is equal to  $\sin \alpha$  for the SCM, as can be observed in equation (7), and it is obtained according to equation (5) for the CORDIC algorithm, but only considering the scaling of the micro-rotations that are carried out.

Finally, it has been shown that it is possible to implement a DCT architecture by performing the rotations  $\pi/16$ ,  $\pi/8$  and  $3\pi/16$  at the last stage of the algorithm [7]. This allows to incorporate the scaling of these rotations to the corresponding constant of the quantizer after the DCT, leading to savings in hardware. These are the three rotations that have been studied in depth in this paper. Consequently, in order to get an optimized hardware architecture for the computation of the DCT, the study presented in this paper allows to choose arbitrarily the number of adders for these rotations depending on the available hardware resources. Then the most efficient rotator that minimizes the error can be simply selected from Table I.

#### IV. CONCLUSIONS

In this work we considered low-complexity realization of constant angle rotators based on shifts, adders, and subtracters.

The results show that redundant CORDIC and scaled constant multiplication are providing the best results, depending on which angle is considered. It is also shown that the precision can vary several bits using the same number of adders and subtracters, and, hence, the correct choice of rotator architecture is crucial for a low-complexity realization.

#### REFERENCES

- [1] A.V. Oppenheim and R.W. Schaffer, *Discrete-Time Signal Processing*, Prentice Hall, 1989.
- [2] N. Ahmed, T. Natarajan and K.R. Rao, "Discrete cosine transform", *IEEE Trans. Comput.*, vol. C-23, pp. 90-93, Jan. 1974.
- [3] J.W. Cooley and J.W. Tukey, "An algorithm for the machine calculation of complex Fourier series", *Math. Comput.*, vol. 19, pp. 297-301, 1965.
- [4] C. Loeffler, A. Ligtenberg and G.S. Moschytz, "Practical fast 1-D DCT algorithms with 11 multiplications", *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'89*, vol. 2, pp. 988-991, Feb. 1989.
- [5] K.K. Parhi, *VLSI Digital Signal Processing Systems*, Wiley-Interscience, 1999.
- [6] Jack E. Volder, "The CORDIC Trigonometric Computing Technique," *IRE Trans. on Electronic Computing*, Sep. 1959.
- [7] Z. Wu, J. Sha, Z. Wang, L. Li and M. Gao, "An improved scaled DCT architecture," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 2, pp. 685-689, 2009.
- [8] M. Garrido and J. Grajal, "Efficient Memoryless CORDIC for FFT Computation," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 2, Apr. 2007, pp. II-113-116.
- [9] J.A. Lee and T. Lang, "Constant-factor redundant CORDIC for angle calculation and rotation", *IEEE Transactions on Computers*, vol. 41 no. 8, pp. 1016-1025, 1992.
- [10] O. Gustafsson, A. G. Dempster, K. Johansson, M. D. Macleod, and L. Wanhammar, "Simplified design of constant coefficient multipliers," *Circuits, Systems and Signal Processing*, vol. 25, no. 2, pp.225-251, Apr. 2006.
- [11] A. G. Dempster and M. D. Macleod, "Multiplication by two integers using the minimum number of adders," in *Proc. IEEE Int. Symp. Circuits Syst.*, Kobe, Japan, May 24-26, 2005, pp. 1814-1817.
- [12] O. Gustafsson and F. Qureshi, "Addition aware quantization for low complexity and high precision constant multiplication," *IEEE Signal Processing Letters*, vol. 17, no. 2, pp. 173-176, Feb. 2010.