

A UNIFIED DESIGN FRAMEWORK FOR VECTOR ROTATIONAL CORDIC FAMILY BASED ON ANGLE QUANTIZATION PROCESS

An-Yeu Wu

Dept. of Elec. Engin., National Taiwan Univ.,
Taipei, 106, Taiwan, R.O.C.

Cheng-Shing Wu

Dept. of Elec. Engin., National Central Univ.,
Chung-Li, 320, Taiwan, R.O.C.

ABSTRACT

Vector rotation is the key operation employed extensively in many digital signal processing applications. In this paper, we introduce a new design concept called *Angle Quantization (AQ)*. It can be used as a design index for vector rotational operation, where the rotational angle is known in advance. Based on the AQ process, we establish a unified design framework for cost-effective low-latency rotational algorithms and architectures. Several existing works, such as conventional CORDIC, AR-CORDIC, MVR-CORDIC, and EEAS-based CORDIC, can be fitted into the design framework, forming a *Vector Rotational CORDIC Family*. Based on the new design framework, we can realize high-speed / low-complexity rotational VLSI circuits, whereas without degrading the precision performance in fixed-point implementations.

1. INTRODUCTION

Vector rotation plays an important role in many digital signal processing (DSP) applications. It is extensively employed as the processing kernel in discrete orthogonal transformations (DCT, DST, and DFT), lattice-based (rotation-based) digital filtering, sinewave generation, and digital modulation/demodulation in communication systems. Let $[x_{in} \ y_{in}]^T$ and $[x_{out} \ y_{out}]^T$ denote the input and output vectors, respectively. Vector rotation of $[x_{in} \ y_{in}]^T$ by a rotational angle θ can be formulated as

$$\begin{bmatrix} x_{out} \\ y_{out} \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \cdot \begin{bmatrix} x_{in} \\ y_{in} \end{bmatrix}, \quad (1)$$

Figure 1 shows the direct implementation of Eq. (1). As one can see, the direct implementation is very area-consuming and low-speed when rotational operations are heavily utilized in VLSI circuits.

In this paper, we propose a novel framework to design high-speed/low-cost vector rotational VLSI circuits. Instead of performing quantization on the coefficient parameters ($\cos \theta$ and $\sin \theta$) in fixed-point implementation, the proposed design framework originates from the concept of *Angle Quantization (AQ)*. The AQ derives the name from the fact that we perform the quantization process on the rotational angle, θ , directly. That is, we *decompose* the original rotational angle θ into several *sub-angles*, θ_i 's. Then, we try to sum up those sub-angles to approximate the original angle as close as possible; or equivalently, we try to minimize the *angle quantization error*

$$\xi_m \triangleq \theta - \sum_{i=0}^{N_A-1} \theta_i, \quad (2)$$

where N_A denotes the number of sub-angles. The AQ process is demonstrated in Fig. 2(a). Based on the AQ process, The vector

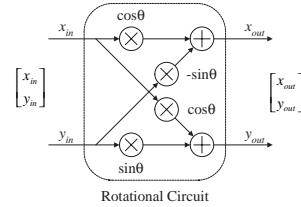


Figure 1: Direct implementation of rotational circuit.

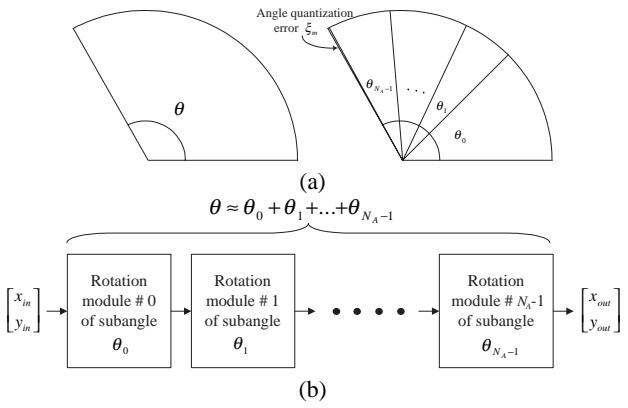


Figure 2: (a) Concept of *Angle Quantization*, where $\theta = (\theta_0 + \theta_1 + \dots + \theta_{N_A-1}) + \xi_m$, (b) Realization of fast vector rotation operation based on the AQ process.

rotation operation can be realized as shown in Fig. 2(b). Each rotation module is dedicated to performing a particular rotation of sub-angle θ_i . Then, the rotation of θ can be accomplished by cascading these N_A rotation modules.

In the AQ process, there are two key design issues:

1. Firstly, we need to *determine (or construct)* the sub-angles, and each θ_i needs to be easy-to-implement in practical VLSI circuits.
2. Secondly, we have to find out how to *select and combine* these sub-angles such that the angle quantization error ξ_m can be suppressed.

In fact, the well-known *COordinate Rotational Digital Computer (CORDIC)* algorithm [1] can be considered as an approach to perform the AQ process. Recall that in the CORDIC algorithm, the rotation of angle θ is performed by *sequentially* rotating elementary angle of $a(i) = \tan^{-1}(2^{-i})$, for $0 \leq i \leq W - 1$, where W denotes the wordlength. The advantageous feature of the elementary angle is that rotation of $a(i)$ requires only two shift-and-

add operators. The easy-to-implement feature of $a(i)$ conforms to the requirements of aforementioned AQ process. In addition, the sequential rotating operation of $a(i)$'s is the way to select and combine those sub-angles in conventional CORDIC.

Next, we can link the AQ process with several existing vector rotation schemes such as Angle Recoding (AR) technique [2], Modified Vector Rotational CORDIC (MVR-CORDIC) algorithm [3] and Extended Elementary Angle Set (EEAS) scheme [4]. We explore their relationship with the proposed AQ process. Then we will derive a unified framework for all these vector rotational operations. That is, all previous schemes can be considered as subsets of the proposed framework. The unified operations and AQ process of these algorithm suggest a family of rotation algorithms. We call it *Vector Rotational CORDIC Family*.

2. DESIGN FRAMEWORK FOR VECTOR ROTATIONAL OPERATIONS

2.1. Conventional CORDIC Algorithm

In conventional CORDIC algorithm, the *elementary angles*, $a(i)$, is defined as $a(i) \triangleq \tan^{-1}(2^{-i})$ [1]. Based on the elementary angles, the conventional CORDIC algorithm can be rewritten as

$$\theta = \sum_{i=0}^{N-1} \mu(i)a(i) + \xi_m, \quad (3)$$

where N denotes the number of elementary angles, $\mu(i) \in \{1, -1\}$ is the *rotation sequence* which determines the i^{th} rotational angle $a(i)$. In general, for data of W -bit wordlength, the iteration number is less than W , *i.e.*, $N \leq W$. Basically, the CORDIC tries to decompose the rotation angle, θ , into the combination of $a(i)$, for $i = 0, 1, \dots, N-1$. The *angle quantization error* of the CORDIC algorithm

$$\xi_{m, \text{CORDIC}} \triangleq \theta - \left[\sum_{i=0}^{N-1} \mu(i)a(i) \right], \quad (4)$$

represents the *residue angle* beyond the resolution of CORDIC algorithm.

2.1.1. Link AQ process with conventional CORDIC algorithm

Next, we would like to define *Elementary Angle Set (EAS)* for the derivation of the proposed vector rotational framework. Basically, EAS consists of all elementary angles used in the rotation algorithms. In the conventional CORDIC algorithm, the EAS comprises of all $a(i)$, for $0 \leq i \leq N-1$, and can be defined as

$$\mathcal{S} = \{a(i) : 0 \leq i \leq N-1\}. \quad (5)$$

With the help of EAS, we can say that the CORDIC algorithm essentially performs the angle quantization. This can be observed from Eq. (3). Given a target rotation angle θ , CORDIC algorithm determines the first rotation sequence $\mu(0)$ for the most significant elementary angle $a(0)$, followed by the determination of $\mu(1)$ for $a(1)$. The process is repeated until the last elementary angle is applied. That is, the CORDIC algorithm tries to perform the rotation through *sequentially* applying micro-rotations of *all* elementary angles.

Referring to Fig. 2, now we can relate AQ to CORDIC algorithm as follows: 1) The sub-angle θ_i in AQ now becomes $\theta_i = \mu(i)a(i)$ in CORDIC algorithm, 2) The number of sub-angles of N_A in AQ is set to be N in CORDIC algorithm, 3) CORDIC algorithm *sequentially* apply *all* θ_i , for $i = 0, 1, \dots, N-1$, to approximate the target angle θ .

2.2. AR Technique [2]

In conventional CORDIC algorithm, the micro-rotations of all elementary angles are performed in a sequential way. On the contrary, in the *Angle Recoding (AR)* technique proposed by Hu and Naganathan [2], certain micro-rotations can be *skipped* depending on the target rotational angle. Specifically, the modification is done by extending the set of $\mu(i)$ from $\{1, -1\}$ to $\{1, -1, 0\}$. One can skip the micro-rotation of the elementary angle $a(i) = \tan^{-1}(2^{-i})$ by setting $\mu(i) = 0$. Now, the angle quantization error of the AR technique, $\xi_{m, AR}$, can be represented as

$$\xi_{m, AR} \triangleq \theta - \left[\sum_{i=0}^{N-1} \mu(i)a(i) \right]. \quad (6)$$

Basically, Eq. (6) is identical to Eq. (4), except for the extended $\mu(i) \in \{1, -1, 0\}$.

2.2.1. Link AQ process with AR technique

To make AR technique fit into our design framework, we reformulate Eq. (6) in a compact form as

$$\xi_{m, AR} = \theta - \left[\sum_{j=0}^{N'-1} \tan^{-1}(\alpha(j) \cdot 2^{-s(j)}) \right] \triangleq \theta - \left[\sum_{j=0}^{N'-1} \tilde{\theta}(j) \right], \quad (7)$$

where $N' \triangleq \sum_{i=0}^{N-1} |\mu(i)|$, denotes the *effective iteration number*, $s(j) \in \{0, 1, \dots, N-1\}$ is the *rotational sequence* that determines the micro-rotation angle in the j^{th} iteration, $\alpha(j) \in \{-1, 0, 1\}$ is the *directional sequence* that controls the direction of the j^{th} micro-rotation of $a(s(j))$, and $\tilde{\theta}(j)$ is the j^{th} micro-rotation angle, defined as $\tilde{\theta}(j) \triangleq \tan^{-1}(\alpha(j) \cdot 2^{-s(j)})$.

As we can see from Eq. (7), the AR technique essentially tries to approximate θ with the combination of selected angle elements from a pre-defined elementary angle set (EAS). The EAS consists of all *possible* values of $\tilde{\theta}(j)$'s, and the EAS \mathcal{S}_1 used in AR technique can be represented as

$$\mathcal{S}_1 = \left\{ \tan^{-1}(\alpha^* \cdot 2^{-s^*}) : \alpha^* \in \{-1, 0, 1\}, s^* \in \{0, 1, \dots, N-1\} \right\}. \quad (8)$$

The use of the subscript 1 will become apparent later in this section. With the EAS \mathcal{S}_1 in hand, now we can easily link AR technique to the AQ process. By comparing Eq. (7) with the AQ approximation equation of Eq. (2), we find that AR technique indeed performs the angle quantization of target angle θ : The sub-angle θ_i now corresponds to $\tilde{\theta}(i) = \tan^{-1}(\alpha(i) \cdot 2^{-s(i)})$ and N_A is set to be N' .

2.2.2. Optimization Problem

We can consider the optimization problem of AR technique from EAS \mathcal{S}_1 point of view. It can be re-stated as: *Given θ , find the combination of elementary angles from EAS \mathcal{S}_1 , such that the angle quantization error $|\xi_{m, AR}| \leq a(N-1)$ and N' is minimized.*

In [2], the *Greedy algorithm* is proposed to solve the optimization problem.

2.3. MVR-CORDIC Algorithm [3]

Based on the AR technique, in the *Modified Vector Rotational CORDIC (MVR-CORDIC) algorithm* [3], two more modifications are proposed.

1. Repeat of elementary angles:

Referring to Eq. (6), in the AR technique, each micro-rotation angle of $a(i) = \tan^{-1}(2^{-i})$ is allowed to be used *only once*. However, in the MVR-CORDIC algorithm, each micro-rotation of elementary angle can be performed repeatedly. The relaxed operation can result in more possible combinations of elementary angles, hence, smaller ξ_m can be expected.

2. Confines of total micro-rotation number:

From Eq. (7), we can see that the effective iteration number N' in the AR technique is not fixed. For certain cases, N' is large and very close to the upper bound of $N/2$ [2]. In the MVR-CORDIC algorithm, we confine the iteration number in the micro-rotation phase to R_m ($R_m \ll W$). The role of R_m is quite similar to the number of non-zero digits, N_D , used in CSD recoding scheme; it will dominate the precision performance and the complexity.

2.3.1. Link AQ Process with MVR-CORDIC algorithm

Putting all the aforementioned modifications together and ignoring the null operations, we can represent the angle quantization error of the MVR-CORDIC algorithm as

$$\xi_{m,MVR} \triangleq \theta - \left[\sum_{i=0}^{R_m-1} \alpha(i) a(s(i)) \right], \quad (9)$$

where $s(i) \in \{0, 1, \dots, W-1\}$ is the *rotational sequence* that determines the micro-rotation angle in the i^{th} iteration, $\alpha(i) \in \{-1, 0, 1\}$ is the *directional sequence* that controls the direction of the i^{th} micro-rotation of $a(s(i))$. As one can find that the sub-angle of $(\alpha(i)a(s(i)))$ in Eq. (9) is exactly the same as the definition of $\tilde{\theta}(j)$ in Eq. (7). Hence, the EAS formed by MVR-CORDIC algorithm is the same as AR technique.

Based on the Eq. (9), it is obvious that MVR-CORDIC algorithm also performs the AQ process as well. The major difference is: 1) The total number of sub-angles N_A in Fig. 2 (*i.e.*, the total iteration number in the micro-rotation phase) is now kept fixed to a pre-defined value of R_m ($N_A = R_m$), 2) The sub-angle θ_i corresponds to $\alpha(i)a(s(i))$ in MVR-CORDIC algorithm, *i.e.*, $\theta_i = \alpha(i)a(s(i)) = \theta(i)$.

2.3.2. Optimization Problem

In the application of MVR-CORDIC algorithm, the optimization problem can be stated from EAS point of view as: *Given θ , find the combination of R_m elementary angles from EAS \mathcal{S}_1 , such that the angle quantization error $|\xi_{m,MVR}|$ is minimized.*

In [3], *Semi-greedy algorithm*, which can provide tradeoffs between computational complexity and performance, is proposed to solve the optimization problem.

2.4. Extended EAS-based CORDIC Algorithm [4]

In *Extended Elementary Angle Set (EEAS)*-based CORDIC algorithm [4], in addition to applying the relaxation on $\mu(i)$, we also relax the constraint of elementary angles by extending EAS \mathcal{S}_1 . Then, we can have more choices (elementary angles) in approximating the target angle θ . It is expectable that the angle quantization error ξ_m can be reduced correspondingly.

2.4.1. Extended EAS

By observing Eq. (8), we can see that the EAS \mathcal{S}_1 are comprised of *arctangent of single signed-power-of-two (SPT) term*. In the problem of SPT-based digital filter design, one effective way to increase the coefficient resolution (hence the filter performance) is to employ more SPT terms to represent the filter coefficients [5]. Motivated by this, we can easily extend the set by representing the elementary angles as the *arctangent of the sum of two SPT terms* [4]. That is,

$$\mathcal{S}_2 = \left\{ \tan^{-1} \left(\alpha_0^* \cdot 2^{-s_0^*} + \alpha_1^* \cdot 2^{-s_1^*} \right) : \alpha_0^*, \alpha_1^* \in \{-1, 0, 1\}, s_0^*, s_1^* \in \{0, 1, \dots, W-1\} \right\} \quad (10)$$

We call it *Extended Elementary-Angle Set \mathcal{S}_2 (EEAS \mathcal{S}_2)*. The subscript is used to denote the number of SPT terms.

Based on the EEAS \mathcal{S}_2 developed in Eq. (10), the sub-angle θ_i in Fig. 2 now can be represented as

$$\theta_i = \tan^{-1} \left(\alpha_0(j) \cdot 2^{-s_0(j)} + \alpha_1(j) \cdot 2^{-s_1(j)} \right), \quad (11)$$

and the number of sub-angles N_A is set to be R_m .

2.4.2. Optimization Problem

With the derived EEAS \mathcal{S}_2 , now the optimization problem of the EEAS-based CORDIC algorithm can be stated as: *Given θ and R_m , find the parameters of $\alpha_0(j)$, $\alpha_1(j)$, $s_0(j)$ and $s_1(j)$ (*i.e.*, the combination of elementary angles from EEAS \mathcal{S}_2), such that the angle quantization error*

$$|\xi_{m,EEAS}| \triangleq \left| \theta - \sum_{j=0}^{R_m-1} \tan^{-1} \left(\alpha_0(j) \cdot 2^{-s_0(j)} + \alpha_1(j) \cdot 2^{-s_1(j)} \right) \right|, \quad (12)$$

can be minimized.

In [6], a novel searching algorithm, called *Trellis-based Searching (TBS) algorithm*, is proposed to solve the optimization problem.

2.5. Generalized EEAS Scheme

By following the similar idea of EEAS scheme, it is straightforward to insert more SPT terms in the representation of elementary angles. Hence, the size of EEAS can be increased. With more than two SPT terms, we call such an extension scheme *Generalized EEAS Scheme*. Specifically, the generalized EEAS with d SPT terms can be represented as

$$\mathcal{S}_d = \left\{ \tan^{-1} \left(\alpha_0^* \cdot 2^{-s_0^*} + \dots + \alpha_{d-1}^* \cdot 2^{-s_{d-1}^*} \right) \right\}, \quad (13)$$

where $\alpha_0^*, \dots, \alpha_{d-1}^* \in \{-1, 0, 1\}$, $s_0^*, \dots, s_{d-1}^* \in \{0, \dots, W-1\}$. As one can expect that the size of the EEAS increases exponentially as d increases. Consequently, with properly chosen design parameters, we can achieve higher precision performance in the AQ process.

Vector Rotation Algorithms	Selection of Rotation Sequence	Elementary Angle Set	Micro-rotations	Angle Quantization	
				θ_i	N_A
Conventional CORDIC Algorithm	$\mu \in \{-1,1\}$	$EAS \mathcal{S}$	Complete	$\theta_i = \mu(i)a(i)$	W Fixed
Angle Recoding Technique	$\mu \in \{-1,0,1\}$	$EAS \mathcal{S}_1$	Selective	$\theta_i = \tan^{-1}(\alpha(i) \cdot 2^{-s(i)})$	N' Varialbe
MVR-CORDIC Algorithm	$\alpha \in \{-1,0,1\}$	$EAS \mathcal{S}_1$	Selective	$\theta_i = \tan^{-1}(\alpha(i) \cdot 2^{-s(i)})$	R_m Fixed
EEAS Scheme	$\alpha_0, \alpha_1 \in \{-1,0,1\}$	$EEAS \mathcal{S}_2$	Selective	$\theta_i = \tan^{-1}(\alpha_0(i) \cdot 2^{-s_0(i)}) + \alpha_1(i) \cdot 2^{-s_1(i)}$	R_m Fixed
Generalized EEAS Scheme	$\alpha_0, \alpha_1, \dots, \alpha_{d-1} \in \{-1,0,1\}$	$EEAS \mathcal{S}_d$ $d \geq 3$	Selective	$\theta_i = \tan^{-1}(\alpha_0(i) \cdot 2^{-s_0(i)}) + \dots + \alpha_{d-1}(i) \cdot 2^{-s_{d-1}(i)}$	R_m Fixed

Table 1: Comparisons of members in the Vector Rotational CORDIC family.

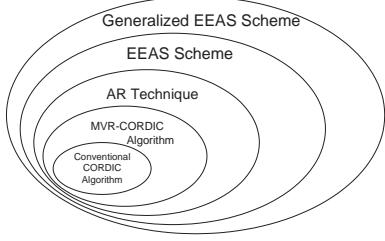


Figure 3: Set diagram of vector rotational CORDIC family.

2.6. Family of Vector Rotational CORDIC Algorithm

So far, we have linked the AQ process with several existing vector rotation approaches, including CORDIC algorithm, Angle Recoding technique, MVR-CORDIC algorithm, EEAS scheme, and Generalized EEAS scheme. All algorithms intend to realize the AQ process with various EAS and suitable combinations of sub-angles. That is, they try to decompose the target rotational angle into several easy-to-implement sub-angles, while minimizing the angle quantization error ξ_m to obtain the best precision performance.

Based on our discussion, now we can link all these rotation algorithms together under a unified design framework, from the AQ point of view. They form a family of vector rotational CORDIC algorithm, called *Vector Rotational CORDIC Family*. They all conform to the AQ process, but each rotational algorithm uses different AQ setting as summarized in Table 1.

Note that EEAS scheme covers MVR-CORDIC algorithm and AR technique due to the fact that MVR-CORDIC and AR employ EAS \mathcal{S}_1 as a searching space that is a subset of EEAS \mathcal{S}_2 . Moreover, MVR-CORDIC algorithm can also be treated as a subset of AR technique due to the fact that we impose one constraint on the total iteration number. Fig. 3 illustrates the relationships among members of vector rotational CORDIC family.

3. DESIGN EXAMPLE

In the design example, we consider the rotation angle of $\theta = 13\pi/32$. All algorithms in vector rotational CORDIC family derived in Section 2 are applied to perform the rotation of θ . Meanwhile, aforementioned searching algorithms are conducted to solve

Rotation Angle $\theta = 13\pi/32$	Searching Algorithm	Combination Type	Rotational Sequence μ and α and Subangle Index s	Angle Approx. Error ξ_m
Conventional CORDIC Algorithm	-----	1	$\bar{\mu} = [1 \ -1 \ 1 \ 1 \ -1 \ 1 \ -1 \ -1 \ 1 \ -1 \ 1 \ 1]$	$.1608 \times 10^{-5}$
Angle Recoding Technique	Greedy Algorithm	2	$\bar{\mu} = [1 \ 0 \ 0 \ -1 \ 0 \ 0 \ -1 \ -1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1]$	$.0110 \times 10^{-5}$
	Greedy Algorithm	3	$\bar{\alpha} = [1 \ -1 \ -1 \ -1]$ $\bar{s} = [0 \ 3 \ 6 \ 7]$	5.2891×10^{-4}
	Semi-greedy Algorithm ($D = 2$)	4	$\bar{\alpha} = [1 \ -1 \ -1 \ 1]$ $\bar{s} = [0 \ 3 \ 5 \ 7]$	5.2033×10^{-4}
	TBS Algorithm	5	$\bar{\alpha} = [1 \ 1 \ -1 \ -1]$ $\bar{s} = [1 \ 2 \ 4 \ 7]$	2.5911×10^{-4}
EEAS Scheme with $R_m = 2$	Greedy Algorithm	6	$\bar{\alpha}_0 = [1 \ -1]$ $\bar{\alpha}_1 = [-1 \ -1]$ $\bar{s}_0 = [0 \ 2]$ $\bar{s}_1 = [8 \ 10]$	4.8233×10^{-4}
	TBS Algorithm	7	$\bar{\alpha}_0 = [1 \ 1]$ $\bar{\alpha}_1 = [-1 \ -1]$ $\bar{s}_0 = [0 \ 6]$ $\bar{s}_1 = [3 \ 5]$	$.7196 \times 10^{-5}$
EEAS Scheme with $R_m = 3$	TBS Algorithm	8	$\bar{\alpha}_0 = [1 \ -1 \ -1]$ $\bar{\alpha}_1 = [1 \ -1 \ 1]$ $\bar{s}_0 = [0 \ 3 \ 7]$ $\bar{s}_1 = [15 \ 6 \ 12]$	3.2503×10^{-7}

Table 2: Design example of rotation angle $\theta = 13\pi/32$, where the wordlength $W = 16$.

the optimization problems. The results are summarized in Table 2.

4. CONCLUSION

In this paper, we introduce a new design index, called Angle Quantization. Following the new index, designers can explore a bigger design space in deriving low-cost/high-performance rotational circuits. As illustrated in [7], most popular DSP algorithms can be realized via rotational circuits. The new framework proposed in this paper can be employed to design the processing kernel of the DSP engine in [7].

References

- [1] Y. H. Hu, "CORDIC-based VLSI architectures for digital signal processing," *IEEE Signal Processing Magazine*, pp. 16–35, July 1992.
- [2] Y. H. Hu and S. Naganathan, "An angle recoding method for CORDIC algorithm implementation," *IEEE Trans. on Computers*, vol. 42, pp. 99–102, Jan. 1993.
- [3] C. S. Wu and A. Y. Wu, "Modified vector rotational CORDIC (MVR-CORDIC) algorithm and its application to FFT," in *Proc. IEEE Int. Symp. Circuits and Systems*, pp. 529 –532, 2000.
- [4] C. S. Wu and A. Y. Wu, "A novel rotational VLSI architecture based on Extended Elementary-angle Set CORDIC algorithm," in *Proc. IEEE 2nd IEEE Asia Pacific Conference on ASICs*, (Cheju, Korea), pp. 111–114, 2000.
- [5] H. Samueli, "An improved search algorithm for the design of multiplierless FIR filters with power-of-two coefficients," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 1044–1047, July 1989.
- [6] C. S. Wu and A. Y. Wu, "A new Trellis-based searching scheme for EEAS-based CORDIC algorithm," submitted to *IEEE Int. Conf. Acoust. Speech, Signal Processing*, Salt Lake City, 2001.
- [7] A. Y. Wu, K. J. R. Liu, and A. Raghupathy, "System architecture of an adaptive reconfigurable DSP computing engine," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 54–73, Feb. 1998.