Concept, Design, and Implementation of Reconfigurable CORDIC

Abstract:

This brief presents the key concept, design strategy, and implementation of reconfigurable coordinate rotation digital computer (CORDIC) architectures that can be configured to operate either for circular or for hyperbolic trajectories in rotation as well as vectoring-modes. It can, therefore, be used to perform all the functions of both circular and hyperbolic CORDIC. We propose three reconfigurable CORDIC designs: 1) a reconfigurable rotation-mode CORDIC that operates either for circular or for hyperbolic trajectory; 2) a reconfigurable vectoring-mode CORDIC for circular and hyperbolic trajectories; and 3) a generalized reconfigurable CORDIC that can operate in any of the modes for both circular and hyperbolic trajectories. The reconfigurable CORDIC can perform the computation of various trigonometric and exponential functions, logarithms, square-root, and so on of circular and hyperbolic CORDIC using either rotation-mode or vectoring-mode CORDIC in one single circuit. It can be used in digital synchronizers, graphics processors, scientific calculators, and so on. It offers substantial saving of area complexity over the conventional design for reconfigurable applications. The proposed architecture of this paper analysis the logic size, area and power consumption using Xilinx 14.2.

Enhancement of the project:

Existing System:

The rotation-mode CORDIC determines the coordinates of any given vector after rotation through a given angle, while in the vectoring-mode it computes the magnitude and phase of the vector. The unified algorithm for linear and hyperbolic CORDICs is an extension of the basic CORDIC algorithm for circular trajectory. It is based on the generalized principle proposed to include hyperbolic and linear trajectories along with the original circular trajectory of operation. A variable m is introduced to modify the rotation matrix and elementary angles as
Where \( m = 1 \) for circular, \( m = 0 \) for linear, and \( m = -1 \) for hyperbolic trajectories, respectively. To guarantee the convergence in hyperbolic mode, the iterations \( i = 4, 13, 40,... \) need to be executed twice. It is shown that the scale-factor converges to a constant \( K = 1.2075 \) (0.60725) for hyperbolic (circular) trajectory. Consequently, we have different scale-factors for the circular and hyperbolic trajectories. The unified CORDIC algorithm supports a range of convergence (RoC) of \([-99^\circ, 99^\circ]\) for circular trajectory, while the RoC for hyperbolic trajectory is \(|\theta| \leq 1.1182 \text{ radians (nearly } 64^\circ)\).

Realization of Vectoring-Mode CORDIC

In vectoring-mode, the phase and magnitude of any given vector are computed by aligning the vector along x-axis. This is achieved by performing micro rotations in the direction, which drives the y-coordinate to zero. Equations (1) and (2) are extended to vectoring-mode by changing the control variable that determines the direction. In rotation-mode, the residual angle \( \theta_i \) acts as the control variable for determining the direction of micro rotations, while in vectoring-mode, the control variable is y-coordinate. If \( y_i \) is positive, the direction of micro rotation is clockwise, while it is anticlockwise if \( y_i \) is negative.

Disadvantages:

- Area coverage is high

Proposed System:

To design a reconfigurable CORDIC architecture with minimum reconfiguration overhead, we need to maximize the sharing of common hardware circuit in different configurations. Therefore, to explore the possibility of reconfigurable CORDIC, we examine, here, the commonalities in three main issues of CORDIC implementation, namely:

1) The coordinate-rotation matrix.
2) Selection of elementary angles.
3) Direction of micro rotations.

Reference Reconfigurable CORDIC

A basic design for reconfigurable CORDIC based on unified CORDIC algorithm was proposed. The major concern with the design of conventional reconfigurable architecture is the incompatibility in RoC of circular and hyperbolic trajectories. The RoC of circular CORDIC is $[-99^\circ, 99^\circ]$, while that of hyperbolic CORDIC is given by $|\theta| \leq 1.1182$ radians. This limits the maximum angle of rotation of the reconfigurable design to $64^\circ$. The incompatible RoC of circular and hyperbolic CORDICs makes it difficult to implement them in the same circuit to perform rotation through $[-180^\circ, 180^\circ]$.

Design Strategy for Proposed Reconfigurable CORDIC

As discussed in Reference Reconfigurable CORDIC, the circular and hyperbolic CORDICs require two different scaling circuits, which is quite costly. Therefore, it is necessary to use a scale-free implementation in the reconfigurable CORDIC. Here, we discuss the scaling-free CORDIC and its limitations, followed by the discussions on our design strategy for a reconfigurable CORDIC.

PROPOSED RECONFIGURABLE CORDIC

The coordinate calculation matrices for circular and hyperbolic CORDICs differ by the sign of operands, and to realize that additions are to be replaced by subtractions and vice-versa. This can be easily realized by a reconfigurable add/subtract circuit. In both cases, the basic-shift could be either 2 or 3, but the number of micro rotations varies with the mode of operation. Besides, each case will have its own circuit to enable the extension of RoC. Based on these observations, we design three reconfigurable CORDIC architectures:

1) Rotation-mode reconfigurable CORDIC.
2) Vectoring-mode reconfigurable CORDIC.
3) Generalized reconfigurable CORDIC.
A) Rotation-Mode Reconfigurable CORDIC

\[ \theta \in [-\pi, \pi] \]

![Diagram](image)

Fig. 1. Proposed reconfigurable rotation-mode CORDIC processor

The proposed design for reconfigurable rotation-mode CORDIC (shown in Fig. 1) consists of three parts:

1) Preprocessing unit.
2) Reconfigurable CORDIC rotation unit.
3) Post processing unit.

![Diagram](image)

Fig. 2. Structure of the proposed reconfigurable recursive CORDIC architectures.

1) Proposed Recursive Architecture: The recursive architecture (shown in Fig. 2) uses a single CORDIC micro rotator to perform all the CORDIC iterations. The reconfigurable coordinate calculation unit (RCCU) is shown in Fig. 3.

Copyright © 2016LeMenizInfotech. All rights reserved
2) Proposed Pipelined Architecture: Fig. 4 shows the reconfigurable CORDIC rotation unit for basic-shift 2. The shift-index \( s_i \) is fixed in every RCCU, and hence the shifters are hardwired and do not involve high complexity barrel-shifters. The implementation of RCCUs varies according to the basic-shift \( s_i \). With slight modifications, the pipeline can be extended for basic-shift 3.

B. Reconfigurable Vectoring-Mode CORDIC

By changing the implementation of the RCCU to implement (8), the recursive architecture of Fig. 2 can be used to realize CORDIC iterations for vectoring-mode. The rollover counter value is 15 for \( s_{\text{basic}} = 2 \), and 17 for \( s_{\text{basic}} = 3 \).
The pipelined architecture of vectoring-mode reconfigurable CORDIC consists of eight stages for $s_{basic} = 2$, as shown in Fig. 5. Similar to reconfigurable rotation-mode CORDIC, for increasing shift-indices, the implementation of RCCUs is simplified for reconfigurable vectoring-mode CORDIC as well.

Fig. 5. Proposed pipeline reconfigurable vectoring-mode CORDIC unit for $s_{basic} = 2$.

Proposed Generalized Reconfigurable CORDIC

The generalized reconfigurable CORDIC can operate either in vectoring-mode or in rotation-mode for both circular and hyperbolic trajectories. The user can select the trajectory of operation using a single bit signal $T$. 
Fig. 6. Structure of CORDIC microrotator for the proposed recursive generalized reconfigurable CORDIC.

The recursive architecture of the proposed generalized reconfigurable CORDIC is implemented by combining the CORDIC microrotators for both rotation-mode and vectoring-mode CORDICs, as shown in Fig. 6. The throughput of the proposed recursive generalized reconfigurable CORDIC is the same as that of the recursive reconfigurable vectoring-mode CORDIC.
The block diagram for pipelined generalized reconfigurable CORDIC using basic-shift $s_{\text{basic}} = 2$ is shown in Fig. 7. It can be easily extended to basic-shift $s_{\text{basic}} = 3$ as is done for reconfigurable rotation-mode and vectoring-mode CORDICs.

**Advantages:**

- Reduce the area
- Maximum operating frequency

**Software implementation:**

- Modelsim
- Xilinx ISE