International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 14
October2016

CORDIC Designs for Fixed Angle of Rotation.

Nitesh Kumar
Vidya Jyoti Institute of Technology, Hyderabad
nitesh.20106@gmail.com

S. Upendra
ASSOCIATE PROFESSOR
Vidya Jyoti Institute of Technology, Hyderabad

Abstract

Rotation of vectors through fixed and known angles has wide applications in robotics, digital signal processing, graphics, games, and animation. But, we do not find any optimized coordinate rotation digital computer (CORDIC) design for vector-rotation through specific angles. Therefore, in this paper, we present optimization schemes and CORDIC circuits for fixed and known rotations with different levels of accuracy. For reducing the area- and time-complexities, we have proposed a hardwired pre-shifting scheme in barrel-shifters of the proposed circuits. Two dedicated CORDIC cells are proposed for the fixed-angle rotations. In one of those cells, micro-rotations and scaling are interleaved, and in the other they are implemented in two separate stages. Pipelined schemes are suggested further for cascading dedicated single-rotation units and bi-rotation CORDIC units for high-throughput and reduced latency implementations. We have obtained the optimized set of micro-rotations for fixed and known angles. The optimized scale-factors are also derived and dedicated shift-add circuits are designed to implement the scaling. The fixed-point mean-squared-error of the proposed CORDIC circuit is analyzed statistically, and strategies for reducing the error are given. We have synthesized the proposed CORDIC cells by Synopsys Design Compiler using TSMC 90-nm library, and shown that the proposed designs offer higher throughput, less latency and less area-delay product than the reference CORDIC design for fixed and known angles of rotation. We find similar results of synthesis for different Xilinx field-programmable gate-array platforms.

Introduction

Digital signal processing (DSP) algorithms exhibit an increasing need for the efficient implementation of complex arithmetic operations.

The computation of trigonometric functions, coordinate transformations or rotations of complex

valued phasors is almost naturally involved with modern DSP algorithms. Popular application examples are algorithms used in digital communication technology and in adaptive signal processing. While in digital communications, the straightforward evaluation of the cited functions is important, numerous matrix based adaptive signal processing algorithms require the solution of systems of linear equations, QR factorization or the computation of eigen values, eigenvectors or singular values. All these tasks can be efficiently implemented using processing elements performing vector rotations.

The COordinate Rotation DIgital Computer algorithm (CORDIC) gives the opportunity to calculate the desired functions in rather simple, and elegant way. CORDIC is a method for computing elementary functions using minimal hardware such as shift's, adds/subs and compares. CORDIC works by rotating the coordinate system through constant angles until the angle is reduces to zero. The angle offsets are selected such that the operations on X and Y are only shifts and adds.

Design Methodology

A hardware circuit can be described in two ways.

- Structural description In this method, the circuit is described as an interconnection of known components.
- Behavioral description In this method, the behavior of the circuit is described by means of Boolean equations and a set of sequential instructions. When a designer is handed over the specifications of a new product, he can start the design using one of the following approaches

Design process

Digital system design starts from the user specifications. The specifications define the terminal behavior of the proposed system and provide a natural language description of how the input signals are transformed into outputs. Translation from this stage to siliconis a giant step. To make it manageable, it has to be broken down into several smaller steps.

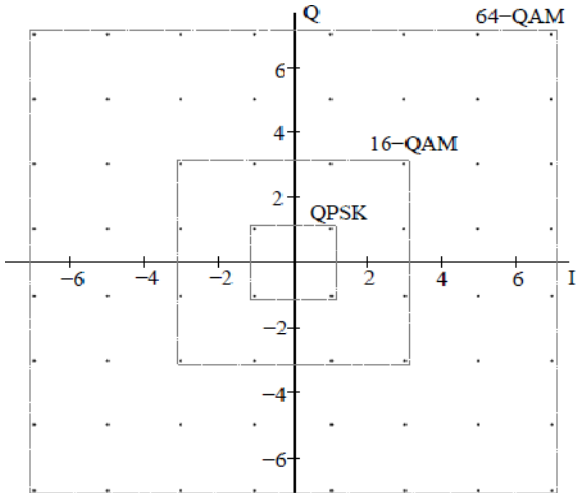
Analysis

First, we need to design the system architecture, identify major blocks and define their interface signals. This is a job, which requires the creative abilities of the right hemisphere of the human brain. We can't expect any computer support at this stage apart from some drawing tools. Having done this, the circuit behavior can be transcribed into a machine-readable form using the behavioral description style of VHDL. At this stage, the emphasis is on obtaining the terminal behavior without bothering about how the circuit will be realized. This description serves as a test environment for part-by-part testing of individual subsystem realizations at lower levels of abstractions. Then each subsystem is rewritten in Register Transfer Level (RTL) form. In this form, the system is described as a set of registers and ALU's interconnected by set of data buses.

The selection of registers on data buses, function selection of ALU and storage of bus contents into selected registers is done in a sequential manner using the control signals generated by a control unit. Today's synthesis tools can understand VHDL code written in RTL style. From this point onwards, it is only the mechanical work of optimization and mapping. If done manually, it is done with the left hemisphere of the brain. This is the area that is most suitable for computerization.

Wireless LAN

Wireless LAN is specified with IEEE 802.11 standard. It was accepted in 1999 and led to organization of local networks development. WLAN is a flexible data communications system implemented to extend or substitute for, a wired LAN. Radio frequency (RF) technology is used by a wireless LAN to transmit and receive data over the air, minimizing the need for wired connections. A WLAN enables data connectivity and user mobility.



DESIGN OF AN CORDIC ALGORITHM

CORDIC (Coordinate Rotation Digital Computer) is a method for computing elementary functions using minimal hardware such as shift's, adds/subs and compares. CORDIC works by rotating the coordinate system through constant angles until the angle is reduces to zero. The angle offsets are selected such that the operations on X and Y are only shifts and adds.

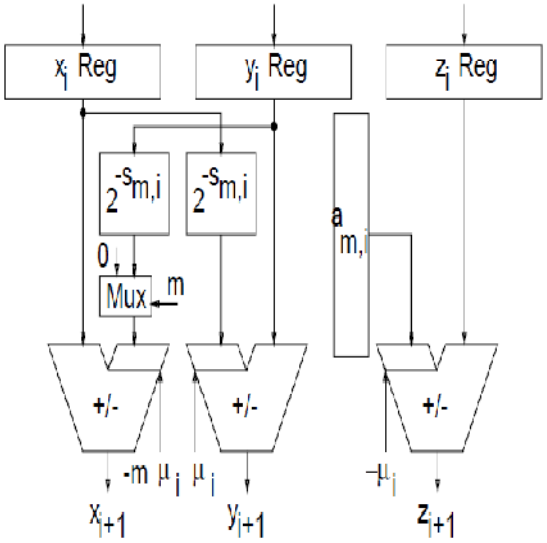
The CORDIC Algorithm is a unified computational scheme to perform

- Computations of the trigonometric functions: *sin*, *cos* and *arctan*
- Computations of the hyperbolic trigonometric functions: *sinh, cosh* and *arctanh*
- and consequently can also compute the exponential function, the natural logarithm and the square root
- Multiplication and division.

CORDIC revolves around the idea of "rotating" the phase of a complex number, by multiplying it by a succession of constant values. However, the "multiplies" can all be powers of 2, so in binary arithmetic they can be done using just shifts and adds; no actual "multiplier" is needed.

Architecture for CORDIC algorithm

The basic architecture for the implementation of the CORDIC algorithm for the computation of sine and cosine functions is shown in below figure. This architecture is simplified according to the original CORDIC algorithm for hardware implementation. It uses only 3 adder/subtractor blocks, 2 shifters, 1 counter, and a ROM table for storing the values of the tangents of the angles. The precision is determined by the maximum count of *i* variable.

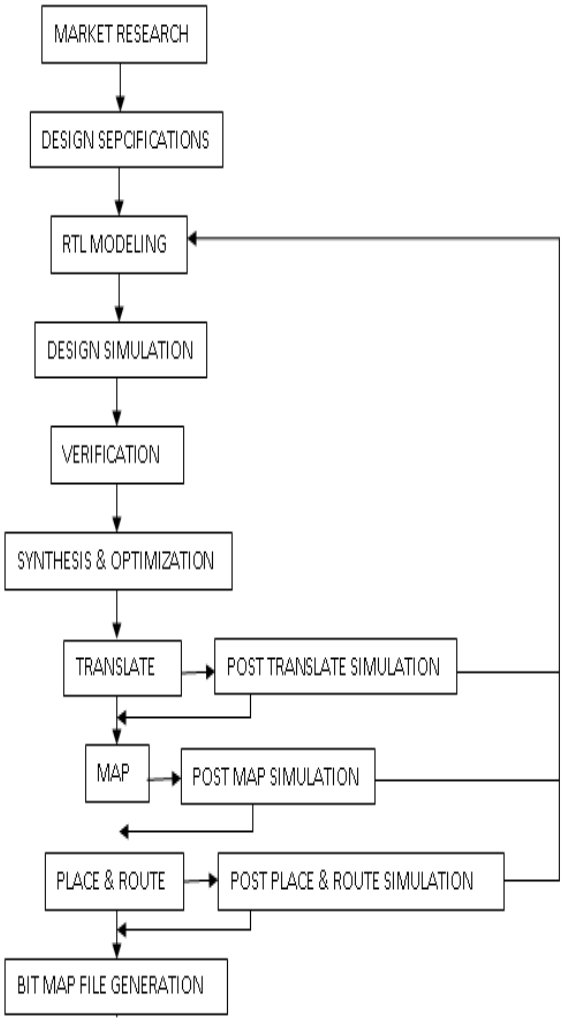


An adder/subtractor (A/S), depending on a selection input, performs an addition or a subtraction. This input indicates whether an operand is negative. The basic cell of A/S is decomposed by two functions with 4 bits input each. One of them is for calculating the output and another to transmit the carry. According to this an N-bit A/S can fit in (2N+1)/2 CLB's (configurable logic block). The additional half CLB is required for introducing the least significant bit (LSB) one in case of the substraction. The critical path here is indicated by the ripple carry propagation and the routing delay of the A/S wire. This net has a fan-out of 2N in this case. It decreases the performance of the circuit and it is the main disadvantage of conventional CORDIC implementations. As the solution to this, redundant arithmetic could be used to increase the speed of the CORDIC. Implementation avoids the carry propagation from the LSB to the most significant bit (MSB), due to its carry-free property. Redundant arithmetic is good to accelerate those operations, which have a long propagation delay. On the other hand redundant arithmetic also has some disadvantages. For example, it is impossible to detect

the sign of a redundant number without checking all the digits which expects propagation from the MSB to the LSB. Another problem, that the redundant arithmetic

VLSI Implementation Flow for CORDIC

The generalized implementation flow diagram of the project is represented as follows.



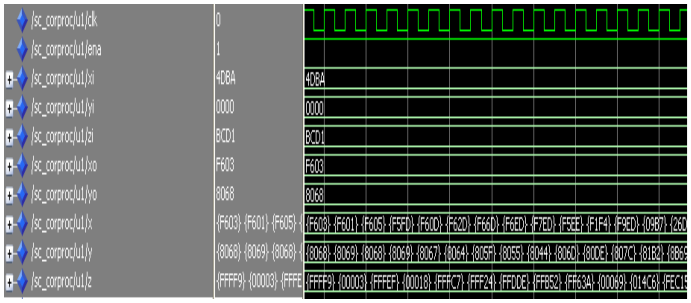
RESULTS AND DISCUSSION

The CORDIC algorithm and its functionality were discussed in the previous chapters. Now this chapter deals with the simulation and synthesis results of the CORDIC. Here Modelsim tool is used in order to simulate the design and checks the functionality of the design. Once the functional verification is done, the design will be taken to the Xilinx tool for Synthesis process and the netlist file generation.

The Appropriate test cases have been identified in order to test this modelled CORDIC design. Based on the identified values, the simulation results which describes the operation of the CORDIC has been achieved. This proves that the modelled design works properly as per the process

Simulation Results

The test bench is developed in order to test the modelled design. This developed test bench will automatically force the inputs and will make the operations of CORDIC.



Introduction to FPGA

FPGA stands for Field Programmable Gate Array which has the array of logic module, I /O module and routing tracks (programmable interconnect). FPGA can be configured by end user to implement specific circuitry. Speed is up to 100 MHz but at present speed is in GHz.

Main applications are DSP, FPGA based computers, logic emulation, ASIC and ASSP. FPGA can be programmed mainly on SRAM (Static Random Access Memory). It is Volatile and main advantage of using SRAM programming technology is re-configurability. Issues in FPGA technology are

complexity of logic element, clock support, IO support and interconnections (Routing).

VHDL LANGUAGE BASICS

VHDL is an acronym for Very High Speed Integrated Circuits Hardware Description Language. The language can be used to model a digital system at many levels of abstraction ranging from the algorithmic level to the Gate level. The complexity of the digital system being modeled would vary from that of a simple gate to a complete digital electronic system. The VHDL Language can be regarded as an integrated amalgamation of sequential language, concurrent language, net list language, timing specifications and waveform generation language.

Design Units in VHDL

VHDL provides five different types of primary constructs called design units. They are:

1. Entity declaration
2. Architecture body
3. Configuration declaration
4. Package declaration
5. Package body

An entity is modeled using an entity declaration and at least one architecture body (A hardware abstraction of the digital system is called an entity). The Entity declaration specifies the name of the entity being modeled and lists the set of interface ports. Ports are signals through which the entity communicates with the other models in its external environment. So entity declaration describes the external view of the entity.

CONCLUSION AND FUTURE SCOPE

In this work regarded to CORDIC implementation in VHDL there has been done a specific research on angle calculation in wireless LAN receiver block. As a result of this research there has been found vector and angle widths necessary and sufficient for an implementation and the number of CORDIC steps for calculating an arctangent function. Then two programs have been written on VHDL language: one for an interleaving and the other for pipelining realizations, and there is found that a pipelining implementation is better than an interleaving because it saves hardware. Both programs have been implemented for FPGA XC3S500E for further implementation.

CORDIC processor was extensively tested with various inputs and result verified. The iteration equations for xi and yi are equal to the usual algorithm. In order to obtain sign(z1) a single initial ripple propagation from MSD to LSD has to be taken into account for the MSD first absolute value calculation. The sign of the iteration variable zi is achieved by differential decoding the sign of zi given the initial sign sign(z0). A negative sign of zi corresponds to a sign change for zi. The successive signs are calculated with a small bit level propagation delay. The resulting parallel architecture for the CORDIC rotation mode is implemented successfully. Compared to the sign estimation approaches a clear advantage is given by the fact that no additional iterations are required for the CORDIC implementation.

Future Scope

In recent research it is proposed to reduce the number of CORDIC iterations by replacing the second half of the iterations with a final multiplication. Further low latency CORDIC

algorithms were derived for parallel implementation of the rotation mode and for word serial recursive implementations of both rotation mode and vectoring mode. Recently, a family of generalized multi dimensional CORDIC algorithms, so called Householder CORDIC algorithms, was derived. Here, a modified iteration leads to scaling factors which are rational functions instead of square roots of rational functions as in conventional CORDIC. This attractive feature can be exploited for the derivation of new architectures.

REFERENCES

[1] Volder, J., "The CORDIC Trigonometric Computing Technique," IRE Tran Electronic Computing, Vol EC-8, pp330-334 Sept 1959.

[2] D.H. Daggett, "Decimal-Binary Conversion in CORDIC", IRE Trans Electronic Computing, Vol EC-8, pp335-339, Sept 1959.

[3] Walther, J.S., "A unified algorithm for elementary functions," Spring Joint ComputerConf., 1971, proc., pp379-385

[4] Andraka, Ray, "A survey of CORDIC algorithms for FPGA based computers,"Proceedings of the 1998 ACM/SIGDA sixth international symposium on Field programmable gate arrays, Feb. 22-24, 1998, Monterey, CA. pp191-200.

[5] Richard Herveille, OPENCORES.ORG, "Cordic Core Specification," Rev. 0.4, December 18,2001.

[6] J. BHASKER, "A VHDL PREMIER 3rd Edition", Prentice Hall PTR 1999.

[7] Xilinx Application Note (XAPP 029) October 27, 1997 (version 1.1), "Serial CodeConversion between BCD and Binary"