International Journal of Engineering and Technical Research (IJETR)
ISSN: 2321-0869, Volume-2, Issue-4, April 2014

Precision Robotic Displacement Using FPGA

Aswathy Surendran.P, V.Saminathan, M.Udhayakumar

Abstract— **Rotation of vectors through fixed and known angles has wide applications in robotics, digital signal processing, graphics, games, and animation. For reducing the area and time complexities, the proposed hardwired pre-shifting scheme in barrel-shifters are implemented. Pipelined schemes are suggested for CORDIC units for high-throughput and reduced latency implementations. It is executed merely by table look-up, shift, and addition operations. By decomposing an arbitrary rotation angle into a sequence of coarse angles and a fine angle which was small enough, this architecture could be implemented by performing a sequence of shift-and-add operations in the radix-2 system without any ROM lookup table or real multiplication requirement. It was suitable to be designed in pipelined architecture for performing the high-speed operations Thus, the corresponding hardware can be implemented in very economic fashion. In this paper CORDIC fixed angle of rotation is implemented in Xilinx FPGA platform.**

Index Terms—CORDIC, LUT, PWM, Vector Rotation, Xilinx FPGA

I. INTRODUCTION

The calculus courses provide us with tools to compute the values of trigonometric functions, for example, via series expansions, polynomial, and rational function approximations. However, these implementations tend to require multiplication and division operations that make them expensive in hardware .In contrast, CORDIC algorithms need only adders, shifters and comparators for computing a wide range of elementary functions. The method is especially efficient when fixed point implementations of signal processing algorithms on hardware are considered. For example, CORDIC is extremely popular in hardware accelerators and also in SIMD realizations. Furthermore, almost all function calculators employ CORDIC. Here the existing method of angle rotation along with the modified method is compared along with the simulation results.

II. CORDIC ALGORITHM

In this section, the basic fundamentals of the CORDIC algorithm are discussed. The co-ordinate representation of the corresponding angle ϕ is explained in detail with respect to the x and y co-ordinates.[1]

Manuscript received April 19, 2014.

Aswathy Surendran P, M.E Student, Department of ECE, Maharaja Engineering College, Avinashi, Tamilnadu,India.
V.Saminathan, Assistant Professor, Department of ECE, Maharaja Engineering College, Avinashi, Tamilnadu,India.
M.Udhayakumar, Assistant Professor, Department of ECE, Maharaja Engineering College, Avinashi, Tamilnadu,India.

A. *Cordic Basics*

In this section, considering computation of vector rotation which maps vector (x, y) to (x′, y′) according to the equations

$$x' = x\cos\phi - y\sin\phi \quad (1)$$
$$y' = y\cos\phi + x\sin\phi \quad (2)$$

Where ϕ is a rotation angle. Note that four multiplications and two additions are needed to compute x′ and y′ provided that the values of cos ϕ and sin ϕ are available. If this is not the case, we might consider computing them digitally by series.

B. *Combination Of Rotations*

As a first step towards the CORDIC implementation, note that if $\phi = \phi_a + \phi_b$, it can be first mapped (x, y) to (x″, y″) using the angle ϕ_a, and then map (x″, y″) to (x′, y′) using the angle ϕ_b. So, it is possible to concatenate mappings for angles ϕi, (i = 0, . . . ,N − 1) in order to evaluate the mapping for $\phi = \Sigma\, \phi_i$. In the following, denote with (x_i, y_i) and (x_{i+1}, y_{i+1}) for the input and output to the rotation by ϕ_i

$$x_{i+1} = x_i\cos\phi_i - y_i\sin\phi_i \quad (3)$$
$$y_{i+1} = y_i\cos\phi_i + x_i\sin\phi_i \quad (4)$$

.

III. MODES OF OPERATION

There are two modes of operations, rotation and vectoring modes. In the rotation mode the coordinate components of a vector and angle of rotation are given, and the coordinate components of the original vector, after rotation through the given angle, are computed. In the second mode, vectoring, the coordinate components of a vector are given and the magnitude and the angular argument of the original vector are computed. Although CORDIC may not be the fastest technique to perform these operations, it is attractive due to the simplicity of its hardware implementation, since the same iterative algorithm could be used for all these applications using the basic shift-add operations. For the sake of simplicity, the trigonometric operations in the CORDIC computer can be functionally described as the digital equivalent of an analog resolver.[10].

In the rotation mode, the coordinate components of a vector and an angle of rotation are given and the coordinate components of the original vector, after rotation through the given angle, are computed. In the second mode, vectoring, the coordinate components of a vector are given and the magnitude and angular argument of the original vector are computed [8]. In essence ,the basic computing technique used in both the rotation and vectoring modes in CORDIC is a step-by-step sequence of pseudo rotations which result in an overall rotation through a given angle(rotation) or result in a final angular argument of zero (vectoring).It is necessary that the angular increments of rotation be computed in a decreasing order. There are several permissible values which may be chosen for the angular magnitude of the first rotation step. The magnitude actually chosen for the first increments 90^0. The expression for a set of coordinate components, Yi and Xi, rotated through plus or minus 90^0 is simply given by the sine and cosine of the corresponding change from the reference position. [9]

A. Existing Cordic Circuit

The inputs to the CORDIC circuit are initial coordinate point (X0,Y0), angle to be rotated in radians as given in Fig.1. The given x and y components are stored in a pair of registers. From the registers the value is given to the adder-subtraction block, one is direct input and the other one is shifted version of the given input [2]. The ROM (require large memory resource) contains the control-bits for the number of shifts corresponding to the micro-rotations to be implemented by the barrel-shifter and the directions of micro-rotations are stored in the sign-bit register. The Data Signals are X_IN, Y_IN, X_OUT and Y_OUT.

The modified CORDIC cell is given in Fig 2, in which it contains the Look up tables for X and Y which contains the default values for the output coordinates .The shifting operation occurs in order to provide addition or subtraction according to the given direction (dir) ;whether the direction is clockwise or anticlockwise. The LUT'S provide the output coordinates Xout & Yout for the given angle. Register 'Z' is used in order to store the given input angle in order to provide the motor action according to the given input angle and direction. Here we use 2 LUT's for both directions ie.,horizontal (X) and vertical (Y).The addition, subtraction and shifting operations are performed accordingly in order to obtain the specified angle of rotation.
For example if the input angle = 10,then the outputs of the LUT'S for the output coordinates are

LUT/X = 0000101100011
LUT/Y= 0011111100000

If angle=20

LUT/X= 0001010111100
LUT/Y=0011110000100

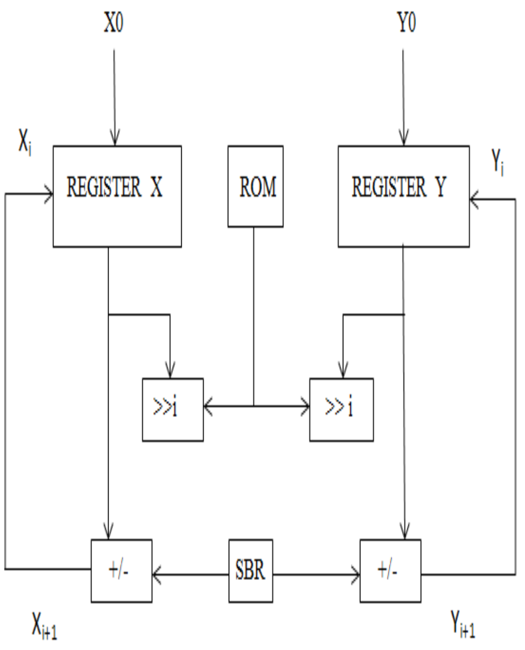

Fig 1: Existing CORDIC circuit

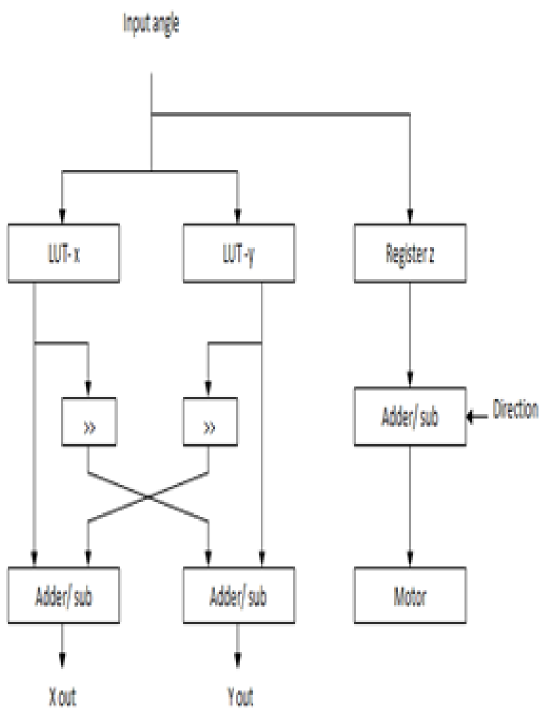

Fig 2: Block diagram of modified CORDIC cell

The integer width is fixed regardless of the word width; the remainder of the bits is used for the fractional portion of the number. Using the Q Numbers Format this representation is described as 1QN where N = word width - 2. It can also be described as Fix(N+2)_N using the System Generator Fix format. Input data signals, X_IN and Y_IN, must be in the range: -1 <= input data signal <= 1. Input data outside this range produces undefined results.[3]
Using a 10-bit word width, +1 and -1 are represented as:
"0100000000" => 01.00000000 => +1.0
"1100000000" => 11.00000000 => - 1.0

When Unsigned Fractional data format has been selected the Data Signals are represented using a unsigned fixed point number with an integer with of 1 bit. The integer width is fixed and the remainder of the word is used to represent the fractional portion of the number. For the Square Root Functional Configuration, the Data Signals, X_IN and X_OUT, are both represented in either Unsigned Fractional or Unsigned Integer data format. Further, the FPGA implementations consume a substantially lower percentage of device components than other implementations.[5]

B. Proposed CORDIC Circuit

IV. FLOWCHART

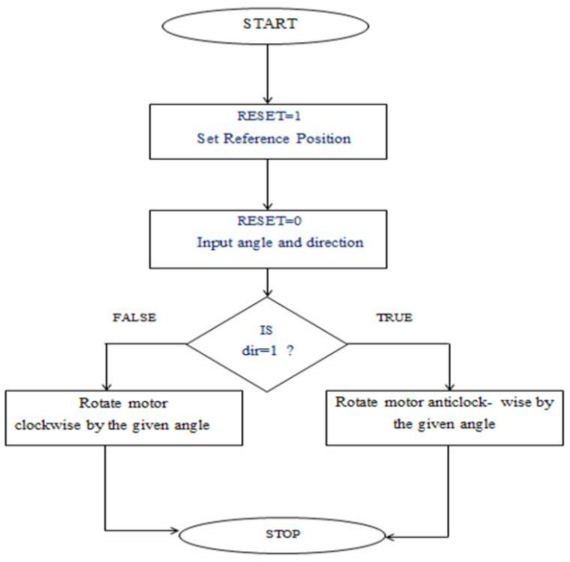
START
RESET=1
Set Reference Position
RESET=0
Input angle and direction
IS dir=1 ?
FALSE
TRUE
Rotate motor clockwise by the given angle
Rotate motor anticlock- wise by the given angle
STOP

Fig 3: Flow chart of motor rotation

From the flowchart shown in Fig-3 the operation of motor rotation can be implemented according to the control signals dir,pwm,reset and the input angle. The motor is rotated to a particular input angle having output coordinates(Xout,Yout). The angle coordinate to be rotated is obtained from the Look-up Table.If reset=1,motor will be in the reference position ,then the angle and the direction to be rotated is entered accordingly .The angle specified takes corresponding (X,Y) Coordinates from the look up table. If dir='0 'motor rotates in the clockwise direction and if dir='1' motor rotates in the anticlockwise direction. Reset option is provided after every angle rotation. The time intervals is the ON and OFF duration of the PWM wave is determined by this. More ON duration keeps the motor to be 'on' for a long time, and the angle is rotated correspondingly & vice versa . Thus the width changes the corresponding angle to be rotated. The width can be less than or greater than the reference time period.

V. SIMULATION RESULTS

A CORDIC module is designed and simulated using Xilinx ISE using VHDL as a synthesis tool. The output of the CORDIC core is analyzed and verified on the test-bench. There are 3 basic VHDL codes which includes LUT,PWM and CORDIC codes.

They are implemented in such a way that the first two codes are port-mapped in order to obtain the final output ie,CORDIC output. The outputs of simulation of the three VHDL programs are shown in figure 4,5 and 6 respectively. In the simulation result shown the signals or the variables used are angle,lut x,lut y,lut/ mem_data_x and lut/mem_data_y.There are certain values for the coordinate values in the directions x and y in the look up table. According to the input angle, the corresponding value for the x and y co-ordinates are chosen from the look up table and since it is stored in the memory locations x and y ,the values are chosen from this memory array.

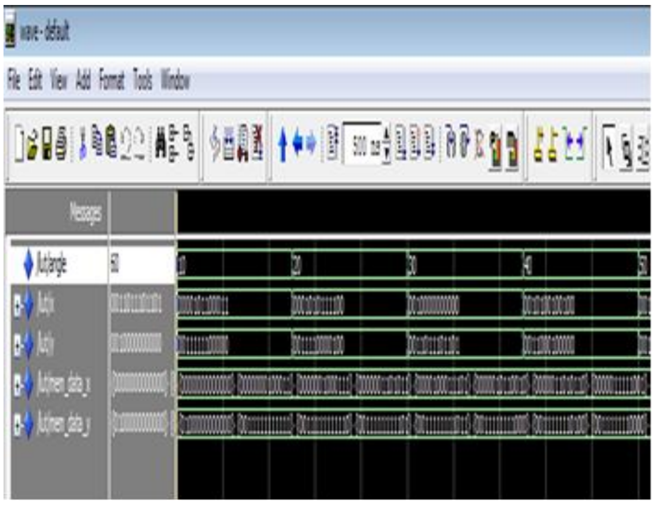

Fig 4: LUT simulation result

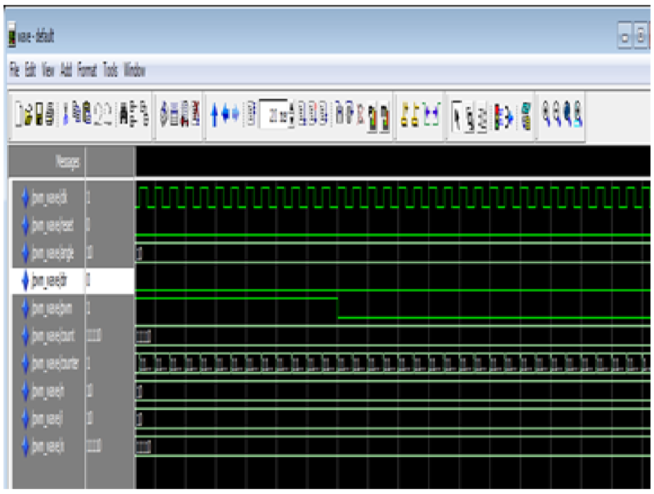

Fig-5: PWM simulation output

In the simulation result of PWM as in Fig 5,the signals used are Clk, angle and reset. According to the angle of rotation the width of the PWM wave is varied and the simulation result shows this change in the width of the modulated wave. When the angle is to be rotated in the clock wise direction, the addition operation takes place and the width will be higher and vice versa.

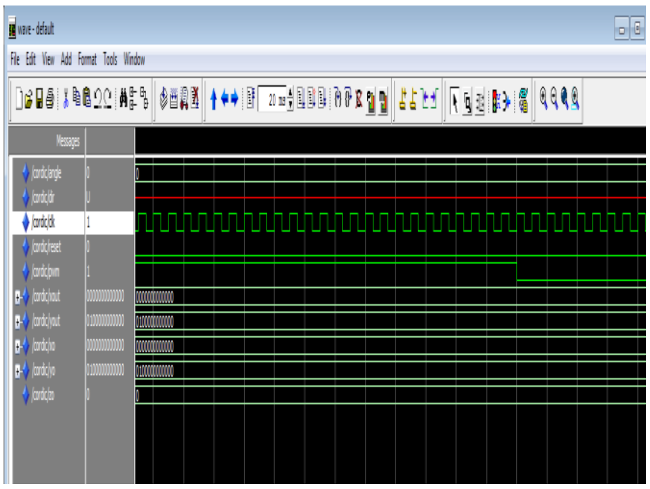

Fig 6: CORDIC simulation output

In the Fig 6,the simulation result for the CORDIC ie., the final output is shown in which port mapping is done with the VHDL programs used in the LUT and s the PWM.In this, the simulation output is the combination of both LUT and PWM.The control signals are reset,angle,dir,pwm,clk,Xout and Yout.

VI. COMPARISON RESULTS

The comparison results of the existing and the proposed systems are given in Table I and II respectively depending on the run data base.xps of the Xilinx design utilization summary.

Table I Design Utilization Summary Of Existing System

LOGIC UTILIZATION	USED	AVAILABLE
NUMBER OF SLICE REGISTERS	101	93120
NUMBER OF SLICE LUT's	410	46560
NUMBER OF FULLY USED LUT-FF PAIRS	95	416
NUMBER OF BONDED IOB's	79	240
NUMBER OF BUFG/BUFGCTRL's	1	32

Table II Design Utilization Summary Of Proposed System

LOGIC UTILIZATION	USED	AVAILABLE
NUMBER OF SLICE REGISTERS	101	93120
NUMBER OF SLICE LUT's	271	46560
NUMBER OF FULLY USED LUT-FF PAIRS	94	278
NUMBER OF BONDED IOB's	37	240
NUMBER OF BUFG/BUFGCTRL's	1	32

VII. CONCLUSION

The COordinate Rotation DIgital Computer algorithm has been used for many years for efficient implementation of vector rotation operations in hardware. It is executed merely by table look-up, shift, and addition operations. Thus, the corresponding hardware can be implemented in very economic fashion. Subsequently, it has been applied for many performance.CORDIC is a good choice for hardware solutions such as FPGA in which cost (gate count) minimization is more important than throughput maximization.

.

ACKNOWLEDGMENT

We are extremely happy in expressing my heartfelt gratitude to the Principal Dr.N.KUPPUSWAMY., Ph.D., F.I.E.,of Maharaja Engineering College ,Avinashi,for providing us the opportunity to undertake this project.

REFERENCES

[1] Cavallaro J.R and F. T. Luk,(1988) 'CORDIC arithmetic for a SVD processor,'J. for Parallel Distrib. Comput., vol. 5, pp. 271–290.
[2] Cani M.P , T. Igarashi, and G.Wyvill(2007), Interactive Shape Design. San Rafael CA:Morgan & Claypool.
[3] Hu Y.H,(1992)'CORDIC-based VLSI architectures for digital signal processing,' IEEE Signal Process. Mag., vol. 9, no. 3, pp. 16–35.
[4] Hu Y.H and S. Naganathan,(1993)'An angle recoding method for CORDIC algorithm implementation' IEEE Trans. Comput., vol. 42, no. 1. pp.99–102
[5] Hu Y.H and H.M. Chern,(1996),'A novel implementation f CORDIC algorithm using backward angle recoding (BAR),' IEEE Trans.Comput.,vol. 45, no. 12, pp. 1370–1378
[6] Jain K,(1989)' Fundamentals of Digital Image Processing'. Englewood Cliffs, NJ: Prentice-Hall
[7] Meher K, J. Valls, T.-B. Juang,K. Sridharan, and K. Maharatna(2009), '50years of CORDIC: Algorithms, architectures and applications,' IEEE Trans. Circuits Syst.I, Reg. Papers, vol. 56, no. 9.
[8] Omondi R(1994),Computer Arithmetic Systems: Algorithms, Architectures and Implementation New York:Prentice Hall.
[9] Volder J E (1959) 'The CORDIC trigonometric computing technique'IRE Trans .Electron.Computer vol.EC-8pp.330-334
[10] Walther J.S (1997), 'A unified algorithm for elementary functions' in Proc.38th Spring Joint Comput.Conf.pp379-385

Aswathy Surendran.P received B.Tech degree in Electronics and Communication Engineering from Vidya Academy Of Science & Technology, affiliated to University Of Calicut, Kerala in 2010.Presently she is pursuing her M.E in Applied Electronics from Maharaja Engineering College Avinashi,Tamil Nadu,affiliated to Anna University.

V.Saminathan Completed his BE Degree in the year 2003 at Government College of Technology Coimbatore and ME in the year 2008 at Maharaja Engineering College and currently pursuing PhD at Anna university in the area of Low Power VLSI Techniques

M.Udhayakumar working as Assistant Professor in the Department of Electronics and Communication Engineering at Maharaja Engineering college, Avinashi, Tamil Nadu, India, affiliated to Anna University Chennai.