Complexity Reduction in the CORDIC Algorithm by using MUXes

By

Yuhang Sun

Department of Electrical and Information Technology
Faculty of Engineering, LTH, Lund University
SE-221 00 Lund, Sweden

Abstract

Nowadays, the CORDIC algorithm plays an important role to deal with the non-linear functions in hardware. In this thesis, a novel methodology is described to reduce the complexity in an unrolled CORDIC architecture, which gives higher speed, lesser area, and lower power consumption. That is, MUXes are used to replace adder stages. Five different unrolled CORDIC architectures have been implemented in ASIC using a 65nm CMOS technology with Low Power High V_T transistors. The area, computational speed, accuracy, error behavior, and power consumption have been analyzed. The design aim is to reduce the power consumption, which is more and more important depending on the area. As a result the area and power consumption get 7.9% lower and 27.2% lower separately, and the speed is 22.9% higher compared to the original unrolled CORDIC architecture.

Keywords: CORDIC, power consumption.

Acknowledgments

This Master's thesis would not exist without the support and guidance of many people, my friends, my parents, especially my supervisors.

I would like to express my deepest gratitude to my supervisors, Professor Peter Nilsson, Erik Hertz, and Rakesh Gangarajaiah who guided me and helped me in this thesis work.

Finally, thanks to my relatives who kept encouraging me and my parents who afford me to study in Sweden

Yuhang Sun

Contents

List of figures

List of tables

List of acronyms

CORDIC	COordinated Rotation DIgital Computer
LUT	Look Up Table
ASIC	Application Specific Integrated Circuit
VHDL	Very High speed integrated circuit Hardware Description Language
CMOS	Complementary Metal Oxide Semiconductor
VCD	Value Change Dump
VLSI	Very Large Scale Integrated circuit

1. Introduction

In most cases, non-linear functions play an important role in hardware. The area, computational speed, accuracy, error behavior, and the power consumption are the important factors that we need to consider in the hardware design. There are several algorithms that can be chosen for implementation of non-linear functions.

Look-Up Table (LUT) is a simple and direct method to compute non-linear functions. However, a look-up table is suitable if the precision is low or the area is not necessarily considered. The size of the table grows exponentially, which makes this method unsuitable for hardware design when the precision goes high.

Another method, Parabolic Synthesis [1] is used to compute unary functions with parabolic functions. It is a novel methodology that came up with a high speed computational technology. The advantage of this method is that the delay is short.

Polynomial approximations [2] are implemented with multipliers and adders, using an iterative algorithm. Which polynomial curves we choose leads to how closely the polynomial curve can follow the special function. That is where the error is generated. Very high order polynomials are used if high accuracy is required. To meet the requirement, the polynomial approximations can be implemented with least squares approximations, which minimize the average error, or least maximum approximations, which minimize the worst-case error.

This thesis work is focus on the CORDIC algorithm. The CORDIC algorithm is an efficient algorithm to compute non-linear functions in hardware.

The COordinate Rotation Digital Computer (CORDIC) algorithm [3], also known as the digit-by-digit method or the Volder's algorithm, was first described by Jack E. Volder in 1959. It is an efficient algorithm to compute non-linear functions especially trigonometric functions in hardware design. Compared to other methods described above, the CORIDC has no multipliers, which means that it is only based on additions, subtractions and bit shifts. Traditionally, iterative CORDICs [4] have been widely used since the cost of area is less.

Nowadays, power consumption is more important than the area parameter, [5] in hardware design, which leads to that unrolled CORDICs are feasible to use. Meanwhile, it is necessary to reduce the complexity since it can also reduce the power consumption, which is the aim of this thesis.

The implementation includes five different unrolled CORDIC architectures in an Application Specific Integrated Circuit (ASIC). One is the original CORDIC architecture and the other four are CORDIC architectures, with reduced complexities on various levels. The designs are written in Very High Speed Integrated Circuit Hardware Description Language (VHDL) using a 65nm CMOS technology with Low Power High V_T transistors. The used supply voltage is V_{DD} = 1.2 volts and the temperature is 25 degrees.

As a result, area, timing, and power consumption, operating under different frequencies, are reported for the five different unrolled CORDIC architectures.

1.1. Thesis outlines

Remaining sections are outlined below:

Section 2 introduces the CORIDC algorithm and methods to reduce the complexity.

Section 3 describes the software simulation, accuracy, and error behavior of the CORIDC algorithm.

Section 4 describes the hardware implementations of the five different unrolled CORIDC architectures.

Section 5 lists and analyzes the result of section 4, consisting of area, timing and power consumption.

Section 6 concludes this thesis work.

Section 7 analyzes possible future work after this thesis work.

2. Theory

In this section, the CORDIC algorithm and the methods to reduce the complexities are introduced. Using two starting angles and four starting angles together with MUXes will reduce one adder stage and two adder stages separately in the unrolled CORDIC architectures.

2.1. The CORDIC algorithm

The CORDIC algorithm, based on vector rotations is used to get an approximation of the non-linear function. In this thesis, the sine functions and cosine functions are the functions that are implemented with the CORDIC algorithm. An iterative CORDIC architecture consists of several adder stages. When the numbers of stages increased, the error between the approximation and the original function gets smaller, which leads to an improved accuracy.

In Fig. 1, a 30 degree angle is taken to be the input angle. There are five vector rotations, which mean there are five stages. The initial vector, a 0 degree angle, is rotated by 45 degrees, which is detected to be larger than the input angle. In next stage, the vector is rotated -27 degrees and so on, to approximate the input angle. The positive degrees means the direction of the rotation is counter clockwise, the negative degrees means the direction of the rotation is clockwise. After five vector rotations, the last vector's angle approximates the input angle.

To improve the accuracy, more stages can be used. This will make the approximation infinitely close to the input angle. The angles 45 and 27 degrees are the coefficient angles, which are shown in Table XXXIII in Appendix A.

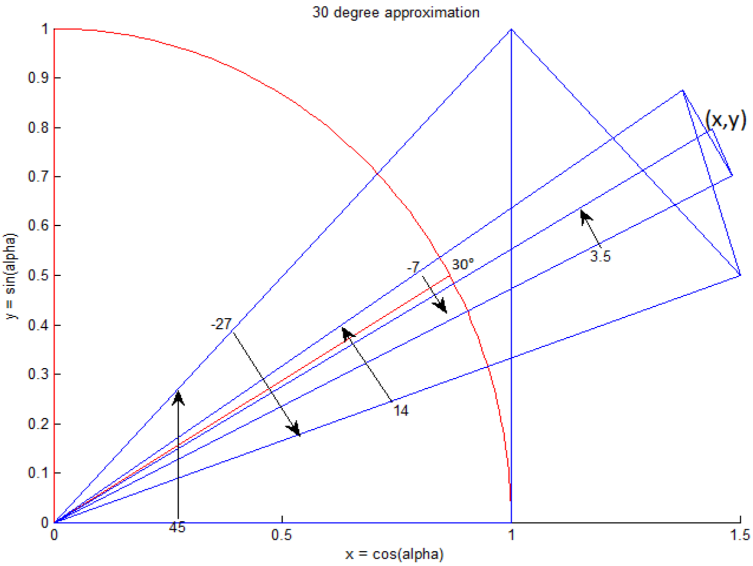

Fig. 1. Vector rotations diagram (five stages)

The function of coefficient angles is shown in (1). Where $angle(i)$ is the coefficient angle, i is the number of stages. The first five coefficient angles are given in (2).

$$agnle(i) = arctan\left(1/2^{i-1}\right) \tag{1}$$

$$\begin{cases} angle(1) = arctan(1/2^{1-1}) = 45 \\ angle(2) = arctan(1/2^{2-1}) = 26.56 \\ angle(3) = arctan(1/2^{3-1}) = 14.04 \\ angle(4) = arctan(1/2^{4-1}) = 7.13 \\ angle(5) = arctan(1/2^{5-1}) = 3.58 \end{cases} \tag{2}$$

The output is the last vector coordinates, (x, y). In this paper, CORDIC algorithm is used to deal with the sine function and cosine function. The approximated sine and cosine value of the input angle are shown in (3) and (4).

$$\sin a = \frac{y}{r} \tag{3}$$

$$\cos a = \frac{x}{r} \tag{4}$$

Where a is the input angle, r is length of the last vector. It can be noted that if $r = 1$, the sine value and cosine value are exactly the coordinates, $x = \cos a\ and\ y = \sin a,$ which means the last vector should end up on the unit circle, as shown in Fig. 2.

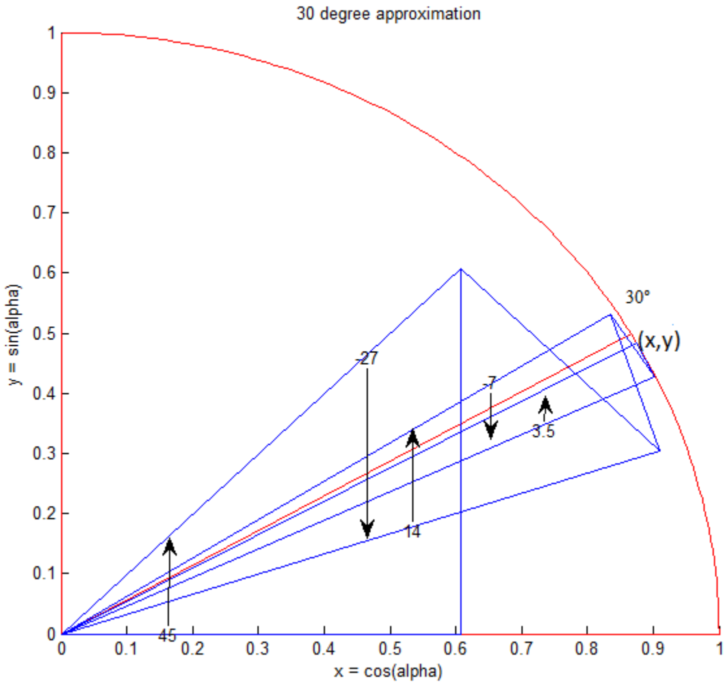

Fig. 2. The last vector ends up on the unit circle (five stages)

To make the last vector end up on the unit circle, the length of the first vector $r(1)$ should be known. There are many ways to calculate $r(1)$. In this thesis, (5) is used to get $r(1)$. When $r(6) = 1$ and $v(5) = 3.58°$, given in (2), the result can be obtained that $r(1) = 0.6088$ and the first vector coordinates is (0.6088, 0). In this case, the output corresponds to the cosine and sine value of the input angle. The architecture in hardware is shown in Fig. 11 in section 4.2.1.

$$r(i-1) = r(i) \times \cos(v(i-1)) \qquad (5)$$

2.2. Two starting angles

The idea of this project is to use more than one start angle. This will reduce the number of stages, while the MUXes will be introduced into the design, which will be discussed in section 4. In Fig. 3, 30 degrees and 60 degrees are used as the two input angles.

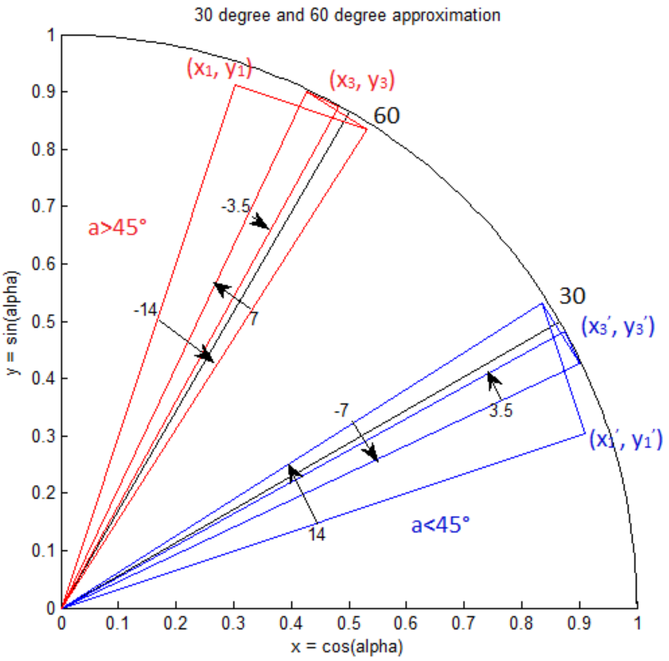

Fig. 3. Two starting angles (three stages)

In Fig. 3, there will be a comparison between the input angle and the 45 degree angle, which is implemented with two MUXes in hardware. The architecture is shown in Fig. 13 in section 4.2.3. The MUXes in hardware are used to determine the starting vector coordinate.

When the input is a 60 degree angle, which is larger than 45 degrees, the vector will rotate in the red region. The starting vector coordinate is (x_1, y_1). After three rotations, the output will be (x_3, y_3).

When the input is a 30 degree angle, which is smaller than 45 degrees, the vector will rotate in the blue region. The starting vector coordinate is (x_1', y_1'). After three rotations, the output will be (x_3', y_3').

When the input angle is exactly 45 degrees, both red lines and blue lines are suitable for the vector rotations.

$$v(1) = \begin{cases} arctan(1) - arctan(1/2) = 18.43 \\ arctan(1) + arctan(1/2) = 71.56 \end{cases} \tag{6}$$

Equation (6) indicates that, the starting vector angles are 18.4349 degrees for the input angle below 45 degrees and 71.5651 degrees for angles above 45 degrees. Fig. 11 in section 4.2.1 shows the original unrolled CORDIC architecture, where the starting vector's coordinates for Fig. 3 can be obtained.

From the left of the architecture in Fig. 11, the first stage, is shown in (7), where x_1 is the $r(1)$ in section 2.1.

$$\begin{cases} y_2 = x_1 \\ x_2 = x_1 \end{cases} \tag{7}$$

$$\begin{cases} y_3 = y_2 + \frac{x_2}{2} = \frac{3x_1}{2} = 0.9132 \\ x_3 = x_2 - \frac{y_2}{2} = \frac{x_1}{2} = 0.3044 \end{cases} \tag{8}$$

$$\begin{cases} y_3 = y_2 - \frac{x_2}{2} = \frac{x_1}{2} = 0.3044 \\ x_3 = x_2 + \frac{y_2}{2} = \frac{3x_1}{2} = 0.9132 \end{cases} \tag{9}$$

In the second stage, the starting vector coordinates can be obtained. When the input is larger than 45 degrees, the coordinates shown in (8) will be used and when lesser than 45 degrees, the coordinate shown in (9) will be used. This will make the architecture in Fig. 13 has two stages lesser than the one in Fig. 11.

2.3. Four starting angles

Fig. 4 shows the four starting angles algorithm. 4 degrees, 30 degrees, 60 degrees, and 86 degrees are used as the four input angles. The rotations depends on the comparisons between the input angle and three different degrees, 18.43 degrees, 71.56 degrees and 45 degrees from (6), which is implemented with six MUXes in a hardware design. The architecture is shown in Fig. 14 in section 4.2.4.

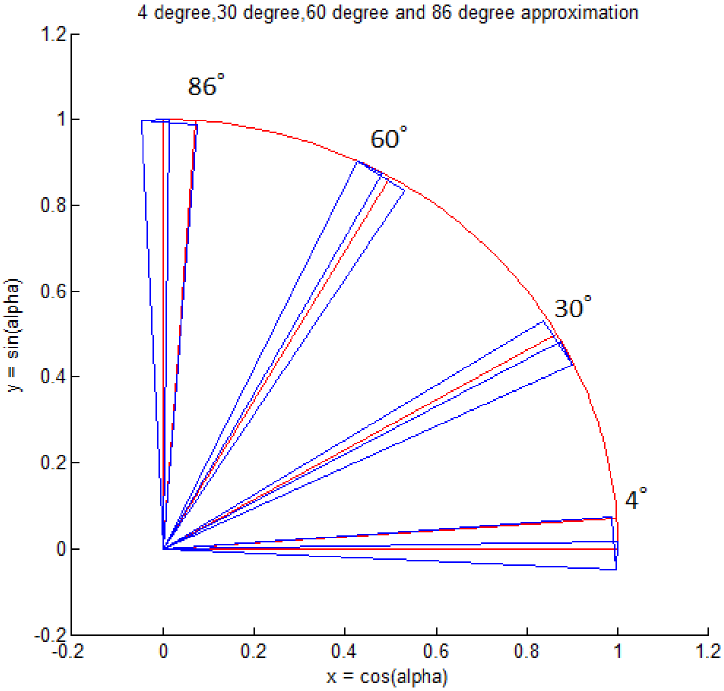

Fig. 4. Four starting angles (two stages)

$$v(1) = \begin{cases} arctan(1) - arctan(1/2) - arctan(1/4) = 4.40 \\ arctan(1) - arctan(1/2) + arctan(1/4) = 32.47 \\ arctan(1) + arctan(1/2) - arctan(1/4) = 57.53 \\ arctan(1) + arctan(1/2) + arctan(1/4) = 85.60 \end{cases} \qquad (10)$$

Equation (10) indicates the starting vector angles for the input angles in four different regions. When the input angle is from 0 degree to 18.43 degrees, the starting vector angle is 4.40 degrees. When the input angle is from 18.43 degrees to 45 degrees, the starting vector angle is 32.47 degrees. When the input angle is from 45 degrees to 71.56 degrees, the starting vector angle is 57.53 degrees. When the input angle is from 71.56 degrees to 90 degrees, the starting vector angle is 85.60 degrees.

With the same methodology, described in section 2.2, the third stage in Fig. 11 is shown in (11), (12), (13), and (14).

$$\begin{cases} y_4 = y_3 - \frac{x_3}{4} = \frac{x_1}{2} - \frac{3x_1}{8} = \frac{x_1}{8} = 0.0761 \\ x_4 = x_3 + \frac{y_3}{4} = \frac{3x_1}{2} + \frac{x_1}{8} = \frac{13x_1}{8} = 0.9893 \end{cases} \tag{11}$$

$$\begin{cases} y_4 = y_3 + \frac{x_3}{4} = \frac{x_1}{2} + \frac{3x_1}{8} = \frac{7x_1}{8} = 0.5327 \\ x_4 = x_3 - \frac{y_3}{4} = \frac{3x_1}{2} - \frac{x_1}{8} = \frac{11x_1}{8} = 0.8371 \end{cases} \tag{12}$$

$$\begin{cases} y_4 = y_3 - \frac{x_3}{4} = \frac{3x_1}{2} - \frac{x_1}{8} = \frac{11x_1}{8} = 0.8371 \\ x_4 = x_3 + \frac{y_3}{4} = \frac{x_1}{2} + \frac{3x_1}{8} = \frac{7x_1}{8} = 0.5327 \end{cases} \tag{13}$$

$$\begin{cases} y_4 = y_3 + \frac{x_3}{4} = \frac{3x_1}{2} + \frac{x_1}{8} = \frac{13x_1}{8} = 0.9893 \\ x_4 = x_3 - \frac{y_3}{4} = \frac{x_1}{2} - \frac{3x_1}{8} = \frac{x_1}{8} = 0.0761 \end{cases} \tag{14}$$

When the input angle is from 0 degree to 18.43 degrees, the coordinate shown in (11) will be used. When the input angle is from 18.43 degrees to 45 degrees, the coordinate shown in (12) will be used. When the input angle is from 45 degrees to 71.56 degrees, the coordinate shown in (13) will be used. When the input angle is from 71.56 degrees to 90 degrees, the coordinate shown in (14) will be used. This will make the architecture in Fig. 14 has three stages lesser than the one in Fig. 11.

3. Software simulation

In this section, the outputs of the designs are simulated in MATLAB. The error behavior and accuracy tests are done in this section as well.

3.1. Outputs simulation

In this section, the original CORDIC architecture, shown in Fig. 11, is simulated in MATLAB. The outputs of the sine and cosine functions are shown in Fig. 5. To improve the accuracy, 18 stages are used. Note that all architectures are simulated with the same result.

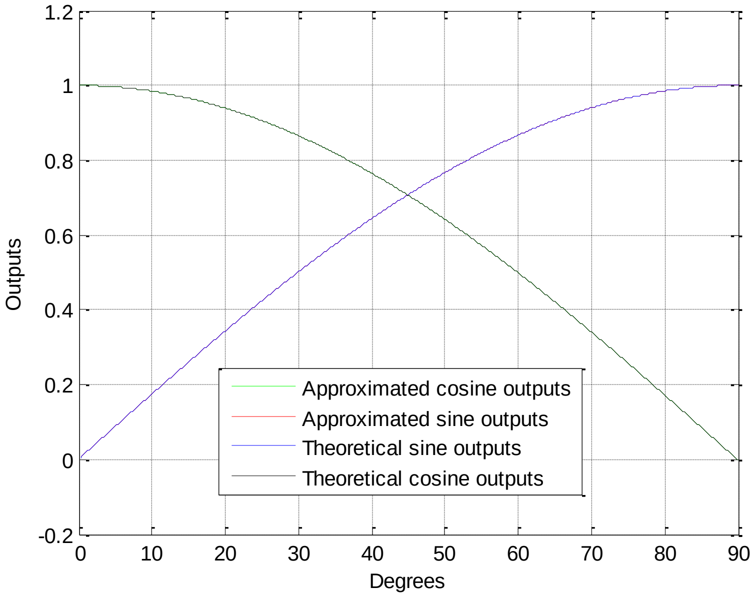

Fig. 5. MATLAB simulation of CORDIC design

Fig. 5 shows the approximated cosine outputs, theoretical cosine outputs, approximated sine outputs, and theoretical sine outputs with various input degrees. The green line and the black line, the blue line and the red line match each other. That is, the simulated CORIDC functions approximate the theoretical (floating point) functions, which is acceptable.

3.2. Error behavior

The errors are tested with all possible 32768 input angles, where $32768 = 2^{15}$. It means 15-bit inputs in hardware. Fig. 6 shows the error of the CORDIC cosine function after truncation to 15 bits.

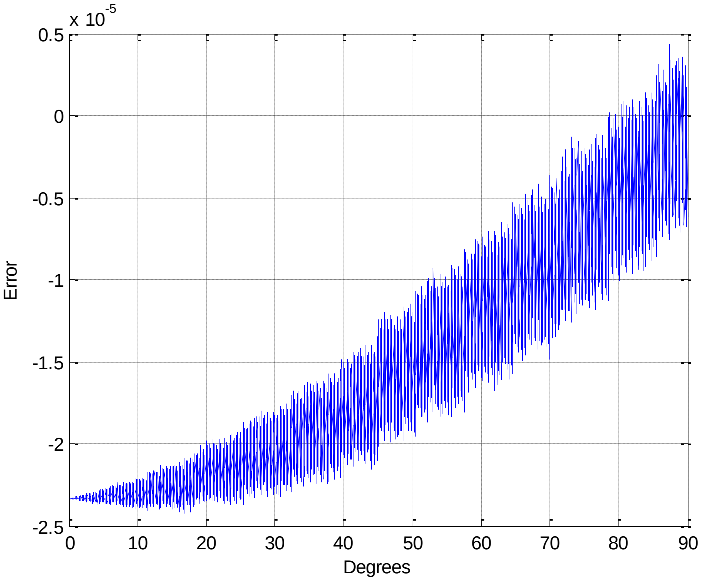

Fig. 6. The cosine error after truncation

Fig. 6 indicates that with the increasing of the input angles, the error is increasing from -2.4×10^{-5} to 0.4×10^{-5}. By displaying a histogram of the error function, the distribution of cosine error can be shown in Fig. 7.

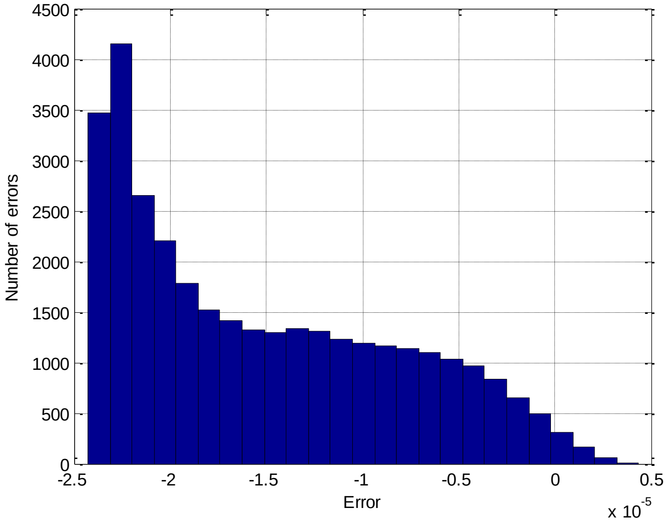

Fig. 7. The cosine error distribution

Fig. 7 indicates that the cosine error peak is placed at -2.4×10^{-5}. Most of the errors are placed away from zero, which is a drawback. The absolute cosine error in dB is shown in Fig. 8.

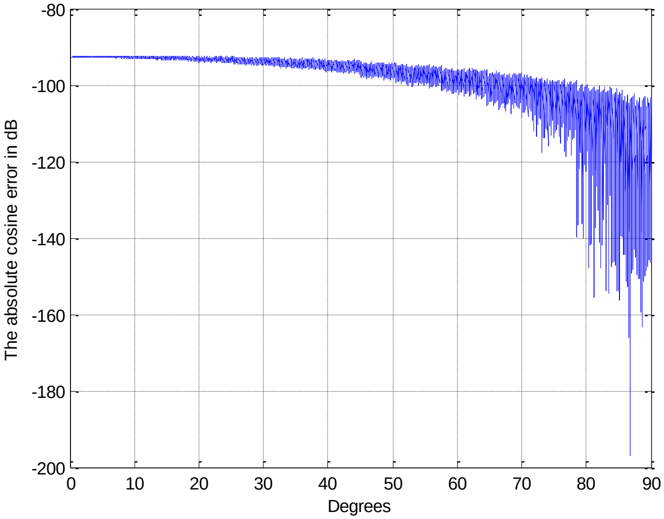

Fig. 8. The absolute cosine error in dB

Since the combination of binary numbers and decibel (dB) matches very well, displaying the errors in dB can simplify the understanding of the errors. The errors in dB are shown in (15), where x is the error.

$$x_{dB} = 20log_{10}|x| \tag{15}$$

3.3. Accuracy

In binary, one bit can presents 2 results, 0 and 1. That is, $20log_{10}(2) \approx 6dB$ corresponds to 1 binary bit in resolution. The logarithmic error divided by $-20log_{10}(2)$ gives the number of the bits, as shown in (16).

$$n = -x_{dB}/20log_{10}(2) \tag{16}$$

Fig. 9 shows the number of error bits on all possible input angles, where 15.33 is the peak, which gives 15.33 bits accuracy.

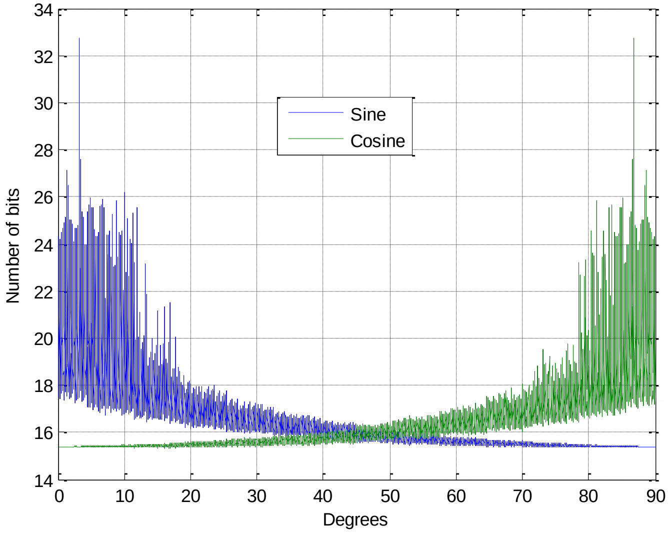

Fig. 9. Degrees and the number of error bits

4. Hardware implementation

In this section, the design flow for the thesis and the five different unrolled CORDIC architectures are implemented in hardware. IO65LPHVT_SF_1V8_50A_7M4X0Y2Z_nom_1.00V_1.80V_25C.db and CORE65LPHVT_nom_1.20V_25C.db are the libraries used in design compiler and prime time. In other words, a low power high V_T (LPHVT) technology is used at a supply voltage at 1.2V.

4.1. Design flow

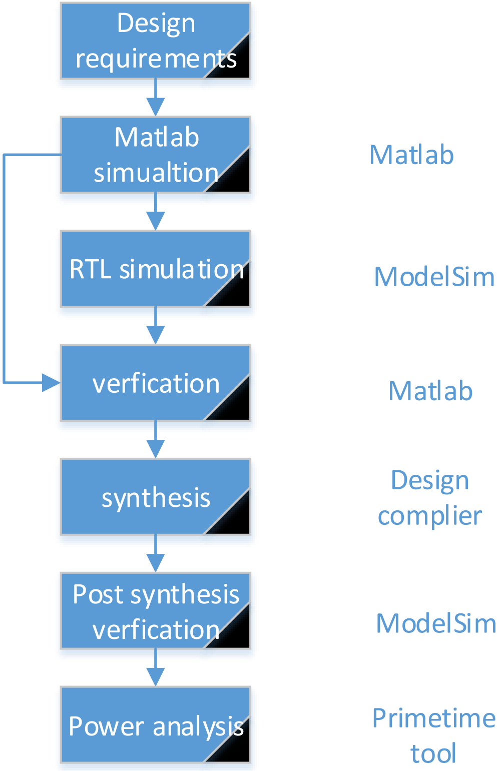

Fig. 10. The design flow

Fig. 10 shows the design flow for the thesis. To implement the CORDIC design in hardware, it should be starting with the MATLAB simulation. The design is coded in VHDL after that the MATLAB model satisfies the requirements of the design. By using ModelSim to simulate the VHDL code, the result will be compared to the MATLAB model. A netlist and a Value Change Dump (VCD) file are generated by design complier and ModelSim separately, which are used for power analysis with the primetime tool.

4.2. Hardware architectures and simulation results

In this section, five different CORDIC architectures are implemented in hardware. One is the original CORDIC architecture. The other four architectures reduce the complexity in different levels.

4.2.1. The original unrolled CORDIC architecture

The original CORDIC architecture, which corresponds to the rotations in Fig. 2, is shown in Fig. 11. Only adders and inverters are used in the architecture. The multiplications such as $1/2$, $1/4$,and $1/8$ will be achieved by hardware wired shifts on the ASIC. Note that the figure only shows the first 5 stages of the 18-stage CORDIC that is used for the simulations.

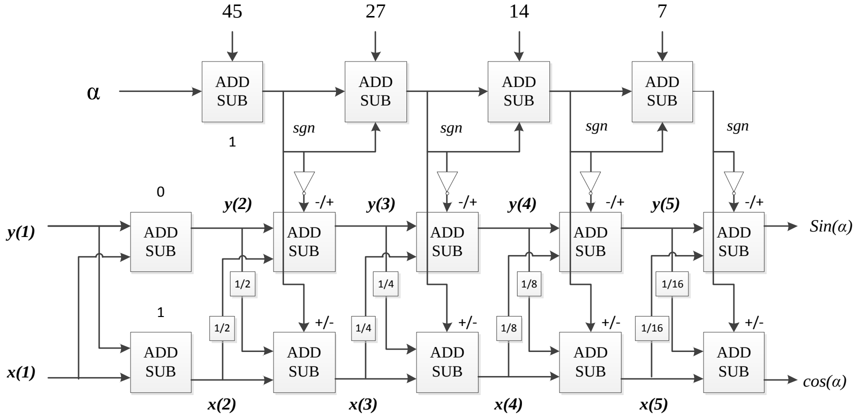

Fig. 11. An unrolled 5-stage CORDIC architecture

In Fig. 11, $x(1) = 0.6088$ and $y(1) = 0$ are the first vector coordinates and α is the input angle. At the top, 45 degrees, 26.5651 degrees, 14.0362 degrees, and 7.1250 degrees are the coefficient angles got from (1). There are 19 coefficient angles in the design. A total of 19 sign bits (*sgn*), is the result from the comparisons between the input angle and the coefficients angle, which is obtained in the upper adder row. The middle and the lower adder row compute the approximations from the left to the right, where the output is generated. The sign bits determine if the middle and the lower row should be added or subtracted.

The power consumption at both 10MHz and maximum frequency, which is 76.7MHz are shown in TABLE I and TABLE II, where the supply voltage is 1.2V.

TABLE I. POWER AT 10MHZ

Power consumption	Frequency	Unit	LPHVT
Net switching power	10MHz	nW	129100
Cell internal power	10MHz	nW	110600
Cell leakage power	10MHz	nW	131.5
Total power	10MHz	nW	239900

TABLE II. POWER AT MAXIMUM FREQUENCY

Power consumption	Frequency	Unit	LPHVT
Net switching power	76.7MHz	nW	1714000
Cell internal power	76.7MHz	nW	1137000
Cell leakage power	76.7MHz	nW	291.8
Total power	76.7MHz	nW	2852000

The results of synthesis are shown in TABLE III, TABLE IV, TABLE V, and TABLE VI, where the highest speed, minimum area, and maximum area are obtained. The timing at minimum area constraints is tested under a 10MHz frequency, which is shown in TABLE IV.

TABLE III. TIMING AT MAXIMUM SPEED CONSTRAINTS

	Unit	LPHVT
Voltage	V	1.2
Speed	MHz	76.7
Time	ns	13.03

TABLE IV. TIMING AT MINIMUM AREA CONSTRAINTS

	Unit	LPHVT
Voltage	V	1.2
Speed	MHz	31
Time	ns	32.22

TABLE V. AREA AT MINIMUM AREA CONSTRAINTS

	Unit	LPHVT
Voltage	V	1.2
Area	um^2	35482

TABLE VI. AREA AT MAXIMUM SPEED CONSTRAINTS

	Unit	LPHVT
Voltage	V	1.2
Area	um^2	100989

4.2.2. First stage removed architecture

A CORDIC architecture where the first stage is removed is shown in Fig. 12. This architecture has 2 adders less than the original one, i.e. 2 times 19 adder cells out of a total of 19 times 19 adder cells, less than the original design. The first vector coordinates are $x(2) = 0.6088$ and $y(2) = 0.6088$.

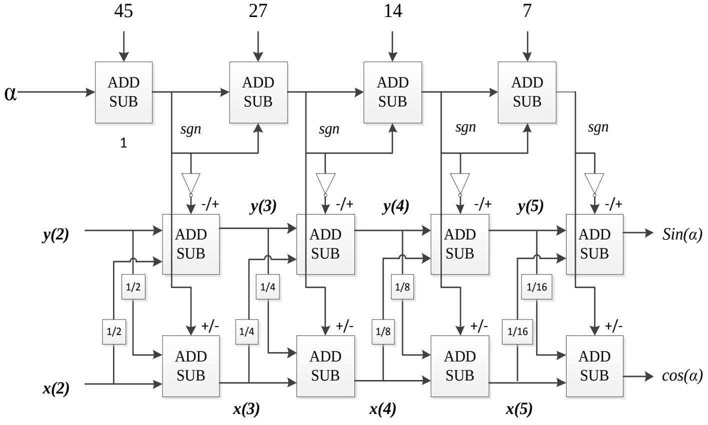

Fig. 12. First stage removed architecture

The power consumption at both 10MHz and maximum frequency, which is 76.7MHz are shown in TABLE VII and TABLE VIII, where the supply voltage is 1.2V.

TABLE VII. POWER AT 10MHZ

Power consumption	Frequency	Unit	LPHVT
Net switching power	10MHz	nW	121700
Cell internal power	10MHz	nW	104500
Cell leakage power	10MHz	nW	131.5
Total power	10MHz	nW	226300

TABLE VIII. POWER AT MAXIMUM FREQUENCY

Power consumption	Frequency	Unit	LPHVT
Net switching power	76.7MHz	nW	1692000
Cell internal power	76.7MHz	nW	1109000
Cell leakage power	76.7MHz	nW	296.2
Total power	76.7MHz	nW	2802000

The results of synthesis are shown in TABLE IX, TABLE X, TABLE XI, and TABLE XII, where the highest speed, minimum area, and maximum area are obtained.

TABLE IX. TIMING AT MAXIMUM SPEED CONSTRAINTS

	Unit	LPHVT
Voltage	V	1.2
Speed	MHz	76.7
Time	ns	13.03

TABLE X. TIMING AT MINIMUM AREA CONSTRAINTS

	Unit	LPHVT
Voltage	V	1.2
Speed	MHz	31
Time	ns	32.21

TABLE XI. AREA AT MINIMUM AREA CONSTRAINTS

	Unit	LPHVT
Voltage	V	1.2
Area	um^2	35394

TABLE XII. AREA AT MAXIMUM SPEED CONSTRAINTS

	Unit	LPHVT
Voltage	V	1.2
Area	um^2	100484

4.2.3. Two stages eliminated architecture

A CORDIC architecture where two stages are removed is shown in Fig. 13, which corresponds to the rotations in Fig. 3. This architecture has four adders less than the original one. Two of these adders are replaced by two MUXes, which are controlled by the first *sgn* bit in the upper adder row. The two hardware wired shifts (1/2) are also removed compared to the original architecture. As described in section 2.2, the first vector coordinates $(x(3), y(3))$ are shown in (8) and (9).

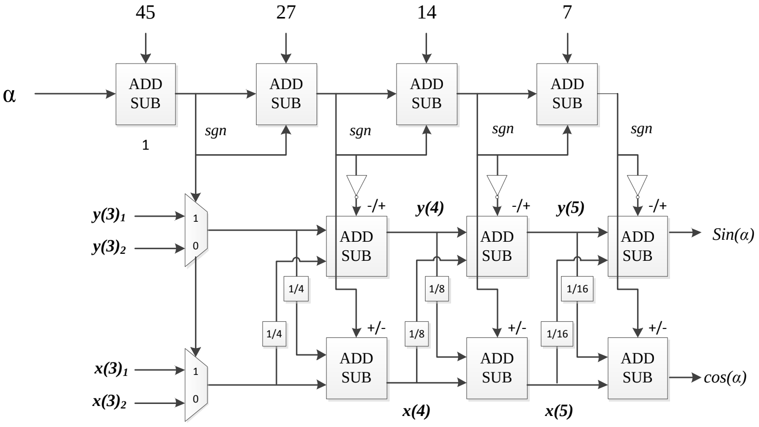

Fig. 13. Two stages are removed in the architecture

The power consumption at both 10MHz and maximum frequency, which is 83.1MHz are shown in TABLE XIII and TABLE XIV, where the supply voltage is 1.2V.

TABLE XIII. POWER AT 10MHZ

Power consumption at	Frequency	Unit	LPHVT
Net switching power	10MHz	nW	115700
Cell internal power	10MHz	nW	96490
Cell leakage power	10MHz	nW	126.2
Total power	10MHz	nW	212400

TABLE XIV. POWER AT MAXIMUM FREQUENCY

Power consumption at	Frequency	Unit	LPHVT
Net switching power	83.1MHz	nW	1615000
Cell internal power	83.1MHz	nW	1071000
Cell leakage power	83.1MHz	nW	319.2
Total power	83.1MHz	nW	2687000

The results of the synthesis are shown in TABLE XV, TABLE XVI, TABLE XVII, and TABLE XVIII, where the highest speed and minimum area, are obtained.

TABLE XV. TIMING AT MAXIMUM SPEED CONSTRAINTS

	Unit	LPHVT
Voltage	V	1.2
Speed	MHz	83.1
Time	ns	12.03

TABLE XVI. TIMING AT MINIMUM AREA CONSTRAINTS

	Unit	LPHVT
Voltage	V	1.2
Speed	MHz	31.5
Time	ns	31.78

TABLE XVII. AREA AT MINIMUM AREA CONSTRAINTS

	Unit	LPHVT
Voltage	V	1.2
Area	um^2	33660

TABLE XVIII. AREA AT MAXIMUM SPEED CONSTRAINTS

	Unit	LPHVT
Voltage	V	1.2
Area	um^2	109840

4.2.4. Three stages eliminated architecture

A CORDIC architecture where three stages are removed is shown in Fig. 14, which corresponds to the rotations in Fig. 4. This architecture has 6 adders less than the original one. Four of these adders are replaced by six MUXes, which are controlled by the first two *sgn*s in the upper adder row. As described in section 2.3, the first vector coordinates $(x(4), y(4))$ are shown in (11), (12), (13), and (14).

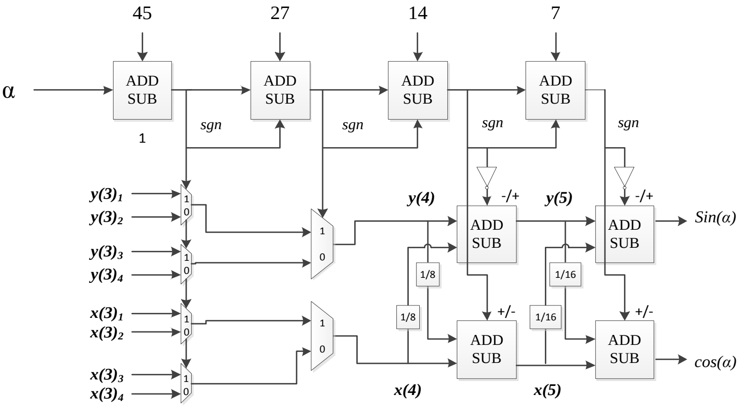

Fig. 14. Three stages are removed architecture

The power consumption at both 10MHz and maximum frequency, which is 90.7MHz are shown in TABLE XIX and TABLE XX, where the supply voltage is 1.2V.

TABLE XIX. POWER AT 10MHZ

Power consumption	Frequency	Unit	LPHVT
Net switching power	10MHz	nW	97410
Cell internal power	10MHz	nW	88820
Cell leakage power	10MHz	nW	123.5
Total power	10MHz	nW	186400

TABLE XX. POWER AT MAXIMUM FREQUENCY

Power consumption	Frequency	Unit	LPHVT
Net switching power	90.7MHz	nW	1396000
Cell internal power	90.7MHz	nW	922900
Cell leakage power	90.7MHz	nW	308.8
Total power	90.7MHz	nW	2319000

The results of synthesis are shown in TABLE XXI, TABLE XXII, TABLE XXIII, and TABLE XXIV, where the highest speed, minimum area, and maximum area are obtained.

TABLE XXI. TIMING AT MAXIMUM SPEED CONSTRAINTS

	Unit	LPHVT
Voltage	V	1.2
Speed	MHz	90.7
Time	ns	11.03

TABLE XXII. TIMING AT MINIMUM AREA CONSTRAINTS

	Unit	LPHVT
Voltage	V	1.2
Speed	MHz	32.3
Time	ns	30.93

TABLE XXIII. AREA AT MINIMUM AREA CONSTRAINTS

	Unit	LPHVT
Voltage	V	1.2
Area	um^2	33407

TABLE XXIV. AREA AT MAXIMUM SPEED CONSTRAINTS

	Unit	LPHVT
Voltage	V	1.2
Area	um^2	119512

4.2.5. Final architecture

A CORDIC architecture, where three stages and one upper adder are removed is shown in Fig. 15. The first adder of the coefficients angle is replaced by a MUX and a Sgn detector compared to the architecture in Fig. 16. This means seven MUXes and a Sgn detector are introduced into the architecture. The first vector coordinates $(x(4), y(4))$ are also presents in (11), (12), (13), and (14).

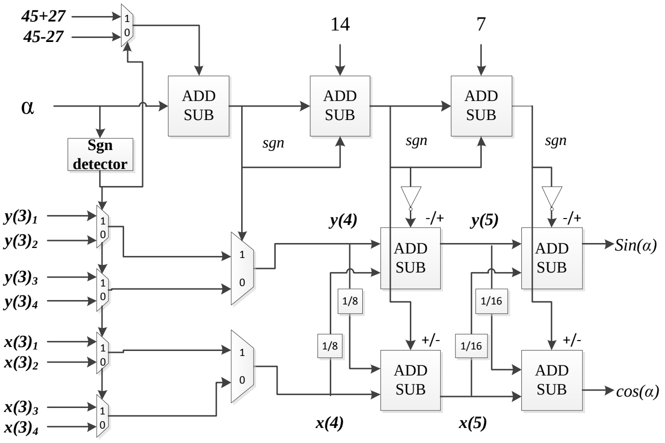

Fig. 15. Final architecture

The Sgn detector is used to detect if the input angle is larger than 45 degrees or not. If the input is larger than 45 degrees, the Sgn detector's output is 1, if not the Sgn detector's output is 0. The Sgn detector is realized by using the upper 6 bits of the input angle and it consists of the inverters and NAND gates in hardware. The architecture of the Sgn detector is shown in Fig. 16.

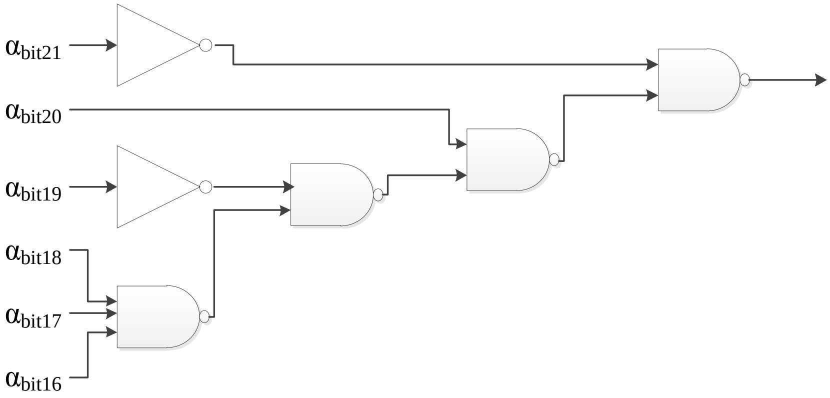

Fig. 16. The Sgn detector

The logic function of the Sgn detector is shown in (17).

$$Sgn = a_{bit21} + a_{bit20}a_{bit19} + a_{bit20}a_{bit18}a_{bit17}a_{bit16} \tag{17}$$

The power consumption at both 10MHz and maximum frequency, which is 99.6MHz are shown in TABLE XXV and TABLE XXVI, where the supply voltage is 1.2V.

TABLE XXV. POWER AT 10MHZ

Power consumption	Frequency	Unit	LPHVT
Net switching power	10MHz	nW	91220
Cell internal power	10MHz	nW	83460
Cell leakage power	10MHz	nW	123.5
Total power	10MHz	nW	174800

TABLE XXVI. POWER AT MAXIMUM FREQUENCY

Power consumption	Frequency	Unit	LPHVT
Net switching power	99.6MHz	nW	1211000
Cell internal power	99.6MHz	nW	932500
Cell leakage power	99.6MHz	nW	316.1
Total power	99.6MHz	nW	2044000

The results of synthesis are shown in TABLE XXVII, TABLE XXVIII, TABLE XXIX, and TABLE XXX, where the highest speed, minimum area, and maximum area are obtained.

TABLE XXVII. TIMING AT MAXIMUM SPEED CONSTRAINTS

	Unit	LPHVT
Voltage	V	1.2
Speed	MHz	99.6
Time	ns	10.04

TABLE XXVIII. TIMING AT MINIMUM AREA CONSTRAINTS

	Unit	LPHVT
Voltage	V	1.2
Speed	MHz	32.3
Time	ns	30.95

TABLE XXIX. AREA AT MINIMUM AREA CONSTRAINTS

	Unit	LPHVT
Voltage	V	1.2
Area	um^2	33394

TABLE XXX. AREA AT MAXIMUM SPEED CONSTRAINTS

	Unit	LPHVT
Voltage	V	1.2
Area	um^2	135193

5. Results

In this section, a test setup for the designs is introduced. The area, timing information, and power consumption are also analyzed.

5.1. Test setup

Fig. 17 shows the test setup of the designs. In this design, the input to the testbench is generated into a text file by software simulation. The output from the top design is also verified by the software.

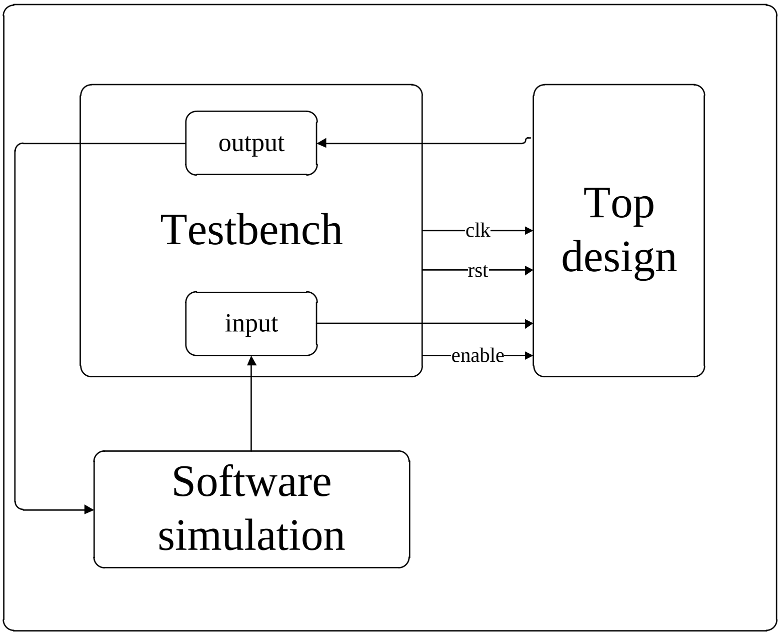

Fig. 17. The design test setup

5.2. The area

The minimum area is estimated by *set_max_area 0* script in design complier. The areas, using the minimum area constraint, for the five CORDIC architectures are shown in Fig. 18.

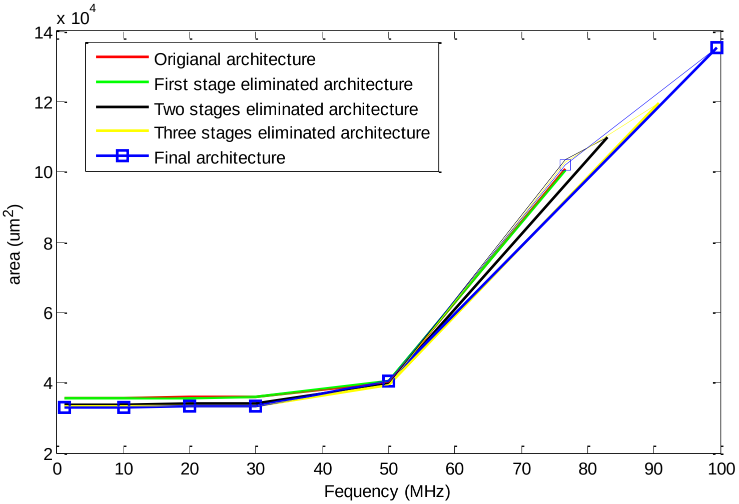

Fig. 18. Area at different frequencies

The minimum areas can be shown in TABLE XXXI. The final architecture has the lowest area, 33200um^2.

TABLE XXXI. AREA AT MINIMUM AREA CONSTRAINTS

Architectures	Minimum area (um^2)
The original architecture	36067
First stage eliminated architecture	35966
Two stages eliminated architecture	33999
Three stages eliminated architecture	33389
Final architecture	33200

TABLE XXXI indicates that with reduced complexity, the minimum areas of the designs are also reduced. The percentage of the area reduction is 7.9%. The areas at maximum speed constraints for the five architectures are shown in TBALE XXXII.

TABLE XXXII. AREA AT MAXIMUM SPEED CONSTRAINTS

Architectures	Area (um^2)
The original architecture	100989
First stage eliminated architecture	100484
Two stages eliminated architecture	109840
Three stages eliminated architecture	119512
Final architecture	135193

5.3. Timing information

The maximum speed is estimated when the area has not been set for any value. When a highest clock frequency is specified, i.e. the slack is zero, the clock frequency corresponds to the maximum speed. The timing information gives the bottleneck at the highest frequency, shown in TABLE XXXIII.

TABLE XXXIII. TIMING AT MAXIMUM SPEED CONSTRAINTS

Architectures	Critical path (ns)	Frequency (MHz)
The original architecture	13.03	76.7
First stage eliminated architecture	13.03	76.7
Two stages eliminated architecture	12.03	83.1
Three stages eliminated architecture	11.03	90.7
Final architecture	10.04	99.6

Table XXXIII indicates that with the complexity reduced, the maximum frequencies of the designs are also increased. More optimizations architecture compared to the others architectures, has the highest speed 10.04ns with 9.96×10^7 Hz frequency. The speed got 22.9% higher performance. The timing at minimum area constraints are shown in TABLEXXXIV.

TABLE XXXIV. TIMING AT MINIMUM AREA CONSTRAINTS

Architectures	Time (ns)	Speed (MHz)
The original architecture	32.22	31
First stage eliminated architecture	32.21	31
Two stages eliminated architecture	31.78	31.5
Three stages eliminated architecture	30.93	32.3
Final architecture	30.95	32.3

5.4. Power consumption

5.4.1. Power analysis

The CMOS transistors' power consists of dynamic power and static power, as shown in (18) [6].

$$P_{tot} = P_{dynamic} + P_{static} \tag{18}$$

The dynamic power consists of the switching power and the internal power, as shown in (19).

$$P_{dynamic} = P_{switching} + P_{internal} \tag{19}$$

The dynamic power can be written in (20).

$$P_{dynamic} = aCV^2f \tag{20}$$

Where the factor a is the switching activity, C is the node capacitance, f is the clock frequency, and V is the supply voltage. The static power comes mainly from the sub threshold leakage current.

TABLE XXXV and TABLE XXXVI indicate the power consumption at the maximum speed constraints and the power consumption at minimum area constraints separately. The power at minimum area constraints is tested under a 10MHz frequency.

TABLE XXXV. POWER AT MAXIMUM SPEED CONSTRAINTS

Architectures	Frequency (MHz)	Switching power (mW)	Internal power (mW)	Leakage (nW)	Total power (mW)
The original architecture	76.7	1.714	1.137	291.8	2.852
First stage eliminated architecture	76.7	1.692	1.109	296.2	2.802
Two stages eliminated architecture	83.1	1.615	1.071	319.2	2.687
Three stages eliminated architecture	90.7	1.396	0.9229	308.8	2.319
Final architecture	99.6	1.211	0.8325	316.1	2.044

TABLE XXXVI. POWER AT MINIMUM AREA CONSTRAINTS

Architectures	Frequency (MHz)	Switchin g power (mW)	Internal power (mW)	Leakage (nW)	Total power (mW)
The original architecture	10	0.1291	0.1106	131.5	0.2399
First stage eliminated architecture	10	0.1217	0.1045	131.5	0.2263
Two stages eliminated architecture	10	0.1157	0.0964	126.2	0.2124
Three stages eliminated architecture	10	0.09741	0.0882	123.5	0.1864
Final architecture	10	0.09122	0.08346	123.5	0.1748

The power consumption for the five different CORDIC architectures at various frequencies is shown in Fig. 19.

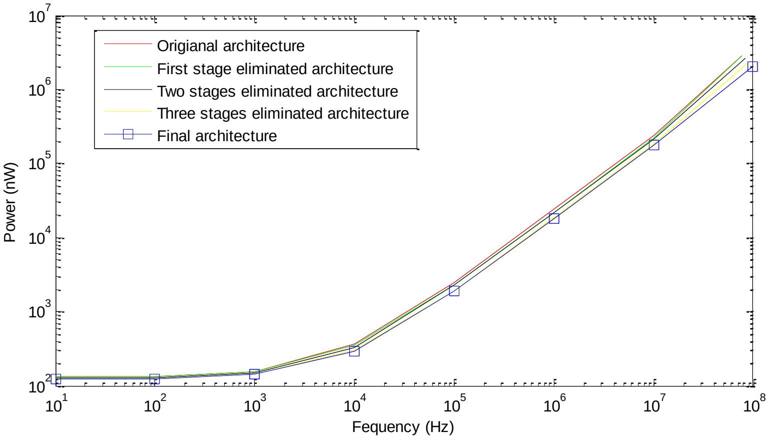

Fig. 19. Power at various frequencies

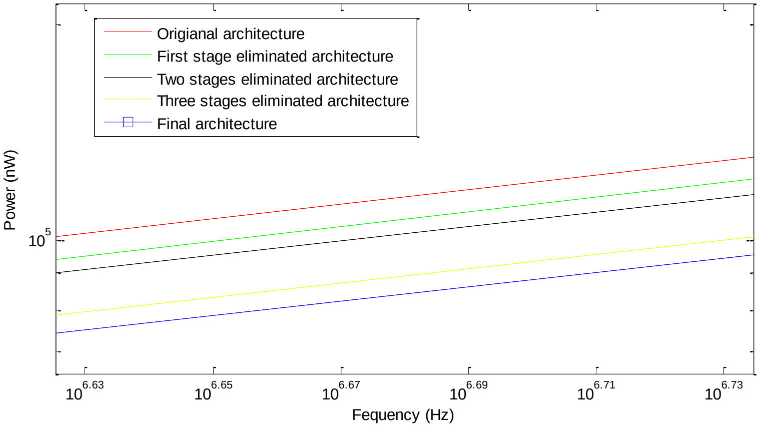

Fig. 20. magnified diagram for Fig. 19

Fig. 19 shows the power consumption at different frequencies for the five architectures described in section 4. Five curves are closed, but in Fig. 20, the magnified diagram for Fig. 19, indicates that the final architecture cost the least dynamic power. We can also get that when the adder stages are eliminated, the cost of power is less. There is a knee in the figure at low frequencies. This is because at low frequencies the static power is much larger than the dynamic power, and it changes only a little with the frequency.

As a result the area is 7.9% lower in the final architecture. A substantial improvement can be seen for the power consumption, which is 27.2% lower in the final architecture compared to the original one and the speed, which is 22.9% higher.

Area much higher for maximum speed, because there are too many logic gates used to achieve higher speed.

6. Conclusions

In this thesis, MUXes are used in hardware to reduce the complexity. Five different CORIDC architectures are implemented with eliminating the stages. The area, computational speed, accuracy, error behavior, and the power consumption have been analyzed under the software simulation and hardware implementation. The speed, minimum area and power consumption have got optimized in different levels. As a result the area and power consumption get 7.9% lower and 27.2% lower separately, and the speed is 22.9% higher compared to the original unrolled CORDIC architecture. It is also proved that in unrolled CORDIC architectures, the reduction of the power consumption can be achieved by reducing the complexity, which meets the aim of this thesis.

7. Future work

In this thesis, three stages are eliminated at most. There are more stages can be eliminated for the reduction of the power consumption.

More than one Sgn detector can be introduced to reduce the coefficients adder.

The number of the iteration should decrease to some extent, which can also reduce the power consumption.

References

[1] Erik Hertz and Peter Nilsson, “Parabolic Synthesis Methodology Implemented on the Sine Function”, in Proceedings of the 2009 International Symposium on Circuits and Systems (ISCAS’09), Taipei, Taiwan, May 2427, 2009 .

[2] Gordon K. Smyth “Polynomial Approximation” in Encyclopedia of Biostatistics ,John Wiley & Sons, Ltd, Chichester, Peter Armitage and Theodore Colton, (ISBN 0471 975761) ,1998.

[3] J. E. Volder, “The CORDIC Trigonometric Computing Technique”, IRE Transactions on Electronic Computers, vol. EC-8, no. 3, 1959, pp. 330–334.

[4] Hue, Y.H., “CORDIC-based VLSI architectures for digital signal processing”, IEEE Signal Processing Magazine, pp. 16-35, ISSN: 1053-5888, July 1992.

[5] Peter Nilsson “Complexity Reductions in Unrolled CORDIC Architectures,” in Proceedings of the IEEE 14th International Conference on Electronics, Circuits and Systems (ICECS 2009), pp. 868-871, Hammamet, Tunisia, December 13-16, 2009.

[6] http://en.wikipedia.org/wiki/CMOS#Power:_switching_and_leakage

Appendix A

TABLE XXXVII. COEFFICIENT ANGLES

	Decimal angle	Binary angle
α_1	45	00101101000000000000000
α_2	26.565032958984375	00011010100100001010011
α_3	14.036224365234375	00001110000010010100011
α_4	7.125000000000000	00000111001000000000000
α_5	3.576324462890625	00000011100100111000101
α_6	1.789886474609375	00000001110010100011011
α_7	0.895172119140625	00000000111001010010101
α_8	0.447601318359375	00000000011100101001011
α_9	0.223785400390625	00000000001110010100101
α_{10}	0.111877441406250	00000000000111001010010
α_{11}	0.055938720703125	00000000000011100101001
α_{12}	0.027954101562500	00000000000001110010100
α_{13}	0.013977050781250	00000000000000111001010
α_{14}	0.006988525390625	00000000000000011100101
α_{15}	0.003479003906250	00000000000000001110010
α_{16}	0.001739501953125	00000000000000000111001
α_{17}	0.000854492187500	00000000000000000011100
α_{18}	0.000427246093750	00000000000000000001110
α_{19}	0.000213623046875	00000000000000000000111