ELSEVIER
Available online at www.sciencedirect.com
SciVerse ScienceDirect
Procedia Engineering 15 (2011) 3848 – 3853
Procedia Engineering
www.elsevier.com/locate/procedia

Advanced in Control Engineering and Information Science

Datapath Design and Full Custom Implementation of Radix-2 CORDIC Processor

Zhuo BI[a] , Yijun DAI [b] a,b*

[a,b]*Coll. of Mecha. Engin. and Automation, Shanghai Univ ,Shanghai 200072, China*

Abstract

The CORDIC algorithm can be implemented efficiently for trigonometric computing which is important in image processing, navigation. And a full custom implementation has better performance in low silicon area and high throughput. This paper presents a 32 bits full custom layout design for CORDIC architecture by using pipelining techniques to increase the throughput of the processor. As a result, the area of CORDIC data-path is 6.0μm^2 in the SIMC 0.13μm 1P8M CMOS process. The critical path delay is 0.639 ns at the Typical-Typical corners(Voltage 1.2V Temperature 25℃), wire delay which approximates twice or three times as the critical path delay should be added in practice that means the frequency can reach 391MHz approximately.

Keywords: CORDIC, pipeline, full custom;

1. Introduction

Trigonometric computing is a fundamental operation in many scientific area and engineering applications, especially in the field of image processing, navigation and so on. High precision, real-time and small area are the important issues in these applications. Co-Ordinate Rotation Digital Computer (CORDIC) algorithm was originally developed as a digital solution for real-time navigation system problems by J.D. Volder in 1959, and generalized by Walther in 1971[1]. In 2004, Tso-Bing Juang and his

* Corresponding author. Tel. :+86-021-56331206-105.
E-mail address: zhuo.bi@shu.edu.cn, daiyijun@shu.edu.cn

colleagues proposed a modified CORDIC algorithm by paralleling to speed up the iterations and have a high precision than before, which were BBR (Binary -To-Bipolar Recoding) and MAR (Microrotation Angle Recoding)[2]. CORDIC algorithm is an iterative algorithm for the calculation of the rotation of a two-dimensional vector in linear, circular and hyperbolic coordinate systems[3]. And each system has two modes which are rotation and vectoring. In nowadays, the trade-off between latency time and the hardware complexity urges designers to use higher radix CORDIC replace the conventional radix-2 CORDIC in high-speed application purposes. From the MATLAB simulations, the precision will decrease by the increase radix. Therefore, this paper chooses radix-2 in a full custom method to achieve high speed, reasonable hardware complexity and high precision.

An existing radix-2 CORDIC algorithm, whose architecture is depicted in detail in the subsequent sections of this paper, is taken into account for the hardware implementation. The rotation is completed in 18 clock cycles for 32 bits precision. The background of CORDIC algorithm will firstly be showed in Section II. The CORDIC RTL (Register Transition Level) model of pipeline structure will be discussed in Section III. In Section IV, it will show the implementation and some discussion between full custom and FPGA on area and speed.

2. CORDIC algorithm

Two computing modes applied in circular system. In the vectoring mode, the coordinate components of a vector are given and the magnitude and angular argument of the original vector are computed. In the rotation mode, the coordinate components of a vector and an angle of rotation are given and the coordinate components of original vector are computed by the given angle after rotation. The rotation mode of CORDIC algorithm can be implemented by the following iterative equations:

$$X_{n+1}=X_n-S_nY_n2^{n+1} \qquad Y_{n+1}=Y_n-S_nY_n2^{n+1} \qquad Z_{n+1}=Z_n-S_n\arctan(2^{-n}) \tag{1}$$

Where 2 means the radix is two, and $S_n=\{-1,+1\}$ is the sign function. Each rotated angle is $Z_n=\theta-\sum_{i=0}^{n-1}S_i\theta_i$. If S_n was 1, then $Z_n\le 0$ and it will rotate in order. Otherwise it will rotate in inverted sequence.

When the initial values are $X_0=K,Y_0=0,Z_0=\theta$, after N times rotations, the solutions of CORDIC arithmetic are $X_{n+1}=\cos\theta, Y_{n+1}=\sin\theta, Z_{n+1}=0$.

$X_{n+1},Y_{n+1}$are the coordinated of the vector resulting from applying n+1 micro rotation and the Z_{n+1} is the angle remaining to be rotated. The final coordinates are scaled by the equation 2:

$$K=\frac{1}{P}=\prod_{n=0}^{N}\cos\theta_n=\prod_{n=0}^{N}\cos\left(\arctan 2^{-n}\right)=\prod_{n=2}^{N}\frac{1}{\sqrt{1+2^{-2n}}} \tag{2}$$

Where K is called the scale factor is decided by the rotation times. It converges to 0.607253 when rotation times tend to infinity.

3. Proposed architecture

This section presents the proposed architecture of the iterative radix-2 CORDIC processor. In order to realize the CORDIC algorithm, there are two hardware architectures which are iteration and pipeline. For an ideal CORDIC architecture, there have many requirements, such as higher speed of operation, lower power, smaller area. But it can not meet all the requirements at the same time in the actual work. For iteration method, the data speed is N times clock cycle, where N is the iteration times. And the values will be achieved after N times rotations. It seems to need a long time to get the values, which looks low efficiency. However, the areas of this method are small. For the second method pipeline, this method needs more areas to get the high efficiency. The clock cycle of the pipeline structure is N times lower than the rotation structure. The number of shifting bits is fixed so that we can use the metal wires to contact two sub-adders directly. Furthermore, the constants are also fixed so that the constants can be implemented by the metal wires instead of look up table. For high precision, this paper chooses the pipeline structure to realize the CORDIC. The proposed architecture is showed in Fig1.

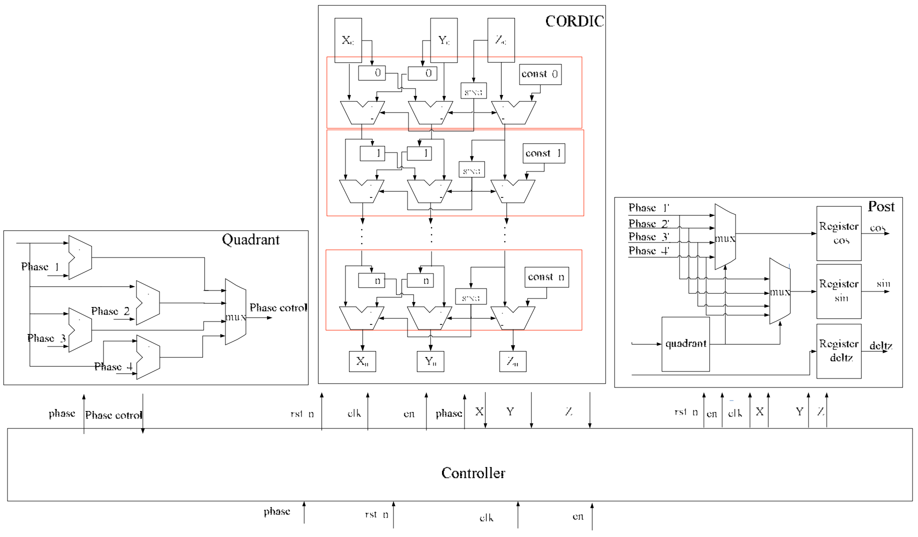

Fig. 1. Proposed CORDIC architecture

The proposed CORDIC architecture (Fig.1) is made up of 3 parts which are quadrant part, CORDIC part, and post part. And the CORDIC part is the key part of this architecture and it is implemented by equation 1.

In the CORDIC part, the highest bit of signal Z is a control signal which determines the X, Y, Z operations in each iteration. 3×N sub-adders, 3×N shifters, 4×N registers are included in pipeline architecture. Constants which store the predetermined angle can be implemented by metal wires in the above analysis. Moreover, the number of shifters is fixed, so the 3*N shifters are also implemented by metal wires. The 32 bits CORDIC architecture is implemented in SMIC 0.13μm 1P8M CMOS technology and table 1 shows the analysis by the timing tool of DC (Design Compiler) of Synopsys in the condition of 1.2V supply and SS (Slow-Slow) corners.

Table 1. Timing analysis of different pipeline stage implementation of the radix-2 CORDIC

CORDIC part stage	1	8	16	32
The total pipeline stage	3	10	18	34

Delay time(ns)	101.86	67.45	6.74	4.34
Cell area(μm²)	222249.2656	437251.9688	610494.8750	664274.250

This paper chooses the total pipeline stages are 18 in the consideration of area and delay. The data will store into the registers after twice sub-adder iteration. The whole architecture needs 3×18 registers, 3×32 sub-adders, and constants, shifters are metal wires. The pipeline needs more area than iteration, whose throughout is enlarged, but the full custom layout makes the area as small as possible.

It needs to change the data format before calculating the cosine and sine by the rotation mode of circular coordinate system of CORDIC architecture. In this paper, the input data are 32 bits which represents an angle. So this paper uses the 2^{32} to represent 360°. The 32 bits output data represent the values of sine and cosine which are between 0 and 1. For the single precision data format, the 31st, 30th bit represents a sign and overflow bit respectively. And the other 30 bits represent 1. The accuracy of the sine and cosine arrive at 10^{-8} after the computing the CORDIC part.

According to J. Walther proposed the order of the iteration of the rotation mode of circular coordinate system of CORDIC architecture was 0,1,2,…n-1, the angle range is[-99°,99°]. In order to get the wide angle range, the Quadrant part makes the 2nd, 3rd, 4th angle change to the first quadrant. In additional, through the relationships of data format between the CORDIC and the IEEE-754, the angle is decided in which quadrant by operating the higher 2 bits. An accurate value of trigonometric functions can be achieved with mathematics knowledge.

4. Implementation and discussion

4.1. Simulation

In order to ensure the compute precision, 32 times iterations are implemented in the CORDIC processor, and the precision of the processor will achieve at $1.334\times 1.334\times 10^{-8}$ approximately. It needs 32 cycles to calculate the sine and cosine for each time. The verification of the CORDIC RTL model is showed in Fig 2.

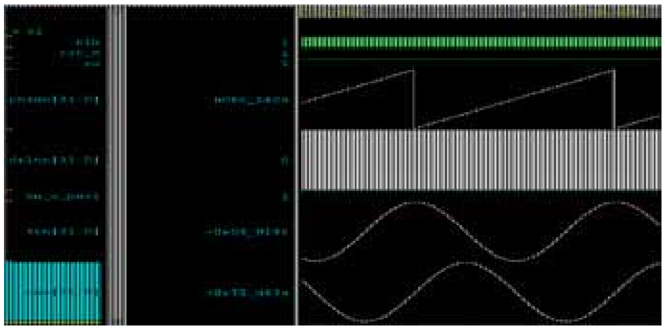

Fig. 2. The input angle increases by 1°from 0°to 360°simulations

Fig 2 shows the outputs signals are cos, sin, deltz and en_o_post, where deltz is the output of Z, which means the difference between the input angle and the remainder angle which is not computed, en_o_post means the whole computing has been finished. It can be found that CORDIC architecture can calculate the sine and cosine values at the same time.

4.2. Full custom

The whole architecture includes additions, registers and multiplexers. The overall delay of the angle computation circuit is determined by the wire delay, the sub-adder circuit and the register circuit. The most important part of the overall delay is the sub-adder circuit. In this paper, we choose the carry look-ahead adder to implement its function. The carry look-ahead adder is similar to the carry-skip adder, but computes group generate signals as well as group propagate signals to avoid waiting for a ripple to determine if the first group generates a carry. The sub-adder (4 bits) layout is showed in Fig.3.

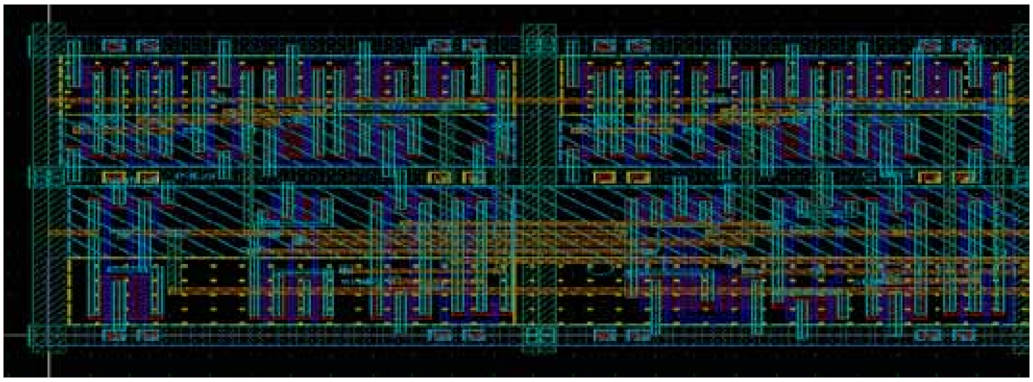

Fig. 3. Sub-adder (4 bits) layout

The critical path delay is 0.639 ns, which includes the delay of two adders and modified registers with 24 transistors, two sub-adders delay. Wire delay which approximates twice or three times as the critical path delay should be added in practice, which means the whole design delay approximates 2.556ns and the frequency can reach 391MHz approximately.

Table 2 shows the comparison of the performance between the proposed in full custom and other techniques published in various references.

Table 2. Performance comparison

	Reference [4]	Reference [2]	Reference [2]	This paper	
Technology(nm)	90	130	130	130	130
Scaled	Radix-4	Radix-2	Radix-2	Radix-2	Radix-2
Implementation	FPGA				Full custom
Target device	XC3S1500-4FG676	EP1C6Q240C8	EP1C6Q240C8	EP1C6Q240C8	Area
Total logic elements	/	4287	603	3741	605284μm^2
Occupancy rate (%)	/	72	10	63	
Architecture	pipeline	pipeline	iteration	Pipeline	Pipeline
Stages		34	1	34	18
Fmax (MHz)	71.2334MHz	134.10	77.37	126.34	391
Delay time(ns)	14.039ns	7.457	12.925	7.915	2.556

Bits precision	20	32	32	32	32

It can be observed in Table 2 which the full custom implementation has better performance than the other designs based on FPGA. We can realize the trigonometric computing in high precision, real-time and small area. Fig.4 shows the full custom layout of CORDIC datapath and the full custom layout is symmetric and organized from the one more detail part of the overview.

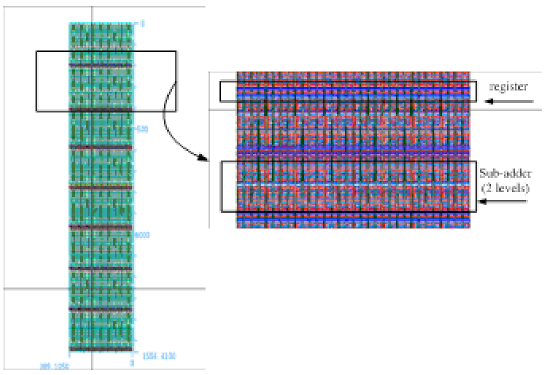

Fig. 4. CORDIC datapath layout

5. Conclusion

A 32 bits radix-2 CORDIC datapath is implemented by full custom design methodology. The corresponding latency of the datapath is 18 clock cycles. The critical path delay is 2.556ns approximately. As a future work based on full custom implementations, it is worthy to involve the dynamic circuits for faster speed.

Acknowledgements

The authors would like to acknowledge the financial support by Shanghai-AM Fund under Grand No.09700714000

References

[1] J. S. Walther. A unified algorithm for elementary functions in Proceedings of Spring[J]. Joint Computer Conference,1971,p: 297-385.

[2] Shiping CHEN, Quan Li, The implementation of 32 Bits Floating sin & cos Functions with FPGA[J], Control and Automation,2008(24), p:176-178

[3] M.G. Buddika Sumananesea. A scale factor correction scheme for the CORDIC algorithm[J]. IEEE Transactions on Computers, 2008(8), pp: 1148-1152

[4] Kaushik B. Rakesh B.. Architectural design and FPGA implementation of radix-4 CORDIC processor[J]. Microprocessors and Microsystems, 2010,p:96-101