Implementation of High Accuracy Trigonometric Function on FPGA by Taylor Expansion

Duoli Zhang [a], Jingju Yu [b] and Yukun Song [c]

Institute of VLSI Design, Hefei University of Technology, Hefei 230009, China

[a]zhangduoli@hfut.edu.cn, [b]yujingju@mail.hfut.edu.cn, [c]songyukun@hfut.edu.cn

Abstract. High-precision implementation of trigonometric function has been important in navigation, engineering, physics and other modern DSP. It presents an implementation of trigonometric function based on Taylor expansion in this paper. This means can calculate the value of trigonometric function at any angle. And the means takes Taylor expansion in four subsections based on the error analysis of Lagrange remainder. The input and output values are radians which meet 32-bit single-precision floating-point numbers of IEEE-754 standard. The algorithm is achieved by an iterative loop hardware circuit on FPGA chip. And the hardware architecture improves the efficiency of data processing by batch processing which is based on register reuse. The results show that this means uses less hardware resources and the accuracy of this circuit reaches 10-6.

Keywords: Taylor expansion; trigonometric function; iterative loop; FPGA.

1. Introduction

Trigonometric function plays an important role in navigation, communication and other modern digital signal processing. And the common implemented methods include look-up table method (LUT), the Taylor series expansion method and coordinate rotation algorithm (CORDIC) [1]. Look-up table method is simple, but the memory consumption is expanding sharply with the increasing of the accuracy requirement. CORDIC algorithm can satisfy a variety of functions, and the precision of CORDIC algorithm is adjustable. With the increase of the accuracy, the iteration number will increase or the pipeline stages will deepen, which leads to the increase of resource consumption and output delay [1,2]. The traditional Taylor series expansion method occupies a lot of resources [3].

A new Taylor expansion method proposed in this paper, whose accuracy reaches 10^{-6}, can calculate the value of trigonometric function at any angle. And it takes less hardware resources. The new method could highly meet the requirements of precision and resource in digital signal processing.

2. Algorithm Principle

The means proposed in this paper takes Taylor expansion in several subsections on average based on the error analysis of Lagrange remainder. And it takes Taylor expansion in the center of each section.

If the function *f(x)* is *n* order continuous derivable in the closed interval *[a, b]*, and is *n*+1 order derivable in the interval *(a,b)*, for any $x \in [a,b]$, the Taylor expansion of the function *f (x)* at $\boldsymbol{x_0}$ shows as[4]

$$f(x) = \frac{f(x_0)}{0!} + \frac{f'(x_0)}{1!}(x - x_0) + \cdots + \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n + R_n(x) \cdot \tag{1}$$

Where $R_n(x)$ is the remainder term of Taylor Formula, and is the higher order infinitesimal of $(x - x_0)^n$.

Lagrange Remainder is

$$R_n(x) = \frac{f^{(n+1)}(\varepsilon)}{(n+1)!}(x - x_0)^{n+1}. \tag{2}$$

Because the absolute value of trigonometric function is less than 1, the Lagrange remainder of trigonometric function Taylor expansion shows as

$$R_n(x) < \frac{1}{(n+1)!}\left(x - x_0\right)^{n+1} . \tag{3}$$

Since the trigonometric function is periodic and symmetric, the curve of trigonometric function in the domain $(-\infty,+\infty)$ can be obtained by the transformation of curve translation and rotation in the $[0,\pi/2]$. The domain of trigonometric function can normalize to $[0,\pi/2]$ by using this nature with the formula $\left|x - int\left(\frac{x}{n}\right) * \pi\right|$, which is conducive to the realization of the hardware.

Supposing the interval $[0,\pi/2]$ is divided into s segments on average, and in each section there are

$$\left|x\text{-}x_0\right| < \frac{\frac{\pi}{2}/s}{2} . \tag{4}$$

Which is

$$\left|x - x_0\right| < \frac{\pi}{4s} . \tag{5}$$

Assuming the error accuracy is 10^{-L}, then it can be get

$$R_n(x) < \frac{1}{(n+1)!}\left(\frac{\pi}{4s}\right)^{n+1} < 10^{-L} \quad \Rightarrow \quad s > \frac{\pi}{4}\frac{\sqrt[n+1]{10^L}}{\sqrt[n+1]{(n+1)!}} \tag{6}$$

In order to meet the accuracy requirement in digital signal processing, the maximum error is set to 10^{-7} considering the standard of single-precision floating-point numbers of IEEE-754. Table 1 shows the relationship between the series of Taylor expansion and the number of segments in $[0,\pi/2]$ [5].

Table 1 The relationship between Taylor series and segments

Taylor series	Segments
3	20
4	8
5	4
6	3
7	2

Taking into account computing resources, storage resources and other constraints, this design chooses the 5 level Taylor expansion, and divides the interval $[0,\pi/2]$ into 4 sections on average.

3. FPGA Implementation of Trigonometric Function

3.1 Algorithm Implementation.

According to the principle of Taylor expansion and the selection of Taylor series, the Taylor expansion formula can be transformed into

$$f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5 = a_0 + x\left\{a_1 + x\left[a_2 + x\left(a_3 + x\left(a_4 + xa_5\right)\right)\right]\right\} . \tag{7}$$

It is obviously that $b+ax$ is the basic processing element (PE). And the algorithm can be realized by the iterative loop of PE. The whole algorithm implementation circuit includes pre-processing unit and processing element.

Pre-processing unit converts the original data into interval $[0,\pi/2]$ according to $\left|x - int\left(\frac{x}{n}\right) * \pi\right|$. It also judges which segment the angle belongs to, and takes Taylor expansion in the center of each section. Part of the pseudo code is as follows:

```
if(data from 0 to pi/8)
    First paragraph();
    X=data-pi/16;
else if(data from pi/8 to pi/4)
    Second paragraph();
    X=data-3*pi/16;
else if(data from pi/4 to 3*pi/8)
    Third paragraph();
    X=data-5*pi/16;
else if(data from 3*pi/8 to pi/2)
    Fourth paragraph();
    X=data-7*pi/16;
```

Processing element is composed of a multiplier, an adder, a source data register set and a coefficient register set. Circuit diagram of processing element shows in Fig. 1. Multiplier selects *a5* or the result of the last iteration output to the port mul_a based on the number of iterations and segment that source data belongs to. And mul_b takes the source data from the source data register set at the same time. The result of multiplication *ax* output to the add_c of the adder. And add_d takes the corresponding coefficient a_i (i=0,1,2,3,4) from the coefficient register set based on the number of iterations and segment that source data belongs to. The result of addition *b*+*ax* is outputted. One iterative operation is completed. From the (7), it is known that the final results of trigonometric function Taylor expansion can be obtained by looping iteration 5 times.

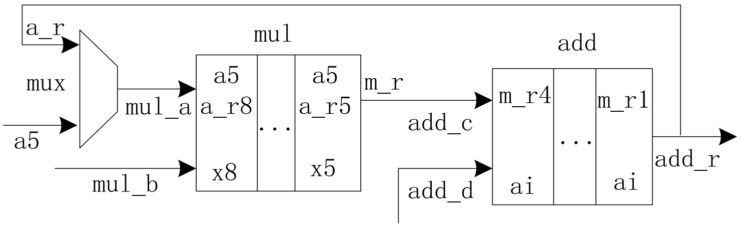


Fig. 1 Circuit diagram of PE

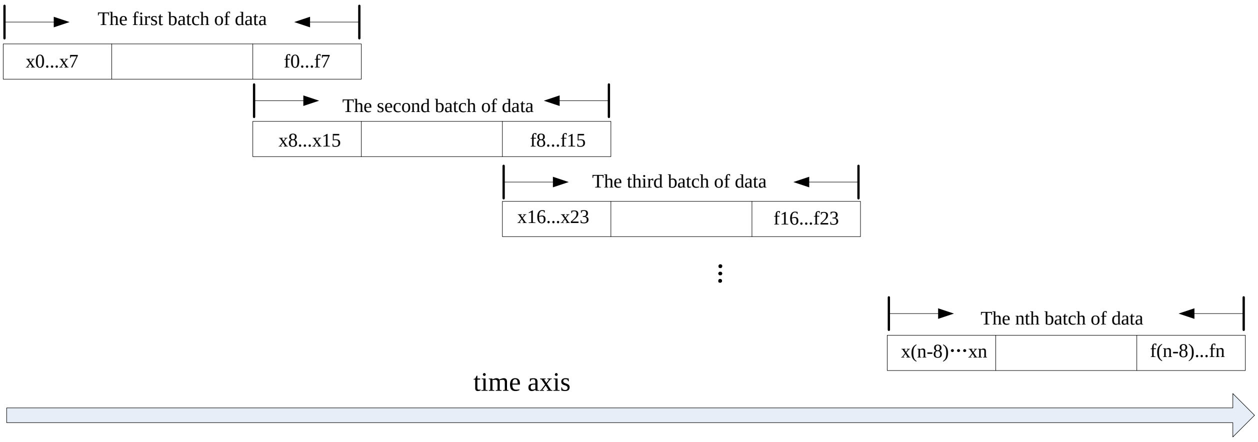


Fig. 2 Timing diagram

Multiplier and adder are set to four-level pipeline in this paper. In order to improve the efficiency of data processing, the processing element processes 8 data each batch through register reuse. And every level pipeline of multiplier and adder work effectively. As the timing diagram shows in Fig. 2, when the results of the batch output, the next batch of source data input at the same time. And it hides the data transmission time. The circuit works effectively all the time.

3.2 Experimental Results and Analysis.

The circuit is completed in Verilog HDL language. And this design is implemented on Xilinx Virtex-5 FPGA family. Synthesized results are shown as Table 2. The maximum frequency of the system is 182.714MHz.

Table 2 Hardware resource consumption

Slice Logic Utilization	Used	Available	Utilization
Slice Registers	1698	69120	2%
Slice LUTs	2126	69120	3%
DSP48Es	6	64	9%

The one PE takes 55 cycles to complete one batch calculation. It is equivalent to get a result data every 7 cycles. From the Fig. 3, it is known that the greater the amount of source data, the shorter the average calculation time of each data. When the amount of data tends to infinity, the average calculation period tends to be 5.88 cycles.

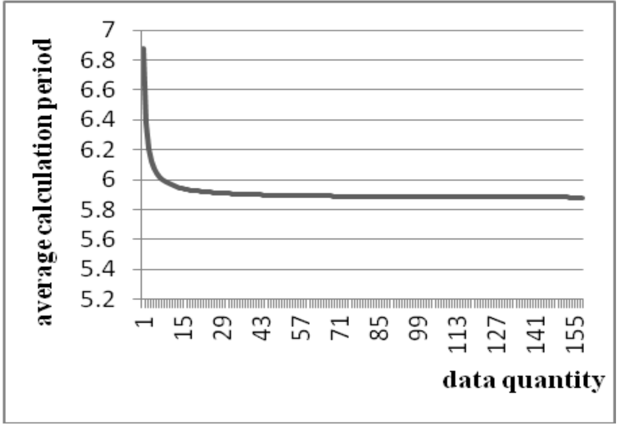


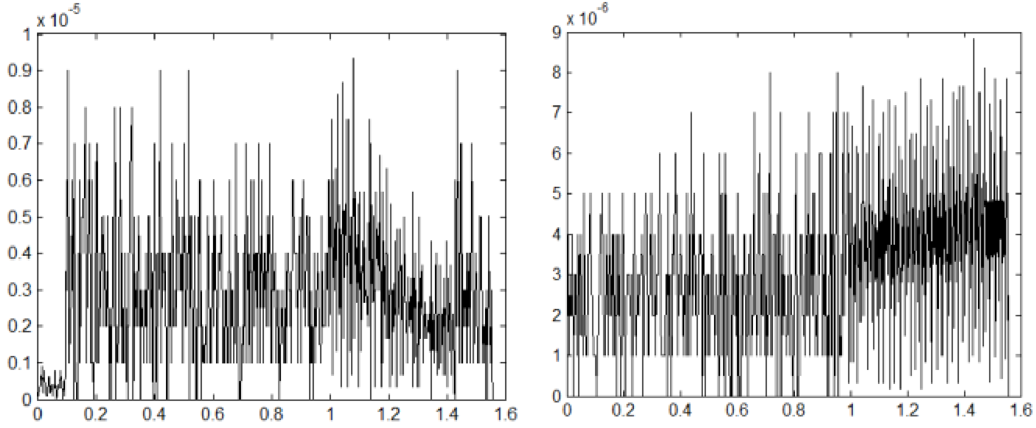
Fig. 3 Average calculation period

Table 3 is the resource consumptions comparison of CORDIC algorithm, LUT + CORDIC algorithm and Taylor series expansion method proposed in this paper on registers, multipliers and adders. Means in this paper uses less resource than CORDIC algorithm and LUT + CORDIC algorithm. DSP48Es reduce by 96.7% than “CORDIC”, and 75% than “LUT+CORDIC”.

Table 3 Comparison of resources

Algorithm	Registers	DSP48Es
CORDIC[6]	11168	64
LUT+CORDIC[7]	347	24
Means in this paper	1698	6

Fig. 4 shows the absolute value of the actual error by comparing the results of the FPGA with the results of MATLAB. The maximum error of sine function is 9.3*10^{-6}, and the maximum error of cosine function is 8.8*10^{-6}. The iterative loop structure can cause the accumulation of errors. The algorithm accuracy itself is 10^{-7}, and the accuracy of the circuit designed in this paper can reach 10^{-6}. It is quite high.



(a) Sine function (b) Cosine function

Fig. 4 Actual error of Taylor expansion algorithm

4. Conclusion

Compared with the common look-up table method and coordinate rotation algorithm, Taylor series expansion method that this paper proposed can calculate the value of trigonometric function with high accuracy, which reaches 10^{-6}, with less resource. It can meet the requirements of resources and precision in modern digital signal processing, and has a strong engineering practical value.

References

[1]. Shengmei Mou, Zhaogang Li. A Computation Method of High-Accuracy Sine & Cosine Function Based on Lookup-table and SF CORDIC Algorithm. Computer & Digital Engineering. Vol.42 (2014) N0.3, p. 359-363.

[2]. R. Andraka. A survey of CORDIC algorithms for FPGA based computers. In Sixth International Symposium on Field-Programmable Gate Arrays.1998, p. 191–200.

[3]. Shanying Xie. Implementation of Arcsine Function Based on FPGA. Chinese Journal of Electron Devices.Vol.33 (2010) No. 3, p. 344-347.

[4]. Xiaoming Liu,Yi Hong. Implementation of Sine & Cosine Function Based on Look-up Table and Taylor Expansion. Modern Electronics Technique. Vol.42 (2009) No. 13, p. 165-166.

[5]. Shaoling Si, Yong Guan. Determination of optimal power of data fitting to characteristic curve of trigonometric function. Computer Engineering and Design. Vol.27 (2006) No. 24, p. 4660-4662.

[6]. Qingda Wu. Implementations of 32-bits Fixed and Floating Point Trigonometric Function with FPGA. Microelectronics & Computer. Vol29 (2012) No. 1, p. 113-116.

[7]. Keyang Chang. Application of CORDIC algorithm in sine & cosine function and its implementation on FPGA. Computer Engineering and Applications. Vol49(2013)No. 7, p. 140-143.