International Journal of Advanced Science and Research
ISSN: 2455-4227
Impact Factor: RJIF 5.12
www.allsciencejournal.com
Volume 2; Issue 5; September 2017; Page No. 86-89

A synthetic division algorithm with positive remainder

S Subha
Department of Information Technology and Engineering, School of Information Technology and Engineering,
Vellore Institute of Technology, Vellore, Tamil Nadu, India

Abstract
Synthetic division gives negative remainders in certain cases. This paper proposes to have zero or positive remainder for synthetic division in accordance with Euclid division algorithm. In case of negative remainders in synthetic division, the remainder is made zero or positive by suitably adding or subtracting the divisor. The quotient is adjusted suitably. The synthetic division algorithm propose in [8] is used. The inputs for the division can be positive or negative. The decimal notation is used for calculations. To the best of the author's knowledge this is the first paper to propose algorithm for positive and negative operands for division. The proposed algorithm is simulated in Quartus 2 Toolkit. An area of 42% with power consumption of 77.64mW and timing of 566.928ns is observed for the chosen simulation parameters.

Keywords: division, non-negative remainder, signed integers, synthetic division

1. Introduction

Arithmetic operations in computers can be for integers or floating point numbers. Integer operations are addition, subtraction, multiplication and division [3]. For division, two algorithms are commonly used. These are restoring division and non-restoring division [1, 3]. The algorithms take 2n bit dividend to be divided into n bit divisor giving n bit quotient and n bit remainder. For both the algorithms, the operands are assumed to be positive integers. The case of negative operands is handled by the following rule [9].

1. Perform the division of positive operands only.
2. Sign of remainder is sign of the divisor
3. Sign of the quotient is product of the signs of divisor and dividend.

Some applications of division algorithm are discussed in [4, 5]. Synthetic division using systolic arrays is proposed in [7]. Algorithm using synthetic division with modifications for obtaining positive remainder for positive operands is proposed in [8]. This paper considers division of integers. The range of numbers is limited by the hardware chosen. The dividend is 2n bits and divisor n bits. The numbers can be positive or negative. The synthetic division is performed on the input by converting it into polynomial of power of two. The remainder can be negative in synthetic division. This condition is corrected by suitably adding/subtracting the divisor from the remainder and adjusting the quotient to obtain positive remainder. The remainder can be greater than the divisor due to this. The synthetic division model proposed in [8] is adapted in this paper. The various cases of dividend and divisor being non-negative and negative are considered. The remainder is made positive by adding or subtracting the divisor number of times say y. The quotient is adjusted with y. The operands are assumed to be in decimal notation in Modelism software. Algorithms are proposed for this. The proposed model is simulated with Quartus2 Toolkit. To the best of the knowledge of the author, the method is the first of its kind. The input is considered to be eight bit dividend, four bit divisor giving eight bit quotient and eight bit remainder in synthetic division. Registers are defined for this. An area of 42% with power consumption of 77.64mW and timing of 566.928ns is observed for the chosen simulation parameters.

The rest of paper is organized as follows. Section 2 gives mathematical background, section 3 proposed algorithm, section 4 simulations, section 5 conclusions followed by references.

2. Mathematical Background

The divrem problem proposed by Euclid [2] is as follows. Given two bignums $A=(a_{m-1,},a_{m-2},...,a_1,a_0)$ and $D=(d_{n-1},d_{n-2},...,d_1,d_0)$with m>n, find bignums Q and R such that A=QD+R with 0<=R<D. The bignums are assumed to be positive.

Consider the synthetic division method [6]. In this method a polynomial is divided into binomial. The steps in this method for dividing polynomial with (x-k) where k is a constant is given below.

1. Write the coefficients of the numerator in descending order of exponent
2. Bring down leading coefficient to the bottom row.
3. Multiply k by value just written in bottom row.
4. Add the result of step 3 to next exponent in the input
5. Continue steps 3-5 till the last exponent i.e. exponent zero
6. The result is the polynomial of degree n-1 with first n-1 coefficients and remainder is the last term of the result in the bottom row.
7. Stop

Table 1: Thus to divide $ax^3 + bx^2 + cx + d$ by x-k the following pattern is used

k	a	b	c	d
		ka		
	a	b+ka		
				Remainder

The first row consists of k and the coefficients of the dividend. In the vertical pattern terms are added. The diagonal terms are multiplied by k. The remainder can be negative in this division and greater than k. The following gives the example of synthetic division.

Table 2: Dividend $x^4 - 10x^2 - 2x + 4$ Divisor x+1

-1	1	0	-10	-2	4
		-1	1	9	-7
	1	-1	-9	7	-3
					Remainder

Answer is quotient = $x^3 - x^2 - 9x + 7$ Remainder: -3. The remainder can be positive or negative in opposition to Euclid theorem in this method. The modified synthetic division proposed in [8] aims at the remainder being positive. Consider the division of polynomial with binomial. Division is performed using synthetic division. Consider the division of eight bit number into four bit number. Both the numbers are signed. The following are the steps to perform the division. Only magnitudes are considered for calculations.

Algorithm Modified Synthetic Division

Given dividend of eight bits and divisor of four bits, this algorithm performs the division using synthetic division concept. The result is eight bit quotient and four bit remainder.

1. Start
2. Express the divisor as the sum of power of two and constant. This can be done from the look up table.
3. Express the dividend as combinations of sum, differences of powers of two and constant. This is assumed to be given as algorithm input.
4. Put the power of two in the denominator as baseval, a variable say y.
5. Express the numerator in terms of y
6. Let the resultant expression be c/(y-d), c is polynomial in y.
7. Perform synthetic division of c/(y-d). Keep track of number of times a zero value is present in input, say count, in this process to the various coefficients, k.
8. Express the result coefficients in terms of baseval. This is quotient q.
9. Let the remainder be r. Let z = baseval + d

a. If the absolute value of r is less than baseval and r is positive, stop.
b. if the absolute value of r is less than baseval and r is negative r=r+z. q=q-1. Stop
c. If r is negative, t = absolute value of r. If r> z, r=k*z + r, q = q-k. Stop.
d. If r is positive and greater than z, r = r-kz, q = q+k. Stop.

10. Quotient = q, Remainder =r
11. Stop

The algorithm makes the remainder positive unlike synthetic division algorithm. This is done in step 9 of the algorithm. The time complexity of above algorithm is equal to the order of degree of polynomial which is seven in this particular case. For storing the polynomial coefficients, the results of the modified synthetic division as described in MODI_SYNTHETIC_DIVISION, separate registers are required. Separate registers to store baseval, z, k are required. The algorithm is scalable. The total time is O(degree of polynomial+constant). The correction performed in step 9 of the algorithm, leads to correct solution.

3. Proposed Model

Consider the synthetic division of two integers. Let D be the dividend, V the divisor, Q the quotient, R reminder. The operands can be positive or negative. The synthetic division proposed in [8] is performed. The operands are 2n bit dividend and n bit divisor. Consider the remainder. According to Euclid division algorithm [2], the remainder is positive. The synthetic division sometimes yields negative remainder. To make the remainder positive, the following steps are performed.

Algorithm Synthetic Division with Positive Remainders: Given the quotient of synthetic division as quotient and remainder as remainder, this algorithm makes the remainder positive if necessary.

1. Start
2. ct = 0; remainder1 = remainder
3. do steps 4 to 6 while (remainder1 < 0)
4. if (divisor > 0) remainder1 = remainder1 + divisor
5. if (divisor <0) remainder1 = remainder1 - divisor
6. ct = ct + 1
7. If divisor > 0

a. if remainder <0, quotient = quotient + ct

8. If divisor < 0

a. if remainder > 0, remainder > divisor, quotient = quotient + ct

b. if remainder < 0, remainder < divisor, quotient = quotient + ct

9. Quotient is in quotient and remainder is in remainder1.
10. stop

In the above algorithm it is possible that the remainder is greater than the divisor as it has to be non-negative. The time complexity of the above algorithm is calculated as follows. The synthetic division algorithm has time complexity equal to the order of degree of polynomial which is seven in this particular case. The steps to make the remainder positive takes ct addition times and one subtract/add time for quotient adjustment. The total time taken is sum of the two time units.

Examples

Consider 43 as dividend and 3 as divisor. The dividend is written as (32+8+2+1) and the divisor as (2+1). Put y=2. The dividend is $y^5 + y^3 + y + 1$ and the divisor is y+1.

1. Positive operands

Consider 43/3. It is resolved as $\frac{y^5 + y^3 + y + 1}{y + 1}$ where y=2

By synthetic division quotient = 15 remainder = 2. Remainder is positive. Hence stop.

2. Positive dividend and negative divisor

Consider 43/(-3). It is resolved as $\frac{\left(-y^5 - y^3 - y + 1\right)}{\left(y - 1\right)}$ where y=-2. By synthetic division the quotient -15 and remainder is -2. The divisor is negative. So, remainder=remainder-divisor once. The result reminder is 1 and quotient is adjusted to -15+1= -14.

3. Negative dividend and positive divisor

Consider (-43)/3. It is resolved as $\frac{-\left(y^5 + y^3 + y + 1\right)}{y + 1}$ where y=2. By synthetic division the quotient is -15 and remainder is 2. The remainder is positive. Hence stop.

4. Negative dividend and negative divisor

Consider (-43)/(-3). It is resolved as $\frac{\left(y^5 + y^3 + y - 1\right)}{\left(y - 1\right)}$ where y=-2. By synthetic division quotient is 15 and remainder is 2. The remainder is positive. Hence stop.

For division by powers of two, the magnitude of dividend is shifted by suitable amount of positive divisor and the sign bit of dividend is XOR of signs of dividend and divisor. The remainder is positive. Hence there is no correction for the remainder. This is true for both positive and negative operands. The following examples illustrate this.

1. Divide 43/2. This is 0101011/010. This gives quotient as 0010101 and remainder one.
2. Divide -43/2. This is 1101011/010. The quotient is 1010101 and remainder is one.
3. Divide 43/-2. This is 0101011/110. This gives quotient as 101010 and remainder as one.
4. Divide -43/-2. This is 1101011/110. This gives quotient as 1010101 and remainder asone.

4. Simulations

The proposed model is simulated in Quartus2 Toolkit. The details of the configurations are given in Table 3. The inputs are eight bit dividend, four bit divisor.

Table 3: Simulation Parameters

Parameter	Value
Processor Family	Cyclone 2
Package	FBGA
Pin Count	484
Speed grade	Fastest
Target Device	Auto select device by Fitter

There are sixteen combinations of the divisor as mentioned in [8]. The Table 4 gives the resolution of the divisor into sum/difference of power of two. It is observed that difference expressions give quicker less complicated results than sum expressions.

Table 4: Divisor Resolution

Value	Resolution
1	Not applicable
2	Shift right once
3	2+1 or 4-1
4	Shift right twice
5	4+1 or 8-3
6	8-2
7	8-1
8	8, shift right three times
9	8+1
10	8+2
11	8+3
12	8+4
13	16-3
14	16-2
15	16-1

The dividend has eight bits and has 256 combinations. It is assumed that the user resolves the dividend into polynomial in 2 based on the choice of the divisor resolution. The simulation results are shown in Table 5.

Table 5: Simulation Results

Parameter	Proposed Algorithm
Area(#slices/total slices)	6059/14448(42%)
Power	77.64mW
Timing	566.928ns

To the best of the knowledge of the author, this is the first paper to propose division involving positive and negative operands.

5. Conclusion

An algorithm to perform division using synthetic division for positive and negative operands is proposed in this paper. The input is 2n bit dividend, n bit divisor with n=4. The result is n bit quotient and n bit remainder usually. In this paper, the quotient and remainder are 2n bits. The algorithm expresses the dividend and divisor in polynomial of powers of two. The synthetic division is performed. The remainder is corrected to be positive with proper adjustment in the quotient. Because of the constraint that the remainder is non-negative it is possible that the remainder is greater than the divisor. The proposed model is simulated with Quartus2 toolkit. An area of 42% with power consumption of 77.64mW and timing of 566.928ns is observed for the chosen simulation parameters. To the best of the knowledge of the author, this is the first paper to assume positive and negative operands for synthetic division resulting in non-negative remainders.

6. References

1. Behrooz Parhami. Computer Arithmetic: Algorithms and Hardware Design, Oxford University Press, 2000.
2. Burton, David M. Elementary Number Theory. McGraw-Hill, 2010
3. Israel Koren, Computer Arithmetic Algorithms, Prentice Hall, NJ, 1993.
4. Josue Saenz S, Juan J, Raygoza P, Edwin C, Susana Ortega Cisneros, Jorge Rivera Dominguez. FPGA Design and Implementation of Radix-2 Fast Fourier Transform Algorithm with 16 and 32 Points, Proceedings of ROPEC (FFT), 2015.
5. Li R, S Y. Fast Constant Division Routines, IEEE Transactions on Computers. 1985; 34(9):866-869.
6. Rajesh Pandey. A Textbook of Engineering Mathematics, WordPress Publications, First Edition, 2010, 1.
7. Stanislaw Zak H, Kai Hwang. Polynomial Division in Systolic Arrays, IEEE Transactions on Computers, 34(6):2985.
8. Subha S. A Modified Synthetic Division Algorithm, IJCAM. 2017; 12(3):691-697.
9. William Stallings, Computer Architecture and Organization, Prentice Hall, 2010.