International Journal of Advanced Science and Research
ISSN: 2455-4227
Impact Factor: RJIF 5.12
www.allsciencejournal.com
Volume 3; Issue 1; January 2018; Page No. 01-02

An improved carry save adder design

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

Abstract
Binary addition is performed using full adders. Carry save adders are studied in literature and have wide applications in numerical applications. This paper proposes binary addition algorithm. The input is two 32-bit numbers. The sum and carry of individual bits are calculated. The results are added according to weightage to the bits using modified full adder logic. The result is n+1 bit number. The proposed model is simulated with Quartus2 toolkit and compared with carry save adder logic. An improvement in area by 27% with power saving of 8.5% with timing improvement of 6.8% in proposed model compared with traditional model is observed.

Keywords: carry save adder, full adder, performance improvement

1. Introduction

Computer ALU performs integer and floating point operations [1]. The integer operations are add, subtract, multiplication and division [2, 3]. Integer addition is performed using full adders. A full adder has three inputs and two outputs. The outputs are sum and carry. Various full adder circuits/models are proposed in literature. Some of them are carry skip adder, ripple carry adder, carry select adder, Carry save adder, parallel prefix adder [2].

In carry save adder, two n-bit numbers are added to give sum and carry. The sum and carry of individual bits are calculated separately. The result is added according to their weights to give the result. Wallace tree is one method to implement carry save adders [2]. This paper proposes method to perform addition of two n-bit numbers using carry save addition principle and modified full adder circuit. The proposed model is simulated using Quartus 2 tookit. An improvement in area by 27% with power saving of 8.5% with timing improvement of 6.8% in proposed model compared with traditional model is observed.

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

2. Mathematical Background

Consider full adder. It takes three inputs A, B, C and gives two outputs Sum and Carryout. The truth table of full adder is given below in Table 1.

Table 1: Full Adder Truth Table

A	B	C	Sum	Carryout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

From Table 1, the expressions for sum and carryout are given below:

$$\text{Sum} = a \text{ XOR } b \text{ XOR } b$$
$$\text{Carryout} = ab + bc + ca \quad (1)$$

The carry save adder is one method to implement the adder circuit. It takes two n-bit numbers and b. The sum and carry of individual bits of the inputs is calculated. The result of bits of same weights is added using full adders to get final result. As an example consider two 4-bit numbers 1011 and 1001. The sum is 0010 and carry is 1001. The result is added as 0010 + 10010. The final sum is 10100. This is represented below.

A: 1 0 1 1
B: 1 0 0 1
Sum: 0 0 1 0
Carry: 1 0 0 1

Now add sum and carry according to weights of bits.

Sum: 0 0 1 0
Carry: 1 0 0 1
Result: 1 0 1 0 0

3. Proposed Model

Consider two n-bit numbers a and b. Find the sum and carry of bits in a and b. Let them be sum and carryout. Add the sum and carryout according to bit weightage. The following algorithm is used for addition.

Algorithm Modified Carry Save Adder Algorithm: Given two n- bit numbers this algorithm gives the sum of the two input numbers a and b.

1. Start
2. Do steps 3-5 for each bit in the inputs.
3. Intiialize sum = b, carry = b
4. If a = 1, sum = !b
5. If a = 0, carry = 0
6. Pair the sum and carry calculated according to their weights.
7. Do steps 7-8 for each set of inputs (sum, carry, cin) where cin is the carry from previous position bits addition.
8. The result and carryout from the addition of (s, carry, cin) is given by the Table1. The Sum and cout give the result and carryout of the operation. The cout is calculated before sum. The addition algorithm is adapted from [4]. The carryout from bit-i is the cin in bit i+1 position.

Table 2: Full Adder Logic

S	Carry	Cin	Sum	cout
S	1	0	!s	s
S	1	1	S	1
S	0	0	S	0
S	0	1	!s	s

9. The bits in position 0 and n need special attention. In bit zero position, there is no input carry cin and Carry. The result is the value of S. In bit position n, there is no S input. The result is the sum of Carry and cin.
10. Stop: Initializations in steps 3-5 and initialization of result in step 7 to Sum reduces the computational time. For n-bit addition, it takes O (n) for calculating the initial sum and carry. It takes O (n) for performing step 7 in algorithm above. The overall time complexity is O (n).

Example: The steps in calculating sum of inputs 1011 and 1001 is given below:

A = 1 0 1 1
B = 10 0 1
Sum = 0 0 1 0
Carry: 1 0 0 1
Result [0] = sum [0] = 0
Result [1] = add (1, 1, 0) Here sum = !1 = 0 and cout = s = 1
Result [2] = add (0, 0, 1). Here sum = !s = 1 and cout = s = 0
Result [3] – add (0, 0, 0). Here sum = s= 0 and cout = 0
Result [4] = add (1, 0). Here sum = 1 and cout = 0
Hence result is 10100 with cout = 0.

4. Simulations

The proposed model was simulated in Quartus2 Toolkit. Input of 32-bits was used for simulation. The circuit was written in Verilog and synthesized. The logic of the code is given next. Let the inputs be a and b. The sum is initialized to the b input. The carry input is initialized to the sum input. If the a input is one, the sum is negated (taking XOR). If the a input is zero, the cinput is made zero. Consider the i-th bit position. If the carry is one, cin is zero, carryout is sum and sum is not(sum). If the carry is one, cin is one, carryout is one. If the carry is zero, carryin is zero, cout is zero. If the carry is zero, carryin is one, carryout is sum and sum is not (sum). The end bits are special cases. The circuit was compared with carry save adder. The carry save adder model is called traditional model. The simulation results are shown in Table 3. As seen from Table3 there is improvement in area by 27% with power saving of 8.5% with timing improvement of 6.8% in proposed model compared with traditional model.

Table 3: Simulations Results

Parameter	Traditional	Proposed	%improvement
Area	136/14448	99/14448	27.02%
Power	85.80mW	78.50mW	8.5%
Timing	24.549ns	22.863	6.86%

5. Conclusion

An improved algorithm for carry save adder is proposed in this paper. The algorithm initializes the results in various steps to one of the operands before computation. The full adder algorithm proposed in [4] is used to calculated the final sum. The proposed model is simulated using Quartus2 Toolkit. An improvement in area by 27% with power saving of 8.5% with timing improvement of 6.8% in proposed model compared with traditional model is observed.

6. References

1. Behrooz Parhami. Computer Arithmetic Algorithms and Hardware Design, Oxford University Press, 2000.
2. Israel Koren. Computer Arithmetic Algorithms, Prentice Hall, NJ, 1999.
3. Morris Mano, Computer System Architecture, 3rd Edition, Prentice Hall, 2007.
4. Subha S. A Gate Level Full Adder Design with Power Saving, IJAER. 2015; 10(17):38384-38386