Constant time Floating Point Adder Circuits

Yuranan Kitrungrotsakul, Prabhas Chongstitvatana
Department of Computer Engineering
Chulalongkorn University
yuranan.kit@student.chula.ac.th, prabhas.c@chula.ac.th

***Abstract*— Floating point unit is commonly used in computers. However, the arithmetic logic unit (ALU) for floating point operations such as addition or subtraction is complicated. Moreover, many floating point ALUs are designed to operate in many clock cycles. Thus, its speed are varied depend on the number of clock cycle. However, in many field, circuits with fix delay time are preferred. This work proposes a constant time floating point adder circuits implemented in Field Programmable Gate Array technology.**

Keywords—floating point operations; find first bit value 1; adder circuits

I. MOTIVATION

The floating point adder is one of very widely used circuits in computing. The most common data format is IEEE 754. There are so many designs even recently, for example, one design for speed [1] another for small area [2]. There are many tradeoff in designing the circuits, mostly on area versus speed [3]. The standard high speed design with custom macro modules is presented in [4]. The design proposed here aims to achieve a constant time delay which makes it very useful for real-time embedded devices. The proposed design achieved its goal using a good binary-search for the first bit value 1 which takes O(log n) time. The design uses IEEE 754 64-bit format with 52-bit fraction. The design composed of only combination circuits.

II. FLOATING POINT ADDER BEHAVIORAL DESCRIPTION

In order to add or subtract floating point together, first the exponents are aligned. If the exponent bits of two data are not equal, the smaller number will increase its exponent until it is equal the other number. Then the fractions bits of the smaller number will change depended on the amount of exponent change. After this de-normalization, two numbers can be added or subtracted depending on the sign bit of operands, i.e. fraction bits. In the last step, the exponent bits will be realigned into IEEE 754 double precision format, so the fraction bits will change too. The realignment of exponents requires finding the first bit that is one in the exponent. The next section shows how to solve this problem.

III. FIND FIRST SET PROBLEM

Finding first bit value 1 can be solved with many algorithms. The major constraints are area and speed of circuits. The purpose of this circuits is a constant time circuits, so the speed will be the most important consideration. The divide and conquer method is used to efficiently finding the first 1 value bit from the bit string. The bit string is divided into 2 substrings with equal length, the most significant substring and the least significant substring. Then, repeat the dividing step with the most significant substring until it has only 2 bits. Then, compare the 2 bits with each other. If it is equal with 0 check another 2 bits with in the same substring generation. If it is equal with 1, the most significant bit is the first 1 value bit from bit string. If it is not equal, the higher one is the first 1 value bit from bit string. This method finds the first 1 value bit in O(log n). Moreover, the area of this method are O(2 log n – 1).

IV. DESIGN

The floating point adder circuits were implemented in hardware description language in order to operating in bit level. Moreover, it is implemented in combination circuits so it has a constant time delay. The design is based on the algorithm described in Section 2. The whole circuit is shown in Figure 1.

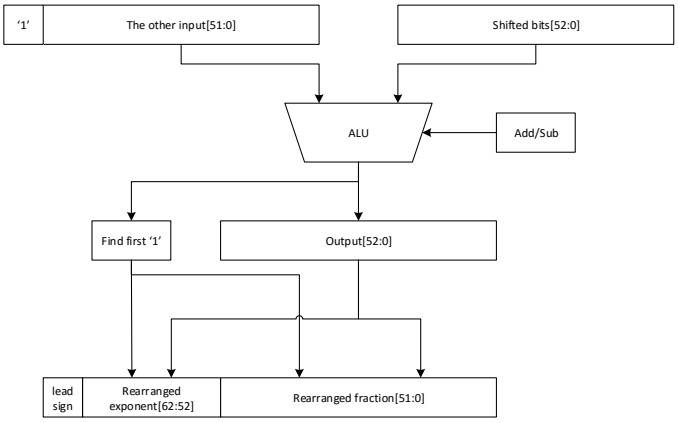

Fig. 1. Flow of whole circuits

Firstly, the operands will be compared with each other to find the smaller number. The smaller fraction bits are aligned by alignment circuits. Then, the operands will be added or subtracted depending on their sign bits. After that, the output will be realigned according to the first bit value 1 of ALU's output. The sign bit is dependent on the larger operand. The alignment circuits were achieved by logical shift circuits that are shown in Figure 2.

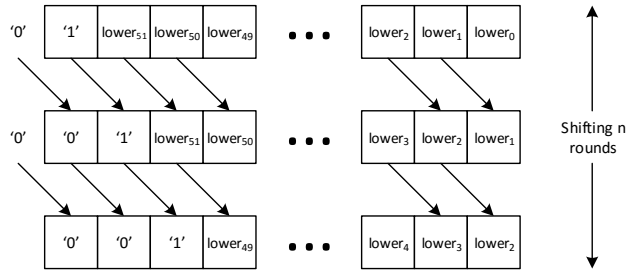

Fig. 2. Alignment circuits

Number of shifting rounds depends on the difference between operands exponent bits. The smaller exponent bits are changed to be equal to the larger exponent bits. Then, the smaller fraction bits are logical right shifted n rounds with one value bit concatenating at the most significant bit in the first round. Find first bit value 1 circuits in Section 3 were implemented by nested 2 to 1 multiplexers. It is shown in Figure 3.

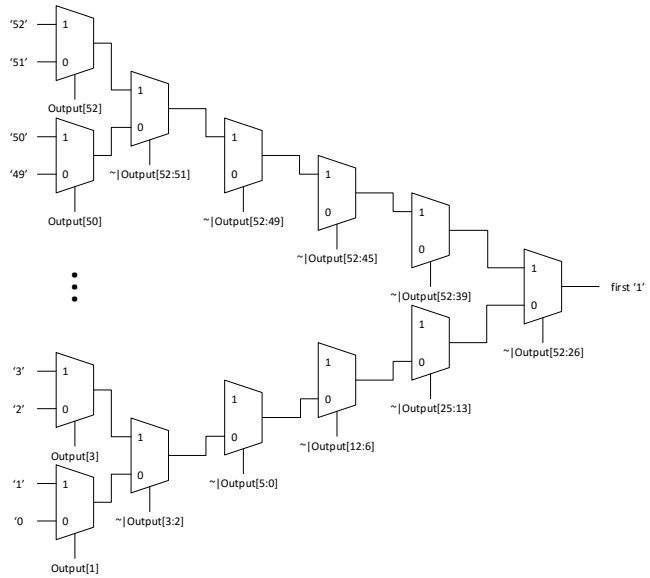

Fig. 3. Find first bit value 1 circuits

Cell name	Amount of CLB
2-Bit Look-Up-Table (LUT2)	113
3-Bit Look-Up-Table (LUT3)	560
4-Bit Look-Up-Table (LUT4)	1249
Inverter (INV)	5
Fast Multiplier AND (MULT_AND)	50
2-to-1 Multiplexer for Carry logic (MUXCY)	327
2-to-1 Look-Up Table Multiplexer (MUXF5)	343
XOR for Carry Logic (XORCY)	256
D Flip-Flop (FD)	8

Fig. 4. Design Summary

The synthesis result using Xilinx xc3s200 is reported in Figure 4. The large amount of resources is mostly consumed by LUT which implemented the alignment circuits and the find first one bit circuits. The speed of our floating point adder is shown in the critical path of 59 ns and has 75 levels of logic. The circuits can be operated at the maximum clock of 288 MHz.

V. Conclusion

After the synthesis that targets a Field Programmable Gate Array, the result shows that it is quite large in terms of area. However, it is fast. The technique proposed here can also be verified for its correctness easily.

References

[1] Sunesh, N.V., Sathishkumar, P., Design and implementation of fast floating point multiplier unit, Int. Conf. VLSI Systems, Architecture, Technology and Applications, 2015, pp.1-5

[2] Ehliar, Andreas, Area efficient floating-point adder and multiplier with IEEE-754 compatible semantics, Int. Conf. Field-Programmable Technology, 2014, pp.131-138.

[3] Malik, A., Dongdong Chen, Younhee Choi, Moon Lee, Seok-Bum Ko, Design tradeoff analysis of floating-point adders in FPGAs, Canadian Journal of E6lectrical and Computer Engineering, vol.33, issue 3/4, 2008, pp.169-175.

[4] Chi Huang, Xinyu Wu, Jinmei Lai, Chengshou Sun, Gang Li, A design of high speed double precision floating point adder using macro modules, Proc. Design Automation Conference, 2005, vol.2, pp.D/11-D/12.