Design a floating-point fused add-subtract unit using verilog

Mayank Sharma, Aswani Sengar and Prince Nagar

ECE Department, Sharda University, Gr. Noida, U.P., India

ABSTRACT

A floating-point fused add-subtract unit is described that performs simultaneous floating-point add and subtract operations on a common pair of single-precision data in about the same time that it takes to perform a single addition with a conventional floating-point adder. Placed and routed in 45nm process. So that there will be less consumption of memory as well as power.

INTRODUCTION

Much research has been done on the floating-point fused multiply add (FMA) unit [1]. It has several advantages in a floating-point unit design. Not only can a fused multiplier-add unit reduce the latency of an application that executes a multiplication followed by an addition, but the unit may entirely replace a floating point co-processor’s floating adder and floating-point multiplier. Many DSP algorithms have been rewritten to take advantage of the presence of FMA units in a given system with FMA systems. For example, in [4] a radix-16 FFT algorithm is presented that speeds up FFTs in systems with FMA units. High-throughput and digital filter implementations are possible with the use of FMA unit. FMA units were utilized in embedded signal processing and graphics applications, used to perform division, argument reduction, and this is why the FMA started to become an integral unit of many commercial processors such as IBM, HP[7] and Intel [9].

Similar to operation performed by a FMA in many algorithms in DSP and other fields both of the sum and difference of a pair of operands are needed for subsequent processing. For example, this is required in computation of the FFT & DCT butterfly operations. In traditional floating-point hardware these operations may be performed in a serial fashion which limits the throughput. The use of a fused adds subtract (FAS) unit accelerates the butterfly operation. Alternatively, the add and subtract may be performed in parallel with two independent floating point adders which is expensive.

This paper investigates implementation of floating point add-subtract unit this paper investigates the implementation of a floating-point fused add-subtract unit shown in figure 1. It performs the following operations:
2. Approach
There are two design approaches that can be taken with discrete floating-point adders to realize the add-subtract function. These are the parallel implementation shown in Figure 2 where two adders operate in parallel (one adding and one subtracting) and the serial implementation shown in Figure 3 where a single adder is used twice (once adding and once subtracting) with the same operands.

In a parallel conventional implementation of the fused add-subtract (such as that shown in Figure 2) two floating-point adders are used to perform the operation. This approach is fast, however, the area and power overhead is large because two floating-point add/subtract units are used.

In a serial conventional implementation of the fused add-subtract (such as that shown in Figure 3) one floating-point adder/subtractor is used to perform the operation in addition to a storage element to store the addition or subtraction result. This approach is very efficient in terms of area. However, due to the serial execution of both operations, the time needed to get both results is twice the time needed by the parallel approach. Also since a storage element is used, it adds slightly to the area and power overhead.

3. Verilog modeling
The fused add-subtract unit, the following floating-point unit were implemented in synthesizable Verilog-RTL

Fused floating add-subtract unit
The Verilog models were synthesized 45nm libraries. The area and the critical timing paths were evaluated. The
floating-point adder and fused add subtract unit were designed to operate on single precision IEEE Std-754 operands.

Figure 4. Floating-Point Fused Add-Subtract Unit [5].

Simulation Results
The Floating point fused add-subtract unit architecture which has been proposed earlier has been implemented and designed through Verilog (ModelSim) and XILINX ISE ISIM Simulator. The output results have been shown below in fig. 5 and fig.6. Here 2 floating point number has been performed through add and subtract operations in the following figure:

Figure 5: Input and Output block diagram
RESULTS

Here using the Xilinx XST tool for floating point fused add-subtract the following netlist has been generated which is shown in following figure 7:

4. Place and route
The floating-point adder and fused add-subtract unit were implemented using an automatic synthesize. Place and route approach. A high performance 45 nm process was used for the implementation with a standard cell library designed for high speed applications.
CONCLUSION

The new fused unit uses the IEEE-754 single-precision format and supports all rounding modes. The implementation results using a 45nm industry standard process and a simulations and synthesis results show that the fused primitives are faster, smaller, use less power and energy. Provide a slightly more accurate result.

Future research can focus on optimizing the fused add-subtract unit using multipath approaches. Last but not least, the fusing concept could be extended to other types of computation extensive applications and might result in delay, area and power consumption reduction.

REFERENCES

[16] www.cadence.com