Indian Journal of Science and Technology, *Vol 9(37), DOI: 10.17485/ijst/2016/v9i37/102131, October 2016*

ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Design and Implementation of Floating Point Unit using 15 nm FIFET

R. Dhanabal* and Sarat Kumar Sahoo

Department of Micro and Nano Electronics, VIT University, Vellore - 632014, Tamil Nadu, India;
rdhanabal@vit.ac.in, sksahoo@vit.ac.in

Abstract

Objectives: To design a 32-bit IEEE 754 floating point standard based MAC unit using 15 nm FinFET. In MAC unit given data is multiplied and accumulated in a register which are processed separately as exponent and mantissa but there are some issues regarding area power and timing, compromising these constraints is bit difficult. **Methods**: The proposed architecture of floating point MAC unit is majorly divided into multiplier and accumulator components in these blocks the sub blocks are designed with Han-Carlson adder, Vedic multiplier, barrel shifter and comparator. **Findings:** The previous work was done in Hardware Description Language (HDL) in which the design will map to CMOS or pass transistor components after synthesis so designing of each component with transistors of 15 nm FINFET becomes vital. The entire design is carried out in cadence virtuoso and layout editor for timing, area and power. The performance can be increased if the computations are performed with less number of transistors. However, the decimal operations have been limited due to the increase in cost and complexity of hardware components. DFP arithmetic is used for the complex computations, it consumes more power because of the area it occupied when implemented in hardware. **Improvements/Applications**: Because of battery driven property low power with high performance are given major importance.

Keywords: Barrel Shifter and Comparator, Han-Carlson Adder, Leading One Detector, MAC Unit (Multiplication and Accumulation Unit), Vedic Multiplier

1. Introduction

In digital signal processing, multimedia systems, convolution and filtering MAC unit plays an important role and manipulation of floating point numbers in that MAC unit will yield final results without loss of precision. The operations performed on the floating point numbers is bit complex compared to real numbers performing these computations consume large number of recourses in FPGA (Field Programmable Gate Arrays). FPGA has logic blocks and switching blocks which are configured while programming, any digital circuit can be implemented with FPGA while coming to ASIC (Application Specific Integrated Circuit) where dealing with power area and timing becomes vital. They reduce the resources consumed for multiplier and adder performance of MAC can be increased. To optimize the hardware consumed there are modifications in previous architectures. Previous works modified the architectures of multiplier from conventional multiplier to Vedic multiplier[1] or booth multiplier or Wallace tree multiplier in Figure 1 and some of the work was previously done on number systems used like residue number system or logarithmic number system and rest of the work was carried on adder architectures full adder, carry save adder, carry look ahead adder, Kogge-Stone adder, Brent kung and Han-Carlson adder[2] in Figure 2 and work on different combinations of the architectures mentioned above like Wallace tree with Kogge-Stone. The entire work was done in hardware description languages but we can still increase the performance by designing each and every component with less number of transistors as possible.

**Author for correspondence*

31	30-23	22-0
Sign	Exponent	Mantissa

Figure 1. 32-bit floating point representation.

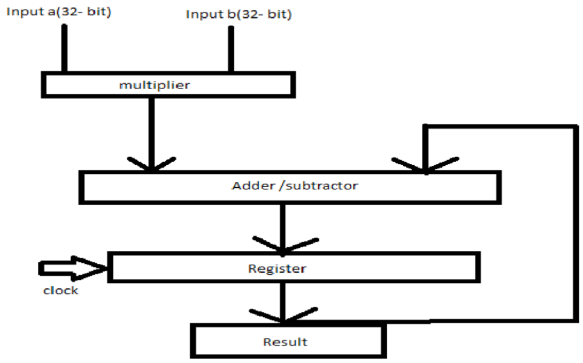

Figure 2. MAC unit basic architecture.

The floating point numbers are represented in IEEE 754 format which have 1 sign bit 8 exponent bits and 23 mantissa bits. In some cases floating point is replaced with fixed point arithmetic but the results are not satisfactory. FINFET is having a major attraction in recent years because there are three major attractions towards it. FINFET is having less leakage current, less off current and short channel effects. Floating-point arithmetic consumes large silicon area on a chip. Thus the circuit consumes large dynamic power and more static power with CMOS. The work done till date is in HDL in which power reduction has reached its limitation. To reduce the area and power much lower we have to excavate new angle of reducing the transistor count which intern reduces dynamic power to reduce static power new device is selected to design called FINFET. The major problem addressed in this thesis is the reduction of area and power floating-point based MAC unit that is capable of delivering high performance.

2. Architecture

The MAC unit is serves a major part in digital signal processing system. MAC is divided into two major parts multiplier in Figure 3 and accumulator before going details about MAC unit IEEE 754 standard should be considered based on this standard design of MAC unit is carried. IEEE 754 standard have 3 types of bit widths 16-bit (half precision), 32-bit (single precision)[3] and 64-bit (double precision). In this paper major interest lies on 32-bit single precision format. As shown in Figure 4 this precision is having a sign bit represents a 0 for positive, 1 for negative. The exponent of base is two. The exponent field contains 127 plus so that representation of both negative and positive exponents becomes easy which is 8 bits wide. The first bit of the mantissa is set as 1 if the given value is normalized 23 bits is allocated for mantissa computation in IEEE 754 floating point standard.

The two basic building blocks of a MAC unit are multiplier and accumulator as shown in Figure 5 which is synchronized with a clock. The operation of the circuit is first two inputs let it be a and b are multiplied passed through adder/Subtraction block and stored in register. Then next two inputs are arrived let it be c and d are multiplied and added to the previous product of a and b and accumulated in the register for next clock cycle likewise this process continues to yield final results.

$$p[i:i] = ai \oplus bi$$

$$g[i:i] = ai \,.\, bi$$

$$g[i:i] = ai \,.\, bi$$ (Process step)

$$p[i:j] = p[i:k+1].p[k:j]$$

$$g[i:j] = g[i:k+1]\ |(g[k:j].p[i:k+1]\ )$$

(Post process)

$$si = p[i:i] \oplus c(i-1)$$

$$ci = g[i:0]\ |(cin\ .p[i:0])$$

There are some other blocks which are designed to normalize the final value. The blocks shifter and leading one detector and comparator in Figure 6. Barrel shifter[4] is used to shift a flock of data left or right based on selection lines and left or right pin shown in Figure 7.

Leading one detector is used to detect leading one of the given input bits which are then passed through an encoder which acts as a selection line for barrel shifter[5]. Block leading one detector is shown in Figure 8. Detector and the outputs from final subtraction is final exponent and outputs from barrel shifter are mantissa of the multiplier. First two sign bits are through XOR gate and taken as a signal for next process in the same clock the data path for sign bit is explained and data path of multiplier is explained. Next exponent data path is explained and finally mantissa data is explained. If application requires area efficiency than fast area-efficient VLSI adders[6] can be used.

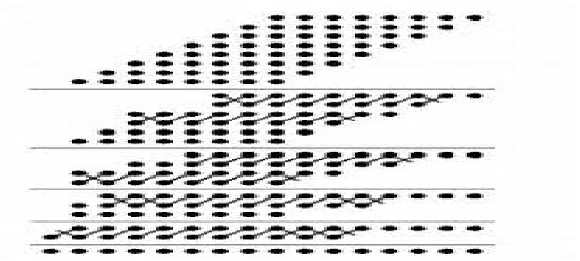

Figure 3. Wallace tree multiplier.

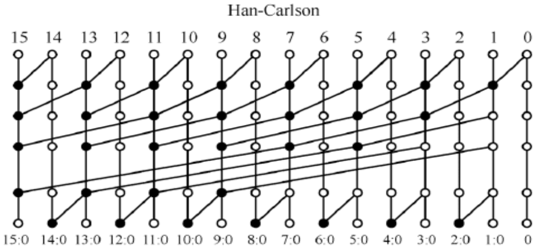

Figure 4. Han-Carlson adder.

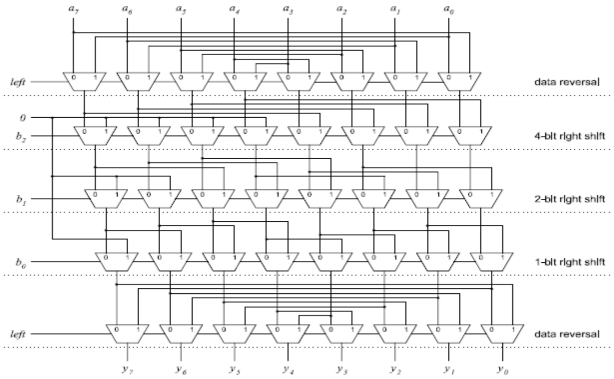

Figure 5. Barrel shifter.

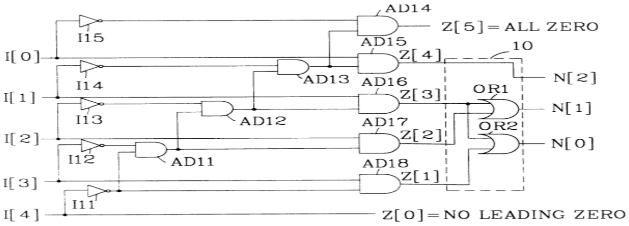

Figure 6. Leading one detector.

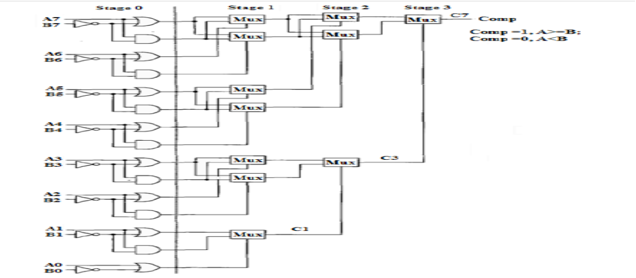

Figure 7. Comparator.

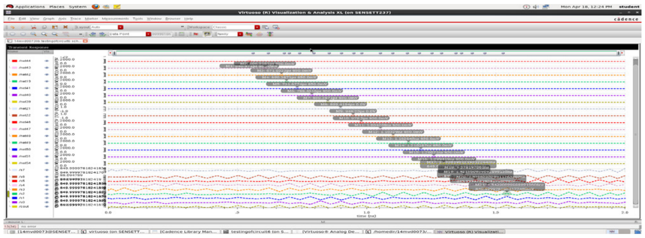

Figure 8. Han-Carlson adder output.

3. Result

The corresponding mantissa bits are shifted and sent to Adder or Subtractor block depends on the sign bits computation obtained from XOR gate the output carry is subtracted from leading one detector bits as shown in Figure 9. And finally the mantissa bits are normalized in barrel shifter as like the operation performed in multiplier block. Implementation of each block with results is followed in next section. Figure 10 shows the Leading One Detector Output obtained from cadence tool. Figure 11 taken from Cadence tool illustrates the MAC Unit Output. Power consumed by the designed architecture is given in Table 1 as 1.26 mW which is suited for low power high performance circuits. Area required for design in term of feature size is (0.36 λ) suitable for portable application. Parallel adder[7,8] can be used for high performance floating point unit design as basic building block. Low-power FINFET[9] or independent gate FINFET[10] can be used for replacing CMOS logic circuits for low power applications.

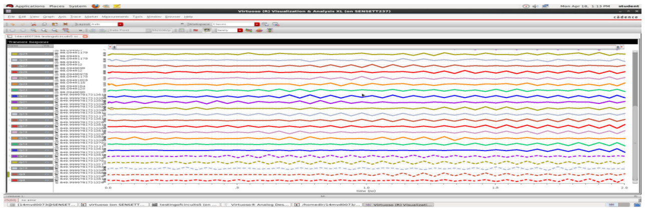

Figure 9. Vedic multiplier output.

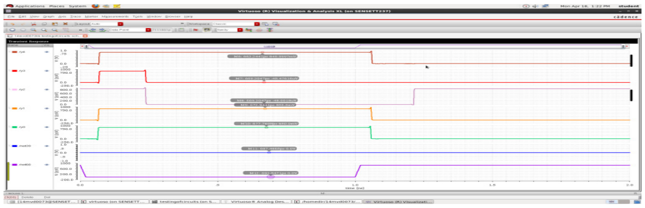

Figure 10. Leading one detector output.

Table 1. Results

Power (mW)	Area(in terms of feature size)
1.26	$(0.36\ \lambda)^2$

D. Simulation Results of MAC Unit

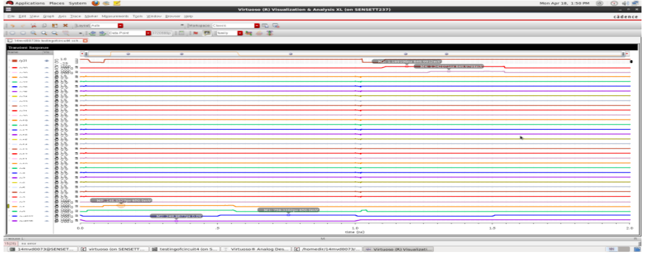
Applications Places System
Mon Apr 18, 1:50 PM
Virtuoso (R) Visualization & Analysis XL (on SENSETT237)
File Edit View Graph Axis Trace Marker Measurements Tools Window Browser Help
cādence
Layout Auto
Workspace Classic
Data Point
Transient Response
0.0
.5
1.0
1.5
2.0
time (ns)
14mvd0073@SENSET...
virtuoso (on SENSETT...
testingofcircuit4 (on S...
Virtuoso® Analog Des...
/homedir/14mvd0073/...
Virtuoso (R) Visualizati...

Figure 11. MAC unit output.

4. Conclusion and Future Scope

The 32-bit MAC unit is implemented in cadence virtuoso and cadence layout editor and the design was carried out in FINFET 15 nm technology. By designing with transistors and with a perfect mix of IEEE 754 standard we obtained power of and the arithmetic is performed with low power. For further improvement in performance in every stage pipelining can be employed. The depth of pipeline can be based on the speed of operation required there by increasing the throughput of the circuit.

Comparator output is passed through AND gate to obtain the final result. The explanation of exponent data path and mantissa data path is carried out in parallel. First the exponent bits are sent to a comparator the output is used to select the highest exponent by using 8 2-bit multiplexers. The two exponents are subtracted and result is used for mantissa stage of barrel shifter before addition.

5. References

1. Krishna VV, Kumar SN. High speed, power and area efficient algorithms for ALU using Vedic mathematics. International Journal of Scientific and Research Publications. 2012 Jul; 2(7):1–6.
2. Binary adder architectures for cell-based VLSI and their synthesis. 2014. Available from: http://www.slideshare.net/prasannaincito/binary-adder-architectures-for-cell-based-vlsi-and-their-synthesis
3. Nesam JJ, Sathasivam S. An efficient single precision floating point multiplier architecture based on classical recoding algorithm. Indian Journal of Science and Technology. 2016 Feb; 9(5):1–7.
4. Begum JT, Naidu SH, Vaishnavi N, Sakana G, Prabhakaran N. Design and Implementation of reconfigurable ALU for signal processing applications. Indian Journal of Science and Technology. 2016 Jan; 9(2):1–6.
5. Barrel shifter or multiply/divide IC structure. 1995. Available from: http://patents.com/us-5465222.html
6. Han T, Carlson DA. Fast area-efficient VLSI adders. 1987 IEEE 8th Symposium on Computer Arithmetic (ARITH); 1987 May. p. 49–56.
7. Brent RP, Kung HT. A regular layout for parallel adders. IEEE Transaction on Computer. 1982 Mar; 31(3):260–4.
8. Kogge PM, Stone HS. A parallel algorithm for the efficient solution of a general class of recurrence equations. IEEE Transactions on Computers. 1973 Aug; C-22(8):786–93.
9. Mishra P, Jha NK. Low-power Fin FET circuit synthesis using surface orientation optimization. Proceedings of the Conference on Design, Automation and Test in European Design and Automation Association; 2010 Mar. p. 311–4.
10. Muttreja A, Agarwal N, Jha NK. CMOS logic design with independent-gate Fin FETs. 25th International Conference on Computer Design, 2007 ICCD; 2007 Oct. p. 560–7.