RESEARCH ARTICLE OPEN ACCESS

Design and Implementation of Fixed Point Arithmetic Unit

S Ramanathan*, G Vinothini*, Ashwini C Chaudhari*, Dr. K Sivasankaran**
**(M.Tech, VLSI Design, School of Electronics Engineering (SENSE), VIT University, Vellore)*
*** (Associate Professor & Head, Micro & Nanoelectronics, School of Electronics Engineering (SENSE), VIT University, Vellore)*

ABSTRACT
This paper aims at Implementation of Fixed Point Arithmetic Unit. The real number is represented in Qn.m format where n is the number of bits to the left of the binary point and m is the number of bits to the right of the binary point. The Fixed Point Arithmetic Unit was designed using Verilog HDL. The Fixed Point Arithmetic Unit incorporates adder, multiplier and subtractor. We carried out the simulations in ModelSim and Cadence IUS, used Cadence RTL Compiler for synthesis and used Cadence SoC Encounter for physical design and targeted 180 nm Technology for ASIC implementation. From the synthesis result it is found that our design consumes 1.524 mW of power and requires area 20823.26 μm^2.

Keywords: ASIC implementation, Cadence IUS, Cadence RTL Compiler, Cadence SoC Encounter, Fixed Point Arithmetic Unit, Qn.m, Verilog HDL.

I. INTRODUCTION

Most of the systems these days are automated, either completely or partially. These are either controlled by one or more controllers or processors which require lot of computations. The controllers and processors need arithmetic unit to perform these computations. The requirement of these computations vary with applications. Some application requires higher performance (speed) with lesser accuracy and some might require higher accuracy with lesser speed. Fixed Point arithmetic finds application that requires higher performance with lesser accuracy. In this paper Section II presents brief introduction to Fixed Point Arithmetic, Section III describes Architecture of Fixed Point Arithmetic Unit, Section IV presents the proposed design, Section V presents the Results and Discussion.

II. FIXED POINT ARITHMETIC UNIT

To define a fixed point number, we need two parameters:

- width of the number representation, and
- the position of binary point within the number

The real number is represented in Qn.m format where n is the number of bits to the left of the binary point and m is the number of bits to the right of the binary point[1]. The Programmer creates the virtual binary point between two locations depending on the requirement[2].

The addition of two fixed point numbers of Qn.m format yields a Qn+1.m number. This is then adjusted to Qn+1.m-1 format by truncating the least significant bit. Similarly the multiplication of two numbers of Qn.m format yields a Q2n.2m number which is then truncated to Q2n.m-n format.

Fixed point numbers are used to represent fractional values in the applications that doesn't require or can't afford to have a floating point arithmetic unit[3]. Floating point arithmetic system is complicated and consumes more power and area with lesser speed of operation as compared to fixed point arithmetic unit. Small processors and low cost processors can't afford to have a floating point arithmetic unit because of these reasons.

Interactive graphics demand speed rather than the accuracy and the floating point arithmetic is accurate but slow. Graphics programs do lot of arithmetic operations and these days we want the system to be as fast as possible. Hence, Fixed Point arithmetic is widely used in Digital Signal Processing (DSP) and game applications as they often give importance to performance than the precision.

The advantages of using Fixed Point Arithmetic is that they are similar to and as efficient as integer arithmetic in computers hence we can perform fixed point arithmetic using the hardware built for integer arithmetic. i.e. no separate hardware is required. The equidistant number distribution results in consistent computation results and they have identical number representation. Their usage can be limited due to limited domain especially in multiplication, less range and precision when compared to floating point arithmetic and bad precision for small numbers in fractional parts.

III. ARCHITECTURE OF FIXED POINT ARITHMETIC UNIT

Due to their similarity with binary operations, generic architectures of adder and multiplier has been targeted to design the Fixed Point Arithmetic Unit.

A. Kogge Stone Adder

Kogge Stone Adder is a parallel prefix form carry look ahead adder. It considered as one of the fastest adders and is widely used in high performance circuits. It generates carry in O(log n) time at the cost of increased area.

An N+1 bit unsigned adder can be used to perform an N bit signed addition.

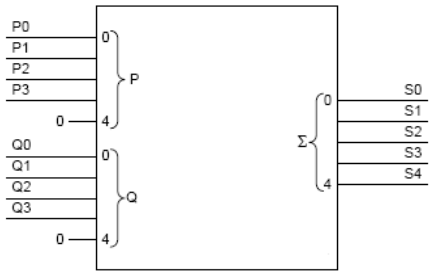

Fig. 1. Example of an N+1 bit adder to perform N bit unsigned addition for N=4.

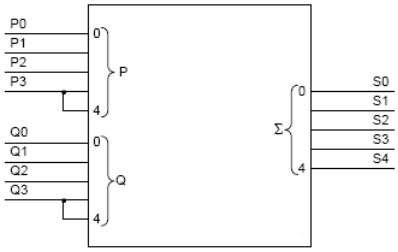

Fig. 2. Example of an N+1 bit adder to perform N bit signed addition for N=4.

B. Booth Multiplier

Booth's algorithm is one of the most widely used algorithm for multiplication of two binary numbers. Due to the similarity in operation with the binary numbers, Booth Multiplier can be used for multiplication in Fixed Point Arithmetic Unit. The multiplication is achieved by using Radix-2 Booth Multiplier.

IV. PROPOSED DESIGN

The proposed design of the Fixed Point Arithmetic Unit is shown in the Fig.3. 17 bit Kogge Stone adder is used to perform 16 bit signed addition. The same module is reused in the Subtracter along with a 2's complement module. Radix-2 Booth Multiplier is used in the multiplier.

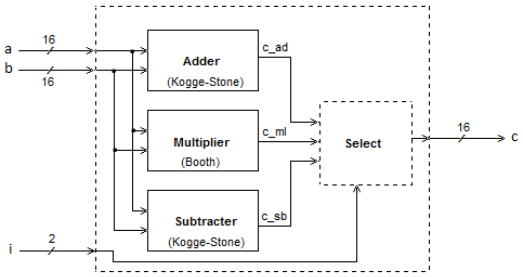

Fig. 3. Block diagram of the proposed design of the Fixed Point Arithmetic Unit.

V. RESULTS AND DISCUSSION

The functional simulation waveform of the Fixed Point Arithmetic Unit is shown in Fig.4.

The following table summarizes the synthesis report.

TABLE I. LOGIC SYNTHESIS SUMMARY

S.No	Parameter	Our Work
1	Area	5131 μm^2
2	Time Slack	165 ps
3	Power	5896.634 nW

TABLE II. PHYSICAL DESIGN SUMMARY

S.No	Parameter	Our Work
1	Area	20823.26 μm^2
2	Time Slack	271 ps
3	Power	1.524 mW

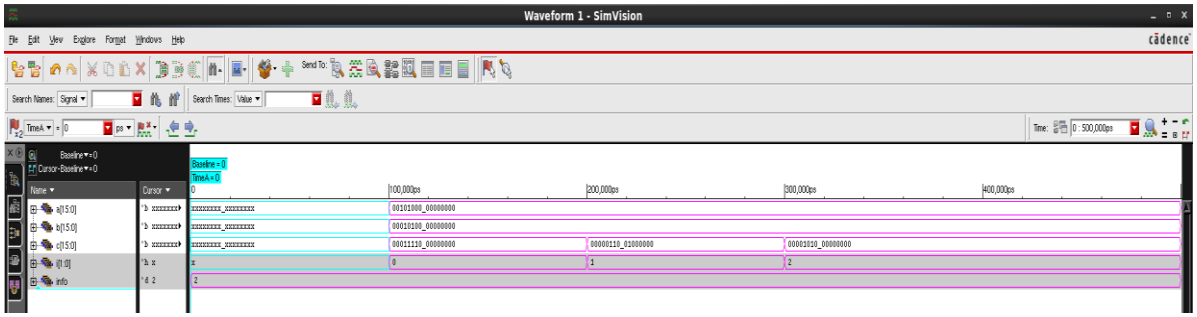

Fig. 4. Functional Simulation wavefirm of Fixed Point Aruthmetic Unit.

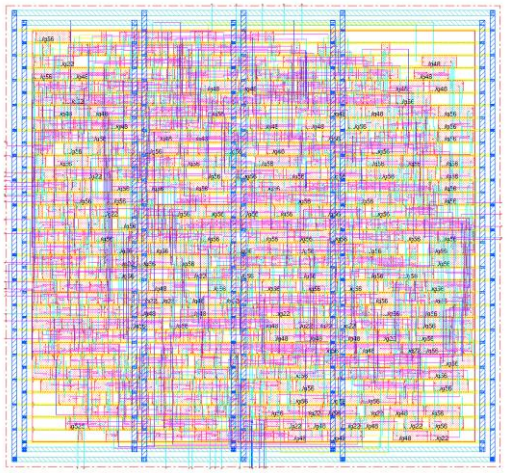

Fig. 5. The phsical design of the final IC

VI. CONCLUSION

In this paper, ASIC implementation of Fixed Point Arithmetic Unit has been presented. The Fixed Point Arithmetic Unit has been designed with modules for Addition, Subtraction and Multiplication. From the synthesis result it is found that our design consumes 1.524 mW of power and requires area 20823.26 μm^2. In future module for division shall be designed.

REFERENCES

Books/Notes:

[1] Mokhtar Aboelaze, *"Verilog Review and Fixed Point Arithmetics"*, York University, 2012

[2] Erick L Oberstar *"Fixed Point Representation And Fractional Math"* Oberstar Consulting

[3] Anton Cervin, *"Fix Point Implementation of Control Algorithms"* Lund University