International Journal of Research and Development in Applied Science and Engineering (IJRDASE)
ISSN: 2454-6844

Design of Floating Point Vedic Multiplier using VHDL

Garima Singh
Electronics Engg. Department
BIIT, Lucknow,India
garima.sitm@rediffmail.com

Suneel Kumar Agrawal
Electronics Engg. Department
BIIT, Lucknow, India
skagrawal10@rediffmail.com

Abstract:
In this paper we proposed the configuration of fast Vedic Multiplier utilizing the strategies of Ancient Indian Vedic Mathematics that have been changed to enhance execution. Vedic Multiplication Technique is utilized to execute IEEE 754 Floating point multiplier. For mantissa increase we are utilizing Urdhvatriyakbhyam sutra for the sub-current and over stream cases are taken care of. The multiplier's inputs are given in IEEE 754, 32 bit group. The Vedic Mathematics is the old arrangement of science which has a one of a kind system of counts in view of 16 Sutras. Our work has demonstrated the effectiveness of Urdhvatriyagbhyam– Vedic technique for duplication which strikes a distinction in the real procedure of augmentation. It empowers parallel era of moderate items, dispenses with undesirable increase ventures with zeros and scaled to higher piece levels utilizing Karatsuba calculation with the similarity to various information sorts. The Urdhva tiryakbhyam Sutra is most effective Sutra (Algorithm), giving least defer for increase of a wide range of numbers, either small or large. We implement this multiplier using VHDL. We implement our work by Xilinx ISE tool i.e. responsible for synthesis also. For simulation we are using Modelsim 10.2a.

Keywords: IEEE 754, Vedic Multiplier, Modelsim, Xilinx, VHDL.

1. Introduction:

Multipliers are key segments of numerous superior frameworks, for example, microchips, FIR channels, computerized signal processors, and so on. Execution of a framework is for the most part dictated by the execution of the multiplier on the grounds that the multiplier is by and large the slowest component in the framework. Since augmentation overwhelms the execution time of most DSP application so there is need of fast multiplier. Moreover, it is by and large the most zone devouring. Thus, enhancing the territory and pace of the multiplier is a noteworthy outline issue. Be that as it may, speed and range are typically clashing imperatives so that enhancing speed comes about for the most part in bigger territories. Thus, an entire range of multipliers with various region speed imperatives has been outlined with completely serial multipliers toward one side of the range and completely parallel Multipliers at the flip side. These multipliers have moderate execution in both pace and zone. Twofold drifting point numbers augmentation is one of the essential capacities utilized as a part of advanced sign handling (DSP) application. The IEEE 754 standard gives the organization to representation of Binary Floating point numbers in PCs. The Binary Floating point numbers are spoken to in Single and Double configurations. The Single accuracy arrangement comprises of 32 bits and the Double exactness position comprises of 64 bits. The organizations are made out of 3 fields; Sign, Exponent and Mantissa. An average focal preparing unit dedicates a lot of handling time in executing math operations, especially augmentation operation. Most superior DSP frameworks depend on equipment duplication to accomplish high information throughput. Augmentation is a critical basic number juggling operation. Execution limitations can likewise be tended to by applying elective innovations. A change at the level of outline usage by the insertion of another innovation can regularly make practical a current minor calculation or design. This venture manages the "Configuration of rapid gliding point multiplier utilizing antiquated strategy". In this task Vedic Multiplication Technique is utilized to actualize IEEE 754 Floating point multiplier. For computation of mantissa unit The Vedic sutra is utilized. A change at the usage level of outline by the insertion of another innovation can regularly make reasonable a current negligible calculation or engineering. Execution limitations can likewise be tended to by applying elective innovations.

2. Related Work:

Sushma S. Mahakalkar et al, they chipped away at the key and the center of all the Digital Signal Processors (DSPs) are its multipliers and the velocity of the DSPs is mostly dictated by the rate of its multiplier. IEEE coasting point arrangement was a standard organization utilized as a part of all preparing components since Binary drifting point numbers augmentation is one of the fundamental capacities utilized as a part of computerized sign handling (DSP) application. In that work VHDL execution of Floating Point Multiplier utilizing old Vedic science is introduced. The thought for planning the multiplier unit is received from old Indian science "Vedas". The Urdhva Tiryakbhyam sutra (strategy) was chosen for execution since it is material to all instances of increase. Augmentation of two no's utilizing Urdhva Tiryakbhyam sutra is performed by vertically and across, transversely implies askew increase and vertically implies straight above duplication and taking their whole. The component is any multi-bit duplication can be diminished down to single piece augmentation and expansion utilizing this strategy. Because of these recipes, the convey proliferation from LSB to MSB is decreases because of one stage era of halfway item [1].

Tariquzzaman et al, took a shot at the FPGA based 64 bit RISC processor with Vedic multiplier component actualized utilizing VHDL .The VHDL code is adaptable dialect and backings FPGA. Framework On-Chip (SoC) and Sparton 3E unit. In today's innovation RISC processor assumes an essential part and RISC framework abbreviate execution time by decreasing the clock cycle per direction and it can address colossal measure of memory up to 16 Exabyte. Augmentation is Fundamental capacity number juggling operation in Digital sign preparing application, for example, FFT, Convolution. Average multiplier requires significant measure of calculation time to actualize result, along these lines productivity of Vedic strategy for augmentation enhances the execution this is the one of a kind system of calculation in view of 16 sutra (Methods). It likewise dispenses with the undesirable duplication step. This work depends on Nikhlam sutra [2].

Aritra Mitra et al, proposed a Vedic Multiplication Technique which used to actualize Floating point multiplier. The Urdhvatriyakbhyam sutra will be utilized for the increase of Mantissa. The undercurrent and over stream cases will be taken care of. The inputs to the multiplier in 32 bit design. The multiplier is outlined in VHDL or VERILOG and reenacted utilizing Modelsim [3].

Bhagyashree Hardiya et al, dealt with augmentation of the drifting point numbers portrayed in IEEE 754 single accuracy substantial. Coasting point multiplier is done utilizing VHDL .Implementation in VHDL(VHSIC Hardware Description Language) is utilized in light of the fact that it permit direct execution on the equipment while in other dialect they need to change over them into HDL then just can be actualized on the equipment. In drifting point duplication, including of the two numbers is finished with the assistance of different sorts of adders yet for increase some additional moving is required. This coasting point increase handles different conditions like flood, sub-current, standardization, adjusting. In this work they utilize IEEE adjusting strategy for perform the adjusting of the came about number. This work surveys the execution of an IEEE 754 single accuracy coasting point multiplier created by numerous scientists [4].

Remadevi R, proposed a calculation for duplicating drifting point numbers which was a basic necessity for DSP applications including substantial element range. This work concentrates just on single accuracy standardized twofold trade design focused for Xilinx Spartan-3 FPGA in light of VHDL. The multiplier was confirmed against Xilinx gliding point multiplier center. It handles the flood and sub-current cases. Adjusting is not executed to give more exactness when utilizing the multiplier as a part of a Multiply and Accumulate (MAC) unit [6].

3. Methodology:

The performance of Mantissa calculation Unit dominates overall performance of the Floating Point Multiplier. This unit requires unsigned multiplier for multiplication of 24x24 BITs. The Vedic Multiplication technique is chosen for the implementation of this unit. This technique gives promising result in terms of speed and power [6].The Vedic multiplication system is based on 16 Vedic sutras or aphorisms, which describes natural ways of solving a whole range of mathematical problems. Out of these 16 Vedic Sutras the Urdhva-triyakbhyam sutra is suitable for this purpose. In this method the partial products are generated simultaneously which itself reduces delay and makes this method fast. The method for multiplication of two, 3 BITs number is shown Figure 3.4. Consider the numbers A and B where A = a2a1a0 and B = b2b1b0. The LSB of A is multiplied with the LSB of B:

s0=a0b0;

Then a0 is multiplied with b1, and b0 is multiplied with a1 and the result are added together as:

c1s1=a1b0+a0b1;

Here c1 is carry and s1 is sum. Next step is to add c1 with the multiplication results of a0 with b2, a1 with b1 and a2 with b0.

c2s2=c1+a2b0+a1b1 + a0b2;

Next step is to add c3 with the multiplication results of a1 with b2 and a2 with b1.

c3s3=c2+a1b2+a2b1;

Similarly the last step

c4s4=c3+a2b2;

Now the final result of multiplication of A and B is c4s4s3s2s1s0.

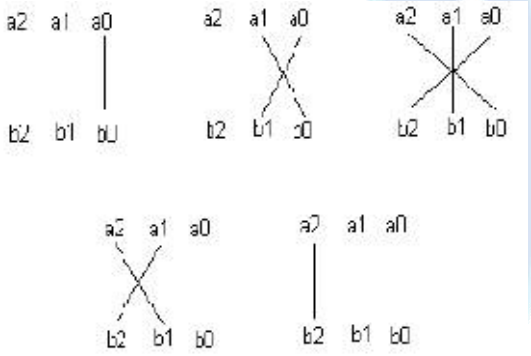

Fig 1. The Vedic Multiplication method

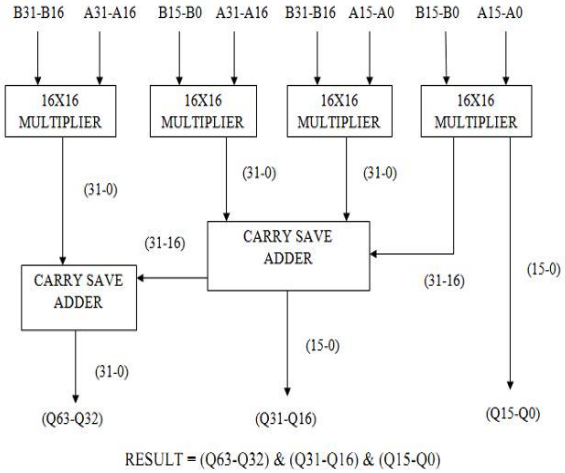

Fig. 2. 32X32 Bits proposed Vedic Multiplier

For Multiplier, first the basic blocks, that are the 2x2 bit multipliers have been made and then, using these blocks, 4x4 block has been made by adding the partial products using carry save adders and then using this 4x4 block, 8x8 bit block, 16x16 bit block and then finally 32 x 32 bit Multiplier as shown in figure 3.5 has been made [5].
The design starts first with Multiplier design, that is 2x2 bit multiplier as shown in figure 2. Here, "Urdhva Tiryakbhyam Sutra" or "Vertically and Crosswise Algorithm"[3] for multiplication has been effectively used to develop digital multiplier architecture. This algorithm is quite different from the traditional method of multiplication, which is to add and shift the partial products.

4. Result and Discussion:

We have taken two inputs 'A' and 'B' as a multiplier and multiplicand these are floating point signed value we are perform multiplier using Vedic Algorithm between these inputs and will be stored in other output port which we have taken as 'Z' all operations are performing on positive edge of clock.
For case I we take value of 'A' is 134.0625 and value of 'B' is -2.25. Here 'A' is unsigned floating pint number and 'B' is Signed Floating Point Number. Now we have to convert value of 'A' to binary format after normalize we get 1.00001100001x2^7 then we have to convert it into IEE-32 floating point format then we get 0 10000110 00001100001000000000000 then convert it into hexadecimal format we get 0x43061000. Now we have to convert value of 'B' to binary format after normalize we get -1.001x2^1 then we have to convert it into IEE-32 floating point format then we get 1 10000000 00100000000000000000000 then convert it into hexadecimal format we get 0xC0100000. After multiplication using Vedic Multiplier we get 0xC396D200 the value of this hexadecimal no. is -301.640625 fig 4.1 shows the simulation result of this data.

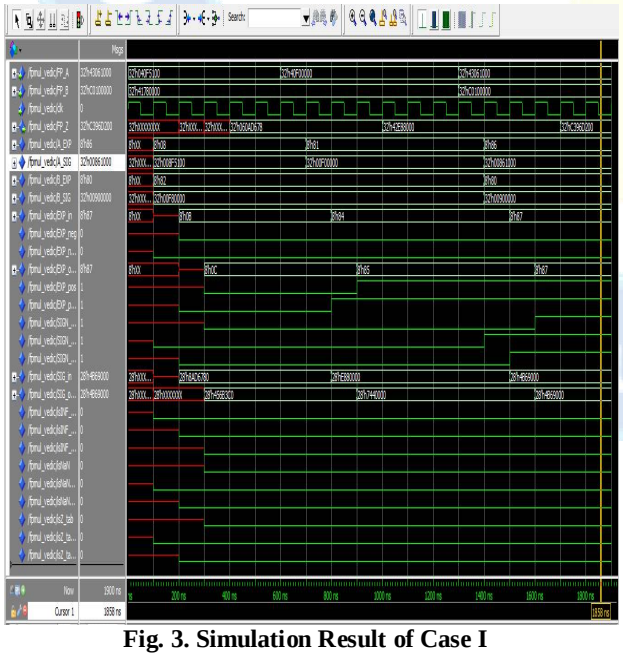

Fig. 3. Simulation Result of Case I

For case II we take value of 'A' is -14.5 and value of 'B' is -0.375. Here 'A' is signed floating pint number and 'B' is also a signed Floating Point Number. Now we have to convert value of 'A' to binary format after normalize we get -1.1101x2^3 then we have to convert it into IEE-32 floating point format then we get 1 10000010 11010000000000000000000 then convert it into hexadecimal format we get 0xC1680000. Now we have to convert value of 'B' to binary format after normalize we get -1.1x2^-2 then we have to convert it into IEE-32 floating point format then we get 1 01111101 10000000000000000000000 then convert it into hexadecimal format we get 0xBEC00000. After multiplication using Vedic Multiplier we get 0x40AE0000 the value of this hexadecimal no. is 5.4375 fig 4.2 shows the simulation result of this data.

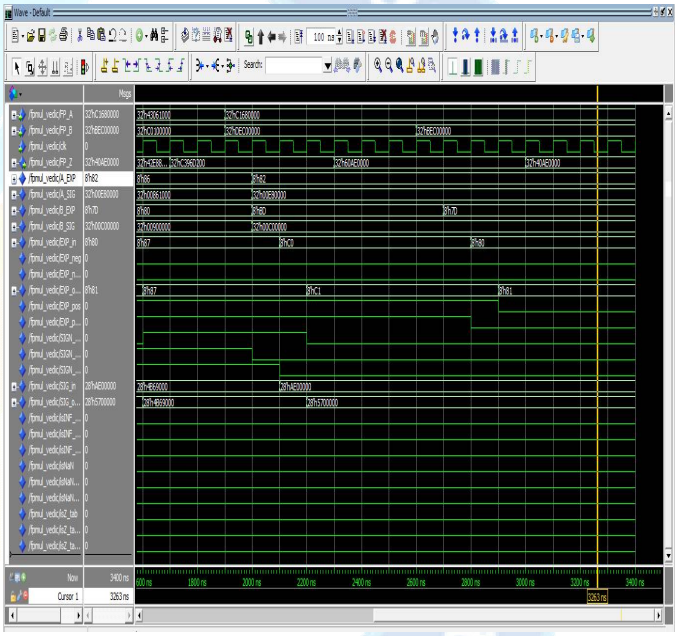

Fig. 4. Simulation Result of Case II

We have implemented our work on FPGA Spartan3E. So we can conclude that our design is ready to implement for hardware shows in fig 5.

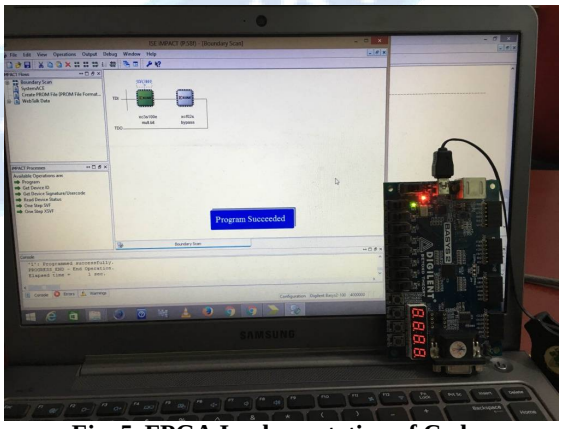

Fig. 5. FPGA Implementation of Code.

5. Conclusion:

The Floating Point numbers are the basic necessity in the current scenario of digital design based systems. Hence we implemented, the design of Floating point number in IEEE32 bit format, on Spartan 3E- XC3S250-5-CP132. The design is based on Vedic method of multiplication. The worst case propagation delay in the Optimized Vedic floating point multiplier case is 4.788 ns. It is therefore seen that the Vedic floating point number multipliers are much faster than the conventional multipliers. This gives us method for hierarchical floating point multiplier design. So the design complexity gets reduced for inputs of large no of bits and modularity gets increased. Urdhva tiryakbhyam sutra algorithm is been used which can reduce the delay and hardware requirements for multiplication of Floating point numbers. FPGA based simulation and Synthesis of this floating point multiplier shows that hardware realization of the Vedic mathematics algorithms is easily possible. The high speed multiplier algorithm exhibits improved efficiency in terms of speed.

References:

[1] Sushma S. Mahakalkar et al, "Review on floating point multiplier using ancient techniques" IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE) e-ISSN: 2278-1676, p-ISSN: 2320-3331 PP 01-04.

[2] Tariquzzaman et al, " FPGA implementation of 64 bit RISC processor with Vedic multiplier using VHDL" IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE) e-ISSN: 2278-1676, p-ISSN: 2320-3331 PP 12-16.

[3] Aritra Mitra et al , "Design of Floating Point Multiplier based on Vedic Multiplication Technique" School of Electronics Engineering Electronics And Telecommuncation Kalinga Institute of Industrial Technology BHUBANESWAR-751024.

[4] Bhagyashree Hardiya et al "Implementation Of Floating Point Multiplier Using VHDL" Technology and Engineering (BEST: IJMITE) Vol. 1, Issue 3, Dec 2013, 199-204 © BEST Journals.

[5] S Venkateswara Reddy et al, "Design And Implementation Of 32 Bit Multipier Using Vedic Mathamatics" International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (An ISO 3297: 2007 Certified Organization)Vol. 2, Issue 8, August 2013.

[6] Remadevi R, " Design and Simulation of Floating Point Multiplier Based on VHDL" Remadevi R / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 3, Issue 2, March - April 2013, pp.283-286.

[7] Dinesh Kumar et al, "Simulation and Synthesis Of 32-Bit Multiplier Using Configurable Devices" International Journal of Advances in Engineering & Technology, Jan. 2013. ©IJAET ISSN: 2231-1963.

[8] Poornima M et al, "Implementation of Multiplier using Vedic Algorithm" International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-2, Issue-6, May 2013.