Journal of Engineering Sciences
Vol 11, Issue 1, JAN/2020
ISSN NO:0377-9254

UNBIASED ROUNDING FOR HUB FLOATING-POINT ADDITION

[1]Indrakanti Tejendra Chary, [2]M Pushpa Latha, [3]Dr. D SubbaRao
[1]PG Scholar, MTech, Dept of ECE, Siddhartha Institute of Engineering & Technology, Ibrahimpatnam, T.S.
charytej@gmail.com
[2]Asso Professor, Dept of ECE, Siddhartha Institute of Engineering & Technology, Ibrahimpatnam, T.S.
[3]Professor & HOD, Dept of ECE, Siddhartha Institute of Engineering & Technology, Ibrahimpatnam, T.S.

ABSTRACT - The increasing complexity of new digital signal processing applications is forcing the use of floating-point numbers in their hardware implementations. In this brief, we investigate the advantages of using HUB formats to implement these floating-point applications on FPGAs. These new floating-point formats allow for the effective elimination of the rounding logic on floating-point arithmetic units. Firstly, we experimentally show that HUB and standard formats provide equivalent SNR on DSP application implementations. We then present a detailed study of the improvement achieved when implementing floating-point adders and multipliers on FPGAs by using HUB numbers. In most of the cases studied, the HUB approach reduces resource use and increases the speed of these FP units, while always providing statistically equivalent accuracy as that of conventional formats. However, for some specific sizes, HUB multipliers require far more resources than the corresponding conventional approach.
Keywords: FIR, Xilinx, FPGA, Synthesize, Implementation, Simulation.

I. INTRODUCTION

These days, numerous Digital Signal Processing (DSP) applications, for example, designs, remote interchanges, mechanical control, and medicinal imaging require the utilization of direct variable-based math or other complex calculations. The utilization of Floating-Point (FP) number-crunching is rapidly turning into a necessity in these applications because of its all-inclusive powerful range and exactness. Consequently, FP number juggling is being presented on FPGA executions, as a delicate center, or even as an equipment obstruct in the most current Altera gadgets. Despite the fact that these implanted equipment squares are more productive and savvier than their equal delicate center structures, the last are still exceptionally helpful. Right off the bat, minimal effort gadgets don't offer this FP inserted squares and it isn't certain that other FPGA brands will incorporate something comparative in their gadgets soon. Also, up to now, just single exactness has been legitimately upheld in DSP squares.

In this way, enhancements to the delicate center executions are of incredible worth. A portion of these arrangements are being intended to keep the IEEE754 standard. In any case, in numerous applications, consistence with this standard is yielded to get increasingly effective executions in regards to territory and execution. In connection to FPGAs, substantially more proficient structures are acquired by utilizing increasingly adaptable usage of FP numbers and in these FP centers, a lot of asset use and postponement is because of the adjusting rationale. Be that as it may, two new groups of configurations, HUB (Half-Unit one-sided) and Round-to-Nearest portrayals, permit RNE to be performed just by truncation, which could make adjusting rationale unimportant. Here, we center around HUB positions. Center point Fixed-point organizations were utilized to improve DSP executions.

A HUB-FP number is a FP number with the end goal that its mantissa (or significand) has an Implicit Least Significant Bit (ILSB) which equivalents one. Contrasted and a standard arrangement, it has a similar number of unequivocal bits and accuracy, however a similar piece vector speaks to a worth one-sided half Unit-in-the-Last-Place.

Objective:

- An exploratory mistake examination of the execution of FIR channels, which shows that the HUB approach gives comparable measurable parameters to those of standard FP usage, including the SNR.
- The aftereffects of FPGA usage of an essential FP snake and multiplier for a wide scope of type and mantissa bit-widths under HUB and traditional methodologies and their examination.
- In a large portion of the cases considered, the HUB organization decreases asset use and builds the speed of these FP units.
- Furthermore, because of its effortlessness, any current delicate or no-nonsense could be effectively upgraded by utilizing the proposed methodology. Along these lines, in light of essential models, our point is to urge analysts to improve their upgraded FP centers or DSP applications by utilizing HUB-FP designs.

• We explored the utilization of HUB-FP configurations to upgrade the execution of DSP applications on FPGA.

II. LITERATURE REVIEW

This paper depicts the engineering and execution, from both the stance of target applications just as circuit plan, of a FPGA DSP Block that can effectively bolster both fixed and drifting point (FP) math. Most contemporary FPGAs install DSP hinders that give straightforward duplicate include based fixed-point math centers. Current skimming point number-crunching FPGA arrangements utilize these solidified DSP assets, together with implanted memory squares and delicate rationale assets, nonetheless, bigger frameworks can't be effectively actualized because of the directing and delicate rationale constraints on the gadgets, bringing about noteworthy territory, execution, and power utilization punishments contrasted with ASIC usage. In this paper we investigate prior proposed implanted skimming point executions, and show why they are not appropriate for a creation FPGA. We differentiate these against our answer – a bound together DSP Block – where (a) the FP multiplier is overlaid on the fixed point builds, (b) the FP Adder/Subtracted is incorporated as a different unit; and (c) the multiplier and snake can be joined in a manner that is both mathematically valuable, yet in addition proficient regarding FPGA directing thickness and blockage. Furthermore, a novel method for consistently joining any number of DSP Blocks in a low idleness structure will be presented. We will show this new approach permits a minimal effort, low power, and high thickness drifting point stage on current 20nm FPGAs.

In this paper, another group of arrangements to manage genuine number for applications requiring round to closest is proposed. They depend on moving the arrangement of precisely spoke to numbers which are utilized in traditional radix-β number frameworks. This system permits performing radix supplement and round to closest without convey spread with immaterial time and equipment cost. Moreover, the proposed configurations have a similar stockpiling cost and exactness as standard ones. Since change to traditional organizations essentially require attaching one extra-digit to the operands, standard circuits might be utilized to perform number juggling activities with operands under the new configuration. We likewise expand the highlights of the RN-portrayal framework and do an exhaustive correlation between both portrayal frameworks. We presume that the proposed portrayal framework is commonly progressively satisfactory to execute frameworks for calculation with genuine number under round-to-closest.

In this paper we depict an open source skimming point viper and multiplier actualized utilizing a 36-piece custom number arrangement dependent on radix-16 and improved for the 7-arrangement FPGAs from Xilinx. Despite the fact that this number configuration isn't indistinguishable from the single-accuracy IEEE-754 arrangement, the drifting point administrators are planned so that the numerical outcomes for a given activity will be indistinguishable from the outcome from an IEEE-754 consistent administrator with help for round-to-closest even, NaNs and Infs, and subnormal numbers. The disadvantage of this number configuration is that the adjusting step is more required than in a normal, radix-2 based administrator. Then again, the utilization of a high radix implies that the territory cost related with standardization and de standardization can be decreased, prompting a net region advantage for the custom number arrangement, under the suspicion that help for subnormal numbers is required. The zone of the skimming point viper in a Kintex-7 FPGA is 261 cut LUTs and the zone of the coasting point multiplier is 235 cut LUTs and 2 DSP48E squares. The viper can work at 319 MHz and the multiplier can work at a recurrence of 305 MHz.

During any composite calculation, there is a steady requirement for adjusting transitional outcomes before they can take an interest in further handling. As of late, a class of number portrayals signified RN-Codings were presented, enabling a fair adjusting to-closest to occur by a straightforward truncation, with the property that issues with twofold adjusting's are maintained a strategic distance from. In this paper, we initially examine a specific encoding of the parallel portrayal. This encoding is summed up to any radix and digit set; nonetheless, radix supplement portrayals for even estimations of the radix end up being especially doable. The encoding is basically a customary radix supplement portrayal with an affixed round-piece, yet at the same time enabling adjusting to-closest by truncation, and in this way dodging issues with twofold roundings. Transformations from radix supplement to these round-to-closest portrayals can be performed in steady time, though change the other way, as a rule, takes in any event logarithmic time. Not exclusively is adjusting to closest a steady time activity, yet so is likewise sign reversal, the two of which are, best case scenario log-time tasks on standard two's supplement portrayals. Expansion and increase on such fixed-point portrayals are first broke down and characterized so that adjusting data can be conveyed along in a significant manner, at insignificant expense. The

investigation is brought through for a smaller (sanctioned) encoding utilizing two's supplement portrayal, provided with a round-piece. In view of the fixed-point encoding, it is demonstrated conceivable to characterize drifting point portrayals, and a sketch of the execution of a FPU is introduced.

This paper breaks down the advantages of utilizing HUB configurations to actualize coasting direct number-crunching under round-toward closest mode from a quantitative perspective. Utilizing HUB arrangements to speak to numbers permits the expulsion of the adjusting rationale of math units, including clingy bit calculation. This is appeared for gliding point adders, multipliers, and converters. Test investigation exhibits that HUB designs and the relating math units keep up a similar precision as ordinary ones. Then again, the execution of these units, in view of essential structures, shows that HUB designs at the same time improve territory, speed, and power utilization. In particular, in light of information got from the amalgamation, a HUB single-exactness viper is about 14% quicker however devours 38% less territory and 26% less power than the regular snake. Additionally, a HUB single-exactness multiplier is 17% quicker, utilizes 22% less region, and expends somewhat less power than ordinary multiplier. At a similar speed, the viper and multiplier accomplish zone and power decreases of up to half and 40%, individually.

III. METHODOLOGY

Existing Module:

- The structure of 32-piece Single Precision Floating Point Multiplier and Adder is reproduced and introduced in this paper.
- The gliding point number can bolster wide scope of qualities. It is spoken to utilizing three fields: sign, example and mantissa. In this paper gliding point expansion, and increase calculations for IEEE-754 (single exactness) is displayed.
- The IEEE-754 converter is utilized to change over decimal gliding point number into Binary drifting point organization and it is additionally used to check the outcomes.
- A square graph of the primary information way bit of the DSP square, two littler 18x18 multipliers, went before by discretionary info pre-adders, can be utilized to execute free multipliers, totals of multipliers, or bigger 27x27 multipliers, utilizing the multiplexers and blowers.
- The last CPA in the DSP square can be part in two on account of individual multipliers. To help the entirety of the fixed-point modes and furthermore FP augmentation, the last CPA was decayed into a lot of prefix structures lined up with both the fixed and FP limits.
- The three FP multiplier steps post pressure: last sub-item decrease, standardization and adjusting were consolidated into a solitary level CPA utilizing a hailed prefix approach.
- Our hailed prefix approach, so as to help both fixed and FP estimations, utilizes a hailed prefix structure, overlaid on the convey select structure, overlaid on a prefix snake structure.

Proposed Module:

- Here, we center around HUB groups. Center Fixed-point organizations were utilized to improve DSP usage, since they permit better word-length advancement.
- The ASIC execution of HUB-FP units has been read for binary16 (half), binary32 (single), and binary64 (twofold), and significant enhancements have been accomplished.
- In this short correspondence, we stretch out this examination to FPGAs over a wide scope of sizes. Contrasted with past articles, we give: An exploratory blunder examination of the execution of FIR channels, which shows that the HUB approach gives comparable measurable parameters to those of standard FP usage, including the SNR.
- The consequences of FPGA usage of an essential FP snake and multiplier for a wide scope of example and mantissa bit-widths under HUB and ordinary methodologies and their examination.
- In the majority of the cases examined, the HUB arrangement lessens asset use and expands the speed of these FP units. Besides, because of its effortlessness, any current delicate or hard
- Core could be effectively upgraded by utilizing the proposed methodology. Subsequently, in light of essential designs, our point is to urge scientists to improve their upgraded FP centers or DSP applications by utilizing HUB-FP positions.

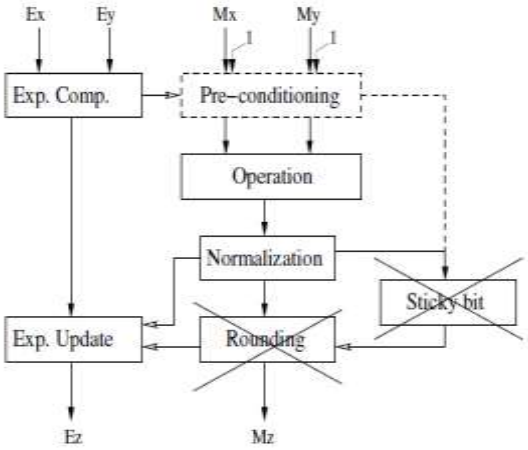
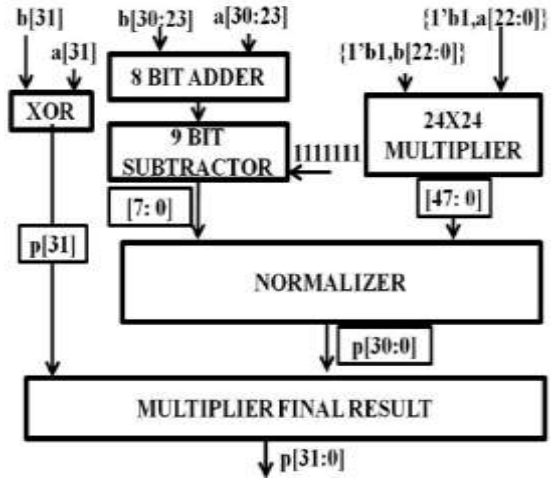


Fig 1 Basic HUB-FP arithmetic architecture

A fundamental general engineering to work HUB-FP numbers is appeared in Fig. 1, where a traditional FP math unit has been helpfully altered. Right off the bat, the ILSBs are attached to the mantissas of the information operands before utilizing them. As result they are changed over to regular configuration. Next, an ordinary information way is utilized, however from the outset the mantissa information way must be one-piece more extensive. Since the conclusive outcome is in HUB design, a straightforward truncation is performed for adjusting. Along these lines, after the math activity itself, customary standardization rationale is used, yet no watchman bit is given at the yield. The adjusting rationale of the regular engineering is just dispensed with (crossed out in Fig. 1). Considering that the ILSB is a consistent, this general design could be additionally streamlined relying upon the particular engineering.

FLOATING POINT MULTIPLICATION



A multiplication of two floating-point numbers is done in four steps:

- Non-signed multiplication of mantissas: it must take account of the integer part, implicit in normalization. The number of bits of the result is twice the Size of the operands (48 bits).
- normalization of the result: the exponent can be modified accordingly
- addition of the exponents, taking into account the bias
- Calculation of the sign.

IEEE 754 standard Floating-point multiplication Algorithm

A brief overview of floating-point multiplication algorithm has been explained below, X1 and X2.

Result X3 = X1 * X2 = $(-1)^{s1}$ (M1 x 2^{E1}) * $(-1)^{s2}$ (M2 x 2^{E2})

S1, S2 => Sign bits of number X1 & X2.

E1, E2: =>Exponent bits of number X1 & X2.

M1, M2 =>Mantissa bits of Number X1 & X2.

1) Check in the event that one/the two operands = 0 or vastness. Set the outcome to 0 or inf. for example types = every one of the "0" or each of the "1".

2) S1, the marked piece of the multiplicand is XOR'd with the multiplier marked piece of S2. The outcome is placed into the resultant sign piece.

3) The mantissa of the Multiplier (M1) and multiplicand (M2) are duplicated and the outcome is set in the resultant field of the mantissa (truncate/round the outcome for 24 bits) =M1 * M2

4) The examples of the Multiplier (E1) and the multiplicand (E2) bits are included and the base worth is subtracted from the additional outcome. The subtracted outcome is placed in the exponential field of the outcome square. =E1+E2-inclination

5) standardize the total, either moving right and increasing the type or moving left and decrementing the example.

6) Check for undercurrent/flood. In the event that Overflow set the yield to boundlessness and for sub-current set to zero.

7) If (E1 + E2 - inclination) >= to Emax then set the item to unendingness.

8) If E1 + E2 - inclination) is lesser than/equivalent to E min at that point set item to zero.

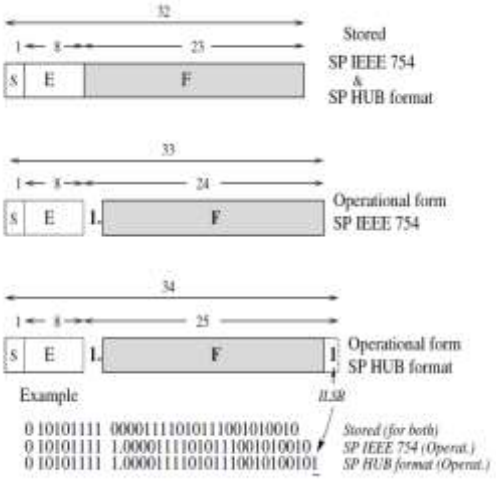
The skimming point duplication includes following advances: Assume that the operands where "man" speaks to mantissa, "exp" speaks to example.

1. Process the indication of the outcome (Aexp ^ Bexp).
2. Increase the mantissa.
3. Standardize the item if necessary.

4. Register the type of the outcome:
Example of the outcome = one-sided type (Aexp) + one-sided type (Bexp) – predisposition
5. Round the outcome to the no. of mantissa bits.

The twofold accuracy gliding point multiplier design is as appeared in Fig. 2. The design contains a multiplier tree which duplicates two 53-piece numbers (1 covered up bit+52 bit mantissa). Yield from the multiplier tree is in convey spare organization and it is passed to a joined include/round stage, where the convey spare item is included and adjusted. Both the sign and type figuring rationale runs in parallel with the mantissa augmentation. To get the last type esteem, the example should be balanced dependent on the adjusting. In some cases there is an opportunity of misunderstanding an incentive for the type as the genuine example worth surpasses the bit range. So to stay away from this in our structure exemption taking care of is incorporated. At whatever point the bit range is high, it creates a special case while performing type rationale or type modifies.

HUB FORMAT



- A HUB-FP number is a FP number with the end goal that its mantissa (or significand) has an Implicit Least Significant Bit (ILSB) which equivalents one. Contrasted and a standard organization, it has a similar number of express bits and exactness, however a similar piece vector speaks to a worth one-sided half Unit-in-the-Last-Place.
- HUB numbers could be effectively worked by customary math circuits, while for all intents and purposes taking out the adjusting rationale. Besides, the effect of the incorporation of the ILSB is restricted since it is consistent.
- A customary FP math unit has been advantageously altered. Initially, the ILSBs are attached to the mantissas of the information operands before utilizing them. As result they are changed over to ordinary organization. Next, a customary information way is utilized, however from the start the mantissa information way must be one-piece more extensive. Since the conclusive outcome is in HUB design, a basic truncation is performed for adjusting.

HUB-FP NUMBERS AND ASSOCIATED CIRCUITS

The fundamental focal points of utilizing the HUB arrangement are that two's supplement is processed by just piece savvy reversal, the truncation of an incentive to get a HUB number delivers a proportionate RNE, and a clingy bit calculation isn't required for generally tasks. Besides, the transformation to a regular arrangement just requires expressly affixing the ILSB to the first number. Along these lines, HUB numbers could be effectively worked by regular math circuits, while for all intents and purposes disposing of the adjusting rationale. Besides, the effect of the incorporation of the ILSB is constrained since it is steady. An essential general design to work HUB-FP numbers is appeared in Fig. 2, where an ordinary FP number juggling unit has been helpfully adjusted. Right off the bat, the ILSBs are affixed to the mantissas of the information operands before utilizing them.

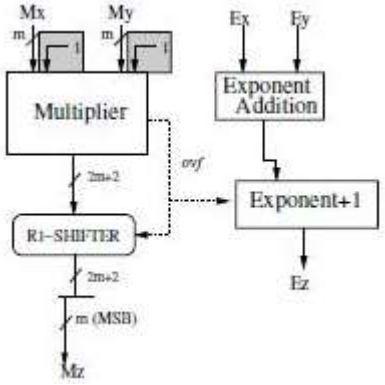


FIG 2 Architecture of the non-normalized HUB multiplier

As outcome they are changed over to traditional arrangement. Next, an ordinary information way is utilized, yet from the start the mantissa information way must be one-piece more extensive. Since the conclusive outcome is in HUB position, a straightforward truncation is performed for adjusting. In this way, after the number-crunching activity itself, regular standardization rationale is used, however no gatekeeper bit is given at the yield. The adjusting rationale of the customary design is just disposed of (crossed out in Fig. 1). Considering that the ILSB is a consistent, this general design could be additionally streamlined relying upon the particular engineering.

ADJUSTING ERROR OF HUB-FP COMPUTATION

Duplication of non-standardized numbers may debase unequivocally the exactness of the outcomes, since the quantity of driving zeros of the outcome is the expansion of the quantity of driving zeros of each information operand. In this manner, we may consider normalizing (at any rate incompletely) the yield of the multiplier to keep the precision sensible cutoff points. As we said over, the quantity of driving zeros at the multiplier yield could be determined by checking them at the info operands.

Fig. 2 depicts a general plan of a multiplier performing standardization utilizing this plan (dim boxes are just utilized in the HUB adaptation). Two Leading Zero Detectors (LODs) are utilized to check the quantity of driving zeros of each info significand in parallel with the increase. What's more, a barrel left-shifter is included at the yield of multiplier. We investigated various alternatives for this plan. Solidly, other than the total standardization, we read two distinct methodologies for halfway standardization to lessen the effect of the standardization circuit. Thus, despite the fact that the design continues as before, both the LOD circuit and the left-shifter circuit are diverse in each approach.

These two methodologies consistently consider flood of the duplication as in M. In one methodology, we execute a more generally standardization utilizing higher radix than two (digits of x-bits), to check zeros and to play out the moving. This is practiced by presenting x-info OR-entryways before checking the zeros and taking out the multiplexors with minimal measure of moving at the barrel-shifter.

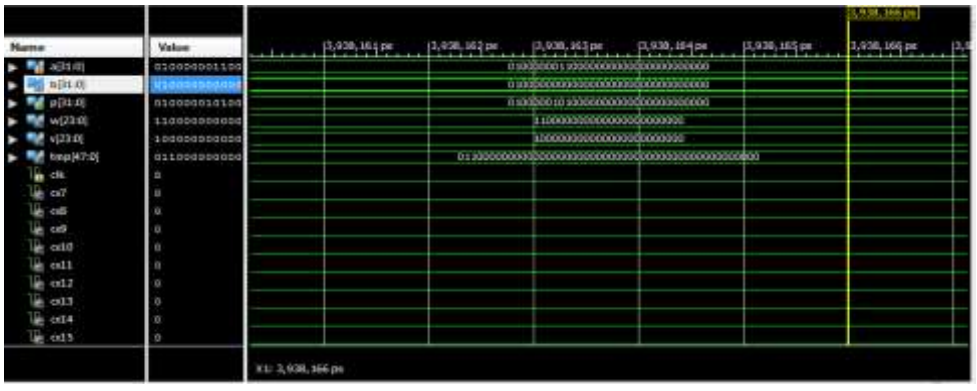
An alternate methodology is to utilize radix-2 yet constraining the most extreme number of left-moving positions, comparatively to A2 viper. As we will find in the following area, in any event, permitting not many moving positions, this incomplete standardization greatly affects exactness. For this situation, the quantity of zeros included in each info significand is restricted to x bits and, as a result, the quantity of left-moving positions is limited to 2x bits. We call these multipliers as MLx (MLxH, for HUB rendition), where x means the greatest number of moving positions permitted. We need to take note of that increase of little operands may deliver undercurrent when the type of the outcome is determined. This circumstance is recognized and the outcome is flushed to zero, which is an alluring conduct for the objective applications.

IV. RESULTS AND DISCUSSION

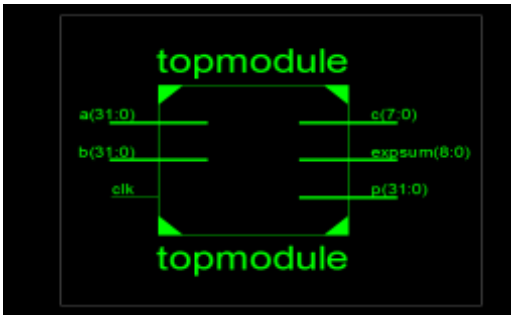
FLOATING POINT MULTIPLICATION

Device utilization summary:

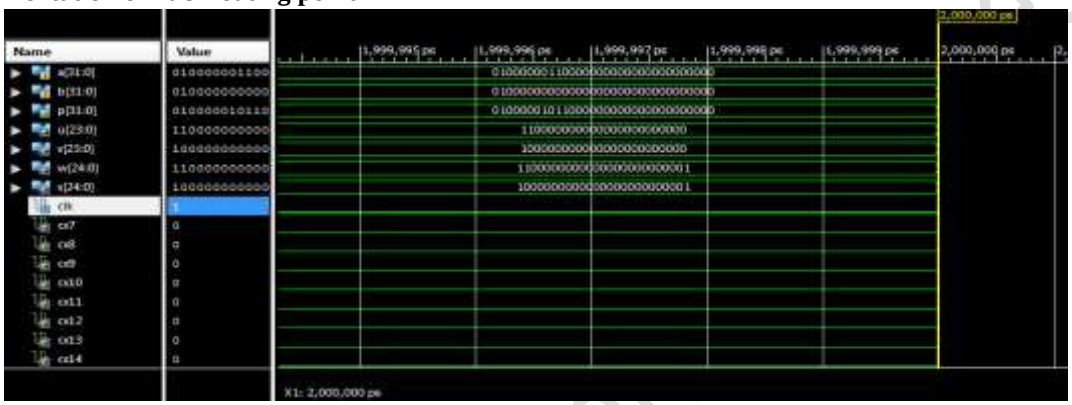
Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	863	5720	15%
Number of fully used LUT-FF pairs	0	863	0%
Number of bonded IOBs	113	102	110%



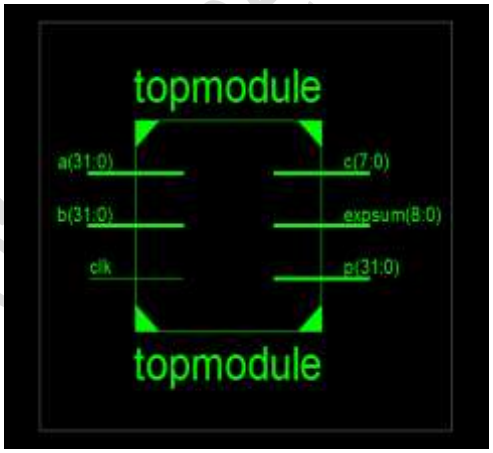
RTL VIEW:



Implementation of hub floating point



RTL VIEW

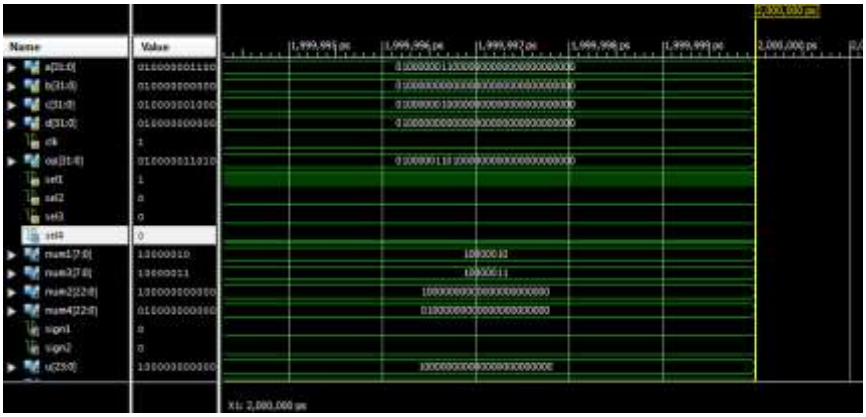


Device utilization summary:

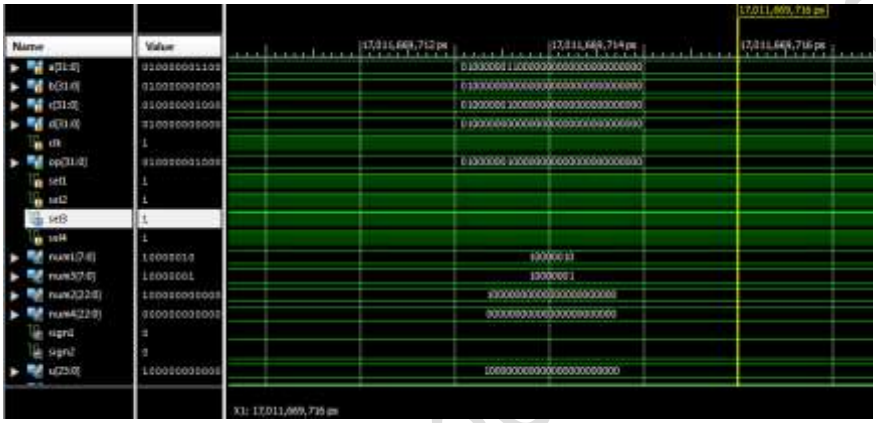
Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	18	2400	0%
Number of fully used LUT-FF pairs	0	18	0%
Number of bonded IOBs	114	102	111%
Number of BUFG/BUFGCTRLs	1	16	6%
Number of DSP48A1s	4	8	50%

IMPLEMENTATION OF FUSED FLOATING DOT PRODUCT:

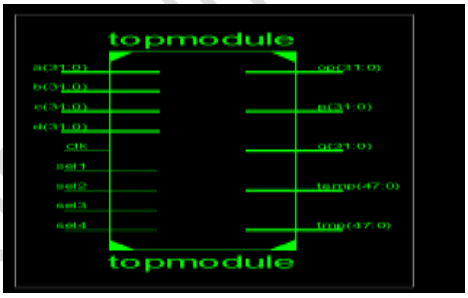
case1: (a.b) + (c.d)



- **Case2: (a.b) - (c.d)**



RTL VIEW:



DESIGN SUMMARY

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	2403	5720	42%
Number of fully used LUT-FF pairs	0	2403	0%
Number of bonded IOBs	324	102	317%

V. CONCLUSION

In this paper, we study the various wellsprings of predisposition when adjusting after FP HUB expansion and FMA. We present and contrast two structures with manage these wellsprings of inclination. From the mistake examination and the usage results we infer that the snake A++ is an answer (with a moderate expanded expense contrasted with one-sided viper A) for applications where the avoidance of the predisposition is required. Then again, the halfway impartial design A+ ought to be for the most part utilized since it forestalls a lot of inclination and includes an irrelevant increment in the equipment cost. Comparable arrangements could be applied to FMA designs.

VI. REFERENCES

[1] E. Alpaydin, Machine learning: the new AI, 2016.

[2] M. Kubat, An Introduction to Machine Learning, 2015, no. August.
[3] J. A. Stankovic, "Research directions for the internet of things," IEEE Internet of Things Journal, vol. 1, no. 1, pp. 3–9, feb 2014.
[4] F. K. Shaikh, S. Zeadally, and E. Exposito, "Enabling technologies for green Internet of Things," IEEE Systems Journal, vol. PP, no. 99, pp. 983–994, jun 2015.
[5] R. Murphy, T. Sterling, and C. Dekate, "Advanced architectures and execution models to support green computing," Computing in Science and Engineering, vol. 12, no. 6, pp. 38–47, nov 2010.
[6] P. P. Pande, A. Ganguly, and K. Chakrabarty, Design technologies for green and sustainable computing systems, P. P. Pande, A. Ganguly, and K. Chakrabarty, Eds. New York, NY: Springer New York, 2013.
[7] Synopsys, "DWFC Flexible Floating-Point Overview," no. August, pp. 1–6, 2016.
[8] "IEEE standard for floating-point arithmetic," IEEE Std 754-2008, pp. 1 –58, 29 2008.
[9] J. Hormigo and J. Villalba, "New formats for computing with real numbers under round-to-nearest, "IEEE Transactions on Computers, vol. 65, no. 7, pp. 2158–2168, 2016.
[10] D. R. Lutz and N. Burgess, "Overcoming double-rounding errors under IEEE 754-2008 using software," in 2010 Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers. IEEE, nov 2010, pp. 1399–1401.
[11] J. Hormigo and J. Villalba, "Measuring Improvement When Using HUB Formats to Implement Floating-Point Systems under Round-to-Nearest," IEEE Transactions on Very Large-Scale Integration (VLSI) Systems,
vol. 24, no. 6, pp. 2369–2377, 2016.
[12] H. L. Garner, "A survey of some recent contributions to computer arithmetic," IEEE Transactions on Computers, vol. C-25, no. 12, pp. 1277–1282, Dec 1976.
[13] W. J. Cody, "Static and dynamic numerical characteristics of floating-point arithmetic," IEEE Transactions on Computers, vol. C-22, no. 6, pp. 598–601, June 1973.
[14] J. F. Reiser and D. E. Knuth, "Evading the drift in floating-point addition," Information Processing Letters, vol. 3, no. 3, pp. 84–87, jan 1975.
[15] H. Hilton, "Bias-free rounding in digital signal processing," Apr. 22 2004, US Patent App. 10/277,419.
[16] J. Villalba-Moreno, "Digit recurrence floating-point division under HUB format," 23rd IEEE Symposium on Computer Arithmetic, Silicom Valley (California, USA), July 2016.
[17] M. D. Ercegovac and T. Lang, Digital Arithmetic. Morgan Kaufmann, San Francisco, 2004.

AUTHOR PROFILE'S

INDRAKANTI TEJENDRA CHARY doing post-graduation in VLSI and Embedded systems from Siddhartha Institute of Engineering and Technology in Department of Electronics and Communication Engineering.

DR. D SUBBA RAO, is a proficient Ph.D person in the research area of wireless communications from Rayalseema University, Kurnool along with initial degrees of Bachelor of Technology in Electronics and Communication Engineering (ECE) from Dr. SGIET, Markapur and Master of Technology in Embedded Systems from SRM University, Chennai. He has 16 years of teaching experience and has published 98 Papers in International Journals, 2 Papers in National Journals and has been noted under 4 International Conferences. He has a fellowship of The Institution of Electronics and Telecommunication Engineers (IETE) along with a Life time membership of Indian Society for Technical Education (ISTE). He is currently bounded as an Associate Professor and is being chaired as Head of the Department for Electronics and Communication Engineering discipline at Siddhartha Institute of Engineering and Technology, Ibrahimpatnam, Hyderabad.

M PUSHPALATHA working as associate professor in ECE branch, Siddhartha Institute of Engineering and Technology, Hyderabad, TS, India. She has completed her Master's degree with specialization in Digital Electronics and communication systems from Mahaveer Institute of Science & Technology affiliated to JNTU Hyderabad in the year 2008. Prior to this has completed bachelor's degree in Electronics and communication engineering from G. Narayanamma Institute of Technology and Science affiliated to JNTU Hyderabad. Pursuing Ph.d in the area of wireless communication, Sri Satya Sai University of technology and medical Science, Bhopal, Madhya Pradesh Her carrier started as a lecturer

and has total 7 years of experience in teaching field. Out of which 3 years worked as Assistant Professor and 4 years as Senior Assistant Professor in a single organization named Abhinav Hi-Tech College of Engineering. Presently working as an Associate Professor for Siddhartha Institute of Engineering and Technology in Department of Electronics and Communication Engineering. She attended and also conducted many workshops and conferences. She is very much interest to do research on DECS, wireless technology and signals.