

Implementation of High Speed Efficient Reversible Floating Point Arithmetic Unit

B. Venkatachalapathi

PG Scholar, Dept of ECE, VLSI-D, Sri Venkateswara College of Engineering, Tirupati, Andhra Pradesh, India
bvchallapathi.105n@gmail.com

Abstract— For computation or portrayal of vast or little numbers, substantial range is basic. These qualities can be spoken to utilizing the IEEE-754 standard based drifting point math portrayal. This venture presents proficient approach towards planning of rapid skimming point unit utilizing reversible rationale. There are different reversible usage of legitimate and number-crunching units have been proposed in the current research, however not very many reversible coasting point plans has been planned. Gliding point operations are utilized as often as possible in about all registering disciplines. The examination of proposed reversible circuit can be actualizing as far as speed and Area.

In existed technique Design single exactness gliding point multiplier, there is prerequisite of proficient 24x24 piece number multiplier. Operand deterioration approach is effective for the proposed reversible plan of 32 bit skimming point multiplier. To outline the reversible 24x24 (AxB) bit multiplier, the qualities are separated into three subgroups of 8 bits each. Hence, the 24x24 piece reversible increase can be performed through nine reversible 8x8 piece Wallace tree multipliers, of which yields are then summed. There are three theoretical stages in Wallace tree duplication incomplete item pressure utilizing 4:2 compressors, Partial item era, full & half adders and after that the last expansion stage to create the item. In this work there is necessity of advancement at each of these three phases. In Proposed outline rather than swell convey snake utilize convey select viper in the last expansion stage to lessen Area.

IEEE 754 Floating-point standard

The IEEE 754 coasting point standard is the most generally utilized standard for skimming point calculations and is followed in the vast majority of the CPU and FPU (Gliding point unit) usage. The standard characterizes a configuration for drifting point numbers, unique numbers, for example, the vast's and NAN's, an arrangement of skimming point operations, the adjusting modes and five special cases. IEEE 754 indicates four arrangements of portrayal: single accuracy (32-bit), twofold exactness (64-bit), single expanded (≥ 43 bits) and twofold broadened precisions (≥ 79 bits).

Under this standard, the drifting point numbers have three segments: a sign, an example and a mantissa. The mantissa has an understood shrouded driving concealed piece and the rest are portion bits. The most utilized arrangements depicted by this standard are the single accuracy and the twofold exactness skimming point number configurations which are appeared in Table 1. In every cell the main number shows the quantity of bits used to speak to every segment, and the numbers in square sections indicate bit positions held for every segment in the single-accuracy and double-precision numbers.

Table 1. Layouts for single and double precision numbers in IEEE 754 format.

Format	Sign	Exponent	Fraction / Mantissa	Bias
Single-precision	1 [31]	8 [30 – 23]	23 [22 – 0]	127
Double-precision	1 [63]	11 [62 – 52]	52 [51 – 0]	1023

The Sign bit: A sign bit value of 0 is used to represent positive numbers and 1 is used to represent negative numbers.

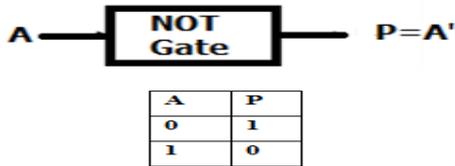
The Type: The example field has 8 bits in single-exactness and 11 bits in twofold accuracy. The esteem is put away in unsigned arrangement and a predisposition is added to the genuine type to get the put away type. For single-accuracy, the predisposition esteem is 127 and for twofold exactness it is 1023. Real type = put away example – 127 for single-accuracy and it is equivalent to put away type – 1023 for twofold exactness. De standardized numbers and zero have each of the zeroes in the type field. The unending and Not a number esteems have all one's in the type field. The scope of the example for single accuracy is from 126 to +127 and for twofold exactness it is - 1022 to +1023.

The Mantissa: Aside from the sign and the example a gliding point number additionally has a size part which is spoken to by the mantissa field. For single-accuracy the quantity of mantissa bits is 23 and for twofold exactness it is 52. Every mantissa has a shrouded bit which is not demonstrated when the coasting point is spoken to in the IEEE organize.

REVERSIBLE GATES

NOT Gate

The most straightforward Reversible entryway is NOT door and is a 1*1 entryway. The Reversible 1*1 door is NOT Entryway with zero Quantum Cost is as appeared in the Figure

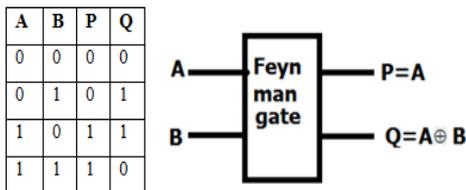


CNOT Gate

CNOT gate is also known as controlled-not gate. It is a 2*2 reversible gate. The CNOT gate can be described as: Iv = (A, B) ; Ov = (P= A, Q= A^B) Iv and Ov are input and output vectors respectively. Quantum cost of CNOT gate is 1.

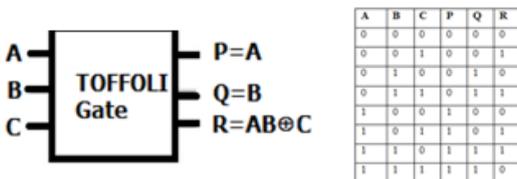
Feynman Gate

The Feynman gate which is a 2*2 gate and is also called as Controlled NOT and it is widely used for fan-out purposes. The inputs (A, B) and outputs P=A, Q= A XOR B. It has quantum cost one



Toffoli Gate

Figure shows a 3*3 Toffoli gate. The input vector is I (A, B, C) and the output vector is O(P,Q,R). The outputs are Fig 4 shows a 3*3 Toffoli gate. The input vector is I (A, B, C) and the output vector is O(P,Q,R). The outputs are defined by P=A, Q=B, R=AB XOR C. Quantum cost of a Toffoli gate is 5



Fredkin Gate

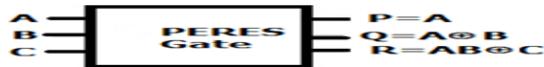
Figure shows a 3*3 Fredkin gate. The input vector is I (A, B, C) and the output vector is O (P, Q, R). The output is defined by P=A, Q=A^B XOR AC and R=A^C XOR AB. Quantum cost of a Fredkin gate is 5



A	B	C	P	Q	R
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	0	1
1	1	1	1	1	1

Peres Gate

Figure shows a 3*3 Peres gate. The input vector is I (A, B, C) and the output vector is O (P, Q, R). The output is defined by P = A, Q = A XOR B and R=AB XOR C. Quantum cost of a Peres gate is 4



A	B	C	P	Q	R
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

Double Peres Gate

Input vector, (Ip)v and output vector, (Op)v of 4x4 Double Peres Gate (DPG) [6] are defined as follows:

(Ip)v = {a,b,c,d} and

(Op)v = {a,a xor b,a xor b xor d,(a xor b)d xor ab xor c}

HNG Gate

Input vector,(Ip)v and output vector, (Op)v of 4x4 HNG are defined as follows:

(Ip)v={a,b,c,d} and

(Op)v={a,b,a xor b xor c,(a xor b)c xor ab xor d}

EXISTING METHODS

FLOATING POINT ADDITION

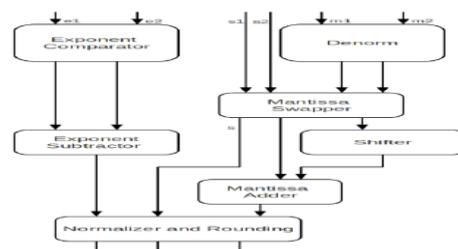


Figure1: Floating-point addition

Drifting point expansion is a straightforward algorithmic process that should be finished in consecutive request. The means associated with this procedure are outlined in figure. The subtraction operation additionally uses a similar equipment with the exception of that the sign piece is altered at the de standardize stage.

Shifter

The example estimation of the littler operand is changed in accordance with coordinate the estimation of the bigger

operand, this is finished by moving the mantissa bits fittingly.

Mantissa Adder

This is a simple process where the mantissa values are added together to generate the mantissa of the result.

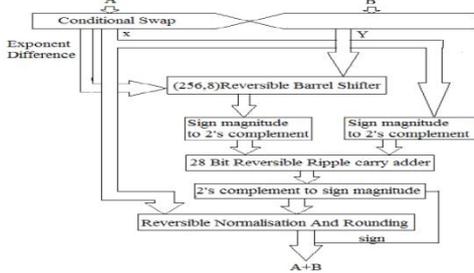


Figure2:

Proposed design of reversible addition architecture

Given two skimming guide numbers toward be included, the IEEE754 Standard for Coasting Point Number-crunching subtle elements how their whole can be found. In the first place, if the types are not equivalent, the littler is augmented until the point when it lines up with the bigger. To adjust the coasting point number with the littler type without modifying its esteem, its particular trailing significant must be moved one place to one side for each time the type is increased. Once the types are equivalent, the significands can be summed. The total is then standardized and adjusted. Fig. outlines the square level schematic of the engineering our proposed reversible skimming point snake configuration employments.

A) Reversible Contingent Swapping

The initial phase in our reversible skimming pointer viper design is to swap the gliding - point operands restrictively and reversibly. Whatever remains of the engineering works accepting that X is the skimming point number with the more noteworthy example, and Y is the drifting point number that perhaps should be lined up with X. The examples of the two gliding point numbers both are unloaded and extended to nine bits. Keeping in mind the end goal to discover their distinction the example that is the minuend is supplemented utilizing two's supplement, and with it the distinction is ascertained. The nine HNG entryways execute the reversible subtracter circuit. The sign piece of the distinction of the examples goes about as the condition (control) by which the two whole floatingpoint numbers are swapped: If $expA < expB$, at that point the drifting point numbers will swap positions, generally $expA \geq expB$ and the skimming point numbers are gone through to the following stage without swapping.

B) Reversible Barrel Shifter-

In our proposed framework we require (256,8) reversible barrel shifter. Be that as it may, because of intricacy and space issue we appeared here outline for (4,2) reversible barrel shifter. The circuit acts as takes after: Each phase

of Fredkin door moves the contribution as per the control estimation of sk. Assume, to plan a (4, 2) shifter which takes i_0, i_1, i_2, i_3 as information inputs and $s_0s_1 = 11$ as select information. So the information will be moved $2^0 + 2^1 = 3$ times to one side. Accordingly the grouping of the move/pivot operation will be $i_1i_2i_3 i_0$ for the main stage and afterward $i_3i_0i_1i_2$ for the following. Then again, for the select info $s_0s_1 = 00$, the information arrangement will stay same for the two phases of multiplexing. Consequently, each Fredkin entryway picks between two info lines it gets and plays out the fitting operation as per the select contribution of that specific stage. Henceforth, for the main (Stage 0) of over (4:2) shifter, the primary Fredkin door will either choose input i_0 or i_1 , the second one will do either i_1 or i_2 et cetera. Every single other stage play out the choice assignment in the same way.

D) 28-Bit Ripple Carry Full Adder-

For this usage, we will be utilizing the Peres door as it is the entryway with the lower quantum cost. The Peres door usage of Full Viper with its relating quantum cost can be seen

below:

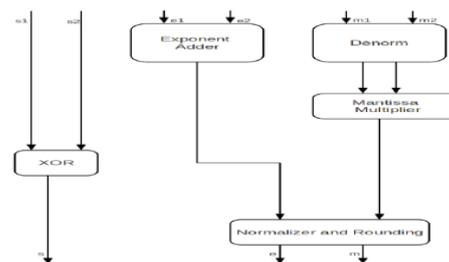


Figure3: Floating-point Multiplication

The mantissa multiplier is the most complex part of the increase pipeline. Since mantissa multiplier works correspondingly to a number multiplier, it is utilized as a part of the FPU of SCOORE to perform marked and unsigned whole number duplications. The initial phase in the duplicate pipeline is like that of expansion, we unload the operands into example, mantissa and sign fields. Much like all FP pipelines, FP duplication comprises of two noteworthy information ways, the type and the mantissa fields.

To produce the resultant example, we have to perform expansion on the one-sided types, guaranteeing that the inclination is subtracted from the outcome. The sign piece of the outcome is gotten by a XOR of the two sign fields of the information operands. The remaining and the most rationale serious stride is to play out the mantissa duplication. The duplication of two 52-bit operands yields a consequence of 104 bits.

Inside, both single and twofold exactness are dealt with the same, for single accuracy number we truncate suitably in the wake of getting the outcome.

The essential strides associated with the mantissa augmentation are halfway item era and fractional item expansion. The skimming point multiplier has a dormancy of three cycles for execution and standardization of the outcome. In the main stage, the halfway items are created and in the following two phases, the incomplete items are included.

Subsequent to getting the outcome, it may be important to standardize the come about by moving the mantissa and altering the type as needs be. The NaN recognition and engendering rationale is like that of the division unit, in this way we have regular rationale that is being utilized by both the pieces. We watch that NaN recognition and engendering can be finished in one cycle yet additional lemon are added to keep up synchronization with the mantissa multiplier pipeline. All the increase operations are finished all together.

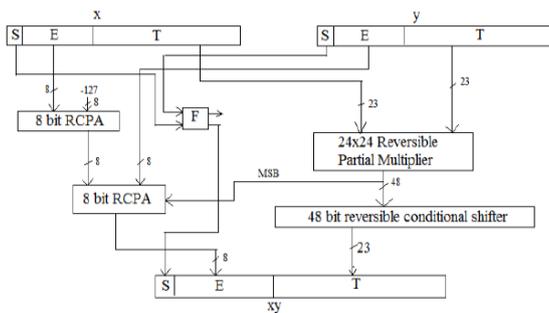


Figure4:Algorithm for floating point multiplication

For to configuration single accuracy gliding point multiplier, there is necessity of proficient 24x24 piece whole number multiplier. Operand disintegration approach is productive for the propose reversible plan of 32 bit coasting point multiplier. To plan the reversible 24x24 (AxB) bit multiplier, the qualities are isolated into three subgroups of 8 bits each. In this way, the 24x24 piece reversible augmentation can be performed through nine reversible 8x8 piece Wallace tree multipliers, of which yields are then summed. There are three calculated stages in Wallace tree augmentation incomplete item pressure utilizing 4:2 compressors, Halfway item era, full & half adders and afterward the last expansion stage to create the item. In this work there is necessity of improvement at each of these three stages.

PROPOSED METHODOLOGY:

The cluster multiplier characterized in the current technique requires vast deferral and colossal equipment for executing it. The swell convey snake actualized in the current technique requires huge convey proliferation delay. The stall multiplier utilized as a part of the skimming point duplication requires low region and

postponement. The convey spare expansion utilized as a part of the gliding point expansion and coasting point multiplier requires insignificant postponement at the cost of expanded hardware.

Stall's calculation analyzes nearby matches of bits of the N-bit multiplier Y in marked two's supplement portrayal, including a certain piece underneath the minimum noteworthy piece, $y_{i-1} = 0$. For each piece y_i , for i running from 0 to $N - 1$, the bits y_i and y_{i-1} are considered. Where these two bits are equivalent, the item gatherer P is left unaltered. Where $y_i = 0$ and $y_{i-1} = 1$, the multiplicand times 2^i is added to P; and where $y_i = 1$ and $y_{i-1} = 0$, the multiplicand times 2^i is subtracted from P. The last estimation of P is the marked product.

The portrayals of the multiplicand and item are not determined; regularly, these are both additionally in two's supplement portrayal, similar to the multiplier, however any number framework that backings option and subtraction will fill in also. As expressed here, the request of the means is not decided. Normally, it continues from LSB to MSB, beginning at $i = 0$; the duplication by 2^i is then commonly supplanted by incremental moving

of the P accumulator to the right between steps; low bits can be shifted out, and subsequent additions and subtractions can then be done just on the highest N bits of P. [1] There are many variations and optimizations on these details.

Y_i	Y_{i-1}	Operation
0	0	NOP
0	1	Addition
1	0	Subtraction
1	1	NOP

The calculation is regularly depicted as changing over series of 1s in the multiplier to a high-arrange +1 and a low-arrange -1 at the finishes of the string. At the point when a string goes through the MSB, there is no high-arrange +1, and the net impact is elucidation as a negative of the fitting value.

The existed coasting point multiplier is adjusted by supplanting the cluster multiplier with corner multiplier and snake via convey spare expansion.

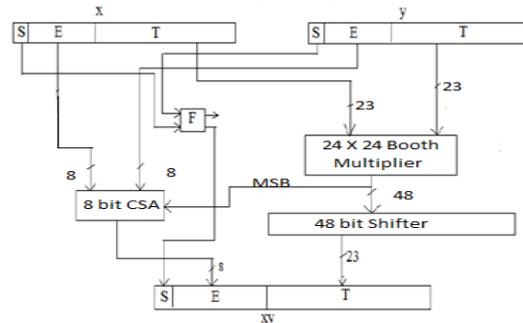


Fig: Proposed flow chart for floating point multiplication

SIMULATION RESULTS

Flow Status	Successful - Sat Jun 17 12:12:14 2017
Quartus II Version	9.1 Build 350 03/24/2010 SP 2 SJ Web Edition
Revision Name	FPAU
Top-level Entity Name	FPADD_RCA
Family	Stratix II
Met timing requirements	Yes
Logic utilization	2 %
Combinational ALUTs	117 / 12,480 (< 1 %)
Dedicated logic registers	172 / 12,480 (1 %)
Total registers	172
Total pins	222 / 343 (65 %)
Total virtual pins	0
Total block memory bits	0 / 419,328 (0 %)
DSP block 9-bit elements	0 / 96 (0 %)
Total PLLs	0 / 6 (0 %)
Total DLLs	0 / 2 (0 %)
Device	EP2S15F484C3
Timing Models	Final

Figure5: Design summary for 32 bit floating point addition

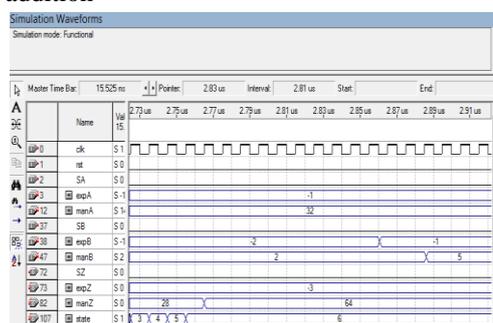


Figure6: Simulation results for 32 bit floating point addition

Flow Status	Successful - Sat Jun 17 12:02:36 2017
Quartus II Version	9.1 Build 350 03/24/2010 SP 2 SJ Web Edition
Revision Name	FPAU
Top-level Entity Name	fpmul
Family	Stratix II
Met timing requirements	Yes
Logic utilization	2 %
Combinational ALUTs	134 / 12,480 (1 %)
Dedicated logic registers	147 / 12,480 (1 %)
Total registers	147
Total pins	258 / 343 (75 %)
Total virtual pins	0
Total block memory bits	0 / 419,328 (0 %)
DSP block 9-bit elements	0 / 96 (0 %)
Total PLLs	0 / 6 (0 %)
Total DLLs	0 / 2 (0 %)
Device	EP2S15F484C3
Timing Models	Final

Figure7: Design summary for 32 bit floating point multiplier

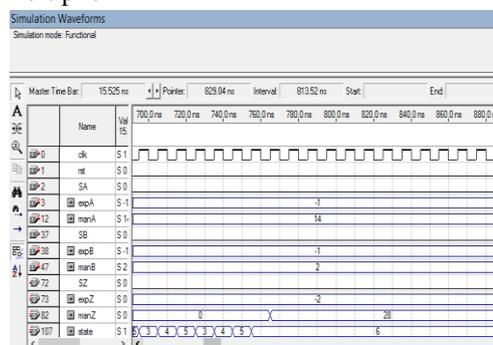


Figure8: Simulation waveform for 32 bit floating point multiplier

CONCLUSION

This venture introduces the gliding point unit as indicated by IEEE 754 Standard. The sum total of what modules

has been planned in reversible approach to diminish Territory and enhance execution. In existed strategy Configuration single accuracy drifting point multiplier, there is prerequisite of effective 24x24 piece whole number multiplier. Operand deterioration approach is proficient for the proposed reversible outline of 32 bit skimming point multiplier. To plan the reversible 24x24 (AxB) bit multiplier, the qualities are partitioned into three subgroups of 8 bits each. In this manner, the 24x24 piece reversible increase can be performed through nine reversible 8x8 piece Wallace tree multipliers, of which yields are then summed. There are three calculated stages in Wallace tree increase incomplete item pressure utilizing 4:2 compressors, Fractional item era, full & half adders and afterward the last expansion stage to create the item.

This venture shows the gliding point unit as per IEEE 754 Standard. The sum total of what modules has been outlined in reversible approach to decrease Range and enhance execution. In existed technique Configuration single accuracy drifting point multiplier, there is necessity of productive 24x24 piece number multiplier. Operand deterioration approach is productive for the proposed reversible plan of 32 bit skimming point multiplier. To plan the reversible 24x24 (AxB) bit multiplier, the qualities are separated into three subgroups of 8 bits each. Hence, the 24x24 piece reversible duplication can be performed through nine reversible 8x8 piece Wallace tree multipliers, of which yields are then summed. There are three reasonable stages in Wallace tree augmentation halfway item pressure utilizing 4:2 compressors, Fractional item era, full & half adders and after that the last expansion stage to create the item of Reversible 32 bit gliding point multiplier is outlined in VHDL and orchestrated and reproduced in ALTERA QUATRUS –II 9.1. From the outcomes it is clear that the proposed configuration utilized 134 Combinational ALUTs and 147 committed rationale registers. In this work there is prerequisite of advancement at each of these three phases.

REFERENCES

[1] R. Landauer, "Irreversibility and heat generation in the computational process," IBM Journal of Research and Development, 5, pp. 183-191, 1961.
 [2] C. H. Bennett, "Logical reversibility of computation," IBM Journal of Research and Development, pp. 525-532, November 1973.
 [3] A. K. Biswas, M. M. Hasan, A. R. Chowdhury, and H. M. H. Babu, "Proficient methodologies for planning reversible paired coded decimal adders," Microelectronics Journal, vol. 39(12), 2008

[4] R. P. Feynman, "Quantum mechanical PCs," *Select. News*, vol. 11(2), pp. 11–20, 1985.

[5] A. Peres, "Reversible rationale and quantum PCs," *Physical survey A*, vol. 32, pp. 3266–3276, 1985.

[6] W. Hung, X. Melody, G. Yang, J. Yang, and M. Perkowski, "Ideal combination of various yield boolean capacities utilizing an arrangement of quantum doors by representative reachability analysis," *IEEE Exchanges on*, vol. 25, no. 9, pp. 1652 –1663, sept. 2006.

[7]"IEEE Standard for Coasting Point Number juggling," *IEEE sexually transmitted disease 754-2008*, vol., no., pp.1-58, Aug. 29 2008.

[8]M. Haghparast, S. Jassbi, K. Navi, and O. Hashemipour, "Plan of a novel reversible multiplier circuit utilizing HNG door in nanotechnology," *World Connected Sciences Diary*, vol. 3, no. 6, pp. 974–978, 2008.

[9] Irina Hashmi, Hafiz Md. Hasan Babu "A Proficient Outline of a Reversible Barrel Shifter" 23rd Bury national Conference on VLSI Design,2010

[10] Alejandro Y. Perez V, " Reversible Snake Usage in VHDL" Computerized Configuration Utilizing HDL (ECE 590) Portland State College, Spring 2006

[11] Lafifa Jamal and Hafiz Md. Hasan Babu, "Productive Ways to deal with Outline a Reversible Gliding Point Divider" Division of Software engineering and Building, College of Dhaka, Dhaka-1000, Bangladesh

[12] David Goldberg, "What Each PC Researcher Should Think About Skimming Point Number-crunching", *ACM Figuring Studies*, Vol 23, No 1, Walk 1991, Xerox Palo Alto Exploration Center, 3333 Coyote Hill Road, Palo Alto, California 94304