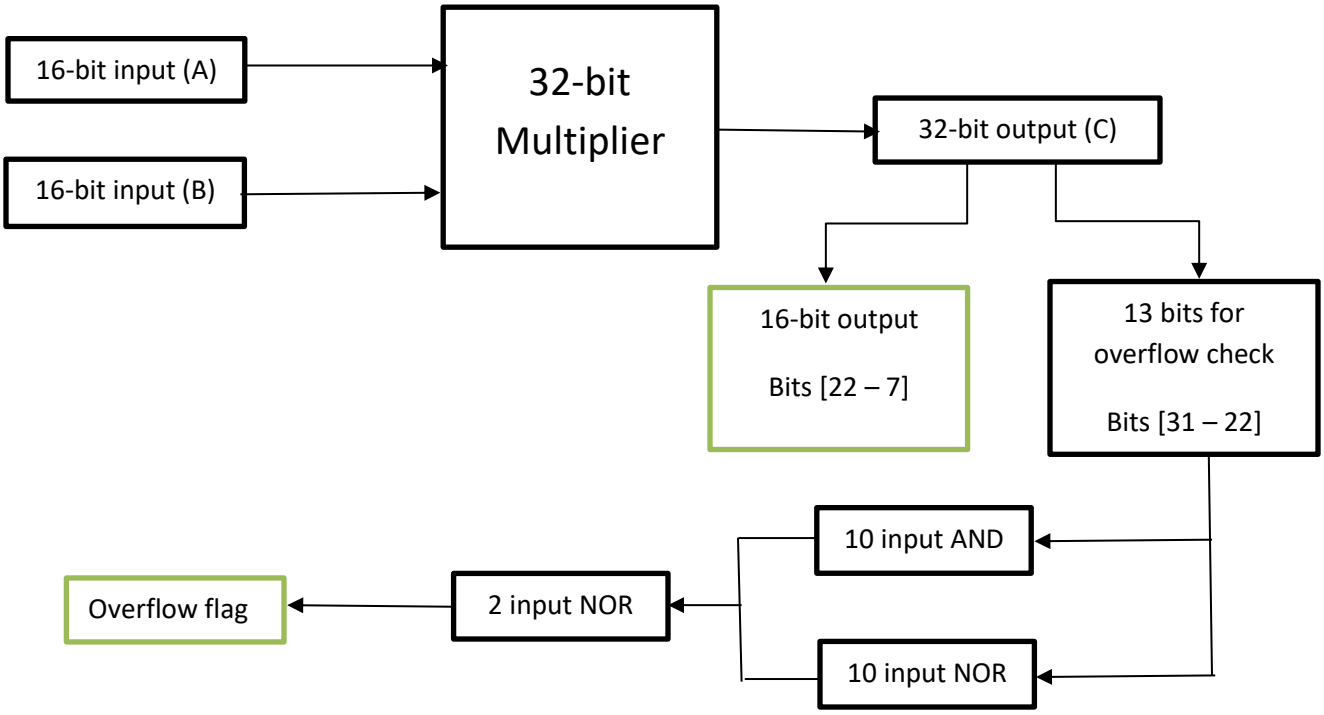
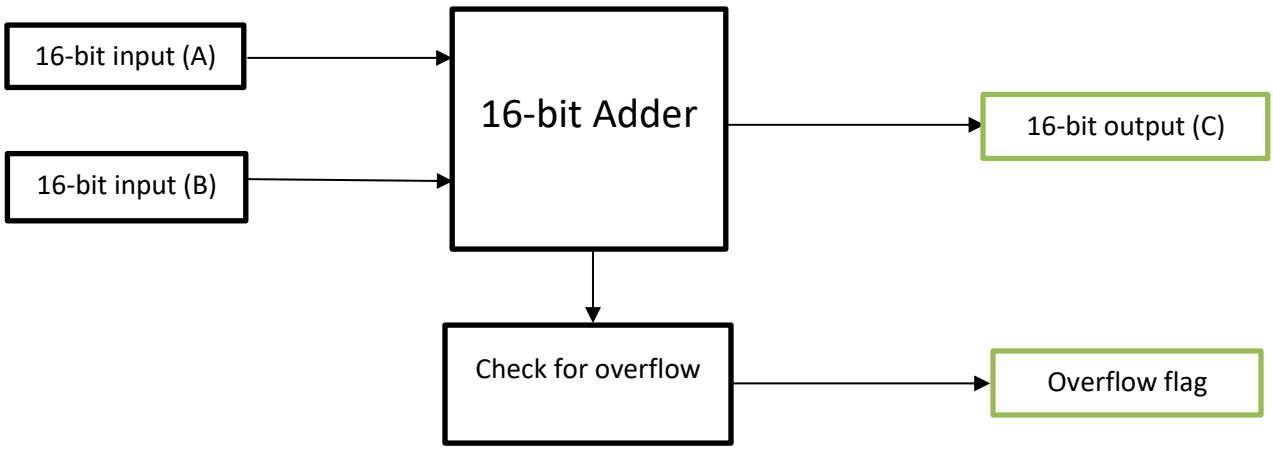
Implementation Report

Multiplication Design:



Addition Design:



Assumptions:

1. We used a constant scale factor of 7.
2. The inputs are turned into signed numbers inside adder and multiplier blocks.
3. Overflow flag in multiplication is calculated from the 10 most significant bits if they are all ones or all zeros then no overflow happens, else then an overflow happened.
4. The multiplication output should be 32 bits but we take 16 bits from it which are in the middle discarding the 7 least significant bits and the 9 most significant bits.

Multiplication Limitations:

1. Safe zone for input is 6 bits in integer part so that when we multiply them, we get valid 12 bits as output in addition to 4 bits for fraction, but we added an overflow flag to indicate such error.

Selected Algorithms:

1. Multiplication algorithms:

1.1 Fixed point algorithms:

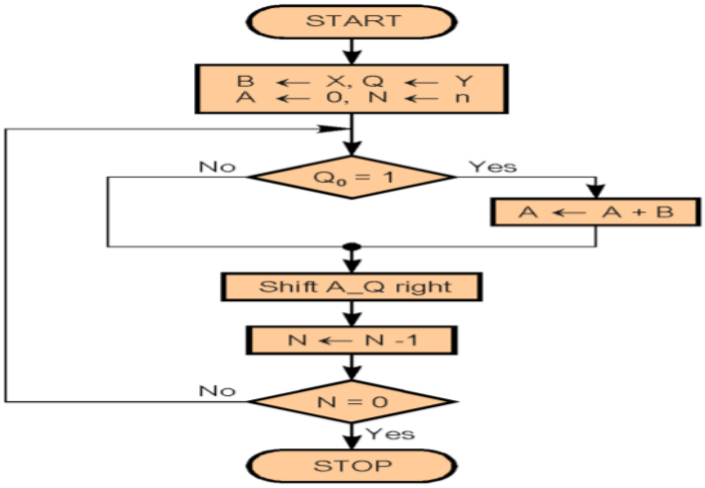
1.1.1 Booth algorithm:

Booths algorithm is a multiplication algorithm that utilizes two's complement notation of signed binary numbers.

The advantage of booth algorithm it reduces the number of partial product and the complexity of the circuit to generate a partial product bit in booth encoding [1].

1.1.2 Shift and Add Multiplier:

The algorithms take the digits of the multiplier one at a time from right to left, multiplying the multiplicand by a single digit of the multiplier and placing the intermediate product in the appropriate positions to the left of the earlier results [2].



1.2 Floating point algorithms:

1.2.1 Dadda algorithm:

proposed a sequence of matrix heights that are predetermined to give the minimum number of reduction stages in order to realize the minimum number of reduction stages, the height of each intermediate matrix is limited to the least integer that is no more than 1.5 times the height of its successor.

The advantage of dada algorithm is that multiplication time is reduced and achieved the speed of 526MHz as mentioned in paper [3].

1.2.2 Double precision multiplication using Vedic multiplier [4].

2. Addition algorithms:

2.1.Floating point algorithms:

2.1.1. Floating point addition using Adder-nor [5].

2.1.2. Two path floating point adder [6].

2.2.Fixed point algorithms:

2.2.1. Carry look Ahead Addition [7].

2.2.2. Carry select Addition [8].

Carry select adder has additional hardware but it introduces less delay. It consists of ripple carry adders and a multiplexer. It performs the addition one with the assumption that the input carry will be one and the other assuming it will be zero. The correct sum and the correct carry are then selected using the multiplexer.

Synthesis reports:

Component	Area	Timing	Power
Multiplier	- Cells: 800 - Area per cell :1286	- Data arrival time = 1010.63 - Slack = 40989	Total power = 1983.71
Addition	- 21 cell - 69 per cell	- Data arrival time = 867.4 - Slack = 41132	Total power = 117.25

Task Distribution:

Name	Task	Hours spent	Problems
Reham Gamal	- VHDL file of Multiplier Test bench - Floating point and fixed points algorithms (Multiplication) - Design software (Multiplication)	- 2 -3 -2	- Understanding the required correctly - Find algorithms for floating point
Mohamed Ahmed Ibrahim	- VHDL file of Multiplier - Floating point and fixed points algorithms (Multiplication) - Design software (Multiplication)	-2 -3 -2	
Mazen Amr Fawzy	- VHDL file of Adder - Floating point and fixed points algorithms (Addition) - VHDL file of Adder Test bench - Synthesis	-4 -3 -1 -1	- At first, we designed for variable scale factor and made dedicated bit for the scale factor in both input and output (Design, Software and VHDL code and testbench). - Then we changed it to fixed as it gave better synthesis results in terms of power, area and timing.
Mahmoud Ahmed Sebak	- VHDL file of Adder - Floating point and fixed points algorithms (Addition) - Design software (Addition) - Synthesis	-4 -3 -2 -1	

References:

[1] https://en.wikipedia.org/wiki/Booth%27s_multiplication_algorithm

[2] http://www.iosrjournals.org/iosr-jvlsi/papers/vol4-issue2/Version-1/E04212935.pdf?fbclid=IwAR0K_jrArhPDxfcvybidnmwP_mrDmxy4icMvsAAMljCoEHPLZb021kW255o

[3] http://wineyard.in/Abstract/mtech/VLSI/bp/WYV54.pdf

[4] http://www.researchpublish.com/download.php?file=Design%20of%20Double%20Precision-2228.pdf&act=book

[5] http://www.arpnjournals.org/jeas/research_papers/rp_2017/jeas_0117_5606.pdf

[6] https://sci-hub.tw/https://ieeexplore.ieee.org/abstract/document/1261822

[7] https://web.stanford.edu/class/ee486/doc/chap3.pdf

[8] https://tams.informatik.uni-hamburg.de/applets/hades/webdemos/20-arithmetic/20-carryselect/adder_carryselect.html