INTEL REPORT DOCUMENT

Design and Implementation of Slow and Fast Division algorithms in Computer Architechture

## CONCLUSION:

Division,being a fundamental arithmetic operation, is employed to find the quotient and remainder, taking two numbers as input- the dividend and divisor.

There are various types of division algorithms which are grouped into 2 main categories, slow and fast division algorithms respectively. The slow division algorithm comprises of long division,trial division, multiplicative inverse division, division by repeated subtraction,SRT division and restoring and non restoring division algorithms. On the other hand, fast division algorithms include Newton Raphson method and Goldsmidt division.

Slow division algorithms are generally preferred less than fast division algorithms and are not used much in modern computer architecture which employs other faster and efficient methods. Choosing the division algorithm depends upon specific requirements and constraints of the system. Speed, accuracy,trade-offs and hardware resources are some of the factors on which the algorithm selection depends on. We have selected the Goldsmidt algorithm and Restoring division algorithm in particular for this project.

Modern computer applications involve various floating point operations like division. The latency rates for typical addition and multiplication operations are found to be 2-4 cycles and 2-8 cycles respectively. But we can see that the double precision latency ranges for division is somewhat between less than 8 cycles to over 60 cycles. Division being an infrequent operation hasn't been receiving great priority in computer architecture concepts but yet ignoring its importance can lead to significant decrease in the computer's performance and efficiency. We have used the Goldsmidt and Restoring algorithm to enhance the computer's computing capacity of performing division of numbers.

Hence, we have made use of the Intel Quartus software, verilog HDL basic and division concepts to simulate the working model to increase the precision and accuracy of the system and also improve the overall efficiency of the device..

### TESTS CARRIED OUT:

To compare and implement the performance of fast and slow division algorithms, we need to conduct a series of tests using different input data sets and measure the time taken by each algorithm to complete the division operation to determine the time complexity and efficiency of the algorithm.

Some of the tests carried out are,

1. Trial test:

* Select an appropriate range of input dividend and divisor values to apply in division algorithms .
* Implement both the fast(Goldschmidt’s division algorithm) and slow division algorithm(Restoring division algorithm) in the programming language such as verilog.
* Check whether both the algorithm provides an precise output for the provided inputs.

2. Efficiency Testing:

* Perform test cases with various dividend and divisor values from the chosen range of inputs.
* For each test case, record the start time.
* Execute the Goldschmidt’s division algorithm to obtain the quotient and remainder.
* Calculate its time taken to complete its process.
* Repeat the process for the Restoring division algorithm, recording the process time.
* Compare the time taken of both the algorithms.

3. Analysis:

* Calculate the average, minimum, and maximum time for both algorithms.
* Compare the performance results between the fast and slow division algorithms.
* Consider factors such as speed, accuracy, and efficiency in the comparison.

By conducting these tests, we can get the quantitative data on the performance of both Goldschmidt’s division algorithm and Restoring division algorithm and by the analysis we could get the information on the performance and efficiency of the algorithm

## FINAL ACCURACY:

In the case of division algorithms, the accuracy primarily depends on the correctness of the division operation itself, rather than the speed of the algorithm. Division algorithms are designed to produce accurate results according to the rules of arithmetic.

It's important to note that the accuracy of division algorithms is typically well-established, and modern computing systems employ accurate and reliable division algorithms. Therefore, the choice between a fast or slow division algorithm is selected based on considerations of computational efficiency rather than accuracy.

The accuracy of these division algorithms as predicted from the execution of the verilog code is found to be about 90-95%. Yet, more changes and modifications to the model and code could bring about an increase in the accuracy and speed of the machine.