**Working of Algorithm**

|  |  |
| --- | --- |
| Image result for utd  ASIC – CE 6306  Homework 5 Report | Abstract  A C++ implementation to accomplish the computation presented in MSDAP paper (PART II).  Anmol Gautam, Soumyadeep Choudhury  Net ID : AXG190014, SXC180056 |

The algorithm mentioned in the second half of the MSDAP paper introduces a new methodology to perform same task as mentioned in part 1 of the paper in a manner which requires less computation and processing power.

The working of our implementation of the same in C++ is described below in the form of operation/steps.

Operation 1 : Coefficient Value is read from the file provided and saved in a two-dimensional array after preprocessing. Preprocessing contains separating the MSB/sign bit and storing it in the 0th index of each row in the array and the rest of the number is converted to integer and stored in the 1st index of each row of the array.

Operation 2 : Rj Value is read from the file provided and saved in a array after converting to integer.

Operation 3 : Input value is read from the file provided and converted to binary and extended to 24 bits and stored in a list of lists.

Operation 4 : For each value of Rj we calculate U(j) using the formula provided in the homework description which uses coefficient to find each input values as much as specified by each Rj value. Each of those are added or subtracted based on the MSB of each coefficient value and then the final U(j) is achieved. Therefore, the number of U(j) values are equal to the number of Rj values.

Operation 5 : Now at this stage all the values ready for operating. We further use the formula below to calculate final output y(n).

*y (n) =* 2-1(….2-1 (2-1 (2-1*u1+ u2*) *+ u3*) *+ ….*) *+ u16*)

At each step we right shift a value and then add/subtract it with its preceding u(j) value.

Operation 6 : Finally the output is converted to hexadecimal format containing 40bits and stored in data.out file.