TASK 3

Aim:

The task is to implement an LMS adaptive filter

Task 1:

To estimate the number of memory words required for the LMS filter.

Objective :

To estimate the approximate number of operations per second for updating and filtering using a LMS filter. Assuming that the sampling frequency is 48 kHz and the LMS filter length is 64 coefficients.

Conclusion :

The number of operations used for FIR filtering are specified and sampling frequency is 48 KHZ

Task 2:

To Implement the LMS filter using float myLMS (float x, float d, float \* e, float mu, float \* state, float \* coeff, const int length)

* **x**: the current input sample
* **d**: the current desired signal sample
* **e**: is a pointer to the output error signal
* **mu**: step size of the LMS filter.
* **state**: the filter state, i.e., memory of previous input samples length is length
* **coeff**: FIR filter coefficients as a vector of length length.
* **length**: the length of the FIR filter

Conclusion :

The implementation of the LMS filter is done by calculating the error signal and output signal y(n) at the received end.

Task 3:

To find  the number of cycles in the simulator for one call to the LMS filter using 64 coefficients.

Objective:

The filter coefficients are updated by multiplying the error signal with input signal and adding the desired signal.

Conclusion:

The number of cycles in the simulator are build and printed with the print f function and the result is compared with the number of memory words.