ECSE 444 Final Project Initial Report

Gauthier, Ian
Computer Engineering
McGill University
260654775
ian.gauthier@mail.mcgill.ca

Misra, Kartik
Computer Engineering
McGill University
Student ID
email

Bieber, Anna
Computer Engineering
McGill University
260678856
anna.bieber@mail.mcgill.ca

Kalia, Paarth
Computer Engineering
McGill University
260662155
paarth.kalia@mail.mcgill.ca

***Abstract* — The goal of this report is to detail the design decisions made by group 1 on the first deliverable of final project of ECSE 444- Microprocessors. This includes how the sine waves were generated, mixed, and output onto the DAC to be able to be seen on the oscilloscope.**

I. Introduction

The goal of the final project, as a whole, is to create an audio application that employs Blind Source Separation using the FastICA algorithm. In essence, the application can be broken down as follows:

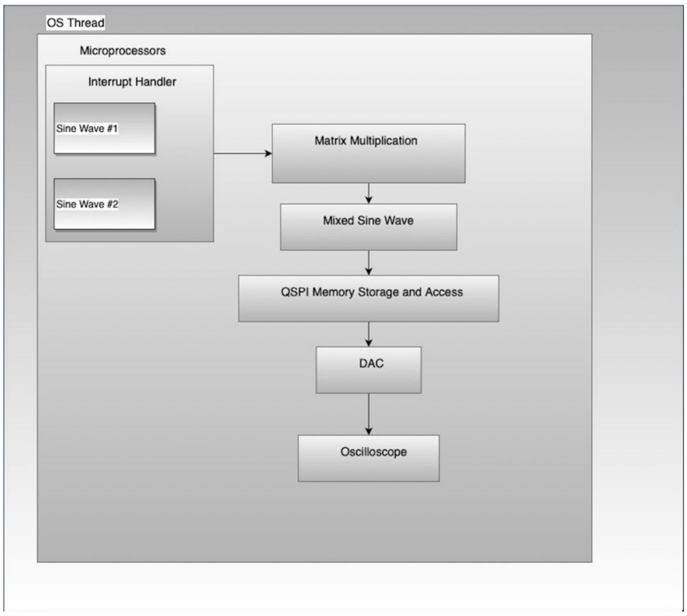
- a system which is able to first generate multiple sine waves at different frequencies
- store them into flash memory
- mix them into a single sine wave
- use the FastICA algorithm to recreate and in turn output the original sine waves

This report will detail the generation, storage and mixing of multiple sine waves.

II. Solution Overview

A. Implementation Approach

At a high-level, the implementation approach was as follows: initialization and starting of a single thread; generation of two sine waves in the thread -- named Sine_Wave_Task -- and mixing them using matrix multiplication; finally, outputting the mixed sine wave onto the microprocessor board so the result could be viewed on an oscilloscope. The diagram below shows the process of the program.



Process Diagram

B. Components of Microprocessors

The following components of the microprocessor were to be initialized to accomplish the aforementioned: Interrupt handler, QSPI and both DAC channels. Interrupt handler was needed to correctly sample the sine wave. Before using the QSPI for memory storage, erasure of the QSPI's data is necessary so that the sine waves may be written onto the flash memory. This action was performed block by block as it was found that calling the Erase_Full_Chip method could be implemented to get the desired output of writing to memory.
Finally, the Sine_Wave_Task thread was begun, which encompassed the functionality, provided by the above components, to effectively output the mixed sine wave onto the oscilloscope.

C. Process

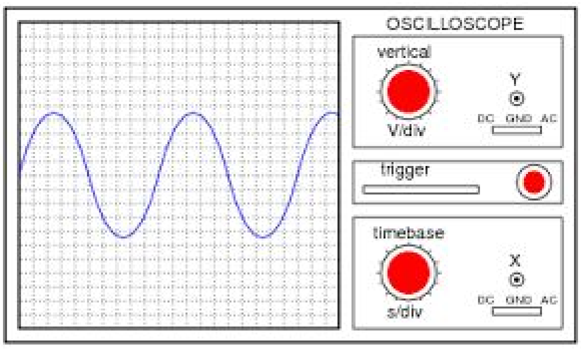
First, the status of the QSPI was checked to see if it was ready - before entering into the infinite for-loop. This status determines if the QSPI is ready to be written to. After, the sine wave generation began for two sine waves of different frequencies. Sine generation is performed using the arm sine

function with the various parameters which are expected by the system. Parameters are given by the following equation:
$s\,(t) = \sin sin\ \ (2\pi f \frac{1}{Ts}\ )$
Where f is the frequency and Ts is the sampling period.

As the DAC only accepts values starting at 0, the sine waves were offset to ensure that they are always positive numbers and thus can be output by the DAC and stored as unsigned numbers.
It is at this point that the system mixed the two signals using matrix multiplication, as is shown in the handout, before writing the signals to the QSPI flash memory. Once all of this was performed, the system checked the timer flag value; this was done so the signal could be sampled correctly, and the proper frequency be outputted by the DAC. Inside this block, the system read the information from the QSPI memory and outputted the sine wave to the DAC so that it can be tested to see that the correct frequency was being output. It is to be mentioned here that writing to the memory was successful, however, we were unable to successfully read from memory and in turn output to the DAC. This is mentioned in detail in the following *Results* section. To overcome this problem, our team directly output the mixed sine wave to the DAC to check the frequency on the oscilloscope.

III. Results

In terms of results, the system is able to perform most of the tasks that are necessary for the completion of the first deliverable. It is able to generate two sine waves - both at the correct frequency. The generated sine waves would be similar to the one on the oscilloscope shown below.



Oscilloscope

The oscilloscope also allowed us to measure the exact frequency of each signal, in their unmixed and mixed form. It is then able to store them and mix them using matrix multiplication before outputting them to the DAC. However, there are some issues present related to the integration of the reading of data from the QSPI memory and outputting that data to the DAC. When using the read from QSPI system, the group is able to observe outputs in the watch window meaning that data is being received from the memory. However, when outputting this data to the DAC, the group has not been able to observe any meaningful information and as such the section is currently unfinished.

IV. Goals Moving Forward

Given that this report is being made at the midpoint of the final project's runtime, it is expected that much of the work will still be unfinished. As stated above in section II, the integration of the QSPI system with the DAC output system needs to be completed correctly so that the group can store the both of the two second sine waves without any loss of information. In addition, the second half of the project - which deals with the implementation of a FastICA algorithm to separate the two sine waves - needs to be completed before the final deliverable.