Formal report - Goertzel Implementation:

(EENGM4120. Deadline: 13rd March 2023 23:59)

Please note that all examples and laboratory exercises conducted during the laboratory sessions are not assessed. This coursework is the only marked assessment for this course.

Tasks to perform:

1. You are required to study the code and implement the Goertzel algorithm in the C language in order to detect one frequency (GTZ_one_freq.zip). **(Total mark 10%)**

2. Complete the GTZ_all_freq.zip project to detect all 15 digits (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, #, A, B, C and D). You are given a data.bin data file and a data_audio.wav audio file, corresponding to an 8-digit tone that you need to decode.

    a Copy the data.bin file into the CCS project.
    b Complete the first 3 TODO sections in main_gtz.c and util.c to implement the Goertzel algorithm and decode the tone in the data file. **(Total mark 30%)**
    c Complete the last TODO section in util.c to generate an audio file based on the decoded tone. This audio should sound the same as the data_audio.wav. **(Total mark 10%)**

3. Modify the C algorithm to increase the performance of the Goertzel algorithm (task 2b main_gtz.c) using intrinsics and compiler switches. Measuring the number of cycles before and after the optimization. An example project of how to time your code is given in timing.zip. **(Total mark 30%).**

4. Write a formal report explaining the implementation and showing/discussing the results. Presentation will be taken into account. **(Total mark 20%).**

The report should follow the template given on Blackboard and should not exceed 6 pages; extra pages can be used in the appendix.

Please submit **one PDF report and one zip file** with all the Goertzel projects. (Please DO NOT submit only the source files). Marks will be deduced if a project will not compile. Only **ONE submission per group** is required, with names and email addresses included in the report. Submission should be through Blackboard. Please note all members of the group are responsible for plagiarism check.

The following two references show examples of implementation [1] [2].

[1] N. Dahnoun, Digital Signal Processing Implementation: Using the TMS320C6000 DSP Platform, Prentice Hall, 2000.

[2] N. Dahnoun, "Multicore DSP: From Algorithms to Real-time Implementation on the TMS320C66x SoC," 2018. [Online and hard copies at University of Bristol Library.], also available: https://www.amazon.co.uk/Multicore-DSP-Algorithms-Real-Time-Implementation/dp/1119003822. [Accessed February 2021].