Integrating Project Pitch

Group 1
Chan, Ethan Lester
Dolon, John Michael
Lu, Andre Giancarlo
Teng, Adriel Shanlley

Introduction

Discrete Fourier Transform (DFT)

- Converts signal from time domain to frequency domain
- Used for filter design and comparing multiple signals

Inverse Discrete Fourier Transform (IDFT)

- Inverse of DFT

Integrating Project Proposal

Plan: To implement DFT and IDFT through a program written in C, and optimize it using CUDA

Inputs: Signal (array of floating-point values), Signal Frequency

Outputs: Input signal transformed through DFT into the frequency domain; Reverted signal from performing IDFT on the signal in the frequency domain

Integrating Project Proposal

Computations to be Parallelized

For DFT:

$$\boxed{X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi kn}{N}} \quad k = 0, 1, \ldots, N-1}$$

For IDFT:

$$\boxed{x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j\frac{2\pi kn}{N}} \quad n = 0, 1, \ldots, N-1}$$

Integrating Project Proposal

How the Processes will be Parallelized

The program starts by allocating each element of our input array to a thread and the program runs on the threads one by one. The program will then split all the threads to blocks where each block will contain 1024 threads. This allows each block to run in parallel with each other, speeding up the process of computing our output array. This results in our program running only 1024 times, but each time will have multiple instances of our computation running at the same time.

Integrating Project Proposal

Existing Implementation

- Java, performed by Douglas Lyon (2009)
- Device used was a 1.5 GHz Celeron with 1 GB RAM and a 1.6.11 version of the JVM that was used as the platform for the code
- The runtime of the DFT as a function of 65,536 array length was 934,375ms, or 93.4375s

Our proposal will be significantly faster since ours will parallelize the computations for the DFT and IDFT

Integrating Project Proposal

Implementation Platform

- C programming language with CUDA
- Visual Studio 2022 as the platform

References

Arar, S. (2017, July 20). *An Introduction to the Discrete Fourier Transform.* All About Circuits. Retrieved from https://www.allaboutcircuits.com/technical-articles/an-introduction-to-the-discrete-fourier-transform/

Delgutte, B., Greenberg, J. (2005). Chapter 4 - *The Discrete Fourier Transform*. Biomedical Signal and Image Processing. Retrieved from https://web.mit.edu/~gari/teaching/6.555/lectures/ch_DFT.pdf

DFT and IDFT (n.d.). Retrieved from https://www.kth.se/social/upload/50919490f2765474b1890dde/lec2.pdf

Lyon, D. (2009, May). The Discrete Fourier Transform, Part 1*. The Journal of Object Technology 8*(3):17-26. Retrieved from (PDF) The Discrete Fourier Transform, Part 1. (researchgate.net)

Project Nariyuki (n.d.). *How to implement the discrete Fourier transform*. Retrieved June 20, 2024, from https://www.nayuki.io/page/how-to-implement-the-discrete-fourier-transform

Sidhu, T., Bhajla, B., Das, S. (2023). Numerical algorithms for protection and metering devices. *Encyclopedia of Electrical and Electronic Power Engineering* (pp. 45-87). Elsevier. https://doi.org/10.1016/B978-0-12-821204-2.00131-8

Thank you for listening!

(/^ ▽ ^)/