EE5811

FPGA

Project Report

Discrete Cosine Transform

Venkatesh Vayyavuru
ee19mtech01010

Shiva Kumar Chintigari
ee19mtech01001

Instructor: Dr.G.V.V. Sharma

April 26, 2019

Contents

1 Motivation

Discrete Cosine Transform(DCT) is one of the important image compression and speech compression algorithms used in image and speech processing application. In speech processing instead of sending total samples of wavefile we can send fewer DCT coefficients and by using IDCT we can reconstruct original speech signal with small loss in data. It involves a lot of multiplications, additions and also has a huge memory requirement. Several algorithms have been proposed over the last couple of decades to reduce the number of computations and memory requirements involved in the DCT computation algorithm. Any algorithm that can reduce the total number of additions, multiplications or memory requirement would be of profound significance to the image processing and speech preocessing domain.The idea behind using transform based coding is signal energy compaction into a small number of coefficients.DCT is similar to the Discrete Fourier Transform since it transforms a signal or image from the spatial domain to the frequency domain. However one primary advantage of the DCT over the DFT is that the former involves only real multiplications, which reduces the total number of required multiplications, unlike the latter. Another advantage lies in the fact that for most images much of the signal energy lies at low frequencies, and are often small - small enough to be neglected with little visible distortion. The DCT does a better job of concentrating energy into lower order coefficients than does the DFT for image data.The another reason for the popularity of DCT is that it has no discontinuity at the boundaries of finite blocks of infinite signals. Where as DFT tends to have discontinuities at the boundaries of finite length sequences. blocks of infinite signals

2 Introduction

A discrete cosine transform (DCT) expresses a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies. The use of cosine rather than sine functions is critical for compression, since it turns out that fewer cosine functions are needed to approximate a typical signal.DCT is similar to the DFT but it operates on real numbers. There are eight standard DCT variants, of which four are common. The most common variant of discrete cosine transform is the type-II DCT, which is often called simply "the DCT".In this experiment we implemented type-II DCT and reconstructed the given sequence using Inverse Discrete Cosine Transform(IDCT).

3 Discription

In my implementation, I explore design points of my hardware implementation using Verilog HDL and evaluate the area-performance trade-off. The design comprises of two modules. One module for DCT computation, anothee for IDCT computation. In the implementation we used Arduino as interface for transmitting and receiving the data from the Raspberry Pi which includes Field

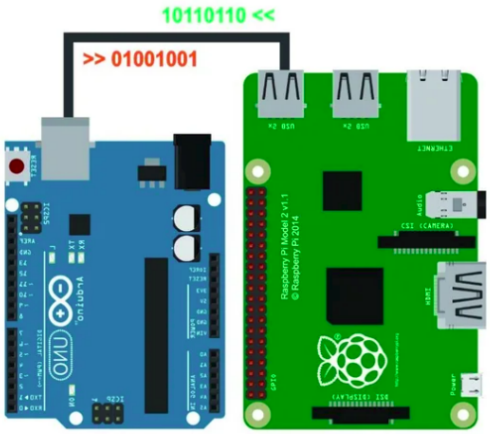

Figure 1: Arduino and Raspberry Pi Communication

Programmable Gate Array(FPGA).The process of implementation describing in following steps.First step in the implementation is sending the real values of signal to the Raspberry Pi by using the Arduino. Next step is implement the type II DCT and inverse type II IDCT using Verilog-Hardware Description Language(HDL) in FPGA.Final step is receiving the real values of signal from Raspberry Pi to Arduino.

4 Theory

4.1 Communication between Arduino and Raspberry Pi

Raspberry Pi and Arduino are the two most popular open source boards in Electronics Community. They are not only popular among Electronics Engineers but also among school students and hobbyists, because of their Easiness and Simplicity. Even some people just started liking Electronics because of Raspberry Pi and Arduino.The communication between Arduino and Raspberry Pi which transfer vital information by sending data one bit at a time.Serial communications are essential for every Micro-controllers to communicate between Micro-controllers and another device. The Micro-controller sends these 1 and 0 (bits) that contain necessary information one by one, or Serially. These bits form together and turn into bytes (composed of 8 bits).

4.2 DCT and IDCT implementation in IcoBoard FPGA

Real-time implementation of the DCT operation is highly computationally intensive. Accordingly, much effort has been directed to the development of suitable cost effective VLSI architectures to perform this. Traditionally the focus has been on reducing the number of multiplications required. Additional design criteria has included minimizing the complexity of control logic, memory requirements, power consumption and complexity of interconnect. In the Four types of DCT the DCT-II is probably the most commonly used form. The mathematical discription of the type II discrete cosine transform is defined as

$$X_k = \sum_{n=0}^{N-1} x_n * cos[(\frac{\pi}{N}) * (n + \frac{1}{2})k] \quad k = 0...N-1 \tag{1}$$

This transform is exactly equivalent to a DFT of 4N real inputs of even symmetry where the even-indexed elements are zero. For given DCT-II coefficients we can recover original signal inverse DCT-II.The mathematical discription of the inverse of DCT-II is defined as

$$x_n = \frac{1}{2} * [\frac{1}{2} * X0 + \sum_{k=1}^{N-1} X_k * cos[(\frac{\pi}{N}(n + \frac{1}{2}]] \quad n = 0...N-1 \tag{2}$$

The DCT-II and DCT-III are transposes of one another. The DCT-III implies the boundary conditions: xn is even around n = 0 and odd around n = N; X_k is even around $k = -\frac{1}{2}$ and even around $k = N - 1/2$.

4.3 Computation in DCT

The computational complexity of the N point DCT is $O(N^2)$ so fast implementations are desired (compare with FFTs O(N log N) complexity). For 4-point DCT direct computations of 16 multiplications and 12 additions required.If we use butterfly algoritm for 4 point DCT 6 multiplications and 12 additions.

5 Results

In verilog we synthesized for 4 point DCT and followed by IDCT. fig-2 shows the original and reconstructed sequence,

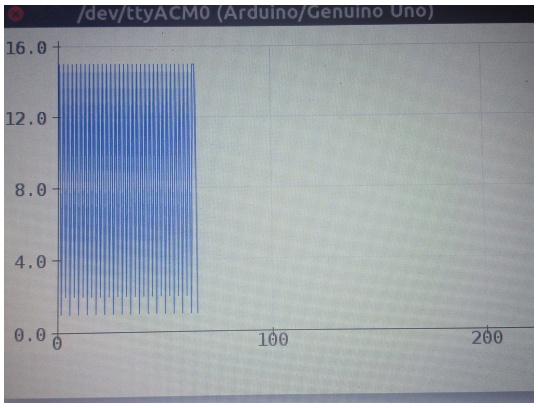

Figure 2: recovered Signal

REFERENCES:

https://github.com/gadepall/EE5811/blob/master/icoboard_fpga/gvv_hemanth_icoboard.pdf

https://en.wikipedia.org/wiki/Discrete_cosine_transform

https://ieeexplore.ieee.org/abstract/document/6376736