PROJECT
ES-203

Implementation of 8 point 1D - DCT and IDCT algorithms on FPGA

Akshay Biju (17110011) Manoj Kumar Kumawat (17110080) Naman Kumar Singh (17110089) Saumitra Sharma (17110135)

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 8-point discrete cosine transform (DCT) is widely used in video and image compression and is a core component in contemporary media standards like JPEG and MPEG. The use of cosine rather than sine functions is critical for compression, since it turns out (as described below) that fewer cosine functions are needed to approximate a typical signal. The standard DCT algorithm involves a lot of arithmetic operations like addition and multiplication, therefore, it requires a lot of memory. Thus, over time faster variants of the DCT algorithm were introduced. One such algorithm was introduced by Arai, Agui and Nakajima (AAN Algorithm).

Objective

Our objective is to implement the fast 8 point 1D-DCT and 1D-IDCT computation algorithms by Arai, Agui, and Nakajima on FPGA boards, for 8 bit inputs, as these boards are tailored specifically to minimise the computation time required.

Background

1-D DCT Algorithm: -

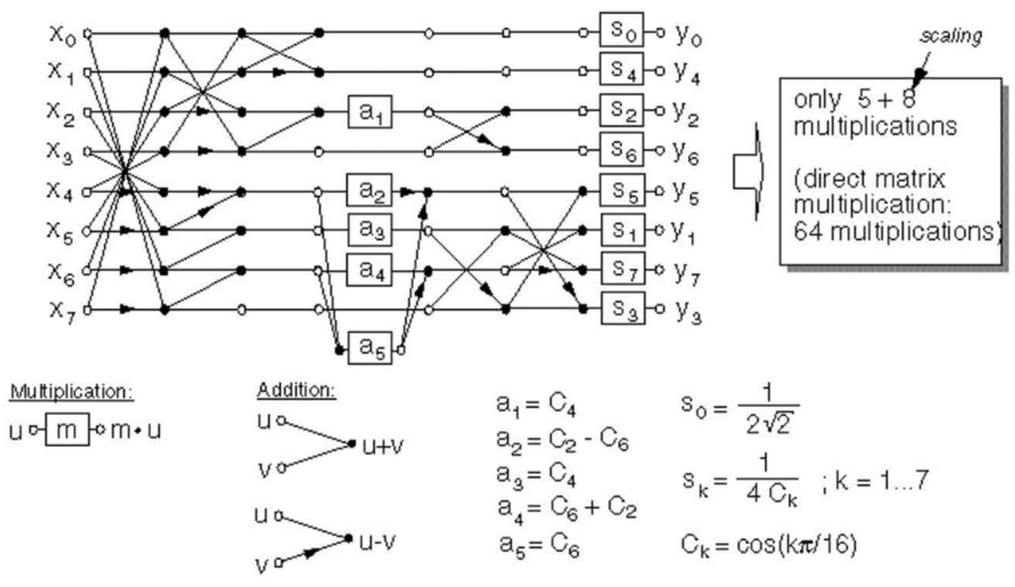
There are several variants of the DCT depending on the type of boundary conditions used to define the finite length transform. To compute 1-D DCT we are using below algorithm which is known as the DCT - II or simply "The DCT".

$$y(k) = \sqrt{\frac{2}{N}} \sum_{n=1}^{N} x(n) \frac{1}{\sqrt{1+\delta_{k1}}} \cos\left(\frac{\pi}{2N}(2n-1)(k-1)\right)$$

where x[n] ∈ R is a data sequence of length N (which is 8 in this case) and coefficients , k = 1,2 . . . , N are expressed by

$$\frac{1}{\sqrt{1+\delta_{k1}}} = \begin{cases} \frac{1}{\sqrt{2}}, & \text{if } k = 1 \\ 1, & \text{otherwise} \end{cases}$$

Block diagram of an 8-point Arai 1-D DCT architecture is given below, in which the indexing starts from k= 0 and n = 0:



1-D IDCT Algorithm :

To compute 1-D IDCT we are using below algorithm which is known as the DCT - III and we are using this because the DCT-II and DCT-III are inverse of each other.

$$y(k) = \sqrt{\frac{2}{N}} \sum_{n=1}^{N} x(n) \frac{1}{\sqrt{1+\delta_{n1}}} \cos\left(\frac{\pi}{2N}(n-1)(2k-1)\right)$$

where x[n] ∈ R is a data sequence of length N (which is 8 in this case) and coefficients , n = 1,2 . . . , N are expressed by

$$\frac{1}{\sqrt{1+\delta_{n1}}} = \begin{cases} \frac{1}{\sqrt{2}}, & \text{if } n = 1 \\ 1, & \text{otherwise} \end{cases}$$

[20/11, 5:06 PM] piyushdewangan06

$$A_0 = \frac{1}{2\sqrt{2}} \approx 0.3535533906,$$

$$A_1 = \frac{\cos(7\pi/16)}{2\sin(3\pi/8) - \sqrt{2}} \approx 0.4499881115,$$

$$A_2 = \frac{\cos(\pi/8)}{\sqrt{2}} \approx 0.6532814824,$$

$$A_3 = \frac{\cos(5\pi/16)}{\sqrt{2} + 2\cos(3\pi/8)} \approx 0.2548977895,$$

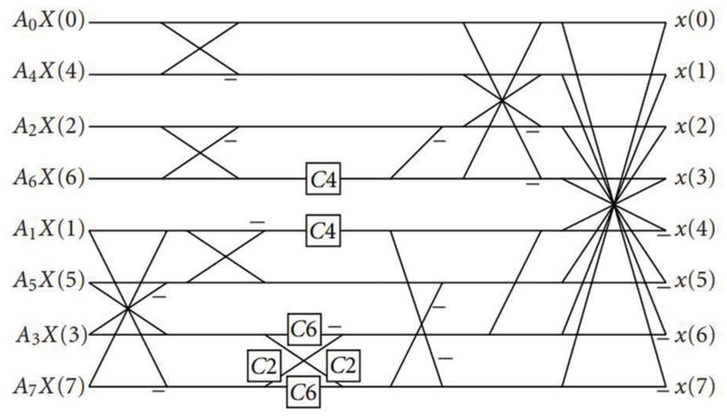
$$A_4 = \frac{1}{2\sqrt{2}} \approx 0.3535533906,$$

$$A_5 = \frac{\cos(3\pi/16)}{\sqrt{2} - 2\cos(3\pi/8)} \approx 1.2814577239,$$

$$A_6 = \frac{\cos(3\pi/8)}{\sqrt{2}} \approx 0.2705980501,$$

$$A_7 = \frac{\cos(\pi/16)}{\sqrt{2} + 2\sin(3\pi/8)} \approx 0.3006724435.$$

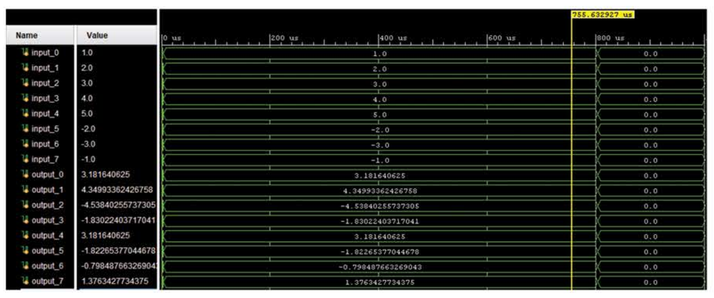
The graphic scheme of data flow the IDCT transform in the Arai's algorithm



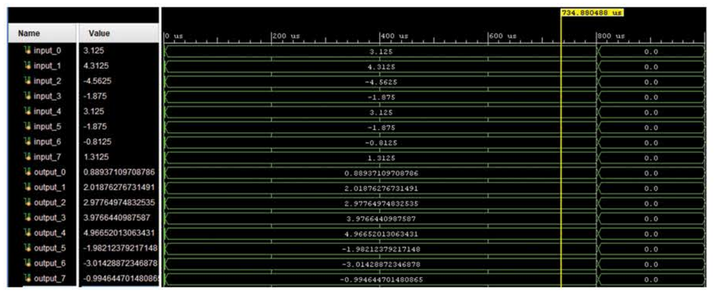
$$Cn = \cos\frac{n\pi}{16}$$

Implementation

First we compared the results of a standard DCT-II MATLAB code with another MATLAB code for the AAN 1D-DCT Algorithm. We did the same for DCT-III (to test 1D-IDCT) algorithm, and found out the results to be comparable.
Next, we implemented this algorithm as a sequential program on verilog. We used verilog's internal arithmetic operations like addition, multiplications etc. in our program.
Then we compared the simulation results against the MATLAB code for both AAN 1D-DCT and 1D-IDCT algorithms, and got accurate results.
For the FPGA implementation (for both algorithms) we gave the 8-bit inputs using the switches from W13 through V17, the inputs are signed and in binary with 4 bits on either side of decimal point.
We used the switches from T1 through V2 to select between the 8 points of the inputs. The switch R2 and the three push buttons (BTNU, BTNL and BTNR) are used to execute the algorithm.
We used seven segment display module in combination with push button BTND and the switches from T1 through V2, and V16 and V17 to display the outputs. BTNC is used as a reset button.

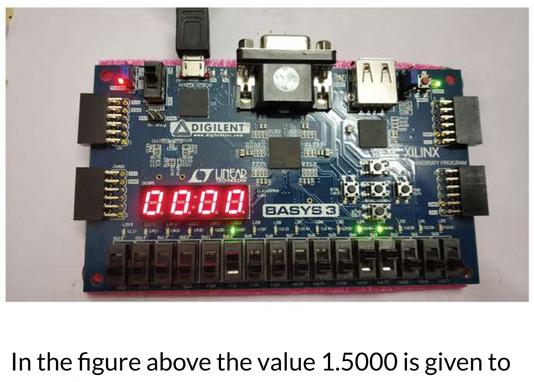


Simulation results of 1D-DCT Algorithm



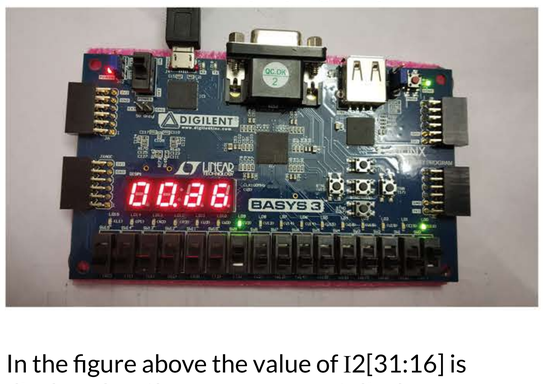
Simulation results of 1D-IDCT Algorithm

While giving input



In the figure above the value 1.5000 is given to input I3.

While displaying output



In the figure above the value of I2[31:16] is displayed on the seven segment display.

Limitations

- Inputs are of limited size i.e. 8 bits.
- As the inputs are signed the range of input is limited i.e. from -8 to 7.9375
- In our implementation we do not get the output instantaneously (we have to hold down the push buton for around 5 seconds to execute the algorithm and retrieve outputs)

References

- https://link.springer.com/content/pdf/10.1155%2F2009%2F485817.pdf
- http://www.cs.cmu.edu/~dongw/final_fantasy/545FinalReport.html
- https://www.mathworks.com/help/signal/ref/dct.html
- https://www.youtube.com/user/Amper6
- https://cs.anu.edu.au/courses/csprojects/16S1/Reports/Artem_Afanasyev_Report.pdf
- https://pdfs.semanticscholar.org/b82d/c072d055a8c758f76b256588aee0097ccf28.pdf