ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 8, Issue 5, May 2021

High Speed Low Power 4-Point Discrete Cosine Transform

[1] Sikharam Giridhar Sai, [2] N.Purnachand

[1][2] School of Electronics Engineering, VIT-AP University, Inavolu, A.P., India

***Abstract*--- DCT (Discrete Cosine Transform) communicates finite succession of information regarding the amount of cosine functions oscillating at various frequencies. The utilization of cosine is basic for compression as incidentally, less cosine capacities are expected to estimate through for differential conditions. Being a transformation technique and one among the complex ones, any n point DCT has complex calculation procedures that also uses matrices. This results in a larger area and power trade-offs. This paper tries to cope up with these particular trade-offs and try to find more efficient ways by using the floating point multiplication/multiplier techniques as the floating point numbers are utilized to address non integer fractional numbers are utilized in most designing and specialized computations. The proposed DCT module operates with very high frequency and with very low dynamic power dissipation.**

***Keywords*--- Discrete cosine transform, floating point numbers and multipliers, data compression, trade-offs**

I. INTRODUCTION

A discrete cosine transform (DCT) demonstrates a finite number of knowledge points in terms of a sum of cosine functions oscillating at different frequencies. The DCT is a largely used transformation technique in signal processing and data compression. "A discrete cosine transform uses n real basis vectors whose coefficients are cosines that are quickly being computed from a Fast Fourier Transform" [2]. "Also, it is a real transform with better computational efficiency than DFT as it compacts the data into sets of discrete blocks" [1] [5].

The utilization of these functions is basic for compaction, a DCT is a Fourier-related change like the (DFT) [6], and are identical to DFTs of generally larger length, working on real information with even symmetry, though in certain variations the information or potentially yield information are moved considerably by half a sample.

"And there are different types of DCT equations as mentioned below" [1].

DCT-1: Corresponds to the boundary conditions

$$X_k = \frac{1}{2}(x_0 + (-1)^k x_{N-1}) + \sum_{n=1}^{N-2} x_n \cos\left[\frac{\pi}{N-1} nk\right] \qquad k = 0, \ldots, N-1.$$

DCT-2: The most commonly used DCT

$$X_k = \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right] \qquad k = 0, \ldots, N-1.$$

DCT-3: Most commonly termed as the Inverse DCT(IDCT)

$$X_k = \frac{1}{2}x_0 + \sum_{n=1}^{N-1} x_n \cos\left[\frac{\pi}{N} n\left(k + \frac{1}{2}\right)\right] \qquad k = 0, \ldots, N-1.$$

DCT-4: Data from various transforms is being overlapped and this is being termed as modified discrete cosine transform(MDCT).

$$X_k = \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}\left(n + \frac{1}{2}\right)\left(k + \frac{1}{2}\right)\right] \qquad k = 0, \ldots, N-1.$$

DCT-5 to 8: These are the higher order types of DCT that concentrate more on the boundary conditions along with them being even/odd functions regarding the point of symmetry.

As the frequently used type of discrete cosine transform is DCT Type-2, we used it for the calculation of a N-point DCT(4-point).This is done in two ways, one being the conventional approach i.e., using the general formula/equation of type-2 DCT as mentioned above where, k=0,1,….N-1 and N is the radix point. And the other way is using the dct butterfly diagram and then comparing the results.

This paper deals with the calculation methods of both the processes.

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 8, Issue 5, May 2021

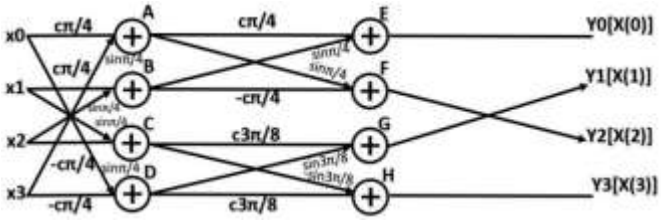

Fig 1: Butterfly diagram for 4-point DCT

"Floating point numbers are one possible way of representing real numbers in binary format" [7]. A number's radix point (decimal point, or, all the more generally in PCs, binary point) can be put at any place comparative with the number of digits. This point is demonstrated as the exponent component, fig 1 and accordingly the floating point portrayal can be contemplated as a sort of insightful documentation. "Multiplication is performed on normalized floating-point operands by following the steps below" [7],

- Check for zero(if any of the given numbers is 0 or not).
- Add the exponents and subtract 127(because exponents are biased and when we add both the exponents, the resultant exponent is biased twice).
- Mantissas are multiplied.
- The sign of the result is decided (xor operation between the signs of two numbers).
- Normalize the resulting value, if necessary.
- Here for multiplying the mantissas, we can use multiplication algorithm on unsigned numbers.

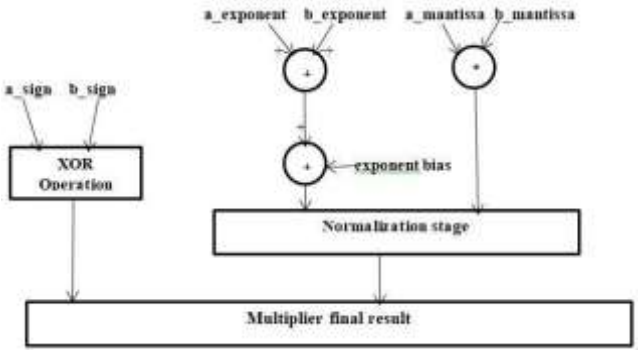

Fig 2: General representation of Floating point multiplier

II. LITERATURE REVIEW

In [1], The authors proposed that, a DCT is defined and a method to compute it using the Fast Fourier transform(FFT) and that DCT, can be used in the area of digital processing for the purposes of pattern recognition.

In [5], The authors proposed that, by considering the fast computation of a 2 dimensional discrete cosine transform of a N*N DCT can be computed using only N 1-D DCT's and hence the number of multiplications of the proposed method is only half of conventional row-column approach.

In [6], The authors proposed a fast 2-D DCT algorithm that reduces the mutliplies by 50 to 75% with comparable number of additions.

In [7], The authors proposed the effective execution of an IEEE 754 single precision floating point multiplier towards carrying out an independent pipelined plan and this execution handles the over and underflow cases.

In [8], The authors proposed about a decimal floating point multiplication is significant in numerous business applications. The curiosity of the plan is that, this proposed first equal floating point multiplier offering low inertness and high throughput.

III. FLOATING POINT CONVERTER

From fig 2 it is clear as to how we can calculate any two numbers, on the basis of IEEE 754 floating point multiplier. Table 1 shows the various calculated results along with their respective mantissas and exponents along with their signs.

	Decimal	Sign	Exponent	Mantissa	Binary Rep
	3.5	0	10000000	1.1E+22	1E+30
	4.5	0	10000001	1E+20	1E+30
Result	15.75	0	10000010	1.11E+22	1E+30
	2.5	1	10000000	1E+21	1E+30
	3.5	0	10000000	1.1E+22	1E+30
Result	8.75	1	10000010	1.1E+19	1E+30
	3.25	0	10000000	1.01E+22	1E+30
	4.25	0	10000001	1E+19	1E+30
Result	13.8125	0	10000010	1.01E+22	1E+30

Table 1: Example calculations using floating point

Here, the calculations are done manually using the floating point multiplier techniques and results are as shown in the Table 1. The exponent and mantissa change on the basis of their respective decimal point equivalent where as the final multiplied result takes the sign of combined result i.e. if any one of the number is negative then the result is negative.

Xilinx Vivado 2018.2.

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 8, Issue 5, May 2021

IV. SIMULATION RESULTS

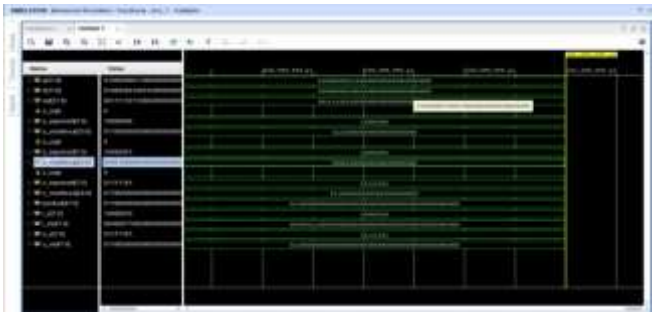

Fig 3: Simulation result for floating point multiplier

In fig 3, two 32 bit numbers are taken and then they are multiplied by using the floating point algorithm and the resultant output is also a 32 bit number. The sign of the output depends on the xor operation between the sign of two numbers.

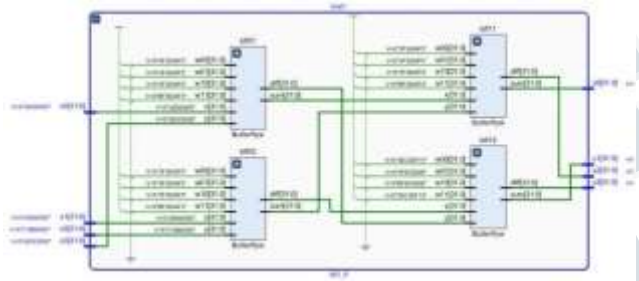

Fig 4: RTL Schematic for 4-point DCT

In fig 4, 4 number of both inputs and outputs, each of 32 bit are taken and the simulation is being performed.

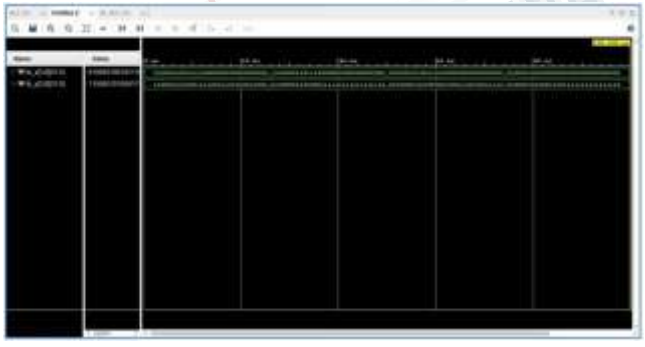

Fig 5: Simulation result of 4 point DCT

In fig 5, the simulation result along with the multiplier is being generated.

The respective DCT along with the floating point multiplier, along with their respective test benches are being coded in Verilog HDL and is being simulated and synthesized in

Table 2: Synthesis Summary Report

Property Name	Value
Slice LUT's(17600)	2960
Slice Registers(35200)	2928
Max Clock frequency	464.037 MHz
Power dissipation(Dynamic)	41 mW

The above result gives the synthesis report, which is done in Xilinx Vivado tool using Zynq Z7 series FPGA library. It is observed that the DCT operates at very high frequency which is 464.037 MHz with very less dynamic power dissipation (41 mW).

V. CONCLUSION

The discrete cosine transform is designed using the Verilog code and the results are simulated along with that of floating point multiplier. From the synthesis report, it is observed that the DCT can operate at very high frequency with low power dissipation. By using the floating point multiplier non integer numbers are also calculated as shown in Table 1 and the cosine and sine terms from the fig 1 i.e., the butterfly diagram calculations are done manually and been verified with the results acquired from the dct type-2 equation.

VI. FUTURE SCOPE

It very well may be utilized to change over the signal (spatial data) into mathematical data ("frequency" or "phantom" information) so that the pictures data exist in a quantitative structure that can be controlled for pressure dependent on a more extensive assortment of areas can be utilized for application, for example, encoding translating video sound multiplexing control signals flagging and simple to computerized transformation. They can likewise be utilized for top notch television (HDTV) encoder/decoder chips. Likewise the area effectiveness isn't undermined, utilizing the floating point multiplier decreases it comparatively. Further more, it can likewise be utilized in FPSOC (SoC FPGA) applications.

REFERENCES

[1] Ahmed, Nasir, T_ Natarajan, and Kamisetty R. Rao. "Discrete cosine transform." *IEEE transactions on Computers* 100.1 (1974): 90-93.

[2] Strang, Gilbert. "The discrete cosine transform." *SIAM review* 41.1 (1999): 135-147.

[3] Zhou, Jianqin, and Ping Chen. "Generalized discrete cosine transform." *2009 Pacific-Asia Conference on Circuits, Communications and Systems*. IEEE, 2009.

[4] Wang, Zhongde, and B. Hunt. "The discrete cosine transform--A new version." In *ICASSP'83. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 8, pp. 1256-1259. IEEE, 1983.

[5] Cho, Nam Ik, and San Uk Lee. "Fast algorithm and implementation of 2-D discrete cosine transform." *IEEE transactions on circuits and systems* 38.3 (1991): 297-305.

[6] Vetterli, Martin. "Fast 2-D discrete cosine transform." *ICASSP'85. IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 10. IEEE, 1985.

[7] Al-Ashrafy, Mohamed, Ashraf Salem, and WagdyAnis. "An efficient implementation of floating point multiplier." *2011 Saudi International Electronics, Communications and Photonics Conference (SIECPC)*. IEEE, 2011.

[8] Hickmann, Brian, Andrew Krioukov, Michael Schulte, and Mark Erle. "A parallel IEEE P754 decimal floating-point multiplier." In *2007 25th International Conference on Computer Design*, pp. 296-303. IEEE, 2007.

[9] Kanhe, Aniruddha, Shishir Kumar Das, and Ankit Kumar Singh. "Design and implementation of floating point multiplier based on vedic multiplication technique." *2012 international conference on communication, information & computing technology (ICCICT)*. IEEE, 2012.

[10] Zhang, Hang, Wei Zhang, and John Lach. "A low-power accuracy- configurable floating point multiplier." *2014 IEEE 32nd International Conference on Computer Design (ICCD)*. IEEE, 2014.

[11] Even, Guy, Silvia M. Mueller, and Peter-Michael Seidel. "A dual precision IEEE floating-point multiplier." *Integration* 29.2 (2000): 167-180.