Arithmetic Cosine Transform Architecture for Computing the Discrete Cosine Transform for Signal Processing Applications

Sangeetha Mohan, Praveen Konda

Abstract— This paper introduces the Arithmetic Cosine Transform (ACT), speedy algorithm for the evaluation of Discrete Cosine Transform (DCT) in digital signal processing. The common algorithms which used to calculate accurate value of DCT includes floating point operations and are mainly concentrated on multiplication which in turn causes the round-off error to occur. The Arithmetic Cosine Transform is proposed for the rapid calculation of DCT where the computation is focused on addition and constant multiplications which reduces the internal errors like round off and truncation. ACT results in less area consumption and low power consuming operations in case of zero-mean input signals. Calculation of ACT can be done with reasonable accuracy, reduction in area and less power using the novel architecture described for non-null mean signals as well. For the computation of eight-point DCT, ten non-uniform sampling instances are required when ACT is introduced. The implementation are done with Xilinx ISE Design suite 10.1 and coded in Verilog HDL. The two architectures are simulated and synthesized using Cadence encounter and the physical design is obtained.

Index Terms— Arithmetic Cosine Transform, Discrete Cosine Transform, Digital Signal Processing, Mean value, Mertens function, Nonnull mean signal, Null mean signal

____ 🌢

1 INTRODUCTION

THE Signal processing is a method to analyze the characteristics of a signal like storage, amplification, compression, reconstruction etc. The Discrete Cosine Transform (DCT) is a signal processing technique which converts a signal from its spatial domain in to frequency components. The existing DCT calculations include floating point operations which lead to computational errors caused by rounding off the values. The Arithmetic Cosine Transform (ACT) algorithm consists of only addition and constant multiplication which in turn reduces the computation error. The ACT can be used to calculate the exact and approximate value of the DCT for null mean and non null mean input sequences respectively. The algorithm is with less area complexity and consumes low power. The main application of DCT is in data compression. Its property states that the DCT coefficient contains most of the relevant information about the image so that it can be used in image compression applications. The DCT is mainly used in JPEG applications which are the algorithms for lossy image compression. Some applications include automatic surveillance, geospatial remote sensing, traffic cameras, satellite based imaging, automotives [1].

2 DISCRETE COSINE TRANSFORM

The Discrete Cosine Transform is a conventional signal processing technique used in number of applications. It's property states that the DCT coefficients contains most of the relevant information about the image so that it can be used in image compression applications. The Arithmetic Cosine Transform algorithms (ACT) are used for quick computation of DCT. The ACT consists of only additions and multiplying with constant value. This overcomes the errors associated with rounding off the values when floating point operations are come in to picture. The exact evaluation of ACT is possible if the input data are non-uniformly sampled and has zero mean. This paper unfolds two main issues (i) calculation of mean value of input signal in case of non-uniformly sampled data and (ii) proposition of efficient architectures of ACT for calculating the 8 point DCT when the input data are considered as only non-uniform samples [1].

2.1 Arithmetic Cosine Transform

The Arithmetic Cosine Transforms (ACT) is a speedy algorithm for evaluating the DCT of non-uniformly sampled input data. The incoming signal to the DCT are generally treated as continuous signal u(t), that are uniformly sampled. This produces the column vector $\mathbf{u} = \{u_n\}_{n=0}^{N-1}$ with dimension N. Its DCT is represented by the vector $\mathbf{U} = \{U_k\}_{k}^{N}$. The corresponding non-uniform samples of the input sequence u (t) are required to compute the vector U. The sampling instants are givenby

$$s = \frac{2rN}{k} - \frac{1}{2} \tag{1}$$

where k = 1, 2, ..., N-1 and r = 0, 1, ..., k - 1

Substituting for r = 0, k = 1 gives s = -1 / 2

For r = 1, k = 2 implies s = 15 / 2

Substituting in the same way, all other values of set S is obtained.

A set S with sampling points as the elements is defined as

International Journal of Scientific & Engineering Research, Volume 7, Issue 5, May-2016 ISSN 2229-5518

$$S = \{\text{All values of S}\}$$
(2)
For an 8-point DCT,
$$s \in S = \left\{-\frac{1}{2}, \frac{25}{14}, \frac{13}{6}, \frac{27}{10}, \frac{7}{2}, \frac{57}{14}, \frac{29}{6}, \frac{59}{10}, \frac{89}{14}, \frac{15}{2}\right\}$$
(3)

Here the ACT algorithm is represented in two ways to compute the DCT for zero mean sequence and non-zero mean sequence. When considering zero-mean sequence, the ACT averages the $A_{\mathbf{k}}$ as

$$A_{k} \triangleq \frac{1}{k} \sum_{r=0}^{k-1} u_{2r\frac{N}{k} - \frac{1}{2}}, \quad k = 1, 2, ..., N - 1$$
(4)

The above ACT averages can be used in the evaluation of DCT of non-uniform input samples by using the expression

$$U_{k} = \sqrt{\frac{N}{2}} \sum_{j=1}^{\left\lfloor \frac{N-k}{k} \right\rfloor} \mu(j) \cdot A_{kj}$$
(5)

Where k = 1, 2, ..., N-1 and $\mu(.)$ is called the Mobius function. The ACT is derived by using the Mobius inversion formula. In case of non-null mean input signal, a correction term is subtracted from the equation of U_k to calculate the DCT coefficients and it follows as

$$U_{k} = \sqrt{\frac{N}{2}} \sum_{j=1}^{\lfloor \frac{N-1}{k} \rfloor} \mu(j) \cdot S_{kj} - \sqrt{\frac{N}{2}} \ \overline{u} \cdot M(\lfloor \frac{N-1}{k} \rfloor) , \qquad (6)$$

where \overline{a} is considered as the arithmetic average of the input uniform samples given by

$$\bar{u} = \frac{1}{8} w_* v_r$$
 (7)

with w as the interpolation weight

$$M(n) = \sum_{r=1}^{n} \mu(r)$$
(8)

where M(n) is the Mertens function.

2.2 ACT Architectures

This paper introduces architectures for the ACT which accepts only non-uniform samples as inputs and calculate the DCT with reduced area complexity and low power consumption. All the above explained methodologies are used for the design of these architectures. There are registers are introduced at different nodes for the temporary storage which gives a fully pipelined structure to the design. This pipelined structure reduces the critical path delay with a slight increse in the latency. Architecture I corresponds to the ACT architecture for computing the DCT of null mean input signals. This architecture can be realized using (4) and (5). This architecture is done with only additions and constant multiplication with integers which reduces the truncation error and complexity. The Architecture I shown in Fig.1 curresponds to N=8 which takes 10 non-uniform samples as inputs according to the values of the set S given by (3). The applications dealing with zero mean input signal uses this architecture with the advantages of less complex computation and area. The simulation result for the Architecture I is shown in Fig.5. The second architecture is used for the calculation of DCT which has non-null mean input signals. It is desired to calculate the mean value of the incoming non-uniform samples. The Mertens correction function is included as per (6). Architecture II consists of Architecture I, mean calculation block and Mertens correction block as shown in Fig.2.

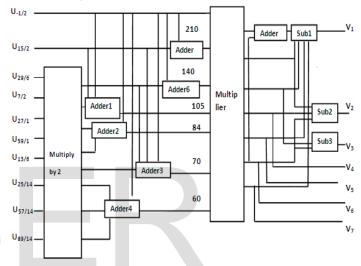
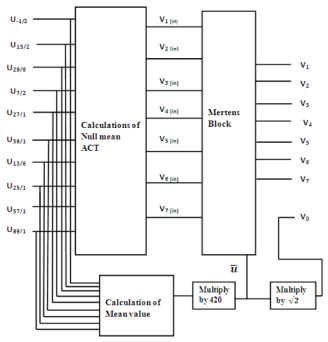
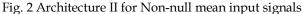


Fig. 1 Architecture I for Null mean input signals

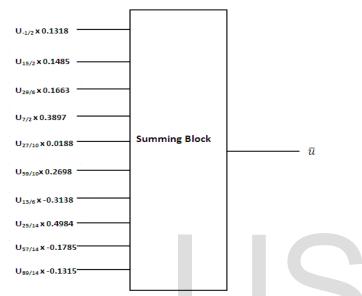




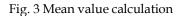
IJSER © 2016 http://www.ijser.org

2.3 Computing Arithmetic Mean

The calculation of mean value is required if the incoming signals are of non null mean type. The architecture in Fig.(3) is realized using (7). Here the input signals are scaled by the sampling instants and are given as input to the mean value calculation block. In the next step, each inputs are multiplied by the corresponding interpolation weight. The final step of mean value computation is to add all these values which will

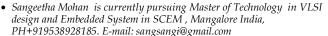


be the mean value of the incoming non null mean sequence.



2.4 Mertens Correction Factor

For non null mean input sequence, it is required to subtract a modification term in order to get the DCT coefficients. This is called the Mertens correction term, M(n). This term is the sum of the Mobius function.



 Praveen Konda is currently working as Associate Professor in Electronics and Communication Engineering Department, SCEM, Mangalore, India, PH +919591306311. E-mail: pkonda57@gmail.com

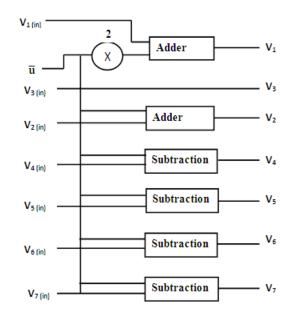


Fig. 4 Mertens correction block

3 RESULTS

The two architectures corresponds to computing DCT for null mean and non null mean inputs are implemented in Verilog HDL using Xilinx 10.1 design suit. The simulation result for the same is shown in Fig. 5 and Fig.6 respectively. The simulation, synthesis and physical design of the two architectures are performed using Cadence Encounter. The synthesis process generates RTL schematic, time, area and power reports. The Cadence result is shown in Fig.7 and Fig.8. The physical design obtained is shown in Fig.9 and Fig.10

Current Simulation Time: 1000 ns		500 ns 600 ns 700 ns 800 ns 900 ns 1000 ns 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
🖬 😽 Out1 (63:0)	6	64h000000034AC000
😐 😽 Out2[63:0]	6	64h00000000009F000
🖪 😽 Out3[63:0]	6	64hFFFFFFFEE6000
🖬 😽 Out4[63:0]	6	64'h000000001347000
😐 😽 Out5[63:0]	6	64'h000000002124000
🖬 😽 Out6[63:0]	6	64'h0000000013F6000
🖽 😽 Out7[63:0]	6	64'h000000001AF4000
😐 😽 In1[63:0]	6	64'h000000000000000000000000000000000000
🖬 😽 in2[63:0]	6	64%000000000000000000000000000000000000
🖬 😽 In3(63:0)	6	64%000000000000000000000000000000000000
🖿 😽 In4[63:0]	6	64'h0000000000000000
🖬 😽 In5[63:0]	6	64%000000000000000000000000000000000000
🖬 😽 In6[63:0]	6	64%000000000000000000000000000000000000
🖿 😽 In7[63:0]	6	64'h00000000000000000
🖬 😽 in8[63:0]	6	64%000000000000000000000000000000000000
🖽 😽 in9[63:0]	6	64'h000000000000000000
🖽 😽 in10(63:0)	6	64'h00000000000000000

Fig. 5 Simulation result of Architecture I in Xilinx

Current Simulation Time: 1000 ns		500 ns 550 ns 600 ns 650 ns 700 ns 750 ns 800 ns 850 ns 900 ns 950 ns 1000 ns
🖬 🔂 w0(63:0)	6	64%000000000550A
🖬 😽 w1(63:0)	6	64%000000000EEEC
🖬 😽 w2[63:0]	6	64%0000000000CEB0
🖬 😽 w3[63:0]	6	6411800000000049D4
🖬 🚮 w4(63:0)	6	64%000000000000000000000000000000000000
84w5(63:0)	6	64500000000001C1DC
🖬 🔂 w6(63.0)	6	64h00000000000000000000
🖬 😽 w7(63:0)	6	64%0000000043A1C
************************************	6	541000000000000000000000000000000000000
 84 v1[63:0] 	6	64%000000000000000000000000000000000000
🖬 🔂 v2[63.0]	6	< 64%000000000000000000000000000000000000
🖬 🔂 v3(63:0)	6	64%000000000000000000000000000000000000
🖬 😽 v4[63.0]	6	641/0000000000000000
64 v5[63:0]	6	64%000000000000000000000000000000000000
 84 v6[63.0] 	6	64%000000000000000000
🖬 🔂 (v7(63:0)	6	64%00000000000
 84 v8[63:0] 	6	64%0000000000000
🖬 🔂 v9[63.0]	6	64h000000000000000000000000000000000000

Fig. 6 Simulation result of Architecture II in Xilinx

	Waveform 1 - SimV	(don 1.1)
Est yes Equa 1	Forget Singleden Hindown Hind	câdence
S PA XD	D×33日本事 ♥++ ***私公社員日日	RL
dt hanes Sapar •	2 8 W Seconters rate - 2 0, 0,	
1moA • • [2]	0 ··· · · · · · · · · · · · · · · · · ·	0-00 999 0 0020-1 Ter \$50:200 0 8.11
LL.		999966
41 %s		19 19 19 19 19 19 19 19 19 19 19 19 19 1
11 Indian-0 17 Centr-Indian-2015	Danima + E	
Numi +	Carace 8 Two Dec Dec Bes Bes Dec Dec	 8m 8m 12m 12m 12m 12m 14m 15m 16m 17m 15m 19m
E A HING	F COMPTEN TOWARD ON ALL DATE AND DATE	An an Ma the first free fee fee for the first free free
0 S 1000	% COMMENT MADE TOMOTON	
A 10820	& COMMERN MACCOME COMMERCIA	
A 100	S CONTRA MOTINE CONCERNMENT	
1000	A COMPLEX INCOME COMPLEX	
A 1100	\$ CONCEPT INCOME CONTENT	
10000	& COMMENT MARCONAL COMPOSE	
1000	S COMPLEX PROCESSING COMPLETE	
1000	* COMMENT AND COMPLETE	
	S DeetTin Rection Chercite	
A Dergisig	A DIRECT MICHING CHARACTER	
1 04000	& COMMENT AND COMPANY	
0.000	A DIRECT MICHAEL DAMA IN	
	s comers meccan prants	
0.480		
	A COMPLEX MOLECUM COMPLEX	
p 💊 oxpoq 8 💊 oxpoq 8 💊 oxpoq	A CONTA MICCON (SPECIA) A CONTA MICCON (SPECIA)	

Fig. 7 Simulation result of Architecture I in Cadence

				We	reform 1	SimVisio									
lie Est Yes Ergion	e Forgat Singlation	Endius Hilo													câdea
by on X	20X 33	f a. z. 🕸	4 mail	*2	総応す	a a fi	10								
Search Names Signar •					1.		3.3								
	and the second se	and proceeding to the state			6.24		-			-					-
TB0A + - 21	a	12				1	+130	10.	24	2	BC 2000	0 Time			8 .11
u														9.0	1446
6 Indered		terment t													
R Cror-Budge *+2	NA.														Tritle - Zine
Name +	Cinit *	E THE ENS	0m 411			in be		1011			385 1481	15m			fähi 15m
E-1 V810	7,403404	10110010.0001000						_							
E-SHIDE	12.0030108	801100.00(1002)													
121824	12.0030308	Second in concises)													
18-00-0800	5.40404	MOMOR CONTINUES													
8 - weter	12-100100	MONDA DIVIDAD													
100-0-8	1.00000	INCOMENT, DISCOMO													
10 -4921	12 10 10 10	Becomme concrete													
8.0.000	To AC341140	101106106_00001000													
8-0 v(822)	10.0030300	\$4134E14_0200.0450													
E-9-489.0	16.0030000	MODELINE CONCLUSIO													
E CANEDR	1.00000	seconda conciliza													
BALLED.	1. 40404	Decidencies Concisiones													
8 400	2 10 10 10	NONDECODE													
10 10 100	to accession	BRODREDR, CORCHANNA													
18 9 148311	2.4034114	100100104_00002a34													
E-Controll	1.0000	MONTH ONCOME.													
B-Contest	1. 101010	AND NET IN CONCINED													
E STOR	's scatter	MONDE CONTACT													
	# CILL? CL	No					_		_	_			_	_	à

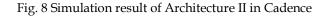
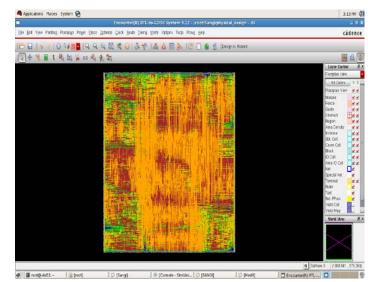


Fig. 9 Physical design of Architecture I using Cadence Encoun-



ter

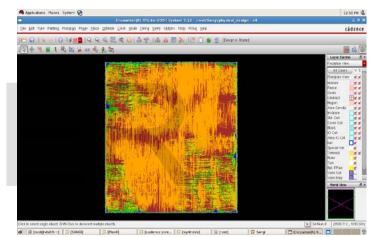


Fig. 10 Physical design of Architecture II using Cadence Encounter

4 CONCLUSION

The various algorithms for the evaluation of Discrete Cosine Transform are analyzed by considering their number of addition operations, number of multiplications needed, computational difficulties, complexity of area, and probability of occurrence of error and power consumption. The Arithmetic Cosine Transform is found to be a fast one for the computation of DCT. It has got reduced architectural complexity with only adders and constant integer multipliers which make the structure to be free from the truncation errors associated with the floating point operations. The two architectures are designed for null mean and non-null mean input signals which are only non-uniformly sampled.

REFERENCES

[1] Nilanka Rajapaksha, Arjuna Madanayake, Renato J.Cintra, Jithra

IJSER © 2016 http://www.ijser.org International Journal of Scientific & Engineering Research, Volume 7, Issue 5, May-2016 ISSN 2229-5518

Adikari, "VLSI computational architectures for the Arithmetic cosine transform," *IEEE Trans. Comput.*, vol. 64, no.9, pp. 2708-2715, Sep.2015.

- [2] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," IEEE Trans. Comput., vol. 23, no. 1, pp. 90–93, Jan. 1974.
- [3] F. A. Kamangar and K. R. Rao, "Fast algorithms for the 2-D discrete cosine transform," *IEEE Trans. Comput.*, vol. 31, no. 9, pp. 899–906, Sep. 1982.
- [4] C. Chakrabarti and J. J_aJ_a, "Systolic architectures for the computation of the discrete Hartley and the discrete cosine transform based on prime factor decomposition," *IEEE Trans. Comput.*, vol. 39, no. 11, pp. 1359–1368, Nov. 1990.
- [5] R. J. Cintra and V. S. Dimitrov, "The arithmetic cosine transform: Exact and approximate algorithms," *IEEE Trans. Signal Process.*, vol. 58, no. 6, pp. 3076–3085, Jun. 2010.

IJSER