An Efficient Distributed Arithmetic based VLSI Architecture for DCT

Vijay Kumar Sharma[1], K. K. Mahapatra[2] and Umesh C. Pati[3]
Dept. of Electronics & Communication Engineering
National Institute of Technology, Rourkela, India-769008
[1]vijay4247@gmail.com, [2]kmaha2@gmail.com, [3]ucpati@nitrkl.ac.in

***Abstract*— Discrete cosine transform (DCT) is widely used in image and video compression standards. This paper presents distributed arithmetic (DA) based VLSI architecture of DCT for low hardware circuit cost as well as low power consumption. Low hardware cost is achieved by exploiting redundant computational units in recent literature. A technique to reduce error introduced by sign extension is also presented. The proposed 1-D DCT architecture is implemented in both the Xilinx FPGA and Synopsys DC using TSMC CLN65GPLUS 65nm technology library. For power and hardware cost comparisons, recent DA based DCT architecture is also implemented. The comparison results indicate the considerable power as well as hardware savings in presented architecture. 2-D DCT is implemented using row column decomposition by the proposed 1-D DCT architecture.**

***Keywords*—2-D Discrete cosine transform (DCT), Distributed Arithmetic (DA), FPGA, Image compression, JPEG.**

I. INTRODUCTION

Multimedia communications such as image and video require high volume of data transmission. Data compression reduces the communication cost in transmitting data over long-haul links because of bandwidth reduction [1]. 8x8 2-D Discrete cosine transform (DCT) is used in image and video compression standards such as JPEG, MPEG-4 and H.263 [2-4]. Dedicated hardware is required to achieve fast processing in real-time applications [5].

DCT is a computation intensive operation. It requires a large number of adders and multipliers for direct implementation. Multipliers consume more power and hence distributed arithmetic (DA) is used to implement multiplication without multiplier [6]. Several advantages of DCT implementation using DA including area saving and high speed operations is mentioned in the literature [7]. Conventional DA implementation achieve fast speed by pre-computing possible values and storing them in ROM. But ROM based DA has the disadvantage of redundancy which is introduced to accommodate all possible combinations of bit patterns of input signals. ROM free low cost DCT implementation using NEw DA (NEDA) is presented in [8]. In [9] compressed adder/subtracter matrix is adopted resulting in adder cost savings than NEDA.

In this paper we have presented efficient DA based VLSI architecture for DCT by exploiting redundancy in previous implementation. The presented architecture achieve considerable hardware as well as power savings which is shown by comparison of Xilinx FPGA and Synopsys DC implementation results. Also a technique to reduce error introduced due to sign extension is presented.

Remainder of this paper is as follows. Section II explains the ROM free DA based 8x1 1-D DCT implementation with a error reduction adder. Area and power efficient VLSI architecture for the 1-D DCT computation is presented in section III. Implementation result and comparisons in both the Xilinx FPGA and Synopsys DC using TSMC CLN65GPLUS 65nm technology is given in section IV. Conclusions are drawn in section V.

II. ROM FREE DA BASED 8X1 1-D DCT AND ERROR REDUCTION TECHNIQUE

For a 2-D data X(i, j), $0 \le i \le 7$ and $0 \le j \le 7$, 8x8 2-D DCT is given by

$$F(u,v)=\frac{2}{8}C(u)C(v)\sum_{i=0}^{7}\sum_{j=0}^{7}X(i,j) \times\cos\left(\frac{(2i+1)u\pi}{16}\right)\cos\left(\frac{(2j+1)v\pi}{16}\right) \quad (1)$$

where $0 \le u \le 7$ and $0 \le v \le 7$ and c(u),c(v) = $1/\sqrt{2}$ for u,v=0, c(u),c(v) =1 otherwise. Implementation computation is reduced by decomposing (1) in two 8x1 1-D DCT given by,

$$F(u)=\frac{1}{2}C(u)\sum_{i=0}^{7}X(i)\cos\left(\frac{(2i+1)u\pi}{16}\right) \quad (2)$$

For 2-D DCT computation of a 8x8 2-D data, first row-wise 8x1 1-D DCT is taken for all rows followed by column-wise 8x1 1-D DCT to all columns. Intermediate results of 1-D DCT are stored in transposition memory [10].

In [9], (2) can be simplified as,

$$F(0)=[X(0)+X(1)+X(2)+X(3)+X(4)+X(5)+X(6)+X(7)]P \quad (3a)$$

$$F(1)=[X(0)-X(7)]A+[X(1)-X(6)]B+[X(2)-X(5)]C+[X(3)-X(4)]D \quad (3b)$$

$$F(2)=[X(0)-X(3)-X(4)+X(7)]M+[X(1)-X(2)-X(5)+X(6)]N \quad (3c)$$

$$F(3)=[X(0)-X(7)]B+[X(1)-X(6)](-D)+[X(2)-X(5)](-A)+[X(3)-X(4)](-C) \quad (3d)$$

$$F(4)=[X(0)-X(1)-X(2)+X(3)+X(4)-X(5)-X(6)+X(7)]P \quad (3e)$$

$$F(5)=[X(0)-X(7)]C+[X(1)-X(6)](-A)+[X(2)-X(5)]D+[X(3)-X(4)]B \quad (3f)$$

$$F(6)=[X(0)-X(3)-X(4)+X(7)]N+[X(1)-X(2)-X(5)+X(6)](-M) \quad (3g)$$

$$F(7)=[X(0)-X(7)]D+[X(1)-X(6)](-C)+[X(2)-X(5)]B+[X(3)-X(4)](-A) \quad (3h)$$

where,

$$M = \frac{1}{2}\cos\frac{\pi}{8},\ N = \frac{1}{2}\cos\frac{3\pi}{8},\ P = \frac{1}{2}\cos\frac{\pi}{4}$$

$$A = \frac{1}{2}\cos\frac{\pi}{16},\ B = \frac{1}{2}\cos\frac{3\pi}{16},\ C = \frac{1}{2}\cos\frac{5\pi}{16},\ D = \frac{1}{2}\cos\frac{7\pi}{16}$$

Distributed arithmetic is used to compute the 8 equations above where cosine terms are expressed in DA form. Implementation is realized by using shift and add operations. Shifting operation is performed by wiring. Each value of F is computed in parallel and hence faster speed is achieved [11]. Shifted data are represented by less number of bits and hence adder bit-width is reduced resulting in less hardware cost. For DCT computation image data is represented in signed 2's complement form range -128 to 127. Bit width of shifted data is determined by number of times shift operation is done. So different bit-width intermediate data are present which are to be added. For 2-input adder both input data width has to be equal and hence sign extension is done in smaller bit-width data. Shifting and addition with sign extension creates error. For example if initial value is -2, in 8-bit 2's complement representation this can be written 11111110. If we take 4 shifted sample of this value a=111111 (shifting 2 times), b=1111 (shifting 4 times), c=111 (shifting 5 times), d=11 (shifting 6 times) all are -1. But all these data should be zero. If we add these values in cascade, result we will have -4 (which should be zero). To overcome this problem we have realized adder as shown in fig.1. If one of the input is -1

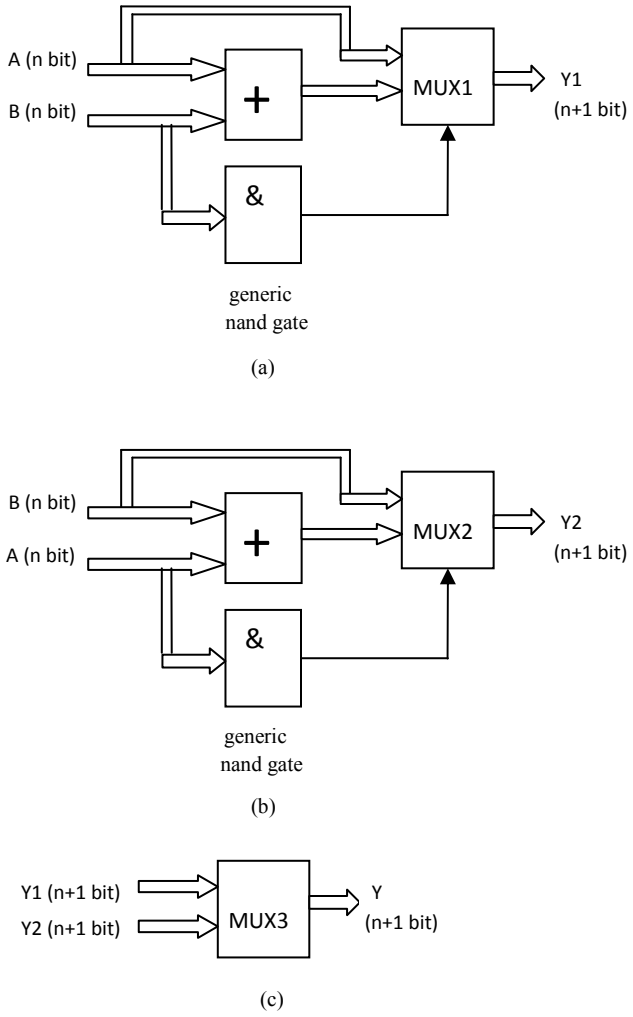

Fig.1. Circuits to reduce sign extension error propagation when number is negative (a) MUX1 selects A if B is -1 else sum of A and B (b) MUX2 select B if A is -1 else sum of A and B, (c) Final sum is from MUX1 or MUX2 output.

then output is other one. To verify this VHDL implementation of 1-D DCT and simulation is done using Xilinx ISE simulator for 8x1 data matrix of X given by,

X = [60, 40, 25, 55, 40, 42, 82, 84]

Matlab simulation result for 1-D DCT gives,

Y = [151.3209 -32.4895 33.1588 -1.7108 17.6777 18.5074 -16.0309 -5.0975]

Fig.2 shows the result of Xilinx ISE simulator using simple adder and proposed adder scheme. It is evident with matlab comparison that error due to sign extension is less in proposed adder scheme.

(a) (b)

Fig.2. VHDL simulation result using Xilinx ISE Simulator of data X for the implementation of 1-D DCT architecture in [9] using (a) simple addition operator and (b) proposed adder.

III. Area And Power Efficient VLSI Architecture of 8x1 1-D DCT

By analyzing (3a) to (3h) it can be seen that there are only seven cosine terms that are to be represented in DA form. So instead of computing F(0) to F(7) in parallel as in [9], they can be computed in pipeline fashion. A general DA based module can be implemented which takes the inputs and gives the multiply and accumulation result.

Let,

a1=X(0)+X(1)+X(2)+X(3)+X(4)+X(5)+X(6)+X(7),

a2=X(0)-X(1)-X(2)+X(3)+X(4)-X(5)-X(6)+X(7),

b1=X(0)-X(7), b2=X(1)-X(6), b3=X(2)-X(5), b4=X(3)-X(4),

c1=X(0)-X(3)-X(4)+X(7) and c2=X(1)-X(2)-X(5)+X(6).

Then from (3a) to (3h), F(0)=a1xP, F(4)=a2xP,

F(1)=b1xA+b2xB+b3xC+b4xD,

F(3)= b1xB-b2xD-b3xA-b4xC,

F(5)= b1xC-b2xA+b3xD+b4xB,

F(7)= b1xD-b2xC+b3xB-b4xA,

F(2)=c1xM+c2xN, and F(6)=c1xN-c2xM

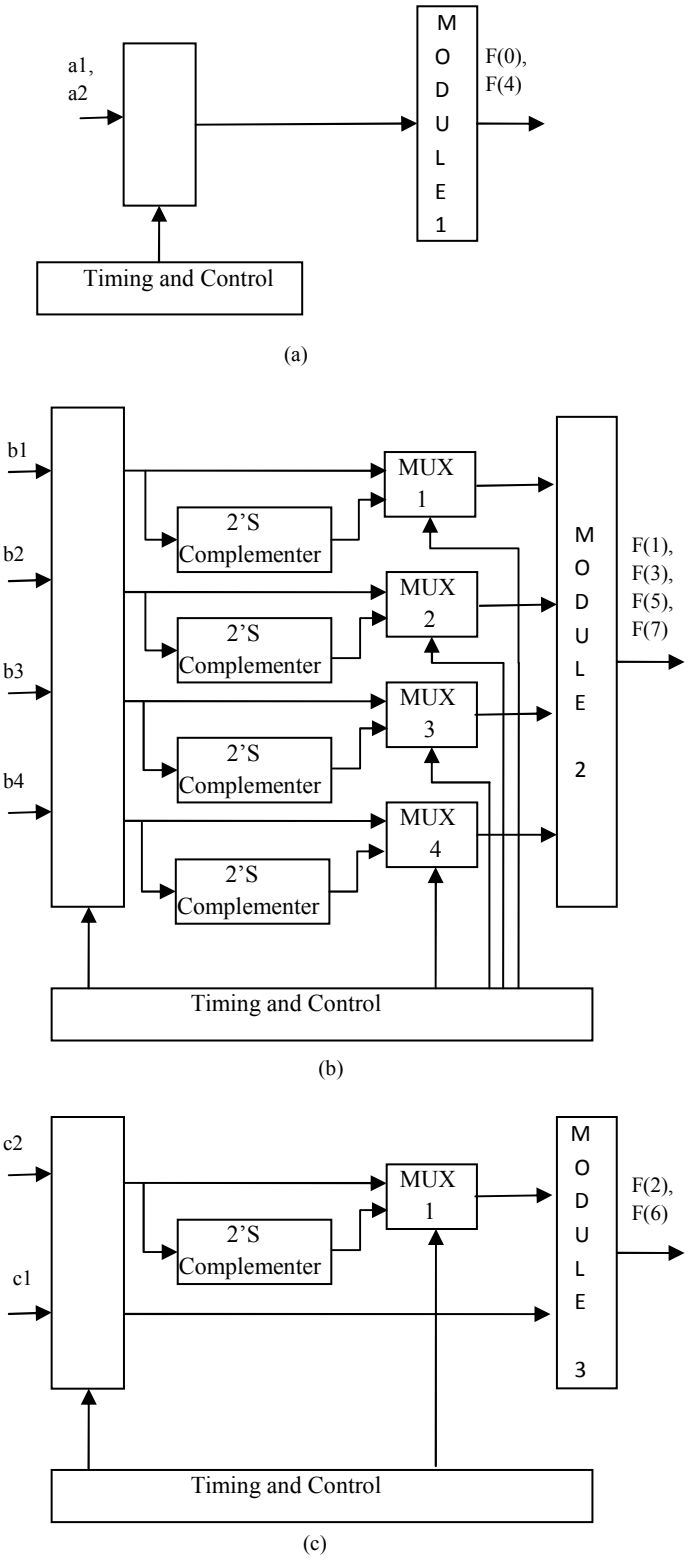

Fig.3. VLSI architecture for computation of 8 point DCT in pipeline manner for (a) computation of F(0) and F(4) (b) computation of F(1), F(3), F(5) and F(7) and (c) computation of F(2) and F(6).

For the 8 coefficients computations three module namely MODULE1, MODULE2 and MODULE3 is constructed as shown in fig.3. For MODULE1 $(1/2)\cos(\pi/4)$ is expressed in DA form with one input. For MODULE2, $1/2\cos(\pi/16)$, $(1/2)\cos(3\pi/16)$, $(1/2)\cos(5\pi/16)$, and $(1/2)\cos(7\pi/16)$ are expressed in DA form with four inputs and for MODULE3 $(1/2)\cos(\pi/8)$ and $(1/2)\cos(3\pi/8)$ is expressed in DA form with two inputs. Table I, II and III show the inputs given to these module on clock cycle basis provided by timing and control unit. Outputs obtained are stored in 8 registers.

TABLE I
PIPELINE COMPUTATION OF DCT COEFFICIENTS F(0) AND F(4)

MODULE 1		
	Clock cycle 1	Clock cycle 2
Input	a1	a2
Output	F(0)	F(4)

TABLE II
PIPELINE COMPUTATION OF DCT COEFFICIENTS F(1), F(3), F(5) AND F(7)

MODULE 2				
	Clock cycle 1	Clock cycle 2	Clock cycle 3	Clock cycle 4
Input 1	b1	-b3	-b2	-b4
Input 2	b2	b1	b4	b3
Input 3	b3	-b4	b1	-b2
Input 3	b4	-b2	b3	b1
Output	F(1)	F(3)	F(5)	F(7)

TABLE III
PIPELINE COMPUTATION OF DCT COEFFICIENTS F(2) AND F(6)

MODULE 3		
	Clock cycle 1	Clock cycle 2
Input 1	c1	-c2
Input 2	c2	c1
Output	F(2)	F(6)

IV. IMPLEMENTATION RESULTS AND COMPARISIONS

VHDL code is written for the implementation of 1-D DCT architecture in (9) and proposed architecture. Adders used in both the implementations are proposed scheme (as explained in section II). Registers are added at the output of 1-D DCT architecture in (9). Code is synthesized in Xilinx xc2vp30 FPGA device as well as Synopsys DC using TSMC CLN65GPLUS 65nm technology library. Table IV shows the device utilization summary for the FPGA implementation and Table V shows the area and power consumption for Synopsys DC synthesis. From the FPGA and ASIC implementation comparison results, it is evident that the proposed architecture is efficient in terms of area (FPGA resources in case of FPGA implementation) and power.

TABLE IV
DEVICE UTILIZATION FOR THE FPGA IMPLEMENTATION

FPGA-chip Xilinx XC2VP30		
	1-D DCT architecture in (9)	Proposed 1-D DCT architecture
# of 4 input LUTs	1268	696
# of slices	694	370
# of slice Flip Flops	0	97
# of IOB Flip Flops	88	0
Min. Period (ns)	32.6	16.29 (Freq. 61.38 Mz)
Power (W)	13.1	2.06

TABLE V
AREA AND POWER COMPARISONS FOR SYNOPSYS DC IMPLEMENTATION

TSMC CLN65GPLUS 65nm technology			
	1-D DCT architecture in [9]	Proposed 1-D DCT architecture	improvement
Total cell area	8259.84	5683.68	31.2 %
Total Dynamic Power (global operating voltage 1.1v)	3.62 mW	2.27 mW	37.3 %
Min. Slack at 500MHz	0.004	0.116	

TABLE VI
DEVICE UTILIZATION SUMMARY FOR 2-D DCT IMPLEMENTATION USING ROW-COLUMN DECOMPOSITION TECHNIQUE OF PROPOSED 1-D DCT ARCHITECTURE

FPGA-chip Xilinx XC2VP30	
# of 4 input LUTs	2522
# of slices	1701
# of slice Flip Flops	1025
Max. Freq.(MHz)	45.173
Power (W)	0.751

8x8 2-D DCT is implemented using row column decomposition technique. Table VI shows FPGA implementation device utilization summary and Table VII shows the Synopsys DC synthesis results.

TABLE VII
2-D DCT ARCHITECTURE IMPLEMENTATION AREA AND POWER

TSMC CLN65GPLUS 65nm technology	
Total cell area	23505.84
Total Dynamic Power (global operating voltage 1.1v)	5.78 mW
Min. Slack at 500 MHz	0.036

V. CONCLUSIONS

An area and power efficient architecture for the computation of 1-D DCT using ROM free DA is proposed and implemented in both Xilinx FPGA and Synopsys DC using TSMC CLN65GPLUS 65nm technology ASIC library. Comparison with recently reported DA based 1-D DCT, a significant area reduction (more than 30 percent) as well as low power consumption (more than 35 percent) is achieved in proposed architecture. With proposed 1-D DCT architecture, 2-D DCT is implemented using row column decomposition technique and area and power results are tabulated. An adder for the error reduction introduced due to shift and add with less number of bit representation is also proposed and comparative HDL simulation is done for the accuracy. Proposed adder shows less error as compared to conventional adder.

ACKNOWLEDGMENT

The authors acknowledge to DIT (Ministry of Information & Communication Technology) for the financial support for carrying out this research work.

REFERENCES

[1] T. Acharya and P. Tsai, "JPEG2000 Standard for Image Compression: Concepts, Algorithms and VLSI Architectures" J. Wiley & sons. NJ, 2005.

[2] Gregory K. Wallace, "The JPEG Still Picture Compression Standard," *IEEE Transactions on Consumer Electronics*, vol.38(I), Feb. 1992.

[3] R. C. Gonzalez, R. E. Woods, "Digital Image Processing," 2nd.Ed.,Prentice Hall, 2002.

[4] F.H.P. Fitzek, M. Reisslein, "MPEG-4 and H.263 Video Traces for Network Performance Evaluation ," *IEEE Network*, vol.15, no.6, pp.40-54, Nov/Dec 2001.

[5] Luciano Volcan Agostini, Ivan Saraiva Silva and Sergio Bampi, "Multiplierless and fully pipelined JPEG compression soft IP targeting FPGAs," *Microprocessors and Microsystems*, vol. 31(8), 3 pp.487-497, Dec. 2007.

[6] S. A. White, "Applications of distributed arithmetic to digital signal processing: a tutorial review," *IEEE ASSP Magazine*, vol.6, no.3, pp.4-19, Jul.1989.

[7] M.-T. Sun, T.-C. Chen, A.M. Gottlieb, ''VLSI Implementation of a 16x16 Discrete Cosine Transform," *IEEE Transactions on Circuits and Systems,* vol.36, no. 4, pp. 610 – 617, Apr. 1989.

[8] A. Shams, A. Chidanandan, W. Pan, and M. Bayoumi, "NEDA: A low power high throughput DCT architecture," *IEEE Transactions on Signal Processing,* vol.54(3), Mar. 2006.

[9] Peng Chungan, Cao Xixin, Yu Dunshan, Zhang Xing, "A 250MHz optimized distributed architecture of 2D 8x8 DCT," 7th International Conference on ASIC, pp. 189 – 192, Oct. 2007.

[10] M. Kovac, N. Ranganathan, "JAGUAR: A Fully Pipelined VLSI Architecture for JPEG Image Compression Standard," *Proceedings of the IEEE*, vol.83, no.2, pp. 247-258,Feb.1995.

[11] Yuan-Ho Chen, Tsin-Yuan Chang, Chung-Yi Li, "High Throughput DA-Based DCT With High Accuracy Error-Compensated Adder Tree," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. PP, issue 99, pp. 1-5, Jan 2010.

[12] A. Kassem, M. Hamad, E. Haidamous, "Image Compression on FPGA using DCT," *International Conference on Advances in Computational Tools for Engineering Applications, 2009*, *ACTEA '09*, pp.320-323, 15-17 July 2009.

[13] Leila Makkaoui, Vincent Lecuire and Jean-Marie Moureaux, "Fast Zonal DCT-based image compression for Wireless Camera Sensor Networks," *2nd International Conference on Image Processing Theory Tools and Applications (IPTA)*, pp. 126-129, 2010.

[14] Byoung-Il Kim and Sotirios G. Ziavras, "Low-Power Multiplierless DCT for Image/Video Coders," *IEEE 13th International Symposium on Consumer Electronics, 2009. ISCE '09,* pp. 133-136.

[15] C. H. Chen, B. D. Liu and J. F. Yang, "Direct Recursive Structures for Computing Radix-r Two-Dimensional DCT/IDCT/DST/IDST", *IEEE Transactions On Circuits And Systems*,I Regular Papers , vol. 51, no. 10, October 2004.

[16] S. An C. Wang, "Recursive algorithm, architectures and FPGA implementation of the two-dimensional discrete cosine transform," *IET Image Process*., vol. 2(6), pp. 286–294, 2008.

[17] Xilinx Inc., Web: www.xilinx.com.

[18] Synopsys Inc.,Web: www.synopsys.com.