Asian Journal of Information Technology 15 (18): 3455-3460, 2016
ISSN: 1682-3915

FPGA Implementation of Mars Based 2-D Fast Cosine Transform

K. Kalyani and S. Rajaram
Department of ECE, Thiagarajar College of Eng., Madurai, India

Abstract: Digital images and videos have become so important and are being used in a wide range of applications including digital cameras, Medical imaging, DVD, digital TV, HDTV and video telephony. These applications are feasible because of the advancement in computing and communication technologies as well as efficient image compression algorithms. Fast Cosine Transform (FCT) is widely used technique in image and video compression algorithms. FCT requires large amount of mathematical computations including memory access, multiplications and accumulations which leads to hardware complexity. The hardware complexity due to memory access can be reduced by the memory access reduction scheme called as MARS. As this Memory Access Reduction Scheme (MARS) is very much suitable for FPGA designs in this work, the 2D-FCT is proposed using MARS based architecture. Then existing architecture for 2D-FCT and the proposed MARS based 2D-FCT are simulated and synthesized in Virtex7-xc7v2000t-2flg1925 FPGA device using Xilinx ISE 14.1. The results of the Conventional 2D-FCT structure is compared with the proposed MARS based 2D-FCT to show the improvement in terms of hardware efficiency of the proposed work.

Key words: FCT, weighting factors, MARS, FPGA, efficiency

INTRODUCTION

The advancement in technologies has increased the use of digital images and videos. Digital images and videos comprises of large amount of data. Reduction in the size of the image data for both storing and transmission of images are becoming increasingly important as they find more applications. Discrete Cosine Transform (DCT) has played a prominent role in image and video applications (Ahmed *et al*., 1974). It has been adopted as primary computation unit in still and moving pictures coding standards such as JPEG, MPEG, ITU's and H.264. But DCT requires large amount of mathematical computations including multiplications and accumulations which leads to increased clock cycles and power. To overcome this Fast Cosine Transform (FCT) is introduced. The Discrete Cosine Transform (DCT) of an N-point real signal is derived by taking the Discrete Fourier Transform (DFT) of a 2N-point even extension of the signal (Makhoul, 1980). However, this approach exhibits higher computational complexity than direct computation due to the matrix transposition. Further a new fast algorithm for 8×8 2D-DCT based on partial sum and its corresponding hardware architecture for VLSI realization (Tian *et al*., 2005) a Loeffler DCT architecture can save 14% addition operations for the same precision requirement and the path delay can be significantly reduced as well. A FPGA based scalable architecture for DCT computation using dynamic partial reconfiguration proposed by Wu *et al*. (2009). The 2D-FCT calculation using quantization and zigzag arrangement is introduced that is used in JPEG image compression (Pradeepthi and Ramesh, 2011). MARS based 2D-FCT is proposed (Liu, 2010) which uses reduced memory access and has less hardware complexity is implemented in DSP Processor (Liu and Bao, 2015). Implementation on DSP Processor is unreliable and not efficient in terms of energy when compared to FPGAs. So, a modified approach to implement 2D-FCT using MARS in FPGA is proposed rather than implementing in DSP.

MATERIALS AND METHODS

Vector radix 2D Fast cosine transform: The 2D-Fast Cosine transform is an image compression algorithm which represents the ordinary cosine transform in terms of the butterfly structures. This FCT method consists of reduced number of multiplications and additions when compared to the DCT (Christopoulos *et al*., 1995). Here, the FCT is represented in decimation in frequency algorithm. The 2D-DCT equation is given by:

$$x(k,1) = \frac{2}{N}\varepsilon_k\varepsilon_1 \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} x[m,n]\cos\left[\frac{\pi(2m+1)l}{2N}\right] \cos\left[\frac{\pi(2n+1)l}{2N}\right] k,l = 0,1,2,3....N-1 \quad (1)$$

where, ε_k, ε_1 $1/\sqrt{2}$ for k, l = 0 otherwise ε_k, ε_1 = 1. In the 2D-Fast Cosine Transform, the first step is input remapping. The input remapping (Skodras and Constantinides, 1991) is done with the following conditions (Eq. 2):

Corresponding Author: K. Kalyani, Department of ECE, Thiagarajar College of Eng., Madurai, India

$$\tilde{x}[m, n] = \begin{cases} x[2m, 2n] & m = 0, ..., \frac{N}{2}-1; n = 0, ..., \frac{N}{2}-1 \\ x[2N-2m-1, 2n] & m = \frac{N}{2},...,N-1; n = 0, ..., \frac{N}{2}-1 \\ x[2m, 2N-2n-1] & m = 0,..., \frac{N}{2}1; n = \frac{N}{2},...,N+1 \\ x[2N, 2m-1, 2N-2n-1] & m = \frac{N}{2},...,N-1; n = \frac{N}{2}, ..., \frac{N}{2}+1 \end{cases} \quad (2)$$

Where:

$\tilde{x}[m, n]$ = The input mapped co-ordinates

N = The matrix size

The 2D-DCT Eq. 1 can be rewritten as:

$$x(k, 1) = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} \tilde{x}[m, n]\cos\left[\frac{\pi(4m+1)k}{2N}\right]\cos\left[\frac{\pi(4n+1)l}{2N}\right]$$

$$k, l = 0, 1, 2, 3....N-1 \quad (3)$$

By decomposing the output sequence X[k, l] into even-even, odd-even, even-odd and odd-odd indexed terms, four (N/2×N/2) point 2D-DCTs given in Eq. 4-7 where:

$$x_1 = \tilde{x}[m, n], x_2 = \tilde{x}\left[m, n+\frac{N}{2}\right],$$

$$x_3 = \tilde{x}\left[m+\frac{N}{2}, n\right] \text{ and } x_4 = \tilde{x}\left[m+\frac{N}{2}, n+\frac{N}{2}\right]$$

$$X[2k\times 21] = \sum_{m=0}^{\frac{1}{2}N-1}\sum_{n=0}^{\frac{1}{2}N-1}\left\{(x1+x2+x3+x4)\cos\left[\frac{\pi(4m+1)k}{2\times\frac{1}{2}N}\right]\cos\left[\frac{\pi(4n+1)l}{2\times\frac{1}{2}N}\right]\right\} \quad (4)$$

$$X[2k\times 21+1] = \sum_{m=0}^{\frac{1}{2}N-1}\sum_{n=0}^{\frac{1}{2}N-1}\left\{2(x1-x2-x3-x4)\cos\left[\frac{\pi(4n+1)}{2\times N}\right]\cos\left[\frac{\pi(4m+1)k}{2\times\frac{1}{2}N}\right]\cos\left[\frac{\pi(4n+1)l}{2\times\frac{1}{2}N}\right]\right\} \quad (5)$$

$$-X[2k-1, 2l]$$

$$X[2k.21+1] = \sum_{m=0}^{\frac{1}{2}N-1}\sum_{n=0}^{\frac{1}{2}N-1}\left\{2(x1+x2+x3+x4)\cos\left[\frac{\pi(4n+1)}{2\times N}\right]\cos\left[\frac{\pi(4m+1)k}{2\times\frac{1}{2}N}\right]\cos\left[\frac{\pi(4n+1)l}{2\times\frac{1}{2}N}\right]\right\} \quad (6)$$

$$-X[2k.2l-1]$$

$$X[2k+1, 2l+1] = \sum_{m=0}^{\frac{1}{2}N-1}\sum_{n=0}^{\frac{1}{2}N-1}\left\{4(x1-x2-x3-x4)\cos\left[\frac{\pi(4m+1)}{2\times N}\right]\cos\left[\frac{\pi(4n+1)k}{2\times N}\right]\cos\left[\frac{\pi(4m+1)l}{2\times\frac{1}{2}N}\right]\cos\left[\frac{\pi(4n+1)l}{2\times\frac{1}{2}N}\right]\right\} \quad (7)$$

$$-X[2k-1, 2l-1]-X[2k+1, 2l-1]-X[2k-1, 2l-1]$$

Using the Eq. 4-7 the butterfly structure is designed.

Weighting factors in Vector Radix 2D-FCT: The cosine functions in Eq. 4-7 are considered as weighting factors. No. of weighting factors at each stage is $N\times N/4^s$. All the required weighting factors in each stage are equally divided into 4 groups such as:

$$C^{p}_{2\times N},\ C^{p}_{2\times N},\ C^{p+\frac{N}{2\log N-s-1}}_{2\times N} \text{ and } C^{q+\frac{N}{2\log N-s-1}}_{2\times N}$$

Where:

$C^{p}_{2\times N}$ = Given as cos pπ/2×N

s = The stage number

From this the stage 1 weighting factors becomes $C^{p}_{2\times N}$, $C^{q}_{2\times N}$, $C^{p+N}_{2\times N}$ and $C^{q+N}_{2\times N}$ The remapped inputs are applied on the butterfly structure as shown in Fig. 1.

Figure 2 represents the weighting factors for each stage for 4×4 FCT. The stage 1 weighting factors are represented by (Cp/2×N, Cq/2×N, Cq+N/2×N, Cp+N/2×N) and stage 2 weighting factors are represented by C2p/2×N C2q/2×N, C. The black circles represent the memory access that occurs during the butterfly computation. The weighting factor applied butterfly structure of a 4×4 FCT is shown in Fig. 3.

In the above method the outputs of the stage 1 are to be stored in order to compute the stage 2 outputs. This increases the memory access which increases the hardware complexity.

Proposed vector radix 2D-FCT using MARS: The proposed method is a Memory Access Reduction Scheme (MARS) based 2D-FCT. In this method, the weighting factors are reduced by means of trigonometric functions. The stage 2 weighting factors are calculated from the stage 1 weighting factors that helps to have reduced memory storage and hence the additional memory access reduced.

Weighting factor reduction in MARS based 2D-FCT: The stage 2 weighting factors are calculated from the 1 stage weighting factor $C^{2p}{}_{2\times N}$ and $C^{2p}{}_{2\times N}$ where $C^{2p}{}_{2\times N} = \cos 2p\pi/2\times N$. Applying trigonometric property:

$$\cos 2x = \cos^2 x - \sin^2 x \tag{8}$$

The weighting factor becomes:

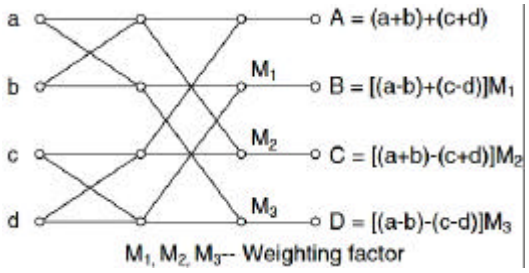

Fig. 1: Butterfly structure; a-d are the remapped inputs and M1-M3 are the weighting factors

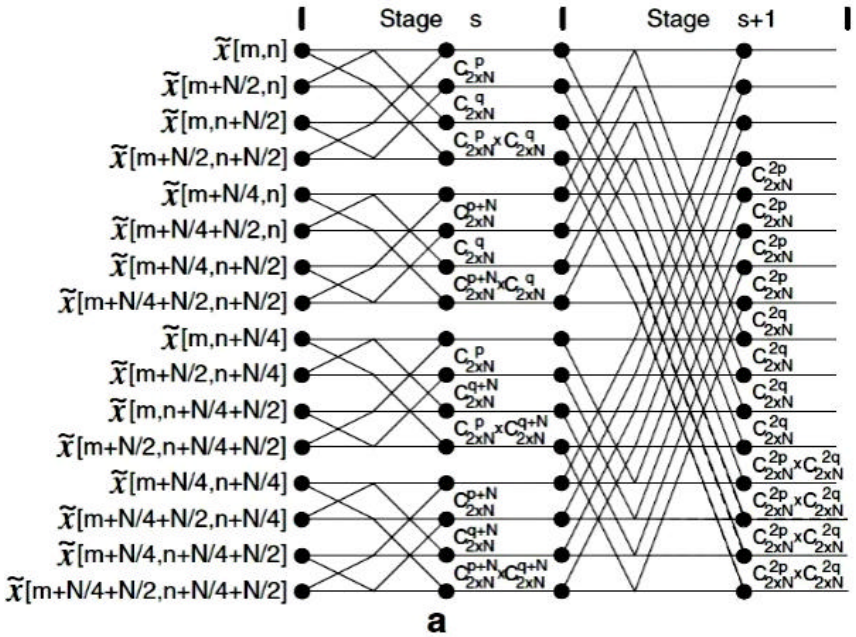

Fig. 2: Butterfly structure for 4×4 matrix

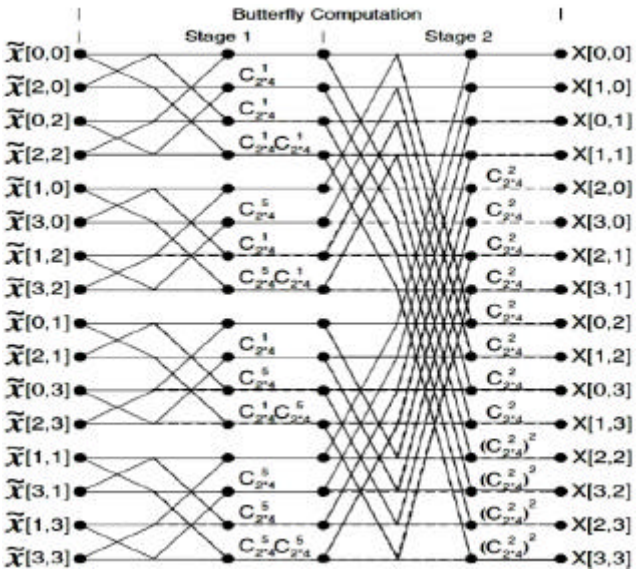

Fig. 3: Butterfly structure with weighting factors applied

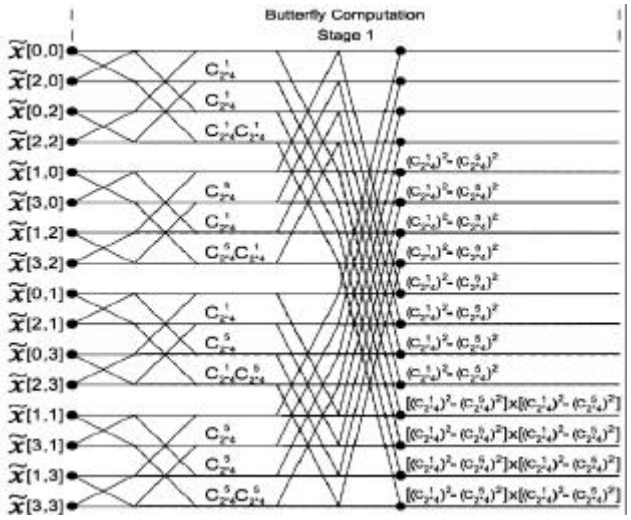

Fig. 4: Butterfly structure with weighting factor reduced

$$\cos^2\frac{p\pi}{2\times N}-\sin^2\frac{p\pi}{2\times N}$$

$$\cos^2\frac{p\pi}{2\times N}=(C^{p}_{2\times N})^2$$

The is $\sin^2 qn/2\times N$ developed by the following process:

$$\begin{aligned}\sin^2\frac{p\pi}{2\times N}&=\cos^2(\pi/2-p\pi/2\times N)\\&=\cos^2(\pi-(\pi/2-p\pi/2\times N)\\&=\cos^2(\pi/2-p\pi/2\times N)\\&=\cos^2(p+N)/2\times N\,\pi)\\&=(C^{p+N}_{2\times N})^2\end{aligned}$$

The derived weighting factor is:

$$C^{2p}_{2\times N}=(C^{p}_{2\times N})^2-(C^{p+N}_{2\times N})^2 \quad (9)$$

$$C^{2q}_{2\times N}=(C^{q}_{2\times N})^2-(C^{q+N}_{2\times N})^2 \quad (10)$$

From (Eq. 9 and 10) the stage 2 weighting factors are calculated from the stage 1. Where $C^{p}_{2\times N}$, $C^{p+N}_{2\times N}$, $C^{q}_{2\times N}$, $C^{q+N}_{2\times N}$ are the stage 1 weighting factors. $C^{2p}_{2\times N}$, $C^{2q}_{2\times N}$ are the stage 2 weighting factors. Thus, the both 1 and 2 stage butterfly structure is combined together as a single stage as shown in Fig. 4.

In this method, there is no need of storage of stage1 outputs because the weighting factors for the second

stage are computed using stage1 weighting factors. Thus, it will reduce the memory accesses compared to the conventional method.

RESULTS AND DISCUSSION

The conventional vector radix 2D-FCT and proposed MARS based 2D-FCT are simulated using Xilinx ISE 14.1 tool as given below.

Simulation of conventional vector radix 2D-FCT: The input matrix is a representation of 8 bit data and the applied 7 weighting factors are in fixed point notation.

In the simulated conventional method as shown in Fig. 5, the butterfly computations are calculated and the stage 1 results are stored and then the stage 2 results are calculated from the stage 1 results. This results in higher number of memory accesses.

Simulation of proposed vector radix 2D-FCT using mars: In the simulated proposed method as shown in Fig. 6, there is no need of storage of stage 1 outputs because stage 2 weighting factors are calculated from stage 1 weighting factors. Hence, there will be less number of memory accesses when compared to the conventional method.

Synthesis report: The conventional Vector Radix 2D-FCT and proposed MARS based 2D-FCT are synthesized in the FPGA device Virtex7-xc7v2000t-2flg1925 using Xilinx ISE 14.1 tool.

From the comparison of synthesis report as shown in Table 1, it is realized that in FCT with MARS based method there is a 41.49% reduction in utilization of slice LUTs, 44.82% of reduced Bonded IOBs and 6% less DSP481As as compared with the conventional 2D-Fast Cosine Transform Algorithm.

Table 1: Comparison of resource utilization of vector radix 2D-FCT and proposed 2D-FCT with mars

Logic utilization	Vector Radix 2D-FCT	Proposed 2D-FCT with MARS	Percentage of improvement (%)
Number of Slice LUTs	1070	626	41.49
Number of bonded IOBs	754	416	44.82
Number of DSP48E1s	33	31	6.06

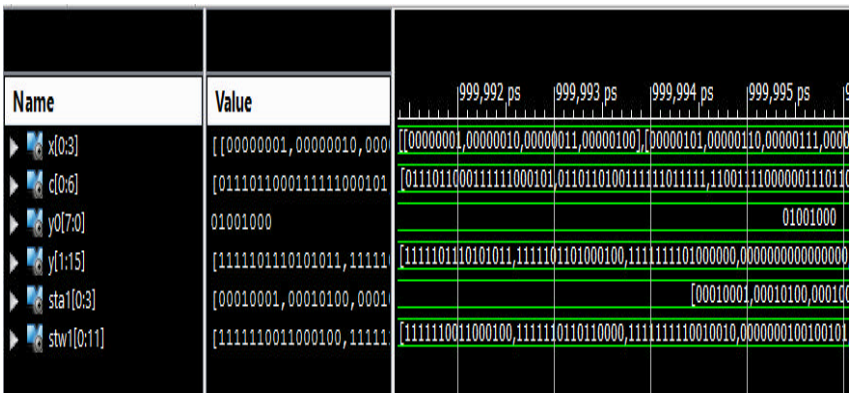

Fig. 5: Simulation of conventional Vector Radix 2D-FCT; x: the input 4×4 Matrix; c: the applied 7 weighting factors sta; stw are the stage1 outputs; y_0-y_{16} are the outputs

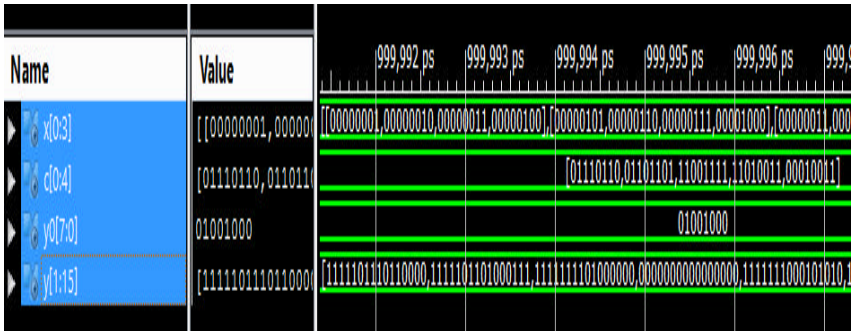

Fig. 6: Simulation of proposed Vector Radix 2D-FCT using MARS; x: the input 4x4 Matrix; c: the applied 5 weighting factors; y_0-y_{16} are the outputs

CONCLUSION

The fast cosine transform being the most commonly used transform technique in compression algorithms requires an efficient architecture in terms of both area and speed. In this research, a low memory access 2D-Fast Cosine Transform with use of MARS architecture is proposed. The VHDL coding for the Conventional FCT and the proposed FCT with MARS are written, simulated and synthesized in Virtex7-xc7v2000t-2flg1925 FPGA device using Xilinx ISE 14.1. The synthesis report reveals that the FCT with MARS based method utilizes less number of logic resources compared with the conventional 2D-Fast Cosine Transform Algorithm. Thus, the proposed MARS based 2D-FCT design is hardware efficient than the conventional 2D-FCT.

REFERENCES

Ahmed, N., T. Natarajan and K.R. Rao, 1974. Discrete cosine transfom. IEEE. Trans. Comput., 23: 90-93.

Christopoulos, C.A., J. Bormans, J. Cornelis and A.N. Skodras, 1995. The vector-radix fast cosine transform: Pruning and complexity analysis. Signal Process., 43: 197-205.

Liu, X. and H. Bao, 2015. Efficient implementation of 2-D FCT with reduced memory access for programmable DSPs. J. Signal Process. Syst., 80: 153-161.

Liu, X., 2010. Memory access reduction method for efficient implementation of fast cosine transform Pruning on DSP. Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), March 14-19, 2010, IEEE, Dallas, Texas, ISBN: 978-1-4244-4295-9, pp: 1490-1493.

Makhoul, J., 1980. A fast cosine transform in one and two dimensions. Acoust. Speech Signal Process. IEEE. Trans., 28: 27-34.

Pradeepthi, T. and A.P. Ramesh, 2011. Pipelined architecture of 2D-DCT, Quantization and Zigzag process for JPEG image compression using VHDL. Intl. J. VLSI Des. Commun. Syst., 2: 99-110.

Skodras, A.N. and A.G. Constantinides, 1991. Efficient input-reordering algorithms for fast DCT. Electron. Lett., 27: 1973-1975.

Tian, M., G.J. Li and Q.Z. Peng, 2005. A new fast algorithm for 8×8 2-D DCT and its VLSI implementation. Proceedings of 2005 IEEE International Workshop on VLSI Design and Video Technology, May 28-30, 2005, IEEE, Sichuan, China, ISBN: 0-7803-9005-9, pp: 179-182.

Wu, Z., J. Sha, Z. Wang, L. Li and M. Gao, 2009. An improved scaled DCT architecture. Consum. Electron. IEEE. Trans., 55: 685-689.