Proc. Natl. Acad. Sci., India, Sect. A Phys. Sci.
https://doi.org/10.1007/s40010-018-0541-3

RESEARCH ARTICLE

Efficient VLSI Architecture for 16-Point Discrete Cosine Transform

M. Thiruveni[1] · D. Shanthi[2]

Received: 14 November 2015 / Revised: 30 July 2018 / Accepted: 9 August 2018

Abstract The Discrete Cosine Transform (DCT) is an important transform in image and video processing systems. The perception of human visualization permits us to be numerically approximate rather than exact with slight compromise in accuracy. In this paper, we have proposed a digital implementation of 16-point approximate DCT architecture based on Modified Gate Diffusion Input (MGDI) technique. The 8-point DCT architecture can be realized in digital Very Large Scale Integration (VLSI) hardware with only 12 additions. The proposed 8-transistor MGDI full adder is used instead of existing 10-transistor MGDI full adder in the DCT architecture. It results in reduced circuit complexity, power and delay. Approximate multiplier-free MGDI DCT is simulated in Tanner SPICE for 90-nm CMOS process technology at 100 MHz. The simulation result shows that 20%, 16% and 7% of area, power and delay are reduced, respectively, when compared with approximate DCT by using 14 additions. The performance was evaluated based on peak signal-to-noise ratio (PSNR), and the proposed architecture shows enhancement in terms of hardware complexity, regularity and modularity with a little compromise in accuracy.

Keywords Discrete Cosine Transform · Modified Gate Diffusion Input · PSNR · Regularity · Modularity

1 Introduction

Signal processing is used universally to extract information from signals or to convert the information-carrying signals from one form to another. In signal processing, image and video signal processing is the most craved area in real world, and it is more convenient and easy to work in frequency domain. The transforms are used to convert time-domain signal into frequency-domain signal. Karhunen–Loève transform (KLT) is the optimum transform but the complexity grows with input and does not have a systematic procedure. Discrete Cosine Transform (DCT) is a very close substitute for the KLT, so it is used to convert a signal from spatial domain to frequency domain. DCT is an essential mathematical tool in both image and video compression due to its high-energy compaction property. As a result, the 16-point DCT was adopted in several imaging standards such as JPEG, MPEG-1, MPEG-2, H.261, H.263 and H.264/AVC. New compression schemes such as the high-efficiency video coding (HEVC) employs DCT-like integer transforms operating at various block sizes. The fast algorithms can significantly reduce the computational complexity of DCT which is frequently used in image and video compression. Since DCT is computationally intensive, all algorithms have been eventual in its efficient calculation [1].

The main aim of the approximation algorithm for DCT is to eliminate multiplications which are strong, power and time-consuming operations and also to obtain meaningful estimation of DCT. The approximation is more pertinent

✉ M. Thiruveni
thiruveniraguraman@gmail.com

D. Shanthi
dshan71@gmail.com

[1] Department of ECE, PSNA College of Engineering and Technology, Dindigul, Tamilnadu, India

[2] Department of CSE, PSNA College of Engineering and Technology, Dindigul, Tamilnadu, India

Published online: 20 September 2018

for higher-size DCT while considering the computational complexity of the DCT which grows nonlinearly.

Gate diffusion input (GDI) design technique [2] was confirmed as a new promising alternate to usual CMOS logic design for low-power digital systems. It is called gate diffusion input technique because one of the inputs is directly diffused into the gates of the transistors of *N* type and *P* type. The drawbacks of GDI technique are as follows: reduced voltage swing and connection of source with bulk. The modified GDI (MGDI) and full-swing GDI techniques [3] are used to solve these problems.

The basic GDI inverter is similar to the standard CMOS inverter, but there are few important differences.

1. The GDI cell contains three inputs: *G* (common gate input of nMOS and pMOS), *P* (input to the source/drain of pMOS) and *N* (input to the source/drain of nMOS).
2. Bulks of both nMOS and pMOS are connected to *N* or *P* (respectively), so it can be arbitrarily biased at contrast with a CMOS inverter.

MGDI [4] cell contains a low-voltage terminal (bulk of PMOS) configured to be connected to a high constant voltage (i.e., supply voltage) and a high-voltage terminal (bulk of NMOS) configured to be connected to a low constant voltage (i.e., ground) in order to get full-swing output.

The idea of this paper is twofold: first, hardware implementation of comparatively good approximation that possesses reduced circuit complexity using MGDI 8T full adder; second, reconfiguration of the architecture for 16-point DCT from 8-point DCT.

2 Literature Survey

Features of approximate DCT are low computational complexity, orthogonality, low error energy and working for higher lengths of DCT.

The transformation matrix of the 8-point DCT approximation methods is mathematically described as

$$[\text{Diagonal matrix}] \times [\text{low-complexity matrix}]. \qquad (1)$$

The diagonal matrix usually contains irrational numbers in the form $1/\sqrt{m}$, where m is a small positive integer. In principle, the irrational numbers required in the diagonal matrix would require an increased computational complexity. Since the entries of the low-complexity matrix comprise only powers of two $\{0, \pm 1/2, \pm 1, \pm 2\}$, null multiplicative complexity is achieved.

In [5], a low-complexity approximate was introduced by Bouguezel et al. and called BAS-2008 Approximation. Its mathematical structure is $C_1 = D_1 \cdot T_1$, where $D_1 = \text{diag}\ (1/\sqrt{8},\ 1/\sqrt{4},\ 1/\sqrt{5},\ 1/\sqrt{2},\ 1/\sqrt{8},\ 1/\sqrt{4},\ 1/\sqrt{5},\ 1/\sqrt{2}\}$. It requires only 18 additions and two shifts for its computation.

The parametric transform proposed in 2011 by Bouguezel–Ahmad–Swamy [6] is an 8-point orthogonal transform containing a single parameter "*a*." Its mathematical structure is $C_1 = D_1 \cdot T_1$, where $D_1 = \text{diag}\ (1/\sqrt{8},\ 1/\sqrt{2},\ 1/\sqrt{\ }\ (4 + 4a^2),\ 1/\sqrt{2},\ 1/\sqrt{8},\ 1/\sqrt{2},\ 1/\sqrt{2},\ 1/\sqrt{(4 + 4a^2)}\}$. It requires only 16 additions for its computation.

In [7], CB-2011 rounding off the elements of exact DCT matrix, a DCT approximation was obtained. The result of 8-point approximation matrix is orthogonal and contains elements only in $\{0, \pm 1\}$. It possesses very low arithmetic complexity. The transformation matrix $C_1 = D_1 \cdot T_1$, where $D_1 = \text{diag}\ (1/\sqrt{8},\ 1/\sqrt{6},\ 1/2,\ 1/\sqrt{6},\ 1/\sqrt{8},\ 1/\sqrt{6},\ 1/\sqrt{2},\ 1/\sqrt{6}\}$. It requires only 22 additions for its computation.

In [8], CB-2011 is modified by replacing elements of the CB-2011 matrix with zeros. The transformation matrix $C_1 = D_1 \cdot T_1$, where $D_1 = \text{diag}\ (1/\sqrt{8},\ 1/\sqrt{2},\ 1/2,\ 1/\sqrt{2},\ 1/\sqrt{8},\ 1/\sqrt{2},\ 1/2,\ 1/\sqrt{2}\}$. It needs only 14 additions for its computation.

In [9], a DCT approximation tailored for a particular radio-frequency (RF) application was obtained in accordance with an exhaustive computational search. The transformation matrix $C_1 = D_1 \cdot T_1$, where $D_1 = 1/2 \cdot \text{diag}\ (1/\sqrt{2},\ 1/\sqrt{3},\ 1/\sqrt{5},\ 1/\sqrt{3},\ 1/\sqrt{2},\ 1/\sqrt{3},\ 1/\sqrt{5},\ 1/\sqrt{3}\}$. It requires only 24 additions and six shifts for its computation.

In [10], a low-complexity approximate DCT defines the cost of a transformation matrix as the number of arithmetic operations required for its computation. Elements of matrix intend to be in $\{0, \pm 1, \pm 2\}$ to insure that resulting multiplicative complexity is null. The transformation matrix $C_1 = D_1 \cdot T_1$, where $D_1 = \text{diag}\ (1/\sqrt{8},\ 1/\sqrt{2},\ 1/2,\ 1/\sqrt{2},\ 1/\sqrt{8},\ 1/\sqrt{2},\ 1/2,\ 1/\sqrt{2}\}$. It requires only 14 additions for its computation.

Some of the existing methods are not efficient in terms of scalability, generalization for higher sizes and orthogonality. Error energy is reduced in the DCT due to orthogonality property. In existing methods, the design cannot be extended for larger transform sizes such as 16 points and 32 points. But large-size DCTs are required for several image processing applications such as tracking and simultaneous compression and encryption.

The reduced complexity of the diagonal matrices reduces the complexity of the approximation. The elements of the low-complexity diagonal matrix are only powers of two, so null multiplicative complexity [11] is obtained. The number of retained coefficients in the transform domain is an important parameter in the image compression routine.

3 Materials and Methods

3.1 Modified Gate Diffusion Input

The CMOS inverter is the reference for any logic family. The GDI inverter is shown in Fig. 1a which uses PMOS and NMOS transistors like CMOS inverter. In both the transistors, source and substrate are connected together. The main drawback of this inverter is that it will not produce full swing at the output.

The MGDI inverter is shown in Fig. 1b which uses PMOS and NMOS. The substrate of both nMOS and pMOS is connected to Vss or Vdd (respectively). This inverter will produce full-swing output.

3.2 *D* Flip-Flop and Latch

Flip-flop is the basic element of memory. The flip-flop shown in Fig. 2 with one input *D* is called the data flip-flop. It is also called delay flip-flop because it delays the propagation of input to output by unit time. If *D* is high, the flip-flop is “Set” and when it is low, the flip-flop will “Reset” i.e., input *D* is copied to the output *Q* during the clock input. Before clock transition occurred, it will not find any change in the state and store whatever the data might produce as an output. The output is “latched” at either logic “0” or logic “1”. The operation table and circuit diagram of *D* flip-flop are given in Table 1 and shown in Fig. 2, respectively.

The 8-bit *D* flip-flop is shown in Fig. 3 and is designed to delay or latch 8-bit input data. When the clock is low, it will not change the state and store whatever data were present on its output before the clock transition occurred. The input is latched to subsequent stage of adders in DCT with unit delay when clock is high.

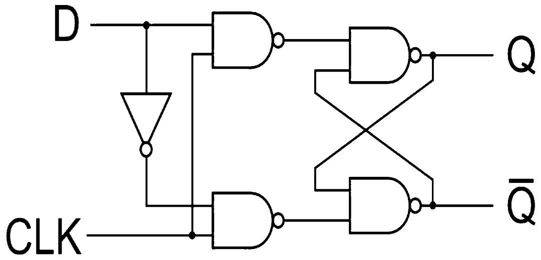


Fig. 2 Circuit diagram of D flip-flop

Table 1 Truth table of *D* flip-flop

D	$Q\ (t+1)$
0	0
1	1
0	0
1	1

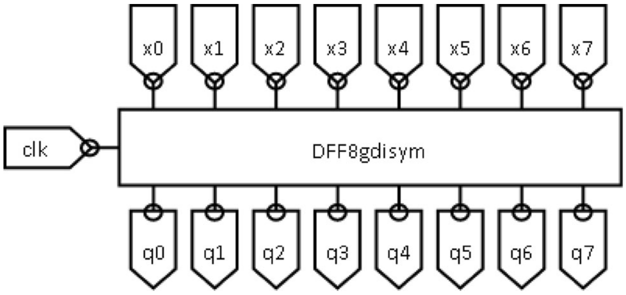


Fig. 3 8-Bit D flip-flop

The 8-bit *D* latch is designed to delay or latch 8-bit input data without waiting for clock. It will change the state according to the change in data. The input is latched to subsequent stage of adders in DCT with unit delay.

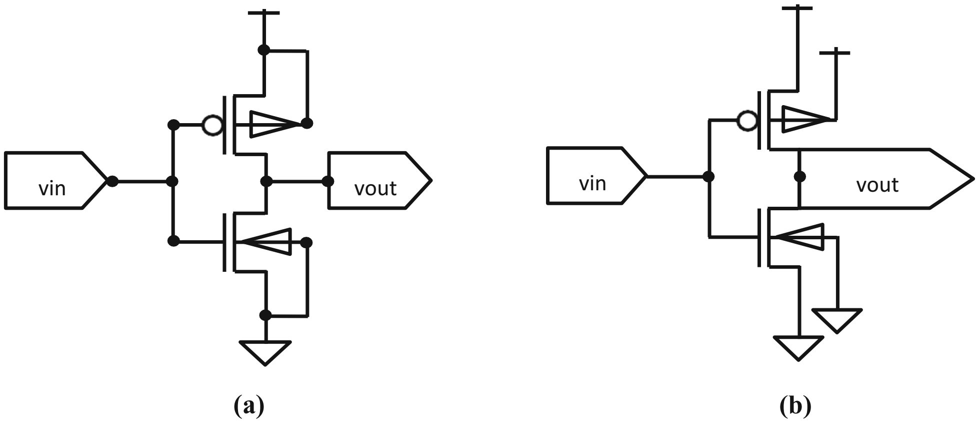


Fig. 1 GDI and modified GDI inverter [13]

3.3 Proposed 8T MGDI Full Adder

The actual operation of the MGDI full adder in Fig. 4 is given in Table 2. Erroneous results are produced with equal strength transistors for input combinations 010, 100 and 111. In order to overcome this issue, the NMOS transistors *N*1 and *N*2 in the 8T MGDI adder are designed as weak transistors when compared with the rest of PMOS and NMOS transistors. Weak transistors are transistors with low-current driving capability. (*W*/*L* ratio is small.)

The ideal equations in three regions of operation are as follows:

(1) Cutoff ($V_{\mathrm{gs}} < V_t$)

$$I_{ds} \approx 0. \tag{2}$$

(2) Linear region ($V_{\mathrm{gs}} > V_t$, $0 < V_{\mathrm{ds}} < V_{\mathrm{gs}} - V_t$)

$$I_{\mathrm{ds}} = \beta\left[\left(v_{\mathrm{gs}} - v_t\right)v_{\mathrm{ds}} - \frac{v_{\mathrm{ds}}^2}{2}\right]. \tag{3}$$

(3) Saturation region ($V_{\mathrm{gs}} > V_t$, $0 > V_{\mathrm{ds}} > V_{\mathrm{gs}} - V_t$)

$$I_{\mathrm{ds}} = \frac{\beta}{2}\left(v_{\mathrm{gs}} - v_t\right)^2, \tag{4}$$

Table 2 Function table of 8T MGDI full adder

Inputs			Erroneous output (with equalized transistors)		Correct output (with weak transistors)	
A	B	C	Cout	SUM	Cout	SUM
0	0	0	0	0	0	0
0	0	1	0	1	0	1
0	1	0	×	1	0	1
0	1	1	1	0	1	0
1	0	0	0	×	0	1
1	0	1	1	0	1	0
1	1	0	1	0	1	0
1	1	1	1	×	1	1

$$\text{where } \beta = \frac{W}{L}.\frac{\mu\varepsilon}{t_{ox}} = k.\frac{W}{L} \text{ , where } k = \frac{\mu\varepsilon}{t_{ox}}. \tag{5}$$

In all three regions of operation, $I_{\mathrm{ds}} \propto \beta$, the transistor gain. The transistor gain $\beta \propto W, \beta \propto \frac{1}{L}$. The gate length specified for a MOSFET technology is the minimum length. While designing, *L* may be larger than the minimum length but not smaller. The *W*/*L* ratio is related to the trans-conductance and the current driving capability, together with the multiplicity factor *k*. A higher *W*/*L* ratio increases the current gain (β) and subsequently a higher

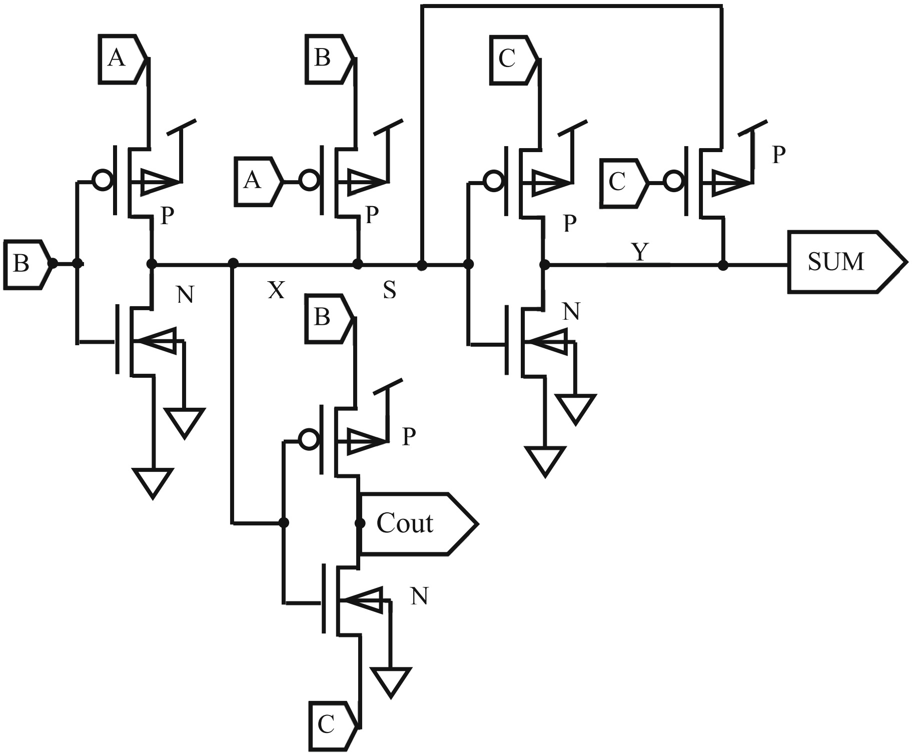


Fig. 4 Proposed 8T MGDI full adder

current for a given V_{gs}. The same is for a higher k that means $k*W/L$.

β decides the current driving capability of any transistor. By choosing transistor width W less than length L, current driving capability of the transistor is reduced.

Full adders are the basic circuit for binary addition. The number of transistors, power and delay are reduced considerably in the MGDI full adder. A modified GDI full adder in Fig. 4 uses only 8 transistors. The power consumption and delay are also reduced than GDI 10T full adder. Lot of adders is available in the literature to perform vector additions of n-bit data.

3.4 Ripple Carry Adder/Subtractor

The 8T MGDI full adder which is efficient than existing CMOS and GDI full adders is used to construct 16-bit ripple carry adder in Fig. 5. It is used for addition of 16-bit inputs.

The binary subtraction can be performed by their complement addition. The vector subtraction can be carried out by ripple carry adder where one of the inputs for addition is complemented with input carry set to logic 1. The circuit for n-bit subtraction is shown in Fig. 6. It is used to find the difference of two 16-bit inputs.

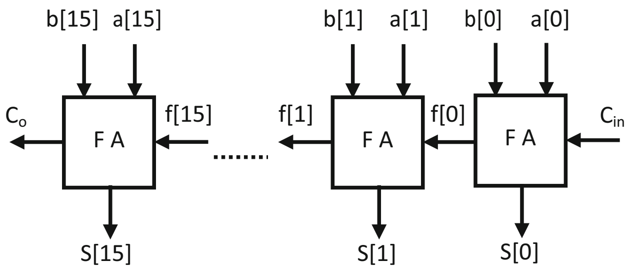


Fig. 5 *n*-Bit ripple carry adder

3.5 Low-Complexity Approximate DCT

A trade-off between accuracy and computational complexity determines the performance of the approximate DCT.

In [12], a low-complexity approximate DCT is obtained by reproducing the butterfly structure. The common computations are identified and shared to remove the redundancy in DCT matrix. Elements of matrix intend to be in {0, ± 1} to insure that resulting multiplicative complexity is null. The transformation matrix $C_1 = D_1 \cdot T_1$, where $D_{1-} = ½\cdot\mathrm{diag}$ (1, 1, 1, 1, 1, 1, 1, 1}. It requires only 12 additions for its computation. All rows of T_1 are non-null. Matrix T_1. T^T must be an orthogonal diagonal matrix.

$$X = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & -1 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

The signal flow graph in Fig. 7 shows the number of additions in the transform [12]. The continuous and dotted lines represent multiplication by + 1 and − 1, respectively.

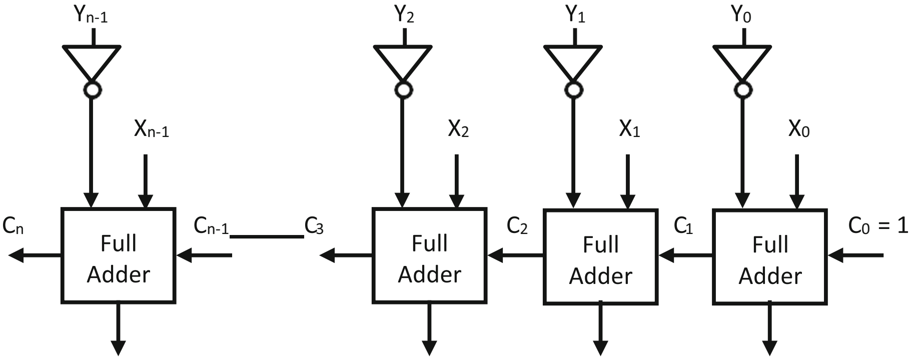
3.6 Architecture of 8-Point Approximate DCT Using MGDI Technique

Low-complexity and power-aware architecture for DCT using MGDI technique is proposed in Fig. 8. It uses 12 additions for computation, i.e., 8 adders and 4 subtractors.

3.7 Architecture of 16-Point Approximate DCT Using MGDI Technique

Two units for the computation of C8 in Fig. 8 are used along with an input adder unit and output permutation unit. The structures of 16-point DCT of Fig. 9 could be extended

Fig. 6 *n*-Bit subtractor design



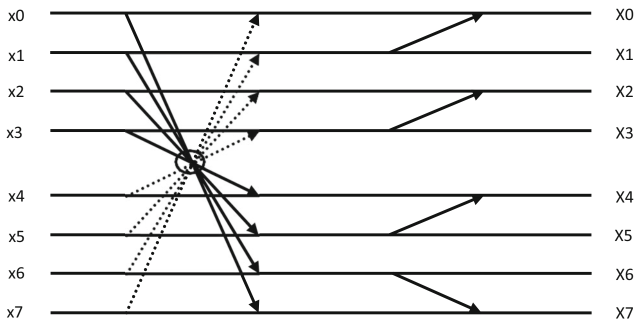


Fig. 7 Signal flow graph of approximate DCT using 12 additions

to obtain the DCT of higher sizes. For example, the structure for the computation of 32-point DCT could be obtained by combining a pair of 16-point DCTs with an input adder block and output permutation block. It can be found that the proposed method requires the lowest number of additions and does not require any shift operations [15]. Shift operation has indirect contribution to the hardware complexity since shift-add operations lead to increase in bit width which leads to higher hardware complexity of arithmetic units. All approximation methods involve significantly less computational complexity than the exact DCT algorithms. According to Loffler algorithm, the exact DCT computation [16] requires 29, 81, 209 and 513 additions along with 11, 31, 79 and 191 multiplications, respectively, for 8-, 16-, 32- and 64-point DCTs, whereas approximate DCT computation requires 12, 40, 112 and 288 additions, respectively, for 8-, 16-, 32- and 64-point with no multiplications. The flow of the proposed work is shown in Fig. 10.

4 Results and Discussion

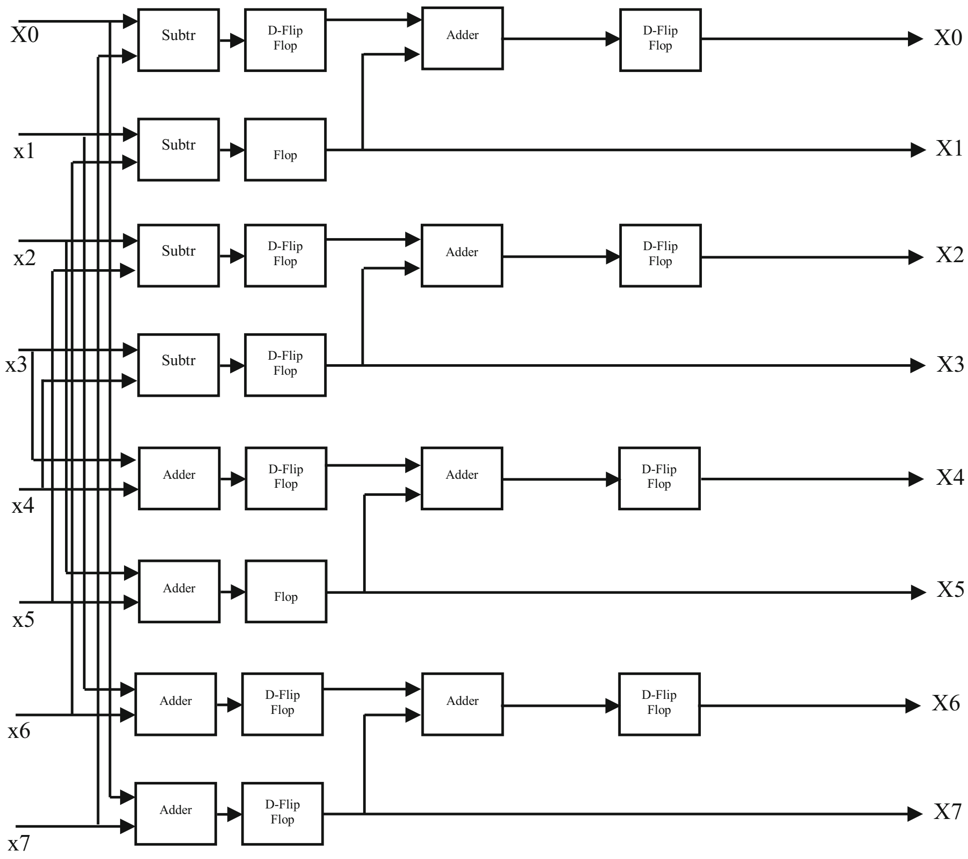
The approximate DCT architectures are simulated in T-spice at 90 nm using predictive technology model (PTM) with the power supply of 1.2 V. Design parameters are given in Table 3. Area, power and delay are reduced for 8- and 16-point approximate DCT than conventional DCT.

Simulation outputs for inverter and MGDI full adder are obtained by T-spice using specified parameters shown in Figs. 11 and 12, respectively.

Simulation output of 16-bit MGDI ripple carry adder for sum bits S_2–S_0 output carry is shown in Fig. 13 as a sample.

The conventional DCT requires 64 additions and 56 multiplications. Approximate DCTs reduce the circuit complexity which can be used for image and videos. The approximate DCT matrix contains only zeros and ones,

Fig. 8 Digital architecture of approximate DCT using 12 additions



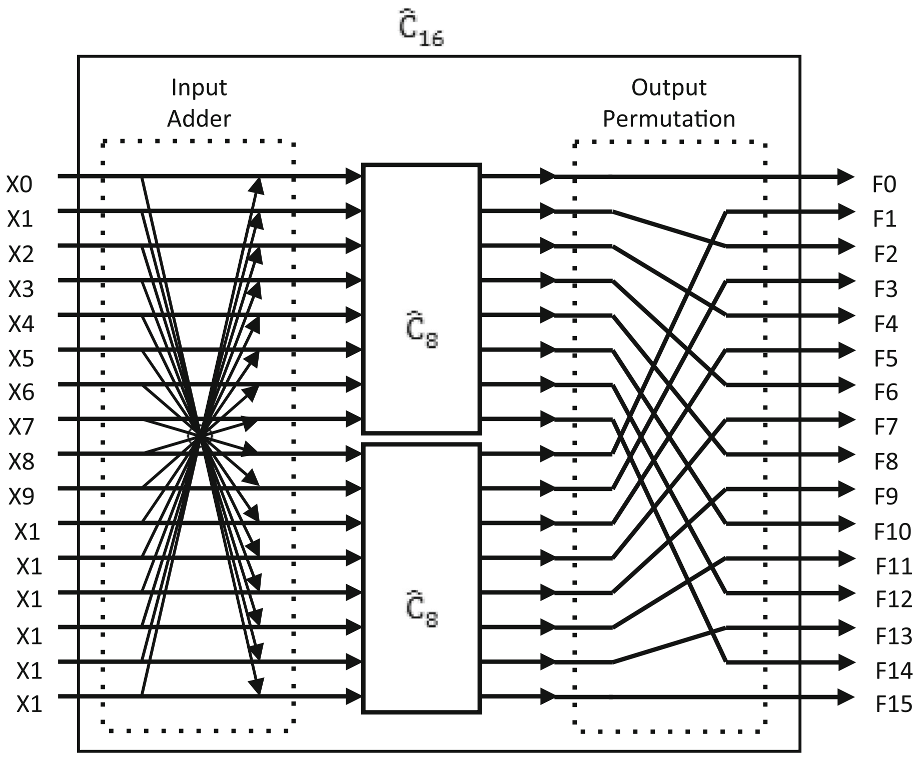


Fig. 9 Block diagram of the approximate 16-point DCT [14]

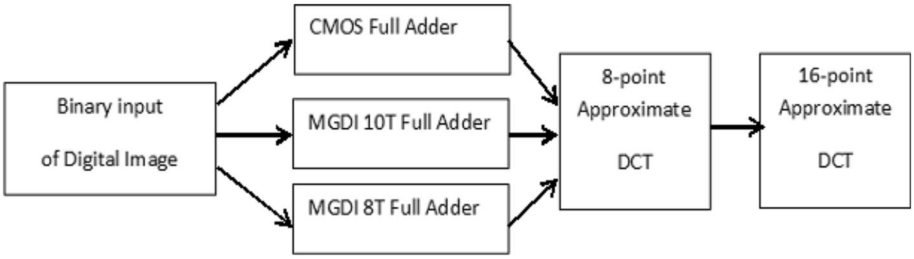


Fig. 10 Flow graph of proposed 16-point DCT

requiring 12 adders reduced to smaller, faster and more energy efficient circuitry design. A proposed modified GDI full adder uses only 8 transistors, whereas existing MGDI full adder uses 10 transistors. So the complexity and power are further reduced by using proposed MGDI full adder for DCT using 12 additions. DCT architecture is extended for 16-point approximate DCT from 8-point approximate DCT.

Simulation output of 8 bits of $X(0)$ of 8-point approximate DCT using 12 additions is shown in Fig. 14. Similarly, the outputs $X(1)$ to $X(7)$ can be obtained.

4.1 8-Point Approximate DCT

The conventional DCT gives accurate results but having high circuit complexity. The architecture for 8-point conventional DCT is implemented using CMOS logic, 10T MGDI adder and 8T MGDI adder technique and compared. Humans are able to perceive and identify the information from slightly erroneous images. So approximate DCTs are used which reduces the circuit complexity with slight compromise in accuracy. The comparison of area, power and delay of various 8-point DCT architectures is given in Table 4.

Table 3 Design parameters

Parameters	Values
Supply voltage	1.2 V
Process technology	90 nm
Model	Predictive technology model
$L_{\min}$	90 nm
W (normal transistor)	180 nm
W (weak transistor)	80 nm

The comparative analysis shows that 96% of area, 99% of power and 97% of delay are reduced in the 8-point

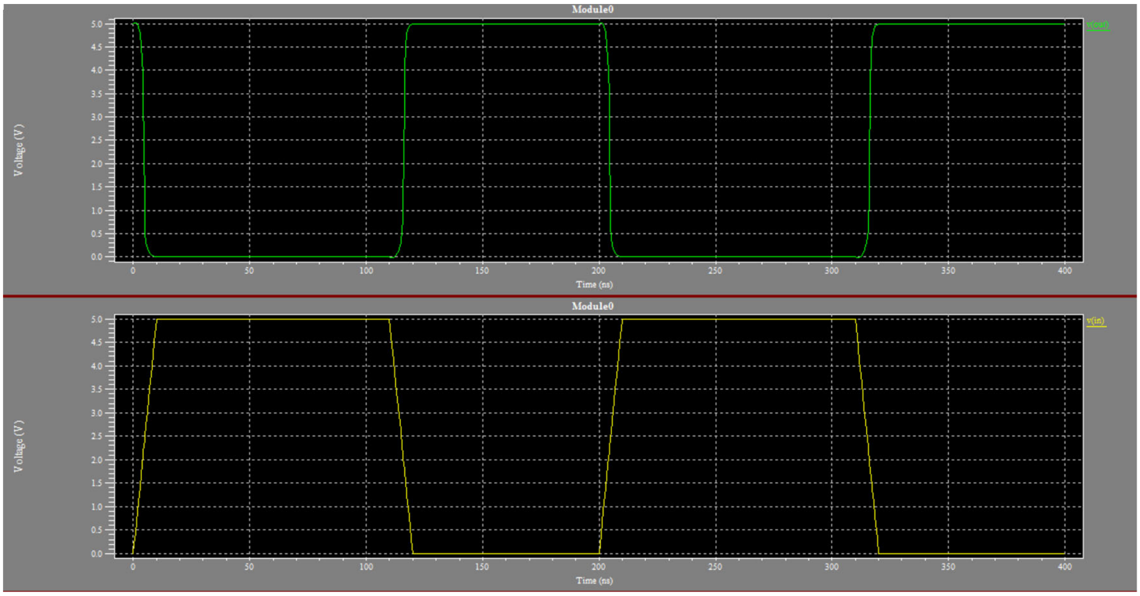


Fig. 11 Simulation output of MGDI inverter

approximate DCT using 8-transistor MGDI full adder compared with conventional DCT. Similarly, area, power and delay of 8-point MGDI approximate DCT using 14 additions are reduced by 23%, 16.6% and 11.2%, respectively, than approximate DCT using 12 additions.

4.2 16-Point Approximate DCT

Two units for the computation of C_8 are used along with an input adder unit and output permutation unit in 16-point approximate DCT. The structures of 16-point DCT could be extended to obtain the DCT of higher sizes. The approximate 16-point DCT architecture using C_8 with 14 and 12 additions, respectively, is implemented using 8T

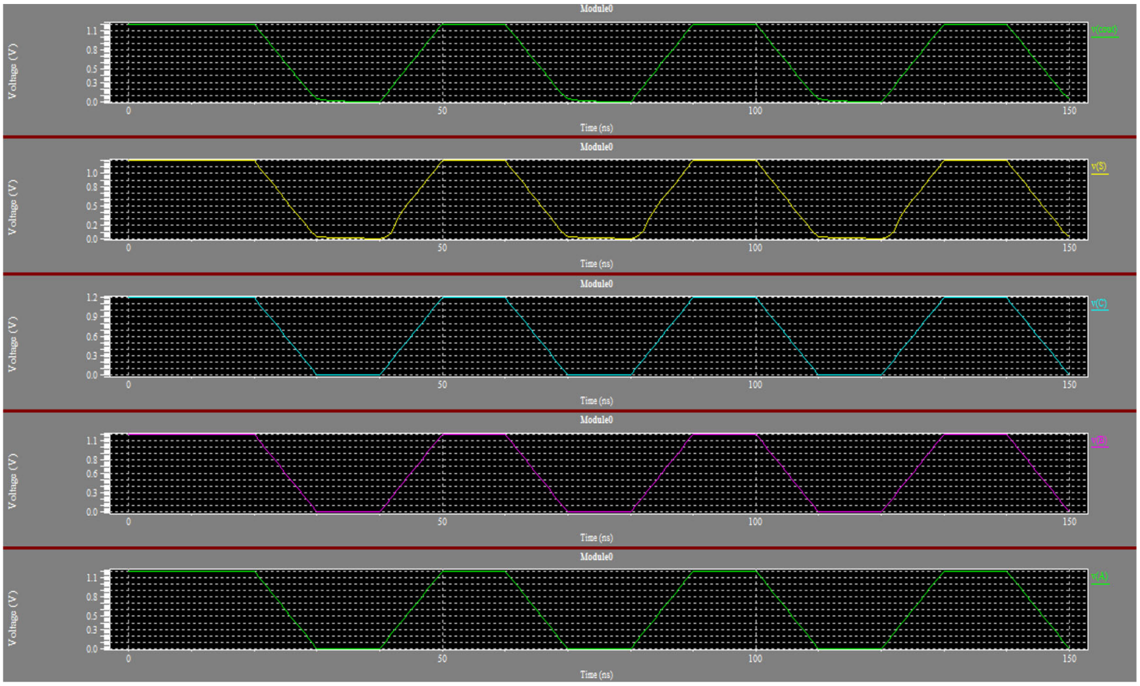


Fig. 12 Simulation output of MGDI full adder

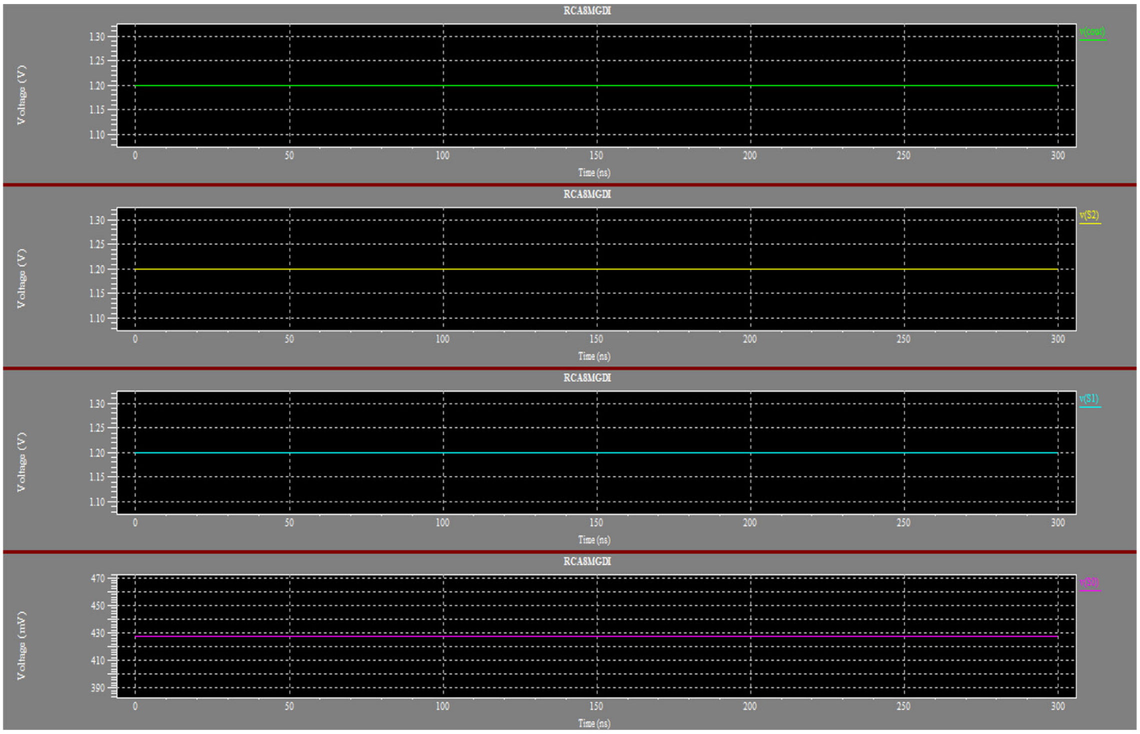


Fig. 13 Simulation output of GDI ripple carry adder

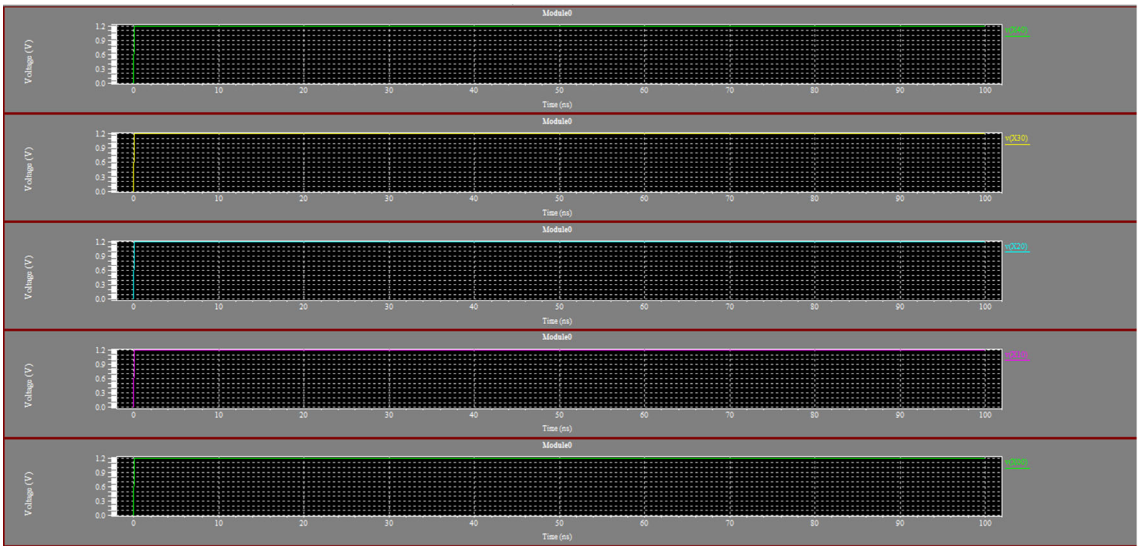


Fig. 14 Simulation output of 8T GDI adder approximate DCT using 12 additions

MDGI adder. The comparison of area, power and delay of 16-bit approximate DCTs is given in Table 5.

Simulation output shows that the area, power and delay of 16-point MGDI approximate DCT architecture using 12 additions are reduced by 20%, 16% and 7%, respectively, than 16-point MGDI approximate DCT using 14 additions with little compromise in accuracy in terms of peak signal-to-noise ratio (PSNR). Reduction in computational complexity of DCT comes at the cost of PSNR. The PSNR computation of existing transforms is given in Table 6. PSNR resulted from Bouguezel et al. [19] is better than other algorithms, but the computational complexity is more. So the transform by Vaithyanathan et al. [22] outperforms in the trade-off of PSNR and reduced computational complexity. To validate the PSNR values, the grayscale images shown in Fig. 15 are used.

Table 4 Comparison result of 8-point DCTs

8-Point DCTs	Area (no. of transistors)	Power (watts)	Delay (s)
CMOS conventional DCT	124,208	3.87e−01	2.69e−04
10T GDI adder conventional DCT	97,792	3.05e−01	2.14e−04
8T GDI adder conventional DCT	87,680	2.17e−01	1.98e−04
8T GDI adder approximate DCT using 14 additions	4112	3.61e−04	8.17e−06
8T GDI adder approximate DCT using 12 additions	3136	3.01e−04	7.25e−06

Table 5 Simulation result of 16-point approximate DCT

16 Point DCT	Area (no. of transistors)	Power (watts)	Delay (s)
8T GDI adder approximate 1D DCT using 14 additions	9376	6.72e−04	4.27e−05
8T GDI adder approximate 1D DCT using 12 additions	7424	5.58e−04	3.98e−05

Table 6 PSNR in (dB) obtained by different 8 × 8 transform matrices [12]

Transform	Lena	Boat	Gold hill	Barbara	Lighthouse
Bouguezel et al. [17]	46.1432	46.5150	45.7904	46.1774	46.5060
Bouguezel et al. [18]	38.4446	38.7755	38.1416	38.4739	38.7539
Bouguezel et al. [19]	50.7247	51.0442	50.3861	50.7648	51.0748
Bouguezel et al. [6] (a = 1)	39.1987	39.5451	38.9039	39.2220	39.5325
Senapati et al. [20]	40.0021	40.247	39.6411	39.7347	40.0654
Cintra and Bayer [7]	40.3542	40.5915	39.9636	40.3566	40.7288
Bayer and Cintra [8]	40.4921	40.8259	40.1348	40.5130	40.8217
Transform in [21]	42.5718	42.9205	42.2187	42.6054	42.9414
Transform in [22]	41.1196	41.4667	40.7955	41.1598	41.4880
Transform in [11]	41.7576	42.1115	41.4527	41.7853	42.1274
Proposed architecture	41.1196	41.4667	40.7955	41.1598	41.4880

Fig. 15 Grayscale images for PSNR computation

5 Conclusion

The DCT is an important transform in image and video processing systems. The digital implementation of 16-point approximate DCT architecture based on Modified Gate Diffusion Input (MGDI) technique is realized in digital VLSI hardware using 8-point DCT architecture. The proposed MGDI full adder with 8 transistors is used in the DCT architecture. Simulation of 16-point approximate multiplier-free MGDI DCT using Tanner SPICE for 90 nm CMOS process technology at 100 MHz shows that 20%, 16% and 7% of area, power and delay are reduced, respectively, when compared with approximate DCT using 14 additions. The proposed architecture enhances results in terms of hardware complexity, regularity and modularity with a little compromise in accuracy.

Compliance with Ethical Standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Rezaei M, Wenger S, Gabbouj M (2005) Video rate control for streaming and local recording optimized for mobile devices. In: Proceedings of IEEE 16th international symposium on personal, indoor mobile radio communications, Berlin, pp 2284–2288
2. Morgenshtein A, Shwartz I, Fish A (2010) Gate diffusion input logic in standard CMOS nanoscale process. In: Proceedings of IEEE 26th convention of electrical and engineers, Eliat, pp 776–780
3. Morgenshtein A, Yuzhaninov V, Kovshilovsky A, Fish A (2014) Full-swing gate diffusion input logic-case-study of low-power CLA adder design. Integr VLSI J 47(1):62–70
4. Singh R, Verma P, Mishra YK (2013) Modified GDI technique: a power efficient method for digital circuit design. Int J Electron Comput Sci Eng 2(4):1071–1080
5. Bouguezel S, Ahmad MO, Swamy MNS (2008) Low-complexity 8 × 8 transform for image compression. Electron Lett 44(21):1249–1250
6. Bouguezel S, Ahmad MO, Swamy MNS (2011) A low-complexity parametric transform for image compression. In: Proceedings of IEEE international symposium on circuits and systems, Rio de Janeiro, pp 2145–2148
7. Cintra RJ, Bayer FM (2011) A DCT approximation for image compression. IEEE Signal Process Lett 18(10):579–582
8. Bayer FM, Cintra RJ (2012) DCT-like transform for image compression requires 14 additions only. Electron Lett 48(15):919–921
9. Potluri US, Madanayake A, Cintra RJ, Bayer FM, Rajapaksha N (2012) Multiplier-free DCT approximations for RF multi-beam digital aperture-array space imaging and directional sensing. Meas Sci Technol 23(11):1–15
10. Potluri US, Madanayake A, Cintra RJ, Bayer FM, Kulasekera S, Edirisuriya A (2014) Improved 8-point approximate DCT for image and video compression requiring only 14 additions. IEEE Trans Circuits Syst I Regul Pap 61(6):1727–1740
11. Edirisuriya A, Madanayake A, Cintra RJ, Bayer FM (2012) A multiplication-free digital architecture for 16 × 16 2-D DCT/DST transform for HEVC. In: Proceedings of IEEE 27th convention of electrical and electronics engineers, Israel, pp 1–5
12. Vaithiyanathan D, Seshasayanan R (2014) Area and power efficient architecture for image compression. EURASIP J Adv Signal Process 180:1–9
13. Fish A, Morgenshtein A, Wagner IA (2002) Gate diffusion input (GDI): a power efficient method for combinational circuits. IEEE Trans Very Large Scale Integr Syst 10(5):566–581
14. Jridi M, Alfalou A, Meher PK (2015) A generalized algorithm and reconfigurable architecture for efficient and scalable orthogonal Approximation of DCT. IEEE Trans Circuits Syst I Regul Pap 62(2):449–457
15. Lengwehasatit K, Ortega A (2004) Scalable variable complexity approximate forward DCT. IEEE Trans Circuits Syst Video Technol 14(11):1236–1248
16. Ahmed N, Natarajan T, Rao KR (1974) Discrete cosine transform. IEEE Trans Comput 23(1):90–93
17. Bouguezel S, Ahmad MO, Swamy MNS (2008) A multiplication-free transform for image compression. In: Proceedings of 2nd international conference on signals, circuits and systems, Monastir, pp 1–4
18. Bouguezel S, Ahmad MO, Swamy MNS (2009) A fast 8 × 8 transform for image compression. In: Proceeding of international conference on microelectronics, Marrakech; pp 74–77
19. Bouguezel S, Ahmad MO, Swamy MNS (2010) A novel transform for image compression. In: Proceeding of IEEE international midwest symposium on circuits and systems, Seattle, pp 509–512
20. Senapati RK, Pati UC, Mahapatra KK (2010) A low complexity orthogonal 8 × 8 transform matrix for fast image compression. In: Proceeding of the annual IEEE India conference, India, pp 1–4
21. Vaithiyanathan D, Seshasayanan R, Anith S, Kunaraj K (2013) Low power DCT architecture for image compression. In: Proceeding of the international conference on advanced computing and communication systems, India, pp 1–6
22. Vaithiyanathan D, Seshasayanan R, Anith S, Kunaraj K (2013) A low-complexity DCT approximation for image compression with 14 additions. In: Proceeding of the international conference on green computing, communication and conservation of energy, India, pp 303–307