

High Speed ASIC Design of DCT for Image Compression

Deepa Yagain, Ashwini, and A. Vijaya Krishna

PES Institute of Technology,
100 ft Ring Road, Bangalore, Karnataka ,India
Deepa.yagain@gmail.com, ashwinib@pes.edu, avijayk@hotmail.com

Abstract. This paper gives the design and implementation of an image data compression method such as DCT(Discrete Cosine Transform) using vedic multiplier. This VLSI hardware can be used in practical coding systems to compress images[1]. Discrete cosine transform (DCT) is one of the most popular schemes because of its compression efficiency and small mean square error. DCT is used specially for the compression of images where tolerable degradation is accepted. In this paper, DCT modules are designed, implemented and verified using 90nm technology library using Tanner EDA. Here various individual cores are designed and connected to implement an ASIC for image compression. The Vedic multiplier in this case performs the multiplication much faster when compared to usual array multiplier approach. Due to this, the speed can be increased. Also since all the simulations and implementations are done in 90nm which is one among the deep submicron technologies, the power, area and length of interconnects taken will be less.

Keywords: Image Compression, ASIC, Discrete Cosine Transform, Vedic Multiplier, Pixel.

1 Introduction

Transform coding constitutes an integral component of contemporary image/video processing applications. The three important features of a suitable transform are its compression efficiency, which relates to concentrating the energy at low frequencies, ease of computation, and minimum mean square error. DCT is the popular technique as it possesses these three advantages and can be represented algorithmically. In a video transmission system, adjacent pixels in consecutive frames show very high correlation. DCT converts data (image pixels) into sets of frequencies. The first frequencies in the set are the most meaningful; the latter, the least. The least meaningful frequencies can be stripped away based on allowable resolution loss.

In this paper 2D DCT is used for image compression which is an invertible linear transform and is widely used in many practical image compression systems. As the DCT related technology becomes prominent in image coding systems[3], an efficient and reliable implementation of the 2D-DCT operation may greatly improve the system performance. Ex: When designing a video codec system, it is important to use a two dimensional DCT functional block in the circuit. By using ASIC as DCT block, performance of codec is improved. A multiplier design using "Urdhva-tiryakbyham" sutras[6][7] has been used to design the multiplier.

2 Analysis and Design DCT Modules

Discrete Cosine Transform. DCT[4] is a real transform that transforms a sequence of real data points into its real spectrum and therefore avoids the problem of redundancy. The most common DCT definition of a 1-D sequence of length N is

$$C(u) = \alpha(u) \sum_{x=0}^{N-1} f(x) \cos \left[\frac{\pi(2x+1)u}{2N} \right] \tag{1}$$

For $u = 0, 1, 2, \dots, N-1$. Similarly, the inverse transformation is defined as,

$$f(x) = \sum_{u=0}^{N-1} \alpha(u) C(u) \cos \left[\frac{\pi(2x+1)u}{2N} \right] \tag{2}$$

For $x = 0, 1, 2, \dots, N-1$. In both the equations $\alpha(u)$ is defined as

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & \text{for } u = 0 \\ \sqrt{\frac{2}{N}} & \text{for } u \neq 0. \end{cases} \tag{3}$$

The 2-D DCT is a direct extension of the 1-D case and is given by

$$F(u, v) = \frac{2}{\sqrt{MN}} C(u) C(v) \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) \cos \left(\frac{(2m+1)u\pi}{2M} \right) \cos \left(\frac{(2n+1)v\pi}{2N} \right) \tag{4}$$

Where $C(x) = \begin{cases} \frac{1}{\sqrt{2}} & x = 0 \\ 1 & \text{otherwise} \end{cases}$

M=Number of rows in the input data set

N= Number of columns in the input data set

m= Row index in the time domain $0 \leq m \leq M-1$

n= Column index in the time domain $0 \leq n \leq N-1$

f(m,n)=time domain data

u=Row index in frequency domain

v=Column index in frequency domain

F(u,v)=Frequency domain coefficient for $u, v = 0, 1, 2, \dots, N-1$

Design of DCT f(x) Module. The Block diagram of DCT f(x) module is as shown in Fig 1(a). It takes f(x) as input signal and Count_Reg enl, Count_Mux_Enable signal as enable signal to give the output Mux_Out. The various blocks used in this module are Registers and Multiplexer. Here any one of the f(x) signal will be selected as Mux_Out depending on Count_Mux_Enable signal whenever clk(Clock) and Count_Reg enables are high. The implementation of Register and Mux are done using MOSFETs. Here pass transistors are used to design DFF which increases the performance which in turn is used to implement a register.

Design of DCT Cos Module. The block diagram of DCT Cos module is as shown in Fig 1(b). The various inputs given to Dct Calc block in DCT cos module are f(x) from DCT_f(x) module, Cos values chosen from various multiplexers. The chosen Cos values are from the memory block. For Ex: Any one cos value from Cos1 to Cos8 will

be chosen depending on Cos_Mux_En. This signal is combined with $f(x)$ which is the output of Dct_f(x) module and given as input to DCT_Calc block which further will generate $F(X)$. The muxes and registers in DCT cos module is implemented similarly as given in DCT f(x) module.

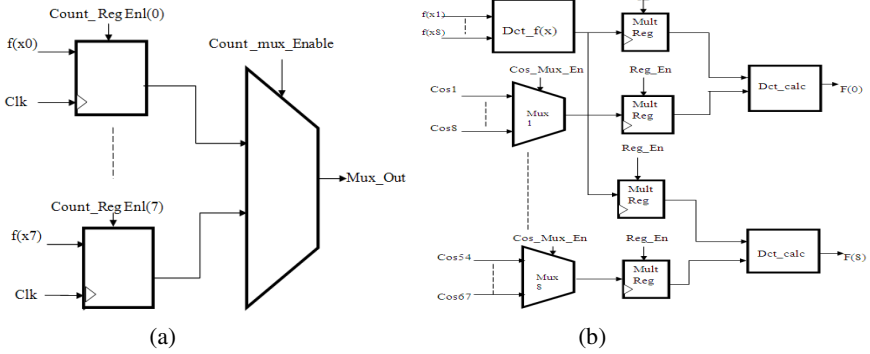


Fig. 1. (a) Block diagram of DCT f(x) Module and (b) Block diagram of DCT Cos Module

The Schematic and symbol of DFF using Pass transistors is as shown in Fig 2. This is implemented using Pass transistors to increase the performance parameters. Similarly the Mux block is implemented using Nand gates and buffers. By cascading 16 DFFs we can implement a register block in DCTf(x) module. Similarly 8:1 mux is implemented and connected to obtain DCT f(x) module.

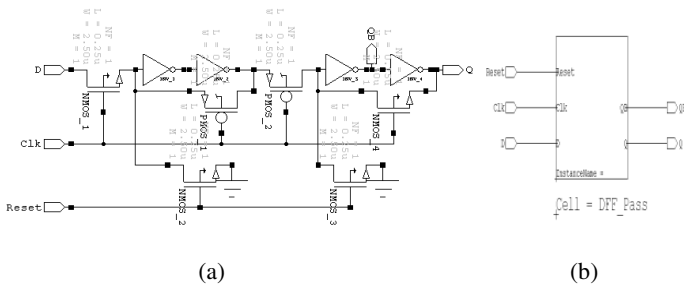
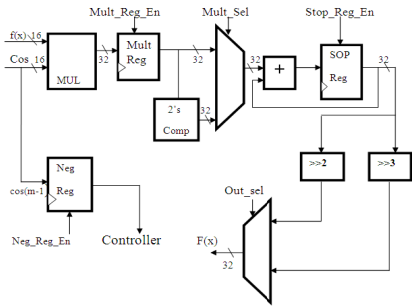


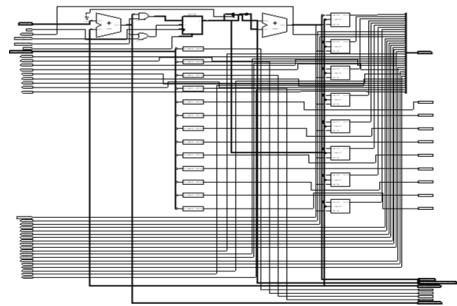
Fig. 2. (a) Schematic of DFF using pass transistors and (b) Symbol of DFF

Design of DCT Calc Module. The block diagram and schematic for DCT Calc block is shown in figure 3(a) and 3(b). Here, multiplication of $f(x)$ and Cos bits are performed in multiplier block. The result is given to the register which is enabled by Mult_Reg_En given from controller. The output of register and its 2's complement value are given as inputs to Mux. Simultaneously MSB of Cos bit stream is sent to controller for sign check. Based on the sign, controller will issue Mult_sel signal to Mux which selects either output of Mult_Reg or its 2's complement. The two's complement values are generated by giving outputs of Mult_Reg to XOR gates. The generated result is added for all the rows (m) and columns (n) of the image block which is indicated by adder and SOP register. According to the formula given in equation (4), here

$$C(x) = \begin{cases} \frac{1}{\sqrt{2}} & x = 0 \\ 1 & \text{otherwise} \end{cases}$$



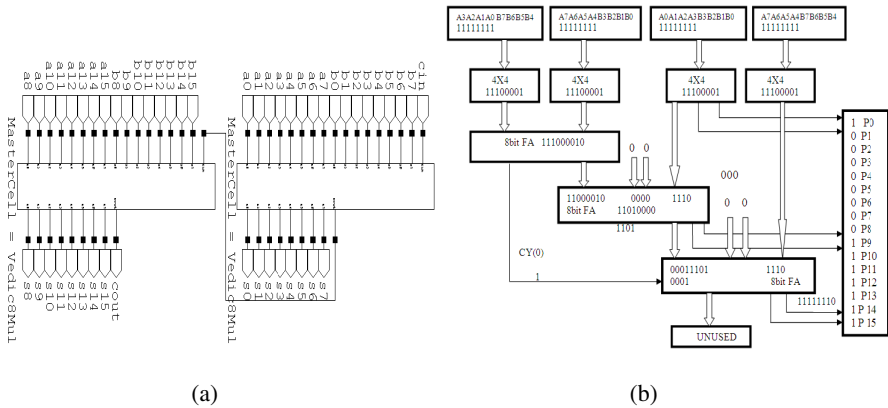
(a)



(b)

Fig. 3. (a) Block Diagram of DCT calc Module and (b) Schematic of DCT calc Module

Thus x can be either zero or a non zero value. Accordingly the result is either right shifted by 2 bits or 3 bits so that the result is either divided by 4 or 8. The out_sel signal from the controller is used to select the shifted values so that final result $F(x)$ is obtained. The basic block in DCT Calc is Multiplier and it is as shown in figure 4 (a) and 4 (b).



(a)

(b)

Fig. 4. (a) Schematic of 16 bit Vedic multiplier and (b) Schematic block of 8 bit Vedic multiplier

Vedic algorithm for multiplication [8] can be applied to various imaging applications. The block diagram of 8 bit vedic multiplier and implementation of 16 bit from 8 bit multipliers is as shown in Figure 5. Here Urdhva Tiryakbham sutra is used in which various vertical products and Cross products of multiplicand and multiplier are added to obtain the final product. For Ex: when 8 bit inputs $A=11111111$, $B=11111111$ and Output is 11111100000000 .

Design of Controller Module. The Block diagram of DCT functional block and state diagram of Controller is as shown in figure 5(a) and 5(b).

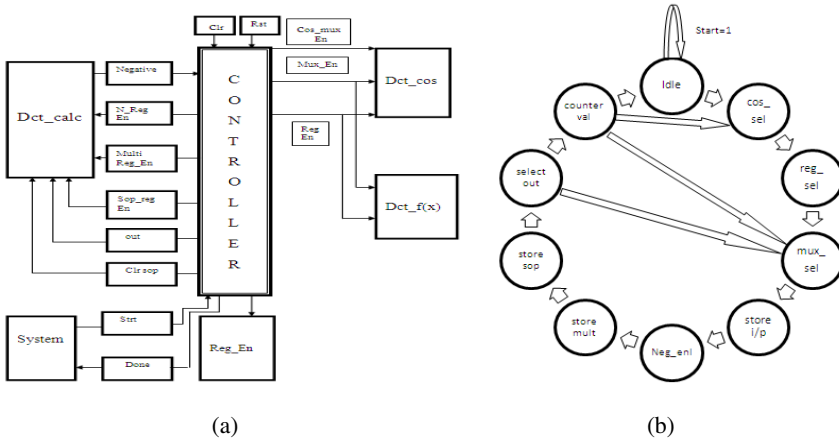


Fig. 5. (a) Block Diagram of DCT functional Block and (b) State machine of Controller Block

Initially, Cos_Mux_En, Reg_En, Neg_Reg_En, Mult_Reg, Mult_Sel, SOP_Reg_En, Clr_SOP_Reg are set to zero. After start is issued, DCT Cos and DCT f(x) module, Mux enables and Reg enables will go high such that the f(x) and cos values are selected and moved to the DCT Calc block. The Neg_reg_enable is made high so that MSB of COS bits are checked for negativity. In the next clock Mult_Reg_En and Mult_Sel signals are made high based on the sign bit as to select either multiplication result or its 2's complement. SOP_Reg_Enable is issued so that the previous sum gets added with the current value. Depending on the out_sel signal, F(X) obtained. This repeats till the Count_val becomes zero. After every iteration, Count_val is decremented by 1. Controller goes to idle state once the count_val becomes zero and remains idle till start becomes high.

3 Simulation and Results

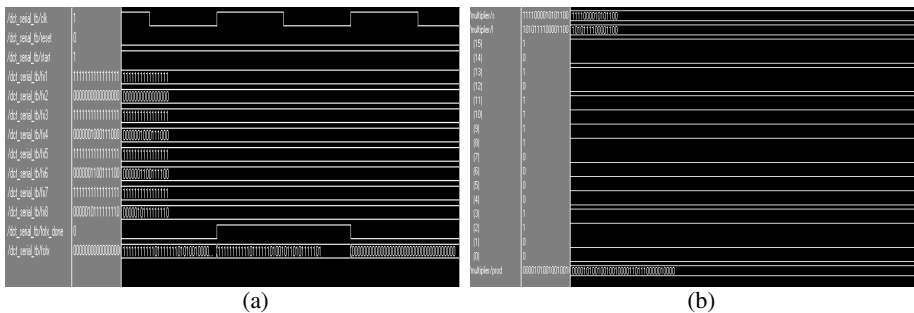


Fig. 6. Simulation results of DCT module showing (a) inputs and (b) F(X) values

Table 1. Comparison results of various multipliers

Parameter	Type of Multiplier		
	Array	Booth	Vedic
Power in Watts	7.634327e-002	5.634327e-002	4.7911e-002
Delay	43ns	37 ns	22ns

4 Conclusions

The entire design procedure of an ASIC 2D-DCT processor is presented in this paper. The important advantage of this core is its usage for various real-time coding systems. As the DCT related technology becomes prominent in image coding systems, an efficient and reliable implementation of the 2D-DCT operation may greatly improve the system performance. The intent of the paper is exploring a 2D-DCT architecture, to regulate the design for different applications, thus expedite the design procedure. The transistors in these modules are designed using MOSIS 90nm technology library.

References

1. Rao, K.R., Yip, P.: Discrete Cosine Transform—Algorithms, Advantages, Applications. Academic Press, London (1990)
2. Ahmed, N., Natarajan, T., Rao, K.R.: Discrete cosine transform. *IEEE Trans. Comput.* C-23(1), 90–93 (1974)
3. Yu, S., Swartzlander Jr., E.E.: DCT implementation with distributed arithmetic. *IEEE Trans. Comput.* 50(9), 985–991 (2001)
4. Wallace, G.K.: The JPEG still picture compression standard. *Communications of the ACM* 34(4), 30–44 (1991)
5. Cho, N.I., Lee, S.U.: Fast Algorithm and Implementation of 2-D DCT. *IEEE Transactions on Circuits and Systems* 38, 297 (1991)
6. Wallace, C.S.: A suggestion for a fast multiplier. *IEE Trans. Electronic Comput.* EC-3, 14–17 (1964)
7. Tiwari, H.D., Gankhuyag, G., Kim, C.M., Cho, Y.B.: Multiplier design based on ancient Indian Vedic Mathematics. In: *Proceedings IEEE International SoC Design Cotiference, Busan, November 24-25, pp. 65–68 (2008)*
8. Thapliyal, H., Srinivas, M.B.: High Speed Efficient N x N Bit Parallel Hierarchical Overlay Multiplier Architecture Based on Ancient Indian Vedic Mathematics. *Transactions on Engineering, Computing and Technology* 2 (2004)