International Journal of Electrical and Computer Engineering (IJECE)
Vol. 7, No. 3, June 2017, pp. 1430 – 1435
ISSN: 2088-8708, DOI: 10.11591/ijece.v7i3.pp1430-1435

VLSI Design of a Fast Pipelined 8x8 Discrete Cosine Transform

Nurulnajah Mohd Zabidi and **Ab Al-Hadi Ab Rahman**
Faculty of Electrical Engineering, Universiti Teknologi Malaysia, Malaysia

Article Info

Article history:
Received Jan 4, 2017
Revised Mar 18, 2017
Accepted Apr 2, 2017

Keyword:

DCT
VLSI
Pipeline
Video Coding

ABSTRACT

This paper presents a Very Large Scale Integrated (VLSI) design and implementation of a fixed-point 8x8 multiplierless Discrete Cosine Transform (DCT) using the ISO/IEC 23002-2 algorithm. The standard DCT algorithm, which is mainly used in image and video compression technology, consists of only adders, subtractors, and shifters, therefore making it efficient for hardware implementation. The VLSI implementation of the algorithm given in this paper further enhances the performance of the transform unit. Furthermore, circuit pipelining has been applied to the base design of the DCT, which significantly improves the performance by reducing the longest path in the non-pipeline design. The DCT has been implemented using semi-custom VLSI design methodology using the TSMC 0.13um process technology. Results show that our DCT designs can run up to around 1.7 Giga pixels/s, which is well above the timing required for real-time ultra-high definition 8K video.

Corresponding Author:
Ab Al-Hadi Ab Rahman
Faculty of Electrical Engineering, Universiti Teknologi Malaysia
81310 Skudai, Johor Bahru, Malaysia
Phone: +6019-7338914
Email: hadi@fke.utm.my

1. INTRODUCTION

Image compression is a process to reduce data redundancies in the image in order to increase the storage capacity. Discrete Cosine Transform (DCT) is widely used for lossy compression. Most image or video coding standards such as Joint Photographic Expert Group (JPEG) and Moving Picture Expert Group (MPEG) use DCT as a standard transform coding scheme. The latest High Efficiency Video Coding (HEVC) which is finalized in 2013 also utilizes the DCT as the main transform unit [1].

DCT basis function is the Cosine, where multiplication and addition are the main arithmetic operations involved. Many DCT-based research has been conducted in the past few years, which has produced different kind of DCT Algorithms, such as Arai DCT scheme, Wang Factorization, Lee DCT for power of two block length, Loeffler algorithm, and Feig-Winograd factorization ([2]). These Algorithms have been used in practical applications. In recent image processing technology, various hardware implementation of DCT are using Arai DCT scheme [3]. It uses only five multiplications and twenty-nine addition, which is less arithmetic operations if compared to other stated algorithms. For the MPEG technology, the International Standards Organization (ISO) released an optimized fixed-point multiplierless version of the DCT algorithm, suitable for image and video compression. The standard which is called the ISO/IEC 23002-2 is described and implemented in the present work [4].

VLSI design of DCT can be found in numerous articles, with an overview given in [5]. For comparison purposes, we have analyzed three similar designs. The work by Mandayake et al in [6] presents a VLSI architecture of the DCT using the Arai DCT scheme. It proposes a fast algorithm by reducing the number of integer channels. The design is implemented using 45nm technology. The work by Wahid et al [7] proposes an area efficient fixed point DCT architecture implemented in 0.18 um CMOS technology. Another interesting work is by Fu et al [8], where a low power implementation is proposed based on algebraic integer encoding technique. This work also utilizes 0.18um CMOS technology. Performance results for these works are given in the results section.

The present paper on the other hand, describes the semi-custom Very Large Scale Integration (VLSI) design of the ISO/IEC 23002-2 DCT algorithm using TSMC 0.13um technology, similar to the design methodology used in

[9]. The design has also been optimized by applying circuit pipelining.

1.1. DCT Theoretical Background

The basic equation of N-point one-dimensional Discrete Cosine Transform (1D-DCT) as defined in [10]:

$$v(k) = \alpha(k) \sum_{n=0}^{(N-1)} u(n) cos \frac{(2n+1)k\pi}{2N}, \quad 0 \leq k \leq N-1 \tag{1}$$

where u(n) denotes the input data, v(k) denotes the output data, whereby

$$\alpha(0) = \sqrt{\frac{1}{N}}; \qquad \alpha(k) = \sqrt{\frac{2}{N}}, \ 1 \leq k \leq N-1 \tag{2}$$

1D-DCT equation for N=8 , can also be written in matrix form as follows:

$$\begin{bmatrix} v(0) \\ v(1) \\ v(2) \\ v(3) \\ v(4) \\ v(5) \\ v(6) \\ v(7) \end{bmatrix} = \begin{bmatrix} c_4 & c_4 & c_4 & c_4 & c_4 & c_4 & c_4 & c_4 \\ c_1 & c_3 & c_5 & c_7 & -c_7 & -c_5 & -c_3 & -c_1 \\ c_2 & c_6 & -c_6 & -c_2 & -c_2 & -c_6 & c_6 & c_2 \\ c_3 & -c_7 & -c_1 & -c_5 & c_5 & c_1 & c_7 & -c_3 \\ c_4 & -c_4 & -c_4 & c_4 & c_4 & -c_4 & -c_4 & c_4 \\ c_5 & -c_1 & c_7 & c_3 & -c_3 & -c_7 & c1 & -c_5 \\ c_6 & -c_2 & c_2 & -c_6 & -c_6 & c_2 & -c_2 & c_6 \\ c_7 & -c_5 & c_3 & -c_1 & c1 & -c_3 & c_5 & -c_7 \end{bmatrix} \begin{bmatrix} u(0) \\ u(1) \\ u(2) \\ u(3) \\ u(4) \\ u(5) \\ u(6) \\ u(7) \end{bmatrix} \tag{3}$$

where

$$c_i = cos \frac{i\pi}{16}, \quad i = (2n+1)k \tag{4}$$

Equation (3) shows that the matrix computes in column wise to produce 1D-DCT. Computing it further in row wise produces the 2D-DCT. Considering the intensive computations required in DCT, many efficient algorithms are proposed and reported in literature as mentioned, including the ISO/IEC 23002-2 used in the preset work. The algorithm is given in Figure 1. The inputs are ind0 to ind7, where each input is 32-bit wide. Several variables are defined for intermediate operations, where finally the results are stored in the outputs outd0 to outd7.

```
x0 := ind0 + ind7;
x1 := ind0 - ind7;
x4 := ind1 + ind6;
x5 := ind1 - ind6;
x2 := ind2 + ind5;
x3 := ind2 - ind5;
x6 := ind3 + ind4;
x7 := ind3 - ind4;

xa := pmul_1_2(x3);
x3 := pmul_1_1(x3);
xb := pmul_1_2(x5);
x5 := pmul_1_1(x5);

x3 := x3 + xb;
x5 := x5 - xa;

xa := pmul_2_2(x1);
x1 := pmul_2_1(x1);
xb := pmul_2_2(x7);
x7 := pmul_2_1(x7);

x1 := x1 - xb;
x7 := x7 + xa;
xa := x1 + x3;
x3 := x1 - x3;
xb := x7 + x5;
```

```
x5 := x7 - x5;
x1 := xa + xb;
x7 := xa - xb;
xa := x0 + x6;
x6 := x0 - x6;
xb := x4 + x2;
x2 := x4 - x2;
x0 := xa + xb;
x4 := xa - xb;

xa := pmul_3_2(x2);
x2 := pmul_3_1(x2);
xb := pmul_3_2(x6);
x6 := pmul_3_1(x6);

x2 := xb + x2;
x6 := x6 - xa;

outd0 := x0;
outd1 := x1;
outd2 := x2;
outd3 := x3;
outd4 := x4;
outd5 := x5;
outd6 := x6;
outd7 := x7;
```

Figure 1. The ISO/IEC 23002-2 1D-DCT algorithm

One of the critical components in the DCT algorithm is the PMUL operation, given in Figure 2. The PMUL performs Polynomial Multiplication. Equation (5) shows an example of polynomial equation, where the highest degree of polynomial in this equation is 11. The exponential could be eliminated by replacing with shift right operations. Equation (6) shows the conversion results of all the exponential in equation (5) into arithmetic shift right.

$$y = (y^3 - y^7) + ((y^3 - y^7) - y^{11})^1 \tag{5}$$

$$y = (y \gg 3 - y \gg 7) + ((y \gg 3 - y \gg 7) - y \gg 11)1 \tag{6}$$

```
pmul_1_1: X - (X >> 3) - (X >> 7);

pmul_1_2: (X >> 3) - (X >> 7) + (((x >> 3) - (x >> 7)) >> 1);

pmul_2_1: (((X >> 9) - X) >> 2) - (X >> 9) - X;

pmul_2_2: X >> 1;

pmul_3_1: ((X + (X >> 5)) >> 2) + (X >> 4);

pmul_3_2: (X + (X >> 5)) - (X + (X >> 5)) >> 2);
```

Figure 2. PMUL components used in the ISO/IEC 23002-2 1D-DCT algorithm

2. PIPELINE CONCEPT

The main concept in circuit pipelining is to split the job process into smaller stages which will help to enhance the performance by reducing the combinatorial critical path. Figure 3 shows the difference between non pipeline and pipeline structure in terms of combinational logic circuit. Non-pipeline circuit is made up of a combinational logic, an input and an output. Pipeline circuit is made up of a combinational logic that has been partitioned into smaller portion and then connected by registers. Essentially, pipelining allows the design to run at a higher operating frequency at a neglible cost in latency caused by initializing the pipeline stages.

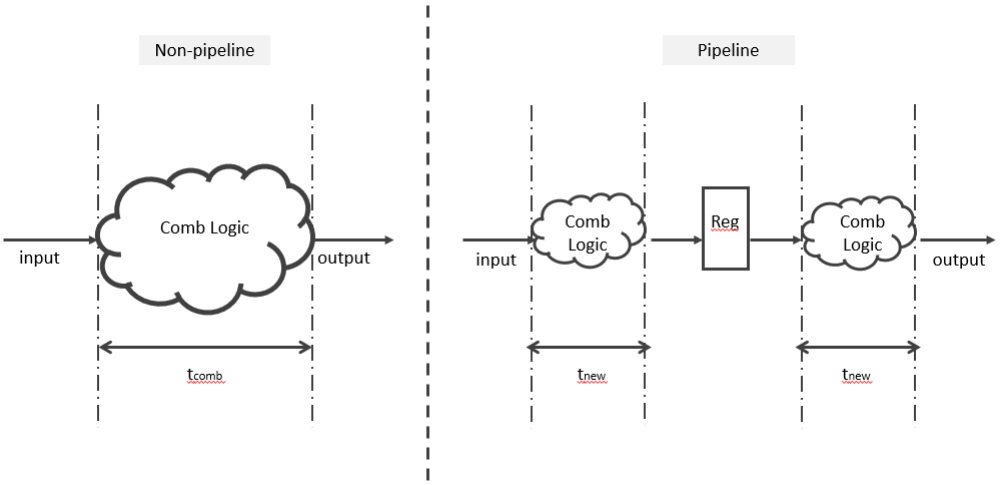

Figure 3. Non-pipeline and pipeline concept

3. PROPOSED DCT HARDWARE ARCHITECTURE

The ISO/IEC 23002-2 algorithm presented in Figure 1 is translated into a hardware architecture via a dataflow graph (DFG), shown in Figure 4. There are a total of 54 intermediate variables, which translates into wires for non-pipeline implementation. Similar to the input and output widths, wires are set at 32-bits each. The algorithm also utilizes 27 subtractors, 19 adders, and 24 shifters. From the DFG, it can be seen that the longest path is 7 arithmetic operators. Therefore, the design can be split for pipelining to reduce the path length. Essentially, the DFG can be partitioned into stages, and at each partition bounday, registers are added to store and hold intermediate data. It should be noted that a very small initial delay is expected to fill in the pipeline registers. Optimum paritioning boundary can be found using some of the proposed algorithms in [11].

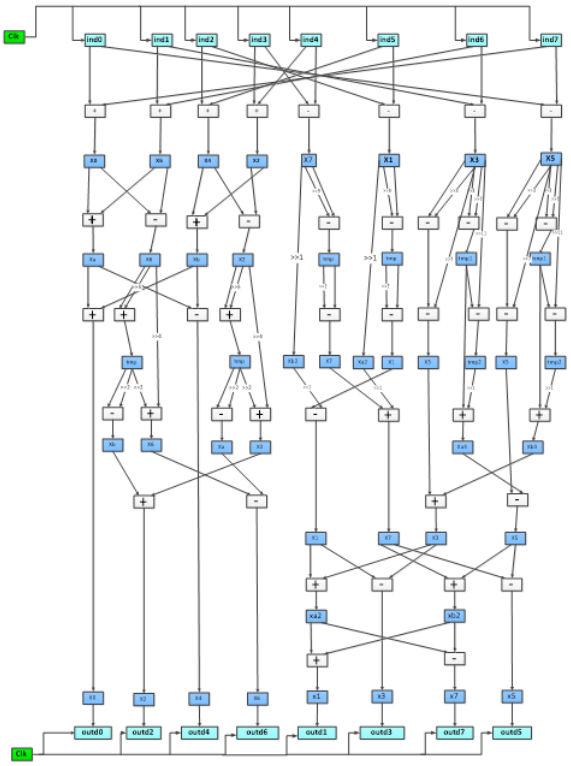

Figure 4. Data Flow Graph of Non-pipeline DCT

4. RESULT AND ANALYSIS

In this work, Mentor Graphics EDA tools with VLSI process technology TSMC 0.13um CMOS are used to design, implement and validate the DCT. Some of the tools used include Pyxis for schematic and physical design; Modelsim for high-level simulation; and Eldonet and EZ wave for low-level SPICE simulation. Semi-custom VLSI design methodology is used, whereby the arithmetic operators are instantiated manually from a standard cell library to form a complete DCT. The simulation results obtained from Modelsim at the RTL level is compared to the one obtained from SPICE simulation to ensure correct results. Here, we show results for non-pipeline and a 2-stage pipeline implementation.

The following is the number of components used for the DCT. There are a total of 19 Adders, 27 subtractors and 24 shifters (all 32-bits) that have been used in both Non-pipeline DCT and pipeline DCT. As for regsiters, 16 32-bit and 42 32-bit registers have been used in Non-pipeline and Pipeline DCT respectively. The pipeline design uses roughly 2.6 times more registers compared to non-pipeline. As for the total number of transistors, non-pipeline DCT consists of 74240, while pipeline DCT consists of 85120, with increase of roughly 14% more resource due to the pipeline registers.

By simulation for two different test patterns called pattern1 and pattern 2 (derived from the Foreman QCIF video frame), the critical data path (cpd) for Non-pipeline DCT is found to be 7.97ns for input pattern 1, and 6.59ns for input pattern 2. As for the Pipeline DCT, it is found that the cpd is 4.61ns for input pattern 1 and 3.83ns for input pattern 2. From this, it can be estimated that the non-pipeline DCT can run at the maximum speed of 125MHz, and the pipeline DCT at 217MHz. In terms of throughput, maximum output rate achieved are 1 Giga pixels/s and 1.7 Giga pixels/s respectively for non-pipeline and pipeline DCTs. The graph in Figure 5 shows the plot of maximum operating frequency vs throughput. Both of these designs can support the speed requirements of well above 8K UHD resolution at real-time. It should be noted however that there are many other factors and components that may affect the speed of this DCT when integrated into a complete video codec.

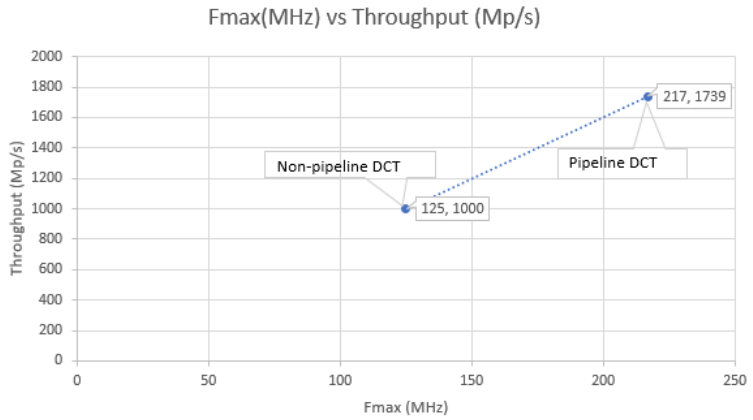

Figure 5. Maximum frequency (Fmax) vs Throughput in Mpixels/s

Table 1 shows the comparison to three similar works in literature: Madanayake et al [6] with 45nm CMOS, Wahid et al [7] with 0.18um CMOS, and Fu et al [8] with 0.18 um CMOS technologies. In terms of throughput, It can be seen that our designs are superior to [7, 8], but trails the design from [6]. This work shows higher performance due to the smaller feature technology used. However, our design has shown to meet the requirements of state-of-the-art video coding resolution even when using a larger technology.

Table 1. Comparison with similar works in literature

	Madanyake et al [6]	Wahid et al [7]	Fu et al [8]	Non-pipeline DCT	Pipeline DCT
CMOS Technology (nm)	45	180	180	130	130
Operating frequency (MHz)	946	194.7	75	116	250
throughput (Mp/s)	7568	194.7	75	1000	1739

5. CONCLUSION

The objective of the present paper is to present results for implementing a fast VLSI pipelined multiplierless fixed-point 8x8 DCT. The algorithm used is the ISO/IEC 23002-2 for the image and video compression technology. The design methodology begins with modeling the algorithm using dataflow graphs in order to analyze for pipelining. The results show that the implementations are feasible to be applied in the latest ultra-high definition 8K video, even when using the relatively large 0.13um technology. We have implemented and compared the DCT using two architectures, non-pipeline and a 2-stage pipeline. Simulation results validated the expectation that throughput can be improved by almost a factor of two, from around 1 Giga pixels/s to 1.8 Giga pixels/s, at a small cost of roughly 14% more resource (i.e. pipeline registers). Future work aims to extend the methodology for implementing different DCT dimensions for other applications.

ACKNOWLEDGEMENT

The authors would like to thank the Malaysia Ministry of Education for providing the funds for the work in this paper (Vote no. 4F659).

REFERENCES

[1] V. Sze, M. Budagavi, G.J. Sullivan, "High Efficiency Video Coding," *Springer*, 2014.
[2] Heyne, B. Gotze, J., A Low Power and High Quality Implementation of The Discrete Cosine Transformation," *Advances in Radio Science*, 2007.
[3] U. S. Potluri, A. Madanayake, R. J. Cintra, F. M. Bayer, S. Kulasekera, and A. Edirisuriya, Improved 8-point approximate DCT for image and video compression requiring only 14 additions,*IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 6, no. 6, pp. 1727-1740, June 2014.
[4] ISO/IEC, Information technology - MPEG video technologies - Part 2: Fixed-point 8x8 inverse discrete cosine transform and discrete cosine transform, *International Standard*, Dec 2014.
[5] A. Madanayake et al., "Low-Power VLSI Architectures for DCT/DWT: Precision vs Approximation for HD Video, Biomedical, and Smart Antenna Applications," *IEEE Circuits and Systems Magazine*, vol. 15, no. 1, pp. 25-47, 2015.
[6] A. Edirisuriya, A. Madanayake, R. Cintra, V. Dimitrov and N. Rajapaksha, "A single-channel architecture for algebraic integer-based 8x8 2-D DCT computation", *IEEE Transactions on Circuits Systems for Video Technology*, vol. 23, no. 12, pp. 2083-2089, Dec. 2013.
[7] K. A. Wahid, M. Martuza, M. Das and C. McCrosky, "Efficient hardware implementation of 8x8 integer cosine transforms for multiple video codecs", *Journal of Real-Time Processing*, pp. 1-8, July 2011.
[8] M. Fu, G. A. Jullien, V. S. Dimitrov and M. Ahmadi, "A low-power DCT IP core based on 2D algebraic integer encoding", *Proceedings of the International Symposium on Circuits Systems (ISCAS)*, vol. 2, pp. 765-768.
[9] A. A. H. Ab-Rahman and I. Kamisian and A. Z. Sha'ameri, VLSI design and implementation of adaptive channel equalizer, *2008 International Conference on Computer and Communication Engineering*, pp. 1121-1124, May 2008.
[10] Rachmad Vidya Wicaksana Putra, Rella Mareta, Nurfitri Anbarsanti , Trio Adiono A New RTL Design Approach for a DCT/IDCT-Based Image Compression Architecture using the mCBE Algorithm, *Journal of ICT Research and Applications*, Vol. 6, No. 2, pp 131-150, 2012
[11] A. Prihozhy and E. Bezati and A. A. H. Ab Rahman and M. Mattavelli Synthesis and Optimization of Pipelines for HW Implementations of Dataflow Programs, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 34, no. 10, pp 1613-1626, 2015.