High-Throughput FPGA Implementation of Sliding DCT using Look-Ahead Pipelining

Geonu Kim[1] and Yong-Ho Cho[a]
Department of Information and Communications Engineering, Mokpo National University
E-mail: [1]geonukim@mnu.ac.kr

***Abstract* – The Discrete Cosine Transform (DCT) is often applied in a sliding window setting. While sliding DCT (SDCT) algorithms take advantage of the computational redundancy inherent from the overlap between consequent processing windows, the recursive computation nature imposes a fundamental limit on realizable throughput. This study presents a high-throughput FPGA implementation of the SDCT algorithm using look-ahead pipelining. A look-ahead transformation of degree two is applied to the original SDCT recursion formula resulting in a threefold reduction of the iteration bound. The resulting maximum clock frequency on a Xilinx FPGA device is increased 2.1x (from 58.4 MHz to 123.4 MHz) by carefully retiming the look-ahead pipeline registers and inserting additional pipeline registers in feed-forward cutsets of the signal-flow graph.**

***Keywords*—Discrete cosine transform, Field programmable gate array, Look-ahead pipelining**

I. INTRODUCTION

The Discrete Cosine Transform (DCT) is one of the most widely used digital signal processing technique with many useful properties such as self-orthogonalization and energy compaction, which follows from the fact that DCT closely approximates the optimal Karhunen-Loeve transform [1]. Furthermore, its data independent nature has led DCT to be utilized in various fields including audio signal processing, image processing, spectral analysis, and adaptive signal processing [2]-[4].

DCT is often applied in a *sliding window* setting like other signal processing algorithms [5]. Some examples of sliding DCT (SDCT) include spectral analysis of non-stationary data and transform domain adaptive filtering [4]. In sliding window processing, a window of fixed size is shifted one sample at a time over the input domain sequence $x(n)$ as shown in Fig. 1. While fast DCT algorithms of complexity $O(N\log N)$ [6] that are similar to the fast Fourier transform exist, they do not take advantage of the inherent redundancy due to data overlap between consequent processing windows. The SDCT algorithm [4] utilizing this inter-window dependency results in recursive computations and is usually more efficient [7]-[10].

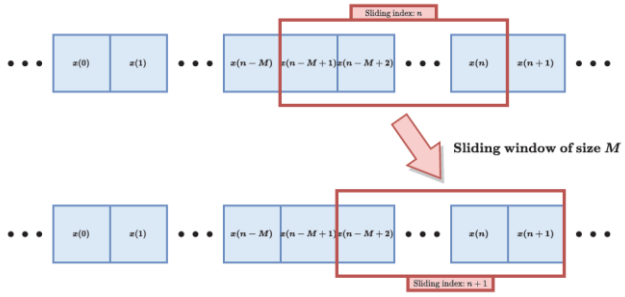

Fig. 1. Sliding window processing.

An important drawback of the SDCT algorithm as well as other recursive algorithms is that the inherent *iteration bound* [11] prevents arbitrary pipelining and restricts high throughput implementations for single data-stream applications. However, this limit can be overcome by using the *look-ahead transformation* technique that has been used in implementing infinite impulse response (IIR) digital filters [11]. This approach is orthogonal to the various formulations found in the literature targeting reduced computation [7]-[10].

In this paper, we present a high-throughput FPGA implementation of the SDCT algorithm using look-ahead pipelining. In section II, the look-ahead transformed SDCT algorithm is shown after the original SDCT algorithm, together with various design details. FPGA synthesis results and analytical discussions are given in section III, which is followed by concluding remarks in section IV.

II. DESIGN METHODOLOGY

A. Sliding DCT algorithm

M-point DCT is defined as follows [4]:

$$X_m = k_m \sum_{i=0}^{M-1} x(i) \cos\left(\frac{m(i+1/2)\pi}{M}\right) \qquad (1)$$

for $m = 0, \ldots, M-1$, where the constant

a. Corresponding author; yonghocho@mnu.ac.kr

Manuscript Received Oct. 11, 2022, Revised Dec. 19, 2022, Accepted Dec. 19, 2022

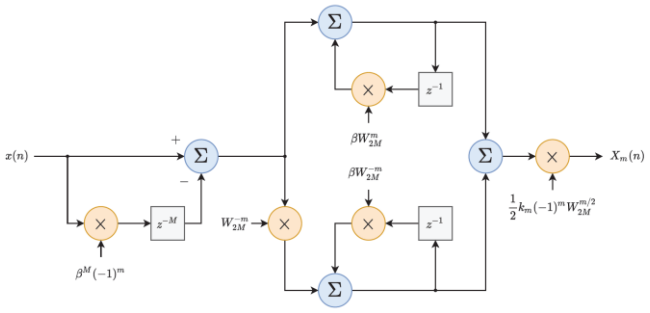

Fig. 2. Block diagram of the SDCT algorithm.

$$k_m = \begin{cases} 1/\sqrt{2}, & m = 0 \\ 1, & otherwise \end{cases}. \tag{2}$$

This particular form is referred to as DCT-II, and is the most commonly used DCT type among several variants.

In the sliding window setting, the SDCT sequence corresponding to the M-point input domain sequence $[x(n), x(n-1), \ldots, x(n-M+1)]$ becomes

$$X_m(n) = k_m \sum_{i=n-M+1}^{n} x(i) \cos\left(\frac{m(i-n+M-1/2)\pi}{M}\right). \tag{3}$$

After some manipulation [4], we get

$$X_m(n) = \frac{1}{2} k_m (-1)^m W_{2M}^{m/2} P_m(n) \tag{4}$$

where the constant

$$W_{2M} = \exp\left(-\frac{j2\pi}{2M}\right), \tag{5}$$

and

$$P_m(n) = P_m^{(1)}(n) + P_m^{(2)}(n) \tag{6}$$

with the following recursions:

$$\begin{aligned} P_m^{(1)}(n) &= W_{2M}^{m} P_m^{(1)}(n-1) \\ &\quad + x(n) - (-1)^m x(n-M), \end{aligned} \tag{7}$$

$$\begin{aligned} P_m^{(2)}(n) &= W_{2M}^{-m} P_m^{(2)}(n-1) \\ &\quad + W_{2M}^{-m}\left(x(n) - (-1)^m x(n-M)\right). \end{aligned} \tag{8}$$

Fig. 2 denotes the block diagram representation of the SDCT algorithm. Note that a multiplicative factor β is applied to each z^{-1} operation [4]. The raw SDCT with $\beta = 1$ is marginally stable due to the position of its poles exactly on the unit circle in the z-plane. Round-off errors may cause instability by pushing the poles outside the unit circle. Choosing a suitable $\beta < 1$ can alleviate this problem by shifting the poles slightly inside the unit circle, with small approximation errors. Equation (7) and (8) are modified accordingly as follows:

$$\begin{aligned} P_m^{(1)}(n) &= \beta W_{2M}^{m} P_m^{(1)}(n-1) \\ &\quad + x(n) - \beta^M (-1)^m x(n-M), \end{aligned} \tag{9}$$

$$\begin{aligned} P_m^{(2)}(n) &= \beta W_{2M}^{-m} P_m^{(2)}(n-1) \\ &\quad + W_{2M}^{-m}\left(x(n) - \beta^M (-1)^m x(n-M)\right). \end{aligned} \tag{10}$$

Note that the left-front part of Fig. 2 computing the common $x(n) - (-1)^m x(n-M)$ expression can be shared within the groups of even and odd values of the transform domain index m.

The iteration bound of the SDCT algorithm is easily identified to be the sum of one (real) multiplication delay and two addition delays, since the computational path through complex multiplication consists of one multiplication and one addition. Throughput or sample rate higher than this iteration bound is infeasible no matter what amount of pipelining is applied.

B. Look-ahead transformation

The look-ahead transformation technique [11] has been used in implementing IIR digital filters to reduce their iteration bound. It can be applied to SDCT as well since SDCT shares the recursive computation structure with IIR filters.

The recursion formulas (9) and (10) are expanded two times as follows:

$$\begin{aligned} P_m^{(1)}(n) &= \beta^3 W_{2M}^{3m} P_m^{(1)}(n-3) \\ &\quad + x(n) - \beta^M (-1)^m x(n-M) \\ &\quad + \beta W_{2M}^{m}\left(x(n-1) - \beta^M (-1)^m x(n-M-1)\right) \\ &\quad + \beta^2 W_{2M}^{2m}\left(x(n-2) - \beta^M (-1)^m x(n-M-2)\right), \end{aligned} \tag{11}$$

$$\begin{aligned} P_m^{(2)}(n) &= \beta^3 W_{2M}^{-3m} P_m^{(2)}(n-3) \\ &\quad + W_{2M}^{-m}\left(x(n) - \beta^M (-1)^m x(n-M)\right) \\ &\quad + \beta W_{2M}^{-2m}\left(x(n-1) - \beta^M (-1)^m x(n-M-1)\right) \\ &\quad + \beta^2 W_{2M}^{-3m}\left(x(n-2) - \beta^M (-1)^m x(n-M-2)\right). \end{aligned} \tag{12}$$

Comparing the resulting block diagram shown in Fig. 3 to the original block diagram in Fig. 2, it is easily seen that the number of delay elements in the critical loop has increased from one to three, which is due to the two-time expansion applied in the recurrence relations (11) and (12). Since the combinational computation delay along the critical loop remains the same, the resulting iteration bound decreases threefold, hence a three-time increase of potential throughput.

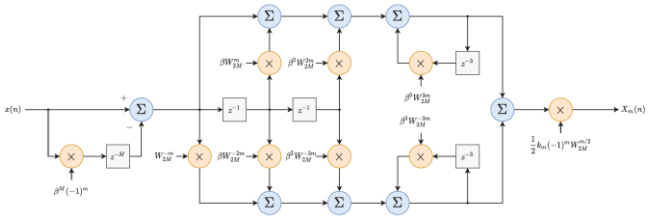

Fig. 3. Block diagram of the SDCT algorithm with look-ahead transformation of degree two.

C. Design details

A pipeline architecture of the SDCT algorithm with look-ahead transformation has been designed and is shown in Fig. 4. A total of eight pipeline stages are introduced to enjoy the reduced iteration bound and achieve high throughput. Four pipeline cuts are inserted each in the upstream and downstream of the critical loop. Note that two among the three delay elements in the critical loop are retimed in the forward direction. Furthermore, redundant delay elements are merged into the delay elements with identical input-output connections.

While two delay elements are shown together after the fully complex multiplication operations in Fig. 4, one is actually placed inside the complex multiplication—between the real multiplication and addition. Overall, to achieve high throughput, every pipeline delay element is placed such that no more than a single real arithmetic operation is performed between each connected delay element pair.

Fixed-point bit widths for every signal are also indicated in the block diagram using the Q notation. For example, Q7.8 means a total bit width of 15 with seven bits for the integer part including the sign bit, and eight bits for the fraction part. The fixed-point configuration inside the complex multiplier, between the real multiplier and adder, simply follows that of the complex multiplier output.

Saturation and rounding are applied whenever required. Saturation arithmetic is chosen since wrapping around for overflows and underflows usually results in notable performance degradation in many communication and signal processing applications. Rather than simple truncation, convergent rounding is performed to suppress quantization noise bias; numbers are rounded to the closest representable number with precedence of even numbers in the case of tie. Further fine grain retiming is performed to minimize the negative impact of saturation and rounding on the critical path delay.

III. RESULTS AND DISCUSSIONS

The designed SDCT look-ahead pipeline architecture has been implemented on a Xilinx Artix-7 FPGA device, xc7a200tlfbg484-2L, using Vivado software. Only synthesis is performed since the designed SDCT is a subblock of a larger system with a more IO ports than the target device pins.

For comparison purposes, a conventional SDCT pipeline architecture has also been designed, which is shown in Fig. 5. Two pipeline stages are introduced to improve throughput and reveal the computational loop that determines the iteration bound as the critical path of the design.

Synthesis timing results, denoted in Table I, show that the critical path delay is reduced from 17.2 ns to 8.1 ns, and the maximum clock frequency increases accordingly from 58.3 MHz to 123.4 MHz. The aforementioned potential threefold improvement in the iteration bound is only possible for both zero clock-to-q delay and setup time. Considering the actual nonzero clock-to-q delay and setup time, together with the practical constraints in placing the pipeline registers, the resulting 2.1x throughput improvement is considered quite reasonable.

FPGA resource allocation results are denoted in Table II. Overhead incurred by look-ahead pipelining can be seen. The number of slice registers increases significantly. In addition to the higher degree of pipelining, this is due to fine grain retiming of the pipeline registers after real

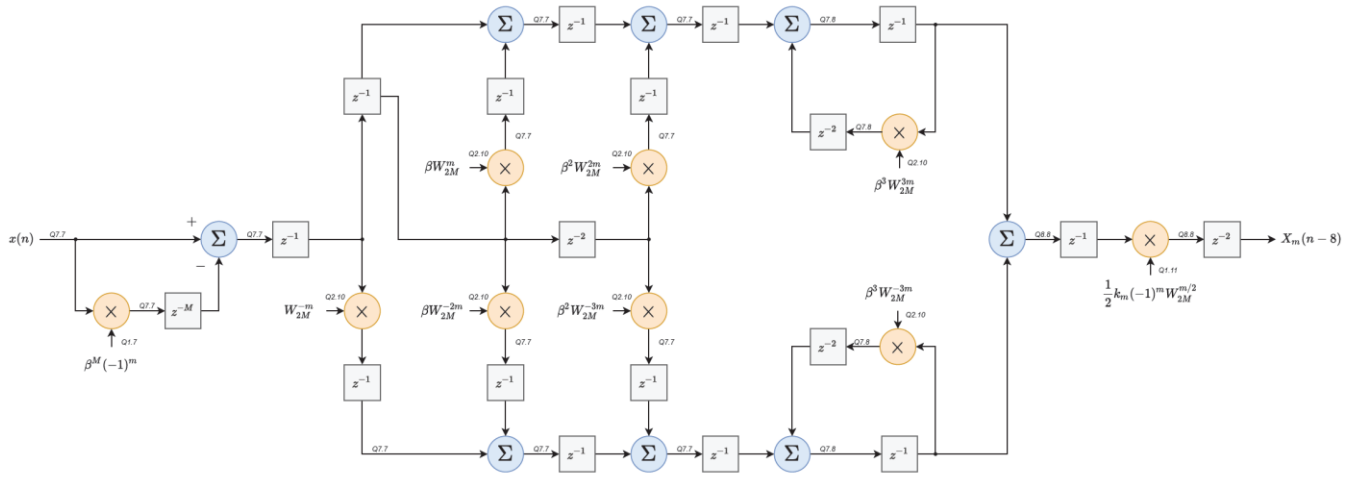

Fig. 4. Block diagram of the designed SDCT architecture with look-ahead transformation; pipelining and retiming also applied.

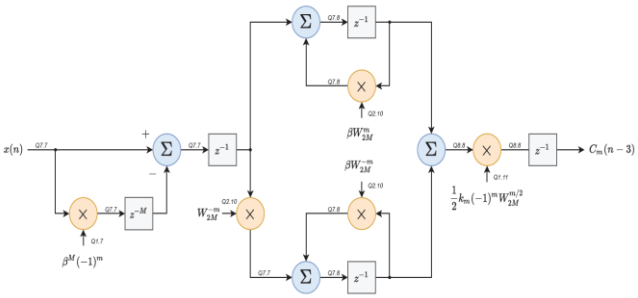

Fig. 6. Block diagram of the conventional SDCT pipeline architecture.

Table I. Critical path delay and throughput comparison.

	Conventional SDCT pipeline	SDCT with look-ahead pipeline
Critical path delay	17.2 ns	8.1 ns
Throughput	58.3 MHz	123.4 MHz

Table II. Allocated FPGA resource comparison

	Conventional SDCT pipeline	SDCT with look-ahead pipeline
Slice LUTs	15380	22653
Slice registers	1907	18350
DSPs	244	268

multiplications. These registers have been placed in front of the convergent rounding operations auxiliary to the multiplication operations—where bit widths are large—to maximally balance the delay of register-to-register paths. The numbers of slice look-up tables (LUTs) and digital signal processing (DSP) blocks, however, increase only moderately. Although the number of multiplication operations incurred by look-ahead transformation is large, each such multiplier in Fig. 4 has one multiplicand constant and the other multiplicand common with a large number of other multipliers. This computational structure is identical to the multiplier block structure of transposed form finite impulse response (FIR) filters, which is highly redundant and can be significantly optimized by FIR filter synthesis techniques [12]. It is supposed that Vivado software has taken advantage of this optimization opportunity.

As each logic slice of an Artix-7 device contains four LUTs and eight flip-flops [13], the required logic slice counts of the conventional and proposed design are 3845 and 5664, respectively. Considering only these logic slices, together with the critical path delays, the area-delay product (ADP) shows a 1.44x improvement. Since the ADP improvement considering only DSP slices is 1.93x, the total ADP improvement can be determined as a number between these two, depending on the area ratio of logic and DSP slices.

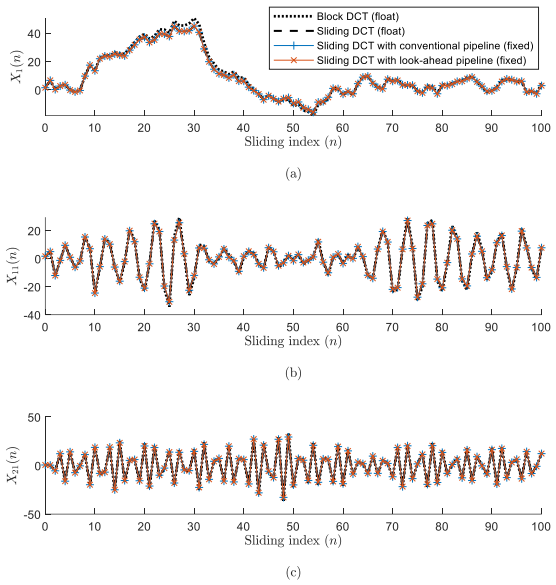

Fig. 7. DCT simulation results for white Gaussian test input; (a) $\boldsymbol{X_1(n)}$ (b) $\boldsymbol{X_{11}(n)}$ (c) $\boldsymbol{X_{21}(n)}$.

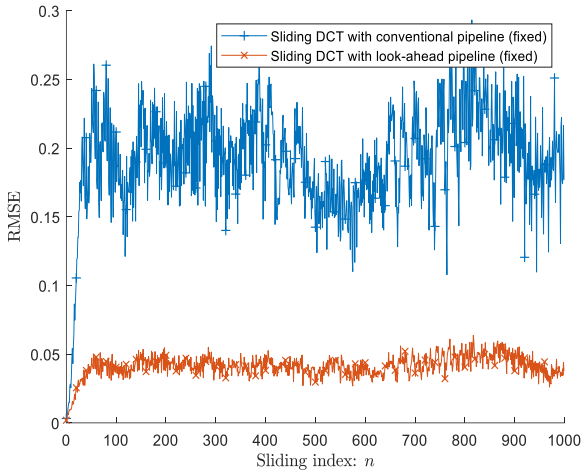

Fig. 5. Fixed-point RMSE of conventional SDCT and look-ahead transformed SDCT.

Fig. 6 shows some simulation results on white Gaussian test input data. Block DCT refers to DCT processing according to (1), i.e., without sliding window recursions. This simulation is done in floating-point MATLAB, in order to be used as golden reference. HDL fixed-point simulation of the designed SDCT look-ahead pipeline architecture as well as fixed and floating-point conventional SDCT simulations are well verified to be correct.

Comparison of fixed-point accuracy of the conventional SDCT and the look-ahead transformed SDCT architectures is denoted in Fig. 7. Root-mean-square error (RMSE) relative to floating-point SDCT is calculated, with mean over all transform domain indexes. Interestingly, the look-ahead transformed SDCT shows much lower RMSE than the conventional SDCT. This RMSE margin can be used to reduce the required FPGA resource of the look-ahead SDCT

pipeline architecture by reducing fixed-point bit widths, or to improve throughput even further by employing a simpler rounding scheme such as rounding half up toward positive infinity.

IV. CONCLUSION

In this study, a high-throughput FPGA implementation of the SDCT algorithm using look-ahead transformation has been demonstrated. The proposed design achieves a maximum clock frequency improvement by 2.1x, from 58.4 MHz to 123.4 MHz, compared to the conventional SDCT pipeline architecture, which is reasonably close to the theoretical threefold reduction of the iteration bound. Various design details including fixed-point configuration and simulation, pipeline retiming, and FPGA synthesis results have been presented, together with detailed analytical discussions.

ACKNOWLEDGMENT

EDA tools were supported by the IC Design Education Center (IDEC), Korea.

REFERENCES

[1] N. Ahmed, T. Natarajan and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. C-23, no. 1, pp. 90-93, Jan. 1974.

[2] H. Ochoa-Dominguez and K. R. Rao, *Discrete Cosine Transform*, CRC Press, 2019.

[3] H. B. Ha, S. U. Park, W. Liu, and Y. B. Cho, "MV–HEVC chip design using high level synthesis," *Journal of Integrated Circuits and Systems*, vol. 6, no. 4, Oct. 2020.

[4] S. Haykin, *Adaptive Filter Theory*, Pearson, 2013.

[5] A. V. Oppenheim and R. W. Schafer, *Discrete-time Signal Processing*, Pearson, 2009

[6] M. Puschel and J. M. F. Moura, "Algebraic signal processing theory: Cooley–Tukey type algorithms for DCTs and DSTs," *IEEE Trans. Signal Process.*, vol. 56, no. 4, pp. 1502-1521, Apr. 2008.

[7] J. Xi and J. F. Chiraro, "Computing running DCT's and DST's based on their second-order shift properties," *IEEE Trans. Circuits. Syst. I*, vol. 47, no 5, pp. 779-783, May 2000.

[8] V. Kober, "Fast algorithms for the computation of sliding discrete sinusoidal transforms," *IEEE Trans. Signal Process.*, vol. 52, no. 6, pp. 1704-1710, Jun. 2004.

[9] Q. Li, L. Shouhua, H. Siyuan, and C. Gong, "Recursive algorithms for direct computation of generalized sliding discrete cosine transforms," *Proc. 3rd Int'l. Congress on Image and Signal Process*., pp. 3017-3020, Oct. 2010.

[10] V. Kober, "Recursive Algorithms for Computing Sliding DCT With Arbitrary Step," *IEEE Sensors*, vol. 21, no. 10, pp. 11507-11513, May 2021.

[11] K. K. Parhi, *VLSI Digital Signal Processing Systems*, Wiley-Interscience, 1999.

[12] H.-J. Kang and I.-C. Park, "FIR filter synthesis algorithms for minimizing the delay and the number of adders," *IEEE Trans. Circuits Syst. II Analog Digit. Signal Process.*, vol. 48, no. 8, pp. 770-777, Aug. 2001.

[13] Xilinx, "7 Series FPGAs Data Sheet: Overview," DS180 (v2.6.1), Sep. 2020. [Online]. Available: https://docs.xilinx.com/v/u/en-US/ds180_7Series_Overview

Geonu Kim received the B.S. and M.S. degrees in Electrical Engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea in 2004 and 2007, and the Ph.D. degree in Electrical Engineering and Computer Science from Seoul National University, Seoul, Korea in 2017. In 2007, he joined SK Hynix, Icheon, Korea, where he has worked in the NAND Technology Development Division as a VLSI design engineer. Since 2020, he has been with the department of Information and Communications Engineering, Mokpo National University, Jeonnam, Korea as an Assistant Professor. His research interests include codes for distributed storage, VLSI signal processing, and, error correction and signal processing algorithms for NAND Flash memory.

Yong Ho Cho received the B.S. degree in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea, in 2004, and the M.S. and Ph.D. degrees in electrical engineering from KAIST, in 2006 and 2013, respectively. From 2013 to 2016, he was a senior researcher with Samsung Electronics in charge of research and development for 5G and IoT communication systems. He was an Assistant Professor with the Department of Information and Communication Engineering, Hoseo University, from 2016 to 2021. He is currently an Associate Professor in the Department of Electronics, Information and Communication Engineering, Mokpo National University. His research interests include 5G and 6G mobile communication systems, the Internet of Things, underwater communication systems and deep learning.