LETTER

A 1.05-V 62-MHz with 0.12-nW standby power SOTB-65 nm chip of 32-point DCT based on adaptive CORDIC

Duc-Hung Le¹a, Trong-Thuc Hoang²b, and Cong-Kha Pham²c)

Abstract In this paper, a Silicon On Thin Buried-oxide (SOTB) implementation of a 32-point Discrete Cosine Transform (DCT) is presented. The architecture is based on the fixed-rotation adaptive COordinate Rotation Digital Computer (CORDIC) algorithm. The SOTB-65 nm process was chosen due to the profound advantages of low-power and high-performance. The core layout contained about 47.2 K gate-count and had the size of about 183 K-µm². The measurement results showed that the highest operating frequency of 62-MHz was achieved at the 1.05-V power supply and consumed about 737-µW and 11.89-pJ/cycle. In the standby mode, the least power consumption of 0.12-nW was achieved at the 0.4-V power supply when the clock-gating technique and reverse back-gate biasing were applied.

Keywords: CORDIC, DCT, low-energy, low-power, SOTB

Classification: Integrated circuits

1. Introduction

Discrete Cosine Transform (DCT) is irreplaceable nowadays, especially in image and video processing applications [1]. For instance, the DCT is required among the most common compression algorithms like JPEG [2], H.263 [3], H.264 [4], and H.265 [5]. Moreover, the high-speed requirement of DCT implementation was becoming more and more necessary in the emerging trend of development. Thus, the research of DCT still plays a vital role in the field of image and video processing.

The performances of a DCT implementation are mainly based on its Signal-Flow-Graph (SFG). The earliest SFGs that had been proposed were the Chen’s in 1977 [6] and Loeffler’s in 1989 [7]. Then from times to times, there were plenty of improved SFGs developed based on the two conventional ones. The main difference between them is that the two classic models [6, 7] utilized the real-value multipliers while the modern ones [8, 9, 10, 12, 13] were usually based on a multiplier-less approach. The well-known works were the BinDCT of J. Liang and T. D. Tran in 2001 [8], approximate DCT of M. Parfieniuk et al. in 2013 [9], integer DCT of P. K. Meher et al. in 2014 [10], scalable orthogonal approximation DCT of M. Jridi et al. in 2015 [11], adaptive approximate DCT of M. Masera et al. in 2016 [12], and optimized real-valued DCT of S. Chatterjee and K. Sarawadekar in 2018 [13]. Besides the above methods, the COordinate Rotation Digital Computer (CORDIC) approach was also among the well-known ways to create a high-speed DCT SFG, and several CORDIC-based DCT implementations had been proposed lately [14, 15, 16, 17]. The idea of Adaptive CORDIC (ACor) was first introduced by Y. H. Hu and S. Naganathan in 1993 [18] to reduce the number of iteration step. Later, various improvements of the ACor implementation had been developed through the years [19, 20, 21, 22]. The latest improvement of the ACor was the work of Hong-Thu Nguyen et al. in 2015 [23], and the DCT implementation that based on it was also introduced in 2018 [24].

In this paper, the measurement result of the 65-nm Silicon On Thin Buried-oxide (SOTB) chip of the 32-point DCT architecture based on Adaptive CORDIC (ACor) is presented. The ACor-based DCT architecture was proposed in the previous work [24] in 2018, and it was proved to have the minimum adder-delay while providing good accuracy outcomes with a similar resources cost in comparison with other works. The SOTB process was chosen because it is on the development trend of low-power Complementary Metal Oxide Semiconductor (CMOS) device; the prominent advantage of the SOTB process was the ultra-low-power aspect [25, 26].

The remainder of this paper is organized as follows. Section 2 describes the fixed-rotation ACor algorithm. Section 3 presents the 32-point DCT architecture. Section 4 shows the measurement results of the SOTB-65 nm chip. Finally, Section 5 gives the conclusion.

2. The adaptive CORDIC algorithm

2.1 The algorithm

The Adaptive CORDIC (ACor) algorithm [23] was proven to have the adequate resources cost while producing high-accuracy results with a few latencies. The key idea is the decision-making step to select only a few micro-rotations instead of all micro-rotations as in the conventional CORDIC. By doing this, the ACor method significantly reduces the iterations number while maintains similar accuracy performances.

Fig. 1 gives the pseudo-code of the ACor method [23]. The algorithm operation is an iteration function revolves around the zj-value with the goal of the zj-value to con-
verge to zero quickly. To do that, an appropriate micro-rotation \( \theta_i = \tan^{-1}(2^{-i}) \) is selected by each step as can be seen in the pseudo-code. To choose the best micro-rotation \( \theta_i \) in each step, the decision-making part of the algorithm utilized the concept of \( c_i = (\theta_i + \theta_{i-1})/2 \) to represent for the range around the \( \theta_i \)-value; except for \( c_0 = \theta_0/2 \).

In the ACor algorithm, after the iteration process, the outputs of \((x, y)\) were unintentionally increased by a factor in comparison with the corrected values. Therefore, in the pseudo-code in Fig. 1, the final outputs of \((X, Y)\) are the results of the multiplications between the \((x, y)\) values and the length-factor \( K \). The \( K \)-value is the product of \( k_i \)-value as seen in the pseudo-code, where \( k_i = 1/\sqrt{1 + \tan^2 \theta_i} \).

To decrease the iterations number further, on the unit circle, the ACor process only applies to the angle of \( [0^\circ, 45^\circ] \). Outside of the \([0^\circ, 45^\circ]\) range, simple trigonometric functions were applied for the conversion. This technique not only reduces the iteration number but also increases the accuracy outcomes.

2.2 The fixed-rotation implementation
The fixed-rotation ACor happened when the input angle of \( z_0 \) is known. Fixed-rotation ACor implementations were common in CORDIC-based DCT SFGs [14, 15, 24]. With the known \( z_0 \)-value, the iteration process of the \( z_j \) becomes a pre-determination process. Hence, the two values sets of \( z_j \) and \( \theta_j \) become constants. Because of this, the number of iteration steps and the length-factor \( K \) also become fixed numbers. As a result, a fixed-rotation ACor implementation has many optimizations in comparison with the variable-rotation ACor design. For example, if the \( z_0 \)-value and the \( N \)-value respectively equal to \( 3\pi/8 \) and \( 16 \), then, according to the pseudo-code in Fig. 1, the \( 3\pi/8 \)-ACor computation process will contain six \( \theta_i \)-value of \( \{\theta_1 - \theta_4 - \theta_7 - \theta_{10} + \theta_{11} + \theta_{12}\} \). Moreover, to reducing the iteration steps further without losing too much of the accuracy performance, the above six \( \theta_i \)-value can be dropped to the first two or three \( \theta_i \)-value only.

\[
i = j = 0; K = 1; x_0 = 1; y_0 = 0; z_0 = \text{input angle};
\]
\[
\text{while } |z_j| > c_{N-1} \text{ and } i < N \text { do }
\]
\[
\text{if } |z_j| \in (c_{i+1}, c_i) \text{ then }
\]
\[
z_{j+1} = z_j - \text{sign}(z_j) \times \theta_i;
\]
\[
x_{j+1} = x_j - \text{sign}(x_j) \times y_j \times 2^{-i};
\]
\[
y_{j+1} = y_j + \text{sign}(y_j) \times x_j \times 2^{-i};
\]
\[
K = K \times k_i; j = j + 1;
\]
\[
\text{end if; } i = i + 1
\]
\[
\text{end while; } X = x_j \times K; Y = y_j \times K;
\]

Fig. 1. The pseudo-code of the ACor algorithm [23].

3. The ACor-based 32-point DCT architecture

3.1 Notations
3.1.1 Twisted-Adder (TA)
A TA is defined as a series of adders/subtractors where the top-half values are twisted and added/subtracted with the bottom-half values. Fig. 2 gives the TA examples of 2-point, 4-point, and 8-point. An adder/subtractor is represented by a dot. A straight line and a dashed line represent for the multiplications with 1 and \(-1\), respectively.

3.1.2 ACor series
is a series of fixed-rotation ACor modules. The \( C^k_n \) notation represents for the \( a/b \)-ACor module, then the notation of \( \hat{C}^{2k+1}_{2k+1} \) is used to describe the ACor series of \( \{C^1_n, C^3_n, \ldots, C^{2k+1}_n\} \). Fig. 3 gives the three ACor series that used in the 8-point, 16-point, and 32-point DCT SFGs.

3.2 The architecture
Fig. 4 gives the SFG of the ACor-based 32-point DCT. In the figure, the SFG used the six submodules of \( M1 \) to \( M6 \) which are described in Fig. 5. Moreover, the SFG of the 16-point DCT used in Fig. 4 is given by Fig. 6. Finally, the 8-point DCT SFG used in Fig. 6 is shown in Fig. 7.

As proven in 2018 [24], when comparing the two ACor settings of 2S and 3S, the 3S-ACor-based 32-point DCT (3S-32p-DCT) had a little advantage in precision, but the cost of adder-delay and number of adders had made the trade-off not worthed. Therefore, the 2S-ACor of 32-point DCT (2S-32p-DCT) was the best design for further implementation.
Fig. 4. The ACor-based 32-point DCT SFG.

Fig. 5. The six submodules used in the ACor-based 32-point DCT SFG.
4. Performance analysis

4.1 Synthesis results comparison

The 2S-ACor-based 32-point DCT (2S-32p-DCT) architecture was implemented with the SOTB-65 nm process library. Table I gives the synthesis result of the core in comparison with other works.

To make a fair comparison between implementations with different technologies, the gate-count number was selected to represent for the area cost of each implementation. The gate-count value is calculated by dividing the total area of the core with the area of the basic NAND2 in its corresponding process library. As seen in Table I, the proposed DCT implementation had the smallest gate-count number of 47.2 K and consumed the least power of 0.68 mW. The result of low power consumption reflects the advantage of the SOTB process over other processes. Therefore, due to the substantial advantage of ultra-low power, the energy result of the core required only 6.82-pJ/cycle, the lowest number in the comparison table.

For the timing performance, because different designs can result in different latencies, therefore the number of processed pixels per cycle was also reported besides the FMax value. According to Table I, although the proposed design achieved only 100-MHz of FMax, its throughput result reached to 100 Mega-Sample-per-second (MSps) due to the parallel design that can process 32-pixel/cycle; each sample is a group of 32-point DCT.

![Fig. 6. The ACor-based 16-point DCT SFG.](image1)

![Fig. 7. The ACor-based 8-point DCT SFG.](image2)

![Fig. 8. The die chip and its features.](image3)

Table I. Synthesis result in comparison with other works.

| Design   | [27] (2012) | [28] (2012) | [29] (2013) | [10] (2014) | [12] (2016) | [30] (2016) | [13] (2018) | 2S-32p DCT |
|----------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|------------|
| Technology | Bulk-180 nm | Bulk-130 nm | Bulk-90 nm  | Bulk-90 nm  | Bulk-90 nm  | SOTB-65 nm  | SOTB-65 nm  |            |
| VDD      | 1.8-V       | 1.2-V       | 1.1-V       | 0.75-V      |             |             |             |            |
| Gate-count | 52.3 K      | 109.2 K     | 63.8 K      | 103 K       | 163 K       | 115.7 K     | 88.6 K      | 47.2 K     |
| FMax (MHz) | 300         | 350         | 270         | 187         | 250         | 200         | 256.4       | 100        |
| Pixel/cycle | 2.12        | 4           | 8           | 16          | 25.7        | 32          | 32          | 32         |
| Throughput (MSps) | 19.88       | 43.75       | 67.5        | 93.5        | 200.78      | 200         | 256.4       | 100        |
| FoM      | 2,631       | 2,496       | 945         | 1,102       | 812         | 579         | 346         | 472        |
| Power (mW) at 100-MHz | -           | -           | -           | 17.63       | -           | -           | -           | -          |
| Energy (pJ/cycle) | -           | -           | -           | 176.3       | -           | -           | 162         | 6.82       |
Furthermore, because different technologies can affect the outcome of \( F_{\text{Max}} \), an overall efficiency Figure-of-Merit (i.e., \( \text{FoM} = \text{Gate-count} / \text{Throughput} \)) was proposed in this paper, and it can be seen as an equivalent to the area-time product. As seen in Table I, the proposed 2S-32p-DCT implementation can achieve a competitive result of 472 FoM. Although that result was still larger than the result [13] in 2018 by about 1.36x, the energy consumption was astonishingly better (i.e., 6.82-pJ/cycle versus 162-pJ/cycle).

4.2 Chip measurement results

The 2S-32p-DCT was fabricated in SOTB-65 nm process. Fig. 8 shows the chip micrograph and the summary result. As shown in Fig. 8, the core chip \( V_{\text{DD}} \) ran at the range of 0.4-V to 1.2-V while its package I/O could operate under various \( V_{\text{DD}} \) settings. A source meter was used to supply the required voltages, and also to measure the active and leakage currents. The clock frequency was provided by an Field-Programmable Gate Array (FPGA) board. The FPGA board also generated the input data and verified the output data of the core.

According to the summary result in Fig. 8, the core reached its highest operating frequency of 62-MHz at 1.05-V \( V_{\text{DD}} \) with approximated 737-\( \mu \text{W} \) power consumption. From 1.1-V to 1.2-V \( V_{\text{DD}} \), the core kept the same value of 62-MHz \( F_{\text{Max}} \) but consumed more power. At the lowest operating \( V_{\text{DD}} \) of 0.4-V, the DCT core can run at 10-MHz with about 17.48-\( \mu \text{W} \) power consumption. At the standby mode when the Clock-Gating (CG) technique was deployed, the power consumption was about 708-nW at 0.4-V \( V_{\text{DD}} \). Furthermore, if the Reverse Back-gate Biasing (RBB) was applied with the voltage of \(-2.5\text{-V} V_{\text{BB}}\), the power consumption can significantly be reduced about 5,900x to 0.12-nW.

Fig. 9 shows the variation of maximum frequency by the changes in the supply voltage. If the 2.5-V Forward Back-gate Biasing (FBB) was applied (i.e., \( V_{\text{BB}} = 2.5\text{-V} \)), the \( F_{\text{Max}} \) and \( P_{\text{active}} \) values at the 0.4-V \( V_{\text{DD}} \) increased about 1.9x and 7.68x to 19-MHz and 134.4-\( \mu \text{W} \), respectively. At 1.05-V \( V_{\text{DD}} \) with 2.5-V FBB, the \( F_{\text{Max}} \) increased only a little bit to 63-MHz, while the \( P_{\text{active}} \) increased about 2.27x to 1676.85-\( \mu \text{W} \). On the other hand, if the \(-2.5\text{-V} RBB \) was applied, the chip could only operate when \( V_{\text{DD}} \geq 0.8\text{-V} \). At 1.05-V \( V_{\text{DD}} \) with \(-2.5\text{-V} V_{\text{BB}} \), the \( F_{\text{Max}} \) and \( P_{\text{active}} \) respectively decreased about 1.35x and 1.44x to 46-MHz and 510.3-\( \mu \text{W} \).

Fig. 10 gives the variation of energy consumption by the changes in the supply voltage. Without Back-gate Biasing (BB), the energy consumption ranged from 1.75-pJ/cycle to 16.15-pJ/cycle at 0.4-V and 1.2-V \( V_{\text{DD}} \), respectively. At 1.05-V \( V_{\text{DD}} \), the energy consumption was 11.89-pJ/cycle. The smallest number of energy consumption was 1.606-pJ/cycle at the condition of 0.4-V \( V_{\text{DD}} \) and \( V_{\text{BB}} = -0.5\text{-V} \). If \( V_{\text{BB}} < -0.5\text{-V} \) at the 0.4-V \( V_{\text{DD}} \), the core cannot function.

Fig. 11 depicts the variation of leakage current in standby mode with the CG technique. The \( I_{\text{leak}} \) value without BB ranged from 1.77-\( \mu \text{A} \) to 3.49-\( \mu \text{A} \) at 0.4-V to 1.2-V \( V_{\text{DD}} \). At 1.05-V \( V_{\text{DD}} \), the leakage current was 3.01-\( \mu \text{A} \). When RBB was applied, the \( I_{\text{leak}} \) value reduced about one order of magnitude at all range of \( V_{\text{DD}} \) when the \( V_{\text{BB}} \) was changing from 0-V to \(-0.5\text{-V} \). However, when lower voltages of \( V_{\text{DD}} \) were applied, the reduction in the \( I_{\text{leak}} \) was not reduced linearly. The main reason is because of the Gate-Induced Drain Leakage (GIDL) phenomenon. The smallest value of \( I_{\text{leak}} \) was 0.3-nA at the setting of 0.4-V \( V_{\text{DD}} \) and \(-2.5\text{-V} V_{\text{BB}} \).

5. Conclusion

The measurement results of the SOTB-65 nm 2S-32p-DCT had been presented in this paper. The paper also reviewed the DCT SFG which was built based on the ACor algorithm. The SOTB-65 nm technology was chosen due to its ultra-low-power and high-performance advantages. The
measurement results showed the substantial advantages of low-power and low-energy of the chip.

Acknowledgement

This research is funded by Vietnam National University Ho Chi Minh City (VNU-HCM) under the grant number of B2017-18-05. This research is also in the cooperation with Pham’s laboratory at the University of Electro-Communications (UEC), Tokyo, Japan. This work is also supported by VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with Synopsys, Inc. and Cadence Design Systems, Inc.

References

[1] A. Madanayake, et al.: “Low-power VLSI architectures for DCTDWT: Precision vs approximation for HD video, biomedical, and smart antenna applications,” IEEE Circuits Syst. Mag. 15 (2015) 25 (DOI: 10.1109/MCAS.2014.2385553).

[2] W. B. Pennebaker and J. L. Mitchell: JPEG: Still Image Data Compression Standard (Van Nostrand Reinhold, Kentucky, USA, 1992) (DOI: 10.1016/0920-5489(93)90038-S).

[3] Int. Telecom. Union - Telecomm. Standardization Sector: T-REC-H.263-200501-I (2005) https://www.itu.int/rec/T-REC-H.263.200501-I/en.

[4] T. Wiegand, et al.: “Overview of the H.264/AVC video coding standard,” IEEE Trans. Circuits Syst. Video Technol. 13 (2003) 560 (DOI: 10.1109/TCSVT.2003.815165).

[5] M. T. Pourazad, et al.: “HEVC: The new gold standard for video compression: How does HEVC compare with H.264/AVC?”, IEEE Consum. Electron. Mag. 1 (2012) 36 (DOI: 10.1109/MCCE.2012.2192754).

[6] W.-H. Chen, et al.: “A fast computational algorithm for the discrete cosine transform,” IEEE Trans. Commun. 25 (1977) 1004 (DOI: 10.1109/TCOM.1977.1093941).

[7] C. Loeffler, et al.: “Practical fast 1-D DCT algorithms with 11 multiplications,” ICASSP (1989) 988 (DOI: 10.1109/ICASSP.1989.266596).

[8] J. Liang and T. D. Tran: “Low-power and high-quality cordic-based DCT architecture based on data priority,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 22 (2014) 1060 (DOI: 10.1109/TVLSI.2013.2263223).

[9] Y. H. Hu and S. Naganathan: “An angle recoding method for CORDIC algorithm implementation,” IEEE Trans. Comput. 42 (1993) 99 (DOI: 10.1109/12.2192217).

[10] P. K. Meher and S. Y. Park: “CORDIC designs for fixed angle of rotation,” IEEE Trans. Very Large Scale Inteleg. (VLSI) Syst. 21 (2013) 217 (DOI: 10.1109/TVLSI.2012.2187080).

[11] R. Shukla and K. C. Ray: “Low latency hybrid CORDIC algorithm,” IEEE Trans. Comput. 63 (2014) 3066 (DOI: 10.1109/TC.2013.173).

[12] H.-T. Nguyen, et al.: “Low-resource low-latency hybrid adaptive CORDIC with floating-point precision,” IEICE Electron. Express 12 (2015) 20150258 (DOI: 10.1587/elex.12.20150258).

[13] T.-T. Hoang, et al.: “Minimum adder-delay architecture of 8/16/32-point DCT based on fixed-rotation adaptive CORDIC,” IEICE Electron. Express 15 (2018) 20180302 (DOI: 10.1587/elex.15.20180302).

[14] R. Tsujiya, et al.: “Silicon on thin BOX: A new paradigm of the CMOSFT for low-power and high-performance application featuring wide-range back-bias control,” IEDM Technical Digest (2004) 631 (DOI: 10.1109/IEDM.2004.1419245).

[15] H. Okuhara, et al.: “Power optimization methodology for ultralow power microcontroller with silicon on thin BOX MOSFET,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 25 (2017) 1578 (DOI: 10.1109/TVLSI.2016.2653675).

[16] J.-S. Park, et al.: “2-D large inverse transform (16 × 16, 32 × 32) for HEVC (high efficiency video coding),” J. Semicond. Technol. Sci. 12 (2012) 203 (DOI: 10.5573/ISTS.2012.12.2.203).

[17] S. Shen, et al.: “A unified 4/8/16/32-point integer IDCT architecture for multiple video coding standards,” IEEE Int. Conf. on Multimedia and Expo (2012) 788 (DOI: 10.1109/ICME.2012.7).

[18] P.-T. Chiang and T. S. Chang: “A reconfigurable inverse transform architecture design for HEVC decoder,” ISCAS (2013) 1006 (DOI: 10.1109/ISCAS.2013.6572019).

[19] C.-W. Chang, et al.: “A fast algorithm-based cost-effective and hardware-efficient unified architecture design of 4 × 4, 8 × 8, 16 × 16, and 32 × 32 inverse core transforms for HEVC,” J. Signal Process. Syst. 82 (2016) 69 (DOI: 10.1007/s11265-015-0982-8).

[20] M.-W. Lee, et al.: “Reconfigurable CORDIC-based low-power DCT architecture based on data priority,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 22 (2014) 1060 (DOI: 10.1109/TVLSI.2013.2263223).

[21] T. K. Rodrigues and E. E. Swartzlander, Jr.: “Adaptive CORDIC: Using parallel angle recoding to accelerate rotations,” IEEE Trans. Comput. 59 (2010) 522 (DOI: 10.1109/TC.2009.190).

[22] P. K. Meher and Y. C. Park: “CORDIC designs for fixed angle of rotation,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 21 (2013) 217 (DOI: 10.1109/TVLSI.2012.2187080).

[23] S. Aggarwal, et al.: “Scale-free hyperbolic CORDIC processor and its application to waveform generation,” IEEE Trans. Circuits Syst. I, Reg. Papers 60 (2013) 314 (DOI: 10.1109/TC.2012.2215778).

[24] M. T. Pourazad, et al.: “Low-power multiplierless DCT architecture using real-valued DCT coefficients,” IEEE Trans. Circuits Syst. Video Technol. 24 (2014) 168 (DOI: 10.1109/TCSVT.2013.2276662).

[25] M. Jridi, et al.: “A generalized algorithm and reconfigurable architecture for efficient and scalable orthogonal approximation of DCT,” IEEE Trans. Circuits and Syst. I, Reg. Papers 62 (2015) 449 (DOI: 10.1109/TCSI.2014.2360763).

[26] M. Masera, et al.: “Adaptive approximated DCT architectures for HEVC,” IEEE Trans. Circuits Syst. Video Technol. 27 (2016) 2714 (DOI: 10.1109/TCSVT.2016.2595320).

[27] S. Chatterjee and K. Sarawadekar: “An optimized architecture of HEVC core transform using real-valued DCT coefficients,” IEEE Trans. Circuits Syst. II, Exp. Briefs 65 (2018) 2052 (DOI: 10.1109/TCSII.2018.2815532).

[28] H. Jeong, et al.: “Low-power multiplierless DCT architecture using image correlation,” IEEE Trans. Consum. Electron. 50 (2004) 262 (DOI: 10.1109/TCE.2004.1277872).

[29] C.-C. Sun, et al.: “Low-power and high-quality cordic-based Loeffler DCT for signal processing,” IET Circuits Dev. Syst. 1 (2007) 453 (DOI: 10.1049/iet-cds:20060289).

[30] H. Huang and L. Xiao: “CORDIC based fast radix-2 DCT algorithm,” IEEE Signal Process. Lett. 20 (2013) 483 (DOI: 10.1109/LSP.2013.2252616).