Chen and Liu *EURASIP Journal on Advances in Signal Processing* (2020) 2020:48
https://doi.org/10.1186/s13634-020-00708-0

EURASIP Journal on Advances in Signal Processing

Research Open Access

# A low-area high-efficiency video coding inverse transform core using resource and time sharing architecture

Yuan-Ho Chen[1,2*] and Chieh-Yang Liu[1,3]

**Abstract**

In this paper, a very-large-scale integration (VLSI) design that can support high-efficiency video coding inverse discrete cosine transform (IDCT) for multiple transform sizes is proposed. The proposed two-dimensional (2-D) IDCT is implemented at a low area by using a single one-dimensional (1-D) IDCT core with a transpose memory. The proposed 1-D IDCT core decomposes a 32-point transform into 16-, 8-, and 4-point matrix products according to the symmetric property of the transform coefficient. Moreover, we use the shift-and-add unit to share hardware resources between multiple transform dimension matrix products. The 1-D IDCT core can simultaneously calculate the first- and second-dimensional data. The results indicate that the proposed 2-D IDCT core has a throughput rate of 250 MP/s, with only 110 K gate counts when implemented into the Taiwan semiconductor manufacturing (TSMC) 90-nm complementary metal-oxide-semiconductor (CMOS) technology. The results show the proposed circuit has the smallest area supporting the multiple transform sizes.

**Keywords:** Small area, High-efficiency video coding (HEVC), Inverse discrete cosine transform (IDCT), Very-large-scale integration (VLSI), Multiple transform dimensions, Shift-and-add unit (SAU)

## 1 Introduction

The video compression technique is utilized in digital image processing to reduce the redundancy of video information and increase the storage capacity and transmission rate efficiently. In recent years, video compression has been widely used in video codec devices, such as video conference equipment, video communication devices, and digital TVs. Groups such as the International Organization for Standardization (ISO) [1], International Telecommunication Union Telecommunication Standardization Sector (ITU-T) [2, 3], and Microsoft Corporation [4, 5] have developed various transform dimensions and coefficients for corresponding standards. The next-generation video coding standard, which is referred to as high-efficiency video coding (HEVC), is expected to provide approximately 50% reduction in the bit rate (at similar visual quality) over the current standard (H.264/AVC). HEVC is intended for larger resolutions and higher frame rates than the H.264/AVC standard [6–10]. In HEVC, the largest coding unit can be up to 64 × 64 in size, and transform sizes of 4 × 4, 8 × 8, 16 × 16, and 32 × 32 are supported [11]. Multiple transform sizes improve the compression performance; however, they also increase the implementation complexity [12–14].

Recently, many researchers have implemented integer transforms, especially for HEVC [15–24]. The row-column decomposition structure is widely used to design a two-dimensional (2-D) transform core. In general, the 2-D transform core is directly implemented using two one-dimensional (1-D) cores and a transposed memory. This method can achieve a high-throughput rate; however, it results in the wastage of a considerable amount of circuit

*Correspondence: chenyh@mail.cgu.edu.tw
[1]Department of Electronics Engineering, Chang Gung University, Taoyuan,Taiwan
[2]Department of Radiation Oncology, Chang Gung Memorial Hospital-LinKou, Taoyuan, Taiwan
Full list of author information is available at the end of the article

area. Many architectures use this structure to implement the inverse transform [15, 16]. According to the area consideration, the multiplexer structure is introduced. The multiplexer controls the 1-D inverse discrete cosine transform (IDCT) core, which calculates the first-dimensional (1st-D) and second-dimensional (2nd-D) operations. The 1-D IDCT core uses matrix decomposition to save the circuit area. Thus, the circuit area can be reduced. However, the throughput rate decreases compared with the original speed [17]. In [18], a single 1-D core was proposed for executing 1st-D and 2nd-D computation simultaneously, which can allow the throughput rate to be maintained the same as the clock rate. To improve the throughput rate, Chen and Ko [19] presented a 32-point IDCT for HEVC. The IDCT utilizes 32 parallel computation paths to reach an ultrahigh-throughput rate of 6.4 giga-pixel per second (GP/s). However, the parallel computation architecture considerably increases the circuit area overhead [19]. The horizontal and vertical line buffer for reference sample is presented in [20], which only costs 0.8K bit and is implemented by register files with SRAM-free. Based on this buffer, the 32-pixel transform unit can achieve a frequency of 400 MHz for a 65-nm process. The resource-sharing pipelined architecture [21] is synthesized by using NanGate OpenPDK 45 nm library achieving a 222-MHz clock rate and supporting real-time decoding of 4096 × 3072 video sequences with 70 fps.

This paper proposes an inverse transform core for HEVC applications supporting multiple transform sizes. The IDCT core utilizes a single 1-D core and transposed memory to achieve a low-area design. The 1-D IDCT core adapts the symmetric property of the transform coefficient matrix, and the 32-point transform can be decomposed into 16-, 8-, and 4-point matrix products. Moreover, the proposed core uses a shift-and-add unit (SAU) to share the hardware resource among multiple transform dimension matrix products and also uses the proposed data control flow to share the computation resource. Thus, the proposed IDCT core can execute 1st-D and 2nd-D computation simultaneously. The proposed circuit can maintain the throughput rate to be the same as the operating frequency. The proposed 2-D IDCT core, which is implemented into 90-nm complementary metal-oxide-semiconductor (CMOS) technology, has a throughput rate of 250 MP/s so that it can meet the full high definition (HD) 1080p requirement with only 110 K gate counts. The main contribution of this work is listed as follows:

- The multiple transform sizes including 32-, 16-, 8-, and 4-point transformations are supported HEVC applications.
- Using a single 1-D core and transposed memory to achieve a low-area design.
- Using the proposed data control flow to share the computation resource, and the IDCT core can execute 1st-D and 2nd-D computation simultaneously achieving a high-throughput rate.

Consequently, the proposed IDCT achieves high-throughput and low-area design supporting multiple transform sizes for HEVC applications.

This paper is organized as follows. Section 2.1 presents the mathematical derivation of the 32-point IDCT. Section 2.2 describes the proposed architecture, which uses resource and timing sharing. This section also describes the hardware architecture based on SAU computation for multiple transform dimensions and the proposed data control flow. Section 3 includes the comparisons and discussion, and the conclusions are presented in Section 4.

## 2 Method

### 2.1 Algorithm of the 32-point IDCT

The transform computation in HEVC uses a set of IDCT transform matrices. In general, a 2-D inverse transform can be obtained by performing two 1-D IDCTs through the row-column decompensation method.

$$\mathbf{y} = \mathbf{C}\mathbf{Z}\mathbf{C}^{\mathbf{T}} \tag{1}$$

$$\mathbf{y}^{\mathbf{T}} = \mathbf{C}^{\mathbf{T}}(\mathbf{C}\mathbf{Z})^{\mathbf{T}} = \mathbf{C}^{\mathbf{T}}\mathbf{x}^{\mathbf{T}} \tag{2}$$

The 32-point 1-D IDCT can be expressed as follows:

$$\mathbf{x} = \mathbf{C}\mathbf{Z} \tag{3}$$

$$\mathbf{x} = \begin{bmatrix} x_0 & x_1 & \dots & x_{31} \end{bmatrix}^T \tag{4}$$

$$\mathbf{Z} = \begin{bmatrix} Z_0 & Z_1 & \dots & Z_{31} \end{bmatrix}^T \tag{5}$$

where $\mathbf{C}$ indicates the 32 × 32 coefficient matrix.

According to the symmetric property, Eq. (3) can be decomposed into two separate equations:

$$\mathbf{x_{16u}} = \mathbf{C_{16e}}\mathbf{Z_{16e}} + \mathbf{C_{16o}}\mathbf{Z_{16o}} = \mathbf{A} + \mathbf{B} \tag{6}$$

$$\mathbf{x_{16d}} = \mathbf{C_{16e}}\mathbf{Z_{16e}} - \mathbf{C_{16o}}\mathbf{Z_{16o}} = \mathbf{A} - \mathbf{B} \tag{7}$$

where

$$\mathbf{x_{16u}} = \begin{bmatrix} x_0 & x_1 & \dots & x_{15} \end{bmatrix}^T \tag{8}$$

$$\mathbf{x_{16d}} = \begin{bmatrix} x_{16} & x_{17} & \dots & x_{31} \end{bmatrix}^T \tag{9}$$

$$\mathbf{Z_{16e}} = \begin{bmatrix} Z_0 & Z_2 & \dots & Z_{30} \end{bmatrix}^T \tag{10}$$

$$\mathbf{Z_{16o}} = \begin{bmatrix} Z_1 & Z_3 & \dots & Z_{31} \end{bmatrix}^T \tag{11}$$

$\mathbf{C_{16e}}$ and $\mathbf{C_{16o}}$ are the 16-point even and odd coefficient matrices, respectively, for the 32-point transform. The coefficient of $\mathbf{C_{16e}}$ is presented in Eq. (12). The 16-point even-part computation can be further divided into 8-point even and odd computations.

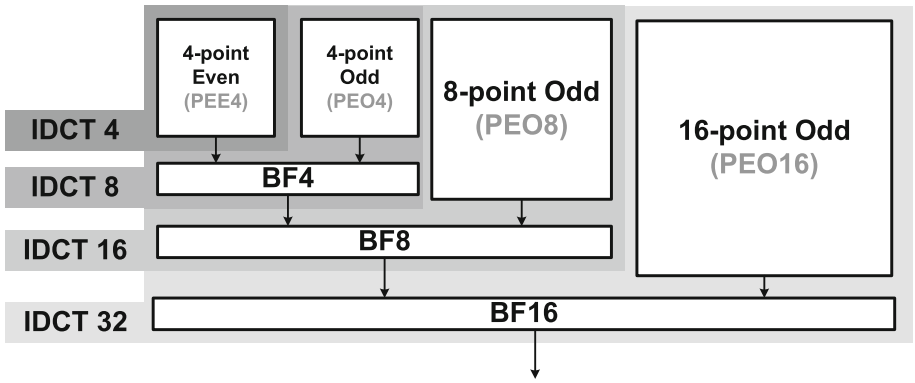


**Fig. 1** Decomposition of the 32-point inverse discrete cosine transform (IDCT) computation

$$\mathbf{C_{16e}} = \begin{bmatrix}
64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 \\
90 & 87 & 80 & 70 & 57 & 43 & 25 & 9 & -9 & -25 & -43 & -57 & -70 & -80 & -87 & 90 \\
89 & 75 & 50 & 18 & -18 & -50 & -75 & -89 & -89 & -75 & -50 & -18 & 18 & 50 & 75 & 89 \\
87 & 57 & 9 & -43 & -80 & -90 & -70 & -25 & 25 & 70 & 90 & 80 & 43 & -9 & -57 & -87 \\
83 & 36 & -36 & -83 & -83 & -36 & 36 & 83 & 83 & 36 & -36 & -83 & -83 & -36 & 36 & 83 \\
80 & 9 & -70 & -87 & -25 & 57 & 90 & 43 & -43 & -90 & -57 & 25 & 87 & 70 & -9 & -80 \\
75 & -18 & -89 & -50 & 50 & 89 & 18 & -75 & -75 & 18 & 89 & 50 & -50 & -89 & -18 & 75 \\
70 & -43 & -87 & 9 & 90 & 25 & -80 & -57 & 57 & 80 & -25 & -90 & -9 & 87 & 43 & -70 \\
64 & -64 & -64 & 64 & 64 & -64 & -64 & 64 & 64 & -64 & -64 & 64 & 64 & -64 & -64 & 64 \\
57 & -80 & -25 & 90 & -9 & -87 & 43 & 70 & -70 & -43 & 87 & 9 & -90 & 25 & 80 & -57 \\
50 & -89 & 18 & 75 & -75 & -18 & 89 & -50 & -50 & 89 & -18 & -75 & 75 & 18 & -89 & 50 \\
43 & -90 & 57 & 25 & -87 & 70 & 9 & -80 & 80 & -9 & -70 & 87 & -25 & -57 & 90 & -43 \\
36 & -83 & 83 & -36 & -36 & 83 & -83 & 36 & 36 & -83 & 83 & -36 & -36 & 83 & -83 & 36 \\
25 & -70 & 90 & -80 & 43 & 9 & -57 & 87 & -87 & 57 & -9 & -43 & 80 & -90 & 70 & -25 \\
18 & -50 & 75 & -89 & 89 & -75 & 50 & -18 & -18 & 50 & -75 & 89 & -89 & 75 & -50 & 18 \\
9 & -25 & 43 & -57 & 70 & -80 & 87 & -90 & 90 & -87 & 80 & -70 & 57 & -43 & 25 & -9
\end{bmatrix} \tag{12}$$

$$\mathbf{x_{16u}} = \begin{bmatrix} \mathbf{x_{8u}} \\ \mathbf{x_{8d}} \end{bmatrix} \tag{13}$$

$$\mathbf{x_{8u}} = \mathbf{C_{8e}Z_{8e}} + \mathbf{C_{8o}Z_{8o}} = \mathbf{a} + \mathbf{b} \tag{14}$$

$$\mathbf{x_{8d}} = \mathbf{C_{8e}Z_{8e}} - \mathbf{C_{8o}Z_{8o}} = \mathbf{a} - \mathbf{b} \tag{15}$$

and

$$\mathbf{x_{8u}} = \begin{bmatrix} x_0 & x_1 & \dots & x_7 \end{bmatrix}^T \tag{16}$$

$$\mathbf{x_{8d}} = \begin{bmatrix} x_8 & x_9 & \dots & x_{15} \end{bmatrix}^T \tag{17}$$

$$\mathbf{Z_{8e}} = \begin{bmatrix} Z_0 & Z_4 & \dots & Z_{28} \end{bmatrix}^T \tag{18}$$

$$\mathbf{Z_{8o}} = \begin{bmatrix} Z_2 & Z_6 & \dots & Z_{30} \end{bmatrix}^T \tag{19}$$

Moreover, the 8-point even-part computation can be divided into the following equations:

$$\mathbf{x_{8u}} = \begin{bmatrix} \mathbf{x_{4u}} \\ \mathbf{x_{4d}} \end{bmatrix} \tag{20}$$

$$\mathbf{x_{4u}} = \mathbf{C_{4e}Z_{4e}} + \mathbf{C_{4o}Z_{4o}} = \alpha + \beta \tag{21}$$

$$\mathbf{x_{4d}} = \mathbf{C_{4e}Z_{4e}} - \mathbf{C_{4o}Z_{4o}} = \alpha - \beta \tag{22}$$

and

$$\mathbf{x_{4u}} = \begin{bmatrix} x_0 & x_1 & x_2 & x_3 \end{bmatrix}^T \tag{23}$$

$$\mathbf{x_{4d}} = \begin{bmatrix} x_4 & x_5 & x_6 & x_7 \end{bmatrix}^T \tag{24}$$

$$\mathbf{Z_{4e}} = \begin{bmatrix} Z_0 & Z_8 & Z_{16} & Z_{24} \end{bmatrix}^T \tag{25}$$

$$\mathbf{Z_{4o}} = \begin{bmatrix} Z_4 & Z_{12} & Z_{20} & Z_{28} \end{bmatrix}^T \tag{26}$$

Thus, the entire 32-point IDCT computation can be decomposed into 16-, 8-, and 4-point operations, as displayed in Fig. 1. The 16-point IDCT computation can be decomposed into 8- and 4-point operations (the 4-point even, 4-point odd, BF4, 8-point odd, and BF8 modules); 8-point IDCT can be calculated by using 4-point even, 4-point odd, and BF4 modules. The 4-point IDCT is implemented as a 4-point even module.

### 2.2 Proposed architecture

Compared to the multiple computation path IDCT [19], the proposed 2-D IDCT core is composed of one 1-D transform core and one transposed memory (TMEM) to achieve a small-area design. The 1-D IDCT core utilizes the proposed data shared in the time scheme such that the throughput rate can be maintained the same as the operation frequency. The 1-D core supports full HD 1080p, which requires $1080 \times 1920 \times 60 = 124,416,000$ pel/s $\simeq$ 125 MP/s. The entire architecture is illustrated in Fig. 2.

#### *2.2.1 1-D IDCT core*

The 1-D 32-point IDCT core comprises a 4-point even-part process element (PEE4), a 4-point odd-part process element (PEO4), an 8-point odd-part process element (PEO8), a 16-point odd-part process element (PEO16), and three butterfly (BF) modules. The process elements (PEs) are designed using add-and-shift to share the hard-

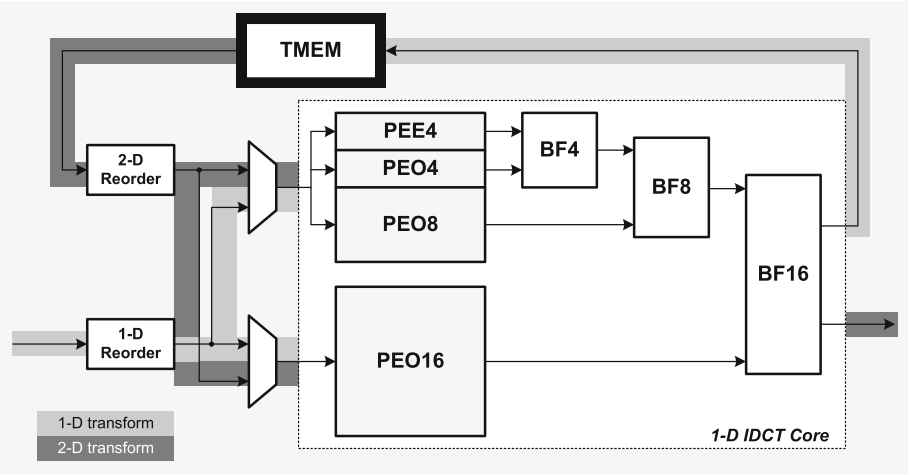


**Fig. 2** Architecture of the proposed two-dimensional (2-D) IDCT

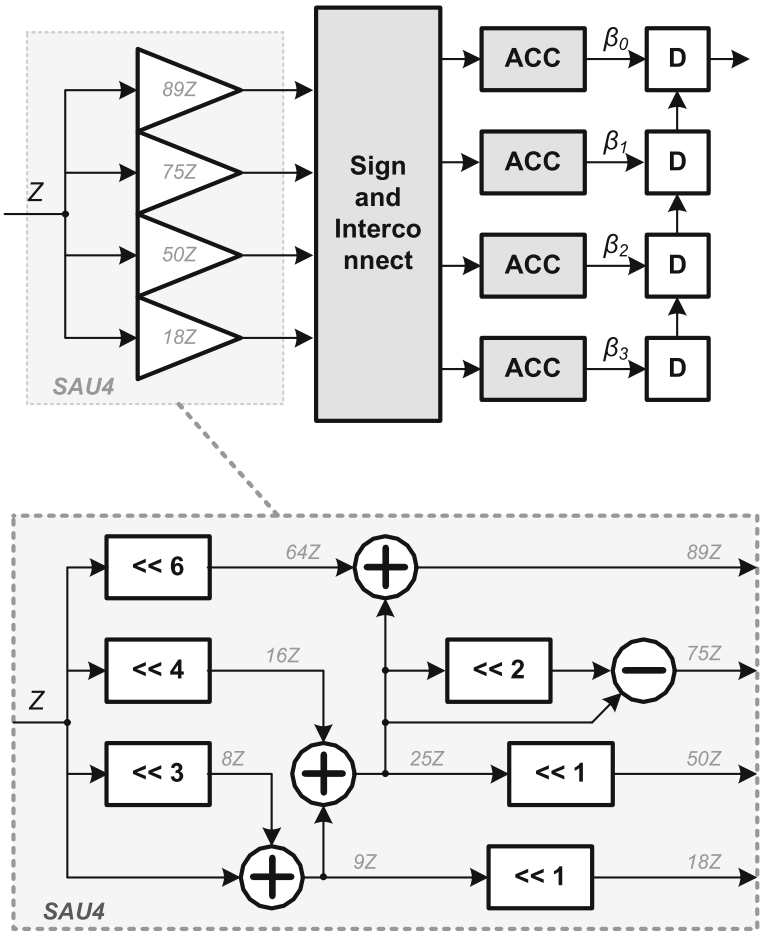


**Fig. 3** Architecture of the PEO4 module

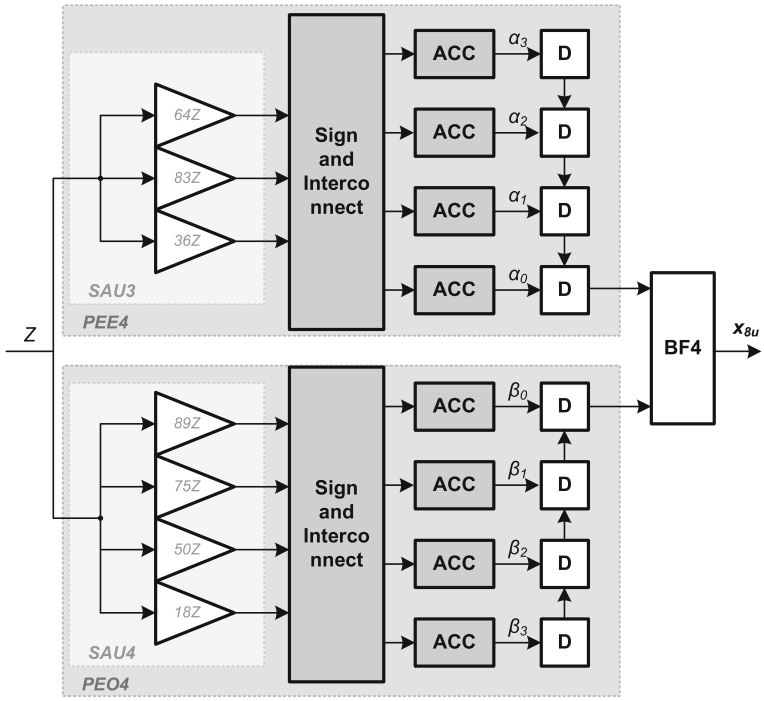


**Fig. 4** Architecture of the 8-point IDCT as PEE8

ware resources. The matrix product of the PEO4 computation can be expressed as follows:

$$\begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} = \begin{bmatrix} 89 & 75 & 50 & 18 \\ 75 & -18 & -89 & -50 \\ 50 & -89 & 18 & 75 \\ 18 & -50 & 75 & -89 \end{bmatrix} \begin{bmatrix} Z_0 \\ Z_1 \\ Z_2 \\ Z_3 \end{bmatrix} \tag{27}$$

Four coefficients $\{89, 75, 50, 18\}$ with different signs are used to multiply the inputs $\begin{bmatrix} Z_0 & Z_1 & Z_2 & Z_3 \end{bmatrix}^T$. Thus, the matrix product operation can be simplified using the multiple constant multiplication technique.

$$89 \cdot Z = 64 \cdot Z + 25 \cdot Z \tag{28}$$

$$75 \cdot Z = (4+1) \cdot 25 \cdot Z \tag{29}$$

$$50 \cdot Z = 2 \cdot 25Z \tag{30}$$

$$18 \cdot Z = 2 \cdot (9 \cdot Z) \tag{31}$$

The sharing architecture called four operands SAU (SAU4) is displayed on the left side of Fig. 3. SAU4 uses the shift-and-add function instead of the multiplier function to reduce the area cost. Furthermore, it shares the same hardware resource among the constant multiplications. Then, the sign-and-interconnection circuit maintains the matrix product. Finally, four accumulators (ACCs) sum the product results for every four clock cycles. Thus, every four clock cycles, the outputs $\beta_0$, $\beta_1$, $\beta_2$, and $\beta_3$ complete the computation in Eq. (27).

#### *2.2.2 Architecture of the 8-, 16-, and 32-point IDCTs*

The architecture of the 8-point IDCT, which is called PEE8, is displayed in Fig. 4. PEE8 consists of the PEE4, PEO4, and BF4 modules, which execute the computations in Eqs. (20)–(22). The PEO4 module executes the matrix product $\mathbf{C_{4o}Z_{4o}}$, as illustrated in Fig. 3. The even-part computation ($\mathbf{C_{4e}Z_{4e}}$) is also implemented in SAU3, sign-and-interconnection circuits, ACCs, and registers (D). The four ACCs and four registers are used to sum the product results for every four clock cycles and send them in the following four clock cycles. The BF4 module adds and subtracts $\mathbf{C_{4o}Z_{4o}}$ and $\mathbf{C_{4e}Z_{4e}}$ to output $\mathbf{x_{4u}}$ and $\mathbf{x_{4d}}$.

Moreover, the 16-point IDCT consists of the PEO8, PEE8, and BF8 modules. The PEO8 module calculates the odd part of the 16-point transformation ($\mathbf{C_{8o}Z_{8o}}$), as indicated in Eqs. (14) and (15). The lower half of Fig. 5 illustrates the architecture of the PEO8 module. The SAU8 module shares the hardware resources by using the shift-and-add architecture, and the BF8 module controls the addition and subtraction output.

The BF16 module calculates the final results before transpose and output. Thus, $\mathbf{C_{16o}Z_{16o}}$ and $\mathbf{C_{16e}Z_{16e}}$ in Eqs. (6) and (7) can be calculated using PEO16 and PEE16, respectively. The architecture of PEO16 is displayed in Fig. 6. The mixed SAU16 (SAU16M) module, which uses the shift-and-add architecture, executes the 16-point matrix product $\mathbf{C_{16o}Z_{16o}}$ as well as the 16-point

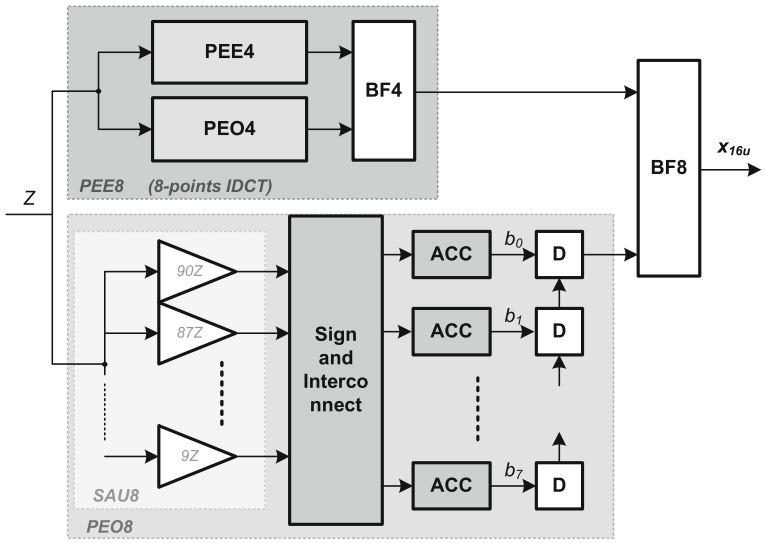


**Fig. 5** Architecture of the 16-point IDCT as PEE16

$\mathbf{C_{16e}Z_{16e}}$, 8-point $\mathbf{C_{8e}Z_{8e}}$, and 4-point $\mathbf{C_{4e}Z_{4e}}$ by supporting variable transform sizes (32-, 16-, 8-, and 4-point matrix products). Thus, $\mathbf{x_{16o}}$, $\mathbf{x_{16u}}$, $\mathbf{x_{8u}}$, and $\mathbf{x_{4u}}$ can be obtained from PEO16 according to the adaptive transform size.

### 2.2.3 *Data flow of the proposed IDCT*

The proposed IDCT core has a 1-D core and TMEM. The 1st-D and 2nd-D computations can be executed in the same 1-D core through the proposed data control scheme to save hardware cost. Thus, the proposed IDCT core can achieve a high throughput and low area.

According to the reorder registers and MUX, the 1st-D/2nd-D data is input into the 16-point odd-/even-part PE during the first 16 cycles of the 32-cycle period. The 1st-D/2nd-D data is then input into the 16-point even-/odd-part PE during the following 16 cycles of the 32-cycle period. Thus, the 1st-D and 2nd-D computations can share the same hardware resources during the 32-cycle period. For the 32-point transform, the PEE4 module executes in the first four clock cycles, the PEO4 module executes in the following four cycles, and the PEE4 module outputs the results to BF4. When the PEO4 module outputs the results to BF4, the BF4 module begins calculating the addition and subtraction as per Eqs. (21) and (22). In the following eight cycles, the PEO8 module calculates the matrix product $\mathbf{C_{8o}Z_{8o}}$ and the BF4 module simultaneously outputs the results. In cycles 1624, the PEO8 module outputs the computation results to BF8 and BF8 executes addition and subtraction. The BF8 module then outputs the addition results in cycles 1624 and the subtraction results in cycles 2432. The PEO16 module executes the matrix product $\mathbf{C_{16o}Z_{16o}}$ when BF8 outputs the addition and subtraction results to BF16. In the

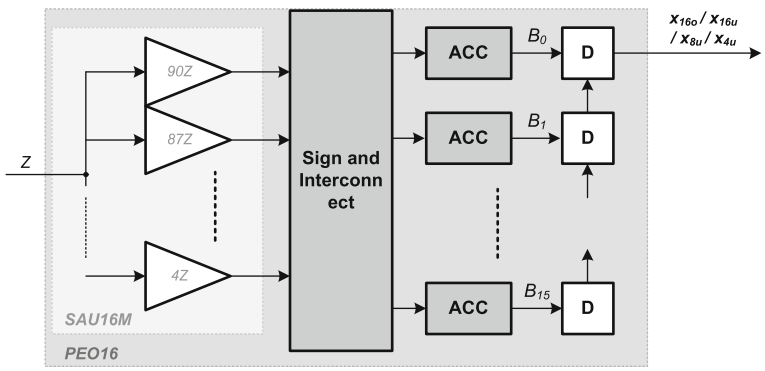


**Fig. 6** Architecture of the PEO16 module

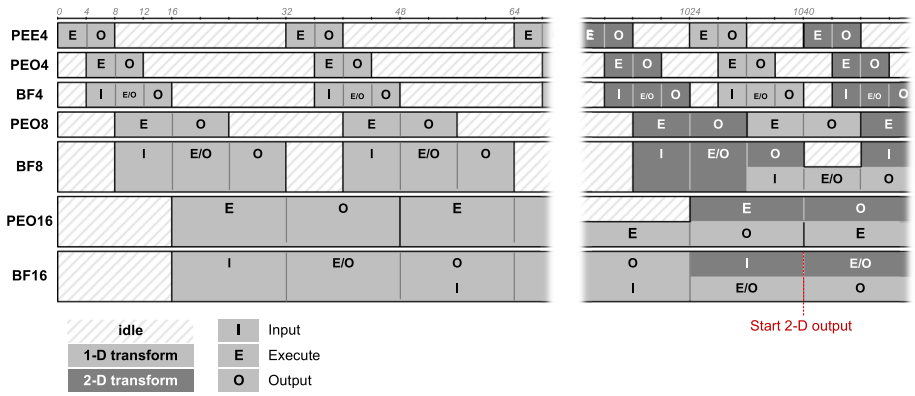


**Fig. 7** Data flow of the proposed IDCT circuit

48th cycle, BF16 outputs the first 32-point 1-D transform data and inputs the following 32-point transform data. After 1008 cycles, the 2nd-D data is output from the TMEM and fed into the PEE4 module. In these 16 cycles, the PEE4, PEO4, BF4, PEO8, and BF8 modules execute the 2nd-D data due to the ideal time of these circuit resources. In the following 16 cycles, the PEE4, PEO4, BF4, PEO8, and BF8 modules execute the 1st-D data and the PEO16 and BF16 modules execute the 2nd-D data. The 2-D transform data is starting output at the 1040 cycle; thus, the latency of the proposed core is 1040 clock cycles. The core takes 2064 cycles to complete the 32 × 32 IDCT transformation. According the proposed data flow (Fig. 7), the proposed circuit can maintain the throughput rate to be the same as the operation frequency.

## 3 Results and discussion

To indicate the performance of the proposed circuit, the very-large-scale integration implementation is described in the following subsection. The proposed circuit is also compared with other circuit designs in the literature.

### 3.1 Chip implementation

The proposed 32-point 2-D IDCT core is implemented in a 1-V Taiwan semiconductor manufacturing (TSMC) 90-nm 1P9M CMOS process. It uses the Synopsys Design Compiler to synthesize the register transfer language code and the Cadence Encounter Digital Implementation for placement and routing (P&R). The proposed IDCT core is operated at 250 MHz with a power consumption of 49 mW to meet the full HD 1080p specifications. The total gate count of the proposed core is 110 K. The gate counts of 1-D IDCT core and TMEM are 80 K and 30 K, respectively. The characteristics of the IDCT are presented in Table 1. The input data is 18-bit and the output data is 14-bit. There are 22 input pins, 14 output pins, and 13 power pins. The layout of the proposed 2-D IDCT core is displayed in Fig. 8, including the 1-D IDCT and TMEM.

### 3.2 Comparison with existing studies

Table 2 presents a comparison of the proposed 2-D inverse transform core with existing methods. In [8] and [16], dual 1-D cores with a transpose memory have been used in the implementation of the 2-D inverse transform. A low-energy HEVC inverse transform core was presented in [8]. The design has 142-K three-input NAND gates without a transpose memory. Park et al. employed high-throughput structures for a 32 × 32 transform and incurred a high area overhead because the memory modules used the register structure [16]. The design only supports the 32- and 16-point inverse transforms, which are insufficient for HEVC applications. The high-performance core associated with

**Table 1** Chip characteristics of the proposed 2-D IDCT core

| | |
|---|---|
| Function | HEVC 4/8/16/32-point. |
| Process technology | TSMC 90-nm CMOS, 1P9M |
| Supply voltage | 1.0 V |
| Max. clock frequency | 250 MHz |
| Throughput rate | 250 MPS |
| Core area | 863 × 858 $\mu m^2$ |
| ⊞ Gate counts | 110 K |
| ⊟ 1-D IDCT | 80 K |
| ⊟ TMEM | 30 K |
| Power consumption | 49 mW @ 250 MHz |
| ⊞ I/O pins | 49 pins |
| ⊟ Input pins | 22 pins |
| ⊟ Output pins | 14 pins |
| ⊟ Power pins | 13 pins |

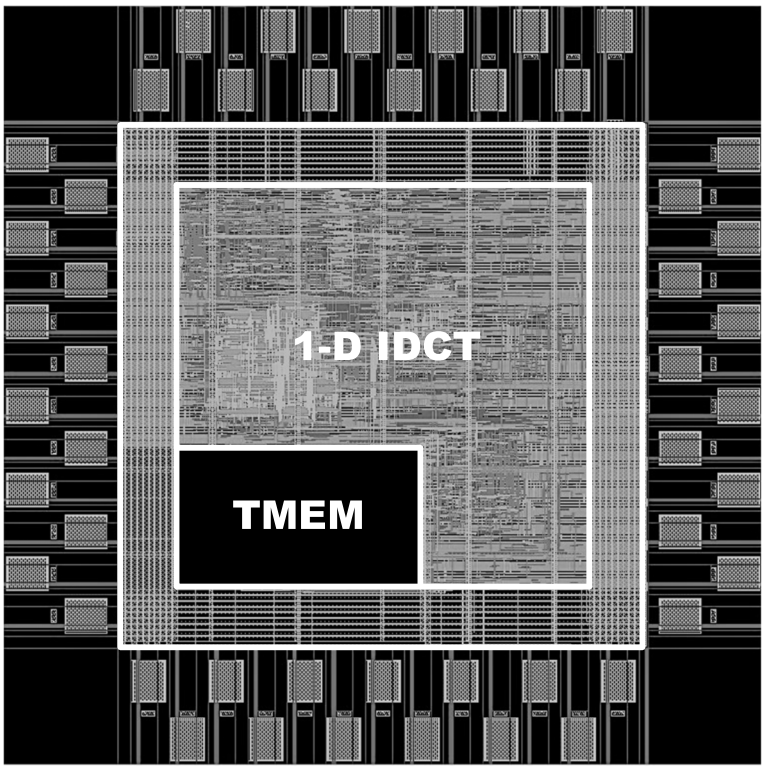


**Fig. 8** Chip layout of the proposed 2-D IDCT core

90-nm technology can support 3840 × 2160@30fps; however, the structure of the multiplexer reduces the operating frequency by half [17]. An ultra-low-cost IDCT employing a single 1-D core with a transposed memory was presented in [18] for the execution of 2-D transforms. This approach considerably reduced the circuit area. However, the design only supports the 32-point HEVC inverse transform, which is insufficient for HEVC application. An ultrahigh-throughput design was presented in [19]. The 16 parallel computation streams achieved a throughput rate of 6.4 GP/s for supporting multiple transform dimensions when implemented into 40-nm CMOS technology. However, a very large area cost is incurred by the design in [19]. The low-area cost design for multiple transform size HEVC applications using shifts and additions is presented in [22], in which 112 K gate counts are required for a 2-D IDCT transform. The 2-D DCT/IDCT [24] computes 2-D 4-/8-/16-/32-point DCT/IDCT and consumes 120 K gates supporting the 4K HEVC video sequences. As presented in Table 2, the proposed design achieves the smallest area cost when supporting multiple transform dimensions.

**Table 2** Comparison of inverse 2-D transform cores

| Method | [8] | [16] | [17] | [18] | [19] | [22] | [24] | Proposed |
|---|---|---|---|---|---|---|---|---|
| Technology | 90 nm | 0.18 μm | 90 nm | 0.18 μm | 40 nm | 0.18 μm | 90 nm | 90 nm |
| Gate count* | 142 K* | 487 K | 133.8 K | 79 K | 335 K | 112 K+16 Kb | 120 K | 110 K |
| Frequency | 150 MHz | 300 MHz | 270 MHz | 125 MHz | 400 MHz | 110 MHz | 149.35 MHz | 250 MHz |
| Pixels/cycle | 2.67 | 2.12 | 0.5 | 2 | 16 | 1 | 32 | 1 |
| Throughput | 400 MPS | 636 MPS | 135 MPS | 250 MPS | 6.4 GPS | 110 MPS | 3.5 GPS | 250 MPS |
| Support standard | HEVC | HEVC | HEVC | HEVC | HEVC | HEVC | HEVC | HEVC |
| Support dimension | 4/8/16/32 | 16/32 | 4/8/16/32 | 32 | 4/8/16/32 | 4/8/16/32 | 4/8/16/32 | 4/8/16/32 |

⋆ Estimate by 2-input NAND gate
* Estimate by 3-input NAND gate and the area excluding the on-chip memory

## 4 Conclusions

This paper proposes a low-area 2-D IDCT core that supports multiple transform sizes for HEVC application. Compared with previously proposed designs, the proposed core has the smallest circuit area when supporting multiple dimension transforms. Moreover, the throughput rate can be maintained the same as the operation frequency (250 MHz). Consequently, the proposed 2-D IDCT core is suitable for HEVC video and next-generation video transform applications.

#### Abbreviations

VLSI: Very-large-scale integration; IDCT: Inverse discrete cosine transform; 2-D: Two-dimensional; 1-D: One-dimensional; HEVC: High-efficiency video coding; SAU: Shift-and-add unit; ISO: International Organization for Standardization ITU-T: International Telecommunication Union Telecommunication Standardization Sector; 1st-D: First-dimensional; 2nd-D: Second-dimensional; HD: High definition; TMEM: Transposed memory; PEE4: 4-point even-part process element; PEO4: 4-point odd-part process element; PEO8: 8-point odd-part process element; PEO16: 16-point odd-part process element; BF: Butterfly; PEs: Process elements; SAU4: Four operands SAU; ACCs: Accumulators; P&R: Placement and routing

#### Acknowledgements

The authors would like to thank the Particle Physics and Beam Delivery Core Laboratory of Institute for Radiological Research, Chang Gung University/Chang Gung Memorial Hospital, Linkou, for their assistance. The authors would like to thank the National Chip Implementation Center (CIC), Taiwan, for providing the circuit design automation tools and chip fabrication.

#### Authors' contributions

Both authors read and approved the final manuscript.

#### Funding

This work was supported in part by the Ministry of Science and Technology of Taiwan (MOST 108-2221-E-182-052, 109-2221-E-182-050), Chang Gung University (BMRPE26), and Chang Gung Memorial Hospital (CMRPD2G0313, CMRPD2H0301, CIRPD2F0014).

#### Availability of data and materials

Not applicable.

#### Ethics approval and consent to participate

Not applicable.

#### Consent for publication

Not applicable.

#### Competing interests

The authors declare that they have no competing interests

#### Author details

[1]Department of Electronics Engineering, Chang Gung University, Taoyuan,Taiwan. [2]Department of Radiation Oncology, Chang Gung Memorial Hospital-LinKou, Taoyuan, Taiwan. [3]Department of Information and Computer Engineering, Chung Yuan Christian University, Taoyuan, Taiwan.

Received: 26 April 2020 Accepted: 17 November 2020
Published online: 30 November 2020

#### References

1. W. Li, Overview of fine granularity scalability in MPEG-4 video standard. IEEE Trans. Circ. Syst. Video Technol. **11**(3), 301–317 (2001)
2. G. Cote, B. Erol, M. Gallant, F. Kossentini, H.263+: video coding at low bit rates. IEEE Trans. Circ. Syst. Video Technol. **8**(7), 849–866 (1998)
3. T. Wiegand, G. Sullivan, G. J. Bjontegaard, A. Luthra, Overview of the H.264/AVC video coding standard. IEEE Trans. Circ. Syst. Video Technol. **13**(7), 560–576 (2003)
4. H. Kalva, J. B. Lee, The VC-1 video coding standard. IEEE Multimed. **14**(4), 88–91 (2007)
5. S. Srinivasan, P. Hsu, T. Holcomb, K. Mukerjee, S. L. Regunathan, B. Lin, J. Liang, M. C. Lee, J. R. Corbera, Windows media video 9: overview and applications. Signal Process. Image Commun. **19**(9), 851–875 (2004)
6. G. Sullivan, J. Ohm, W.-J. Han, T. Wiegand, Overview of the high efficiency video coding (HEVC) standard. IEEE Trans. Circ. Syst. Video Technol. **22**(12), 1649–1668 (2012)
7. Z. Pan, L. Chen, X. Sun, Low complexity HEVC encoder for visual sensor networks. Sensors. **15**(12), 30115–30125 (2015)
8. E. Kalali, E. Ozcan, O. M. Yalcinkaya, I. Hamzaoglu, A low energy HEVC inverse transform hardware. IEEE Trans. Consum. Electron. **60**(4), 754–761 (2014)
9. P. Garrido Abenza, M. Malumbres, P. Piol, O. Lpez-Granado, Source coding options to improve HEVC video streaming in vehicular networks. Sensors. **18**(9), 3107 (2018)
10. Y. Tseng, Y. Chen, Cost-effective multi-standard video transform core using time-sharing architecture. EURASIP J. Adv. Signal Process. **49**, 1–9 (2019)
11. M. Zhang, J. Qu, H. Bai, Entropy-based fast largest coding unit partition algorithm in high-efficiency video coding. Entropy. **15**(6), 2277–2287 (2013)
12. H. Sun, D. Zhou, P. Liu, S. Goto, A low-cost VLSI architecture of multiple-size IDCT for H.265/HEVC. IEICE Trans. Fundamentals. **E97-A**(12), 2467–2476 (2014)
13. D. Coelho, R. Cintra, F. Bayer, S. Kulasekera, A. Madanayake, P. Martinez, T. Silveira, R. Oliveira, V. Dimitrov, Low-complexity loeffler DCT approximations for image and video coding. J. Low Power Electron. Appl. **8**(4), 46 (2018)
14. G. Pastuszak, Flexible architecture design for H.265/HEVC inverse transform. Circ. Syst. Signal Process. **34**(6), 1931–1945 (2015)
15. P. Meher, S. Y. Park, B. Mohanty, K. S. Lim, C. Yeo, Efficient integer DCT architectures for HEVC. IEEE Trans. Circ. Syst. Video Technol. **24**(1), 168–178 (2014)
16. J. S. Park, W. J. Nam, S. M. Han, S. Lee, 2-D large inverse transform (16×16, 32×32) for HEVC (high efficiency video coding). J.. Semicond Technol. Sci. **12**(2), 203–211 (2012)
17. P. T. Chiang, T. S. Chang, A reconfigurable inverse transform architecture design for HEVC decoder. Proc. IEEE Int. Symp. Circ. Syst., 1006–1009 (2013)
18. Y. H. Chen, C. Y. Liu, Area-efficient video transform for HEVC applications. Electron. Lett. **51**(14), 1065–1067 (2015)
19. Y.-H. Chen, Y.-F. Ko, High-throughput IDCT architecture for high-efficiency video coding (HEVC). Int. J. Circ. Theor. Appl. **45**, 2260–2269 (2017)
20. Y. Fan, G. Tang, X. X. Zeng, A compact 32-pixel TU-oriented and SRAM-free intra prediction VLSI architecture for HEVC decoder. IEEE Access. **7**, 149097–149104 (2019)
21. H. R. Nayeri, F. Zargari, Hardware solution for implementing the entire inverse IDCTs in HEVC decoder. Int. J. Electron. **105**(8), 1261–1272 (2018)
22. A. Ben Atitallah, M. Kammoun, R. Ben Atitallah, An optimized FPGA design of inverse quantization and transform for HEVC decoding blocks and validation in an SW/HW environment. Turk. J. Electr. Eng. Comput. Sci. **28**(3), 1656–1672 (2020)
23. A. Ben Atitallah, M. Kammoun, K. M. A. Ali, R. Ben Atitallah, An FPGA comparative study of high-level and low-level combined designs for HEVC intra, inverse quantization, and IDCT/IDST 2D modules. Int. J. Circ Theor. Appl. **48**(8), 1274–1290 (2020)
24. A. Singhadia, M. Mamillapalli, I. Chakrabarti, Hardware-efficient 2D-DCT/IDCT architecture for portable HEVC-compliant devices. IEEE Trans. Consum. Electron. **66**(3), 203–212 (2020)

## Publisher's Note