Alliance for Open Media
Codec Working Group

Document: **CWG-E230_**<u>**v4**</u>

On Core Transform for AV2

November 20, 2024

Status:	Input document
Purpose:	Proposal
Author(s):	Xin Zhao, Aki Kuusela, Jaehong Chon, Alexis Tourapis, Krishna Rapaka, Felix Fernandes, Hilmi Egilmez, Alican Nalci, Yeqing Wu, Yunfei Zheng
Email(s):	xinzhao@apple.com
Source:	Apple Inc.

Abstract

In AVM v8.0.0, the core transform employs multi-stage full butterfly arithmetic operations for the Discrete Cosine Transform (DCT) and the Asymmetric Discrete Sine Transform (ADST), with 12-bit multipliers and bitdepth+8 range clippings between row and column inverse transforms. In this contribution, a redesign of the core transform, that is characterized by a) single-stage arithmetic, b) 8-bit multipliers, and c) 16-bit range clipping between the row and column inverse transforms, is proposed.

Experiments based on AVM v8.0.0 under the v7 Common Test Conditions (CTC v7) are reported as follows:
Average (w/o B2):

AI	0.01% (PSNR-Avg)	-0.01% (SSIM)	0.02% (VMAF)	0.01 (APSNR-Avg)
RA	-0.05% (PSNR-Avg)	-0.10% (SSIM)	-0.21% (VMAF)	-0.06% (APSNR -Avg)
LD	-0.06% (PSNR-Avg)	-0.16% (SSIM)	0.00% (VMAF)	-0.07% (APSNR-Avg)

It is reported that, the proposed core transforms reduce per-transform silicon area on average (from 4-point to 32-point) by 22**%** (up to 45%) while cutting data pipeline length (latency) by 45% (up to 57%).

It is reported that, when SIMD optimizations related to transforms are disabled, the proposed core transforms implementation on top of AVM v8.0.0 reduces average SW decoding runtimes by 13% for AI, 4% for RA and 9% for LD.

In v2 of this contribution, an alternative design of 32-point DCT-2 is described (other transforms are kept the same as v1), which reduces per-transform silicon area on average (from 4-point to 32-point) by 27**%** (up to 45%) while cutting data pipeline length (latency) by 38% (up to 57%).

<u>In v4 of this contribution, per the discussion in HW SG, the coding performance of the proposed method, excluding above feature c) 16-bit range clipping between the row and column inverse transforms, is reported (see below additional test #2). In this test, the AV1</u>

internal bit depth (bitdepth + 8) is used for the clipping between the row and column inverse transforms.

Introduction

In AV1 and the AVM v8.0.0, the core transform applies various transform types with their main features (in terms of size, arithmetic operations, and bit width of multipliers) listed in Table 2. The changes on transform coding from AV1 to AVM v8.0.0 are highlighted in the yellow cells, including the modified ADST transform types proposed in [1] and the data-driven transform [2].

Regarding the aspect ratio of transform block shapes, on top of AV1, which covers the 1:1, 1:2/2:1 and 1:4/4:1 ratios, the ratios of 1:8/8:1 and 1:16/16:1 are also supported in the AVM.

Regarding the combinations of transform types, up to 16 combinations of transform types are supported, which is same as in AV1.

In addition, in both the AV1 and the AVM inverse transform, the row transform is performed first, followed by the column transform. Clipping operations are performed between the inverse row transform and the inverse column transform, and the clipping range is $[2^{bd+7}, 2^{bd+7}-1]$, where bd is the internal operational bit depth of the decoder.

It is reported in this contribution that the transforms in AV1 and the AVM are costly for implementations with 1) multi-stage operations with clamping operations between stages and 2) large multipliers being the major complexity contributors.

Table 1: Core transform types in AV1

Tx name	Tx type	Size	Arithmetic	Multipliers bit width
DCT	DCT-2	4-/8-/16-/32-/64-point	Multi-stage BTF	Fwd: 10 ~ 13-bit, Inv: 12-bit
ADST	DST-7	4-point	Multi-stage BTF	Fwd: 12 ~ 13-bit, Inv: 12-bit
	DST-4	8-/16-point	Multi-stage BTF	Fwd: 12 ~ 13-bit, Inv: 12-bit
IDT	-	4-/8-/16-/32-point	Single Multiply	Fwd: 13-bit, Inv: 13-bit

Table 2: Core transform types in AVM v8.0.0

Tx name	Tx type	Size	Arithmetic	Multipliers Bit width
DCT	DCT-2	4-/8-/16-/32-/64-point	Multi-stage BTF	Fwd: 10 ~ 13-bit, Inv: 12-bit
ADST	DST-4	4-point	Multi-stage BTF	Fwd: 12 ~ 13-bit, Inv: 12-bit
	LGT (1.5)	8-point	Matrix multiply	Fwd: 12-bit, Inv: 8-bit
	DST-7	16-point	Matrix multiply	Fwd: 12-bit, Inv: 8-bit
	DDT	8-/16-point	Matrix multiply	Fwd: 14-bit, Inv: 9-bit
IDT	-	4-/8-/16-/32-point	Single Multiply	Fwd: 13-bit, Inv: 13-bit

Flow chart of the 32-point DCT-2 in AV1/AVM is shown below

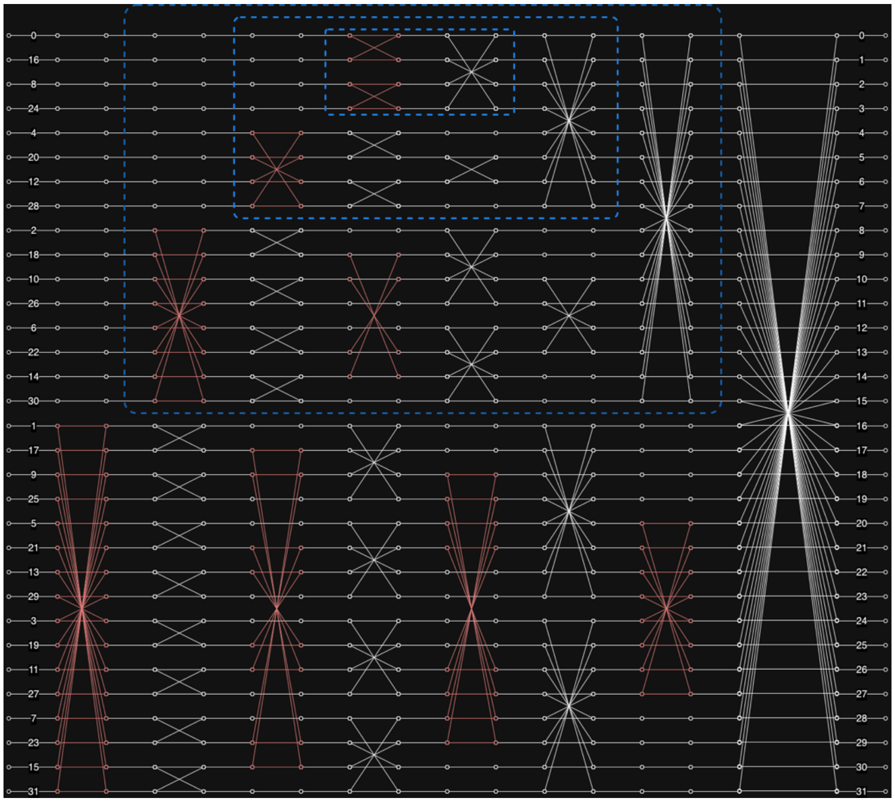

Figure 1: Flow chart of 32-point inverse DCT-2 in AV1/AVM

Experiment description

Algorithm description

A redesign of the core transform is proposed, with the main features summarized in Table 3.

Table 3: Proposed core transforms

Tx name	Tx type	Size	Arithmetic	Multipliers Bit width
DCT	DCT-2	4-/8-/16-/32-/64-point	BTF/Matrix multiply	Fwd/Inv: 8-bit
ADST	DST-4	4-point	BTF/Matrix multiply	Fwd/Inv: 8-bit
	LGT (1.5)	8-point	Matrix multiply	Fwd/Inv: 8-bit
	DST-7	16-point	Matrix multiply	Fwd/Inv: 8-bit
	DDT	4-/8-/16-point	Matrix multiply	Fwd/Inv: 8-bit
IDT	-	4-/8-/16-/32-point	Single Multiply	Fwd/Inv: 8-bit

The main features of the proposed core transforms are described as follows:

1. All the supported core transform types in the AVM can be implemented using direct matrix multiplication with 8-bit multipliers, while a fast butterfly implementation is also available for DCT-2 with no impact to the bitstream compared to direct matrix multiplication.
2. The intermediate clipping range is kept as 16 bit, i.e., $[-2^{15}, 2^{15}-1]$, regardless of the internal bit depth.

3. The transform order is kept unchanged, i.e., the column transform is performed first for the forward transform and the row transform is performed first for the inverse transform. This is the same with both AV1 and the current AVM (v8.0.0).
4. Quantization is kept unchanged and is the same as with the current AVM (v8.0.0).
5. The zero-out process of the transform coefficients for 64xN and Nx64 is kept unchanged and is again the same as with the current AVM (v8.0.0).

The transform kernels are listed as follows,

4-point DCT-2:

```
{ 64, 64, 64, 64 }
{ 83, 35,-35,-83 }
{ 64,-64,-64, 64 }
{ 35,-83, 83,-35 }
```

8-point DCT-2:

```
{ 64, 64, 64, 64, 64, 64, 64, 64 }
{ 89, 75, 50, 18,-18,-50,-75,-89 }
{ 83, 35,-35,-83,-83,-35, 35, 83 }
{ 75,-18,-89,-50, 50, 89, 18,-75 }
{ 64,-64,-64, 64, 64,-64,-64, 64 }
{ 50,-89, 18, 75,-75,-18, 89,-50 }
{ 35,-83, 83,-35,-35, 83,-83, 35 }
{ 18,-50, 75,-89, 89,-75, 50,-18 }
```

16-point DCT-2:

```
{ 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64 }
{ 90, 87, 80, 70, 57, 43, 26,  9, -9,-26,-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,-26, 26, 70, 90, 80, 43, -9,-57,-87 }
{ 83, 35,-35,-83,-83,-35, 35, 83, 83, 35,-35,-83,-83,-35, 35, 83 }
{ 80,  9,-70,-87,-26, 57, 90, 43,-43,-90,-57, 26, 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, 26,-80,-57, 57, 80,-26,-90, -9, 87, 43,-70 }
{ 64,-64,-64, 64, 64,-64,-64, 64, 64,-64,-64, 64, 64,-64,-64, 64 }
{ 57,-80,-26, 90, -9,-87, 43, 70,-70,-43, 87,  9,-90, 26, 80,-57 }
{ 50,-89, 18, 75,-75,-18, 89,-50,-50, 89,-18,-75, 75, 18,-89, 50 }
{ 43,-90, 57, 26,-87, 70,  9,-80, 80, -9,-70, 87,-26,-57, 90,-43 }
{ 35,-83, 83,-35,-35, 83,-83, 35, 35,-83, 83,-35,-35, 83,-83, 35 }
{ 26,-70, 90,-80, 43,  9,-57, 87,-87, 57, -9,-43, 80,-90, 70,-26 }
{ 18,-50, 75,-89, 89,-75, 50,-18,-18, 50,-75, 89,-89, 75,-50, 18 }
{  9,-26, 43,-57, 70,-80, 87,-90, 90,-87, 80,-70, 57,-43, 26, -9 }
```

32-point DCT-2 (only columns 0 ~ 16 are provided below because of page size limitation, columns 17 ~ 31 can be derived based on the symmetricity of DCT-2 transform):

```
{ 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64 }
{ 90, 90, 88, 85, 82, 78, 73, 67, 61, 54, 47, 39, 30, 22, 13,  4 }
{ 90, 87, 80, 70, 57, 43, 26,  9, -9,-26,-43,-57,-70,-80,-87,-90 }
{ 90, 82, 67, 47, 22, -4,-30,-54,-73,-85,-90,-88,-78,-61,-39,-13 }
{ 89, 75, 50, 18,-18,-50,-75,-89,-89,-75,-50,-18, 18, 50, 75, 89 }
{ 88, 67, 30,-13,-54,-82,-90,-78,-47, -4, 39, 73, 90, 85, 61, 22 }
{ 87, 57,  9,-43,-80,-90,-70,-26, 26, 70, 90, 80, 43, -9,-57,-87 }
{ 85, 47,-13,-67,-90,-73,-22, 39, 82, 88, 54, -4,-61,-90,-78,-30 }
{ 83, 35,-35,-83,-83,-35, 35, 83, 83, 35,-35,-83,-83,-35, 35, 83 }
{ 82, 22,-54,-90,-61, 13, 78, 85, 30,-47,-90,-67,  4, 73, 88, 39 }
{ 80,  9,-70,-87,-26, 57, 90, 43,-43,-90,-57, 26, 87, 70, -9,-80 }
{ 78, -4,-82,-73, 13, 85, 67,-22,-88,-61, 30, 90, 54,-39,-90,-47 }
{ 75,-18,-89,-50, 50, 89, 18,-75,-75, 18, 89, 50,-50,-89,-18, 75 }
{ 73,-30,-90,-22, 78, 67,-39,-90,-13, 82, 61,-47,-88, -4, 85, 54 }
{ 70,-43,-87,  9, 90, 26,-80,-57, 57, 80,-26,-90, -9, 87, 43,-70 }
{ 67,-54,-78, 39, 85,-22,-90,  4, 90, 13,-88,-30, 82, 47,-73,-61 }
```

```
{ 64,-64,-64, 64, 64,-64,-64, 64, 64,-64,-64, 64, 64,-64,-64, 64 }
{ 61,-73,-47, 82, 30,-88,-13, 90, -4,-90, 22, 85,-39,-78, 54, 67 }
{ 57,-80,-26, 90, -9,-87, 43, 70,-70,-43, 87,  9,-90, 26, 80,-57 }
{ 54,-85, -4, 88,-47,-61, 82, 13,-90, 39, 67,-78,-22, 90,-30,-73 }
{ 50,-89, 18, 75,-75,-18, 89,-50,-50, 89,-18,-75, 75, 18,-89, 50 }
{ 47,-90, 39, 54,-90, 30, 61,-88, 22, 67,-85, 13, 73,-82,  4, 78 }
{ 43,-90, 57, 26,-87, 70,  9,-80, 80, -9,-70, 87,-26,-57, 90,-43 }
{ 39,-88, 73, -4,-67, 90,-47,-30, 85,-78, 13, 61,-90, 54, 22,-82 }
{ 35,-83, 83,-35,-35, 83,-83, 35, 35,-83, 83,-35,-35, 83,-83, 35 }
{ 30,-78, 90,-61,  4, 54,-88, 82,-39,-22, 73,-90, 67,-13,-47, 85 }
{ 26,-70, 90,-80, 43,  9,-57, 87,-87, 57, -9,-43, 80,-90, 70,-26 }
{ 22,-61, 85,-90, 73,-39, -4, 47,-78, 90,-82, 54,-13,-30, 67,-88 }
{ 18,-50, 75,-89, 89,-75, 50,-18,-18, 50,-75, 89,-89, 75,-50, 18 }
{ 13,-39, 61,-78, 88,-90, 85,-73, 54,-30,  4, 22,-47, 67,-82, 90 }
{  9,-26, 43,-57, 70,-80, 87,-90, 90,-87, 80,-70, 57,-43, 26, -9 }
{  4,-13, 22,-30, 39,-47, 54,-61, 67,-73, 78,-82, 85,-88, 90,-90 }
```

64-point DCT-2 (columns 0 ~ 16 and 17 ~ 31 are provided below separately because of page size limitation, columns 32 ~ 63 can be derived based on the symmetricity of DCT-2 transform):

Columns 0 ~ 16:

```
{ 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64 }
{ 90, 90, 90, 89, 88, 87, 86, 84, 83, 81, 79, 76, 74, 71, 69, 66 }
{ 90, 90, 88, 85, 82, 78, 73, 67, 61, 54, 47, 39, 30, 22, 13,  4 }
{ 90, 88, 84, 79, 71, 62, 52, 41, 28, 15,  2,-11,-24,-37,-48,-59 }
{ 90, 87, 80, 70, 57, 43, 26,  9, -9,-26,-43,-57,-70,-80,-87,-90 }
{ 90, 84, 74, 59, 41, 20, -2,-24,-45,-62,-76,-86,-90,-89,-83,-71 }
{ 90, 82, 67, 47, 22, -4,-30,-54,-73,-85,-90,-88,-78,-61,-39,-13 }
{ 89, 79, 59, 33,  2,-28,-56,-76,-88,-90,-81,-62,-37, -7, 24, 52 }
{ 89, 75, 50, 18,-18,-50,-75,-89,-89,-75,-50,-18, 18, 50, 75, 89 }
{ 88, 71, 41,  2,-37,-69,-87,-89,-74,-45, -7, 33, 66, 86, 90, 76 }
{ 88, 67, 30,-13,-54,-82,-90,-78,-47, -4, 39, 73, 90, 85, 61, 22 }
{ 87, 62, 20,-28,-69,-89,-84,-56,-11, 37, 74, 90, 81, 48,  2,-45 }
{ 87, 57,  9,-43,-80,-90,-70,-26, 26, 70, 90, 80, 43, -9,-57,-87 }
{ 86, 52, -2,-56,-87,-84,-48,  7, 59, 88, 83, 45,-11,-62,-89,-81 }
{ 85, 47,-13,-67,-90,-73,-22, 39, 82, 88, 54, -4,-61,-90,-78,-30 }
{ 84, 41,-24,-76,-89,-56,  7, 66, 90, 69, 11,-52,-88,-79,-28, 37 }
{ 83, 35,-35,-83,-83,-35, 35, 83, 83, 35,-35,-83,-83,-35, 35, 83 }
{ 83, 28,-45,-88,-74,-11, 59, 90, 62, -7,-71,-89,-48, 24, 81, 84 }
{ 82, 22,-54,-90,-61, 13, 78, 85, 30,-47,-90,-67,  4, 73, 88, 39 }
{ 81, 15,-62,-90,-45, 37, 88, 69, -7,-76,-84,-24, 56, 90, 52,-28 }
{ 80,  9,-70,-87,-26, 57, 90, 43,-43,-90,-57, 26, 87, 70, -9,-80 }
{ 79,  2,-76,-81, -7, 74, 83, 11,-71,-84,-15, 69, 86, 20,-66,-87 }
{ 78, -4,-82,-73, 13, 85, 67,-22,-88,-61, 30, 90, 54,-39,-90,-47 }
{ 76,-11,-86,-62, 33, 90, 45,-52,-89,-24, 69, 83,  2,-81,-71, 20 }
{ 75,-18,-89,-50, 50, 89, 18,-75,-75, 18, 89, 50,-50,-89,-18, 75 }
{ 74,-24,-90,-37, 66, 81,-11,-88,-48, 56, 86,  2,-84,-59, 45, 89 }
{ 73,-30,-90,-22, 78, 67,-39,-90,-13, 82, 61,-47,-88, -4, 85, 54 }
{ 71,-37,-89, -7, 86, 48,-62,-79, 24, 90, 20,-81,-59, 52, 84,-11 }
{ 70,-43,-87,  9, 90, 26,-80,-57, 57, 80,-26,-90, -9, 87, 43,-70 }
{ 69,-48,-83, 24, 90,  2,-89,-28, 81, 52,-66,-71, 45, 84,-20,-90 }
{ 67,-54,-78, 39, 85,-22,-90,  4, 90, 13,-88,-30, 82, 47,-73,-61 }
{ 66,-59,-71, 52, 76,-45,-81, 37, 84,-28,-87, 20, 89,-11,-90,  2 }
```

Columns 17 ~ 31:

```
{ 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64 }
{ 62, 59, 56, 52, 48, 45, 41, 37, 33, 28, 24, 20, 15, 11,  7,  2 }
{ -4,-13,-22,-30,-39,-47,-54,-61,-67,-73,-78,-82,-85,-88,-90,-90 }
{-69,-76,-83,-87,-90,-90,-89,-86,-81,-74,-66,-56,-45,-33,-20, -7 }
{-90,-87,-80,-70,-57,-43,-26, -9,  9, 26, 43, 57, 70, 80, 87, 90 }
{-56,-37,-15,  7, 28, 48, 66, 79, 87, 90, 88, 81, 69, 52, 33, 11 }
{ 13, 39, 61, 78, 88, 90, 85, 73, 54, 30,  4,-22,-47,-67,-82,-90 }
{ 74, 87, 90, 83, 66, 41, 11,-20,-48,-71,-86,-90,-84,-69,-45,-15 }
```

```
{ 89, 75, 50, 18,-18,-50,-75,-89,-89,-75,-50,-18, 18, 50, 75, 89 }
{ 48, 11,-28,-62,-84,-90,-79,-52,-15, 24, 59, 83, 90, 81, 56, 20 }
{-22,-61,-85,-90,-73,-39,  4, 47, 78, 90, 82, 54, 13,-30,-67,-88 }
{-79,-90,-76,-41,  7, 52, 83, 90, 71, 33,-15,-59,-86,-88,-66,-24 }
{-87,-57, -9, 43, 80, 90, 70, 26,-26,-70,-90,-80,-43,  9, 57, 87 }
{-41, 15, 66, 90, 79, 37,-20,-69,-90,-76,-33, 24, 71, 90, 74, 28 }
{ 30, 78, 90, 61,  4,-54,-88,-82,-39, 22, 73, 90, 67, 13,-47,-85 }
{ 83, 86, 45,-20,-74,-90,-59,  2, 62, 90, 71, 15,-48,-87,-81,-33 }
{ 83, 35,-35,-83,-83,-35, 35, 83, 83, 35,-35,-83,-83,-35, 35, 83 }
{ 33,-41,-87,-76,-15, 56, 90, 66, -2,-69,-90,-52, 20, 79, 86, 37 }
{-39,-88,-73, -4, 67, 90, 47,-30,-85,-78,-13, 61, 90, 54,-22,-82 }
{-86,-74, -2, 71, 87, 33,-48,-90,-59, 20, 83, 79, 11,-66,-89,-41 }
{-80, -9, 70, 87, 26,-57,-90,-43, 43, 90, 57,-26,-87,-70,  9, 80 }
{-24, 62, 88, 28,-59,-89,-33, 56, 90, 37,-52,-90,-41, 48, 90, 45 }
{ 47, 90, 39,-54,-90,-30, 61, 88, 22,-67,-85,-13, 73, 82,  4,-78 }
{ 88, 56,-41,-90,-37, 59, 87, 15,-74,-79,  7, 84, 66,-28,-90,-48 }
{ 75,-18,-89,-50, 50, 89, 18,-75,-75, 18, 89, 50,-50,-89,-18, 75 }
{ 15,-79,-69, 33, 90, 28,-71,-76, 20, 90, 41,-62,-83,  7, 87, 52 }
{-54,-85,  4, 88, 47,-61,-82, 13, 90, 39,-67,-78, 22, 90, 30,-73 }
{-90,-33, 74, 69,-41,-88, -2, 87, 45,-66,-76, 28, 90, 15,-83,-56 }
{-70, 43, 87, -9,-90,-26, 80, 57,-57,-80, 26, 90,  9,-87,-43, 70 }
{ -7, 88, 33,-79,-56, 62, 74,-41,-86, 15, 90, 11,-87,-37, 76, 59 }
{ 61, 73,-47,-82, 30, 88,-13,-90, -4, 90, 22,-85,-39, 78, 54,-67 }
{ 90,  7,-90,-15, 88, 24,-86,-33, 83, 41,-79,-48, 74, 56,-69,-62 }
```

4-point DST-4:

```
{  18,  50,  75,  89 }
{  50,  89,  18, -75 }
{  75,  18, -89,  50 }
{  89, -75,  50, -18 }
```

8-point LGT (self-loop rate 1.5):

```
{  11,  28,  44,  58,  70,  79,  86,  89 }
{  34,  74,  89,  76,  39, -12, -58, -86 }
{  54,  89,  48, -34, -87, -66,  12,  79 }
{  71,  68, -41, -86,   1,  87,  38, -70 }
{  84,  17, -89,  10,  86, -35, -75,  58 }
{  88, -44, -44,  88, -44, -44,  88, -44 }
{  79, -83,  50,   6, -59,  86, -74,  29 }
{  50, -69,  81, -84,  78, -62,  40, -14 }
```

16-point DST-7:

```
{   8,  17,  25,  33,  41,  48,  55,  62,  67,  73,  77,  81,  84,  87,  88,  89 }
{  25,  48,  67,  81,  88,  88,  81,  67,  48,  25,   0, -25, -48, -67, -81, -88 }
{  41,  73,  88,  84,  62,  25, -17, -55, -81, -89, -77, -48,  -8,  33,  67,  87 }
{  55,  87,  81,  41, -17, -67, -89, -73, -25,  33,  77,  88,  62,   8, -48, -84 }
{  67,  88,  48, -25, -81, -81, -25,  48,  88,  67,   0, -67, -88, -48,  25,  81 }
{  77,  77,   0, -77, -77,   0,  77,  77,   0, -77, -77,   0,  77,  77,   0, -77 }
{  84,  55, -48, -87,  -8,  81,  62, -41, -88, -17,  77,  67, -33, -89, -25,  73 }
{  88,  25, -81, -48,  67,  67, -48, -81,  25,  88,   0, -88, -25,  81,  48, -67 }
{  89,  -8, -88,  17,  87, -25, -84,  33,  81, -41, -77,  48,  73, -55, -67,  62 }
{  87, -41, -67,  73,  33, -88,   8,  84, -48, -62,  77,  25, -89,  17,  81, -55 }
{  81, -67, -25,  88, -48, -48,  88, -25, -67,  81,   0, -81,  67,  25, -88,  48 }
{  73, -84,  25,  55, -89,  48,  33, -87,  67,   8, -77,  81, -17, -62,  88, -41 }
{  62, -89,  67,  -8, -55,  88, -73,  17,  48, -87,  77, -25, -41,  84, -81,  33 }
{  48, -81,  88, -67,  25,  25, -67,  88, -81,  48,   0, -48,  81, -88,  67, -25 }
{  33, -62,  81, -89,  84, -67,  41,  -8, -25,  55, -77,  88, -87,  73, -48,  17 }
{  17, -33,  48, -62,  73, -81,  87, -89,  88, -84,  77, -67,  55, -41,  25,  -8 }
```

4-point DDT:

```
{   2,  20,  72, 104 }
{  14,  68,  81, -69 }
{  67,  86, -61,  25 }
{ 108, -61,  27,  -8 }
```

8-point DDT:
```
{   4,   7,  15,  33,  65,  98, 106,  80 }
{   6,  14,  36,  77, 100,  45, -57, -98 }
{  22,  48,  85,  88,   0, -86, -23,  82 }
{  57,  94,  76, -26, -73,  34,  54, -66 }
{  96,  73, -43, -69,  55,  20, -71,  53 }
{ 103, -17, -80,  56,  15, -66,  75, -41 }
{  78, -79,   7,  56, -82,  79, -56,  26 }
{  56, -96,  98, -77,  54, -33,  19,  -6 }
```

16-point DDT:
```
{  12,  15,  19,  23,  30,  39,  51,  66,  78,  88,  94,  97,  93,  83,  68,  50 }
{  17,  23,  30,  38,  48,  61,  76,  87,  83,  59,  19, -30, -73, -99, -99, -76 }
{  37,  49,  60,  69,  75,  75,  61,  29, -18, -67, -96, -83, -28,  40,  84,  83 }
{  45,  60,  69,  73,  66,  40,  -8, -65, -91, -57,  21,  86,  81,   8, -69, -90 }
{  47,  60,  61,  49,  19, -29, -77, -83, -16,  75,  93,   3, -92, -74,  32,  97 }
{  60,  74,  64,  28, -31, -87, -82,   4,  88,  54, -55, -77,  29,  88,   3, -86 }
{  64,  70,  40, -19, -79, -78,  11,  92,  28, -85, -41,  82,  39, -83, -37,  83 }
{  82,  73,   3, -80, -91,  10,  94,  18, -84,  -5,  80, -17, -70,  47,  55, -68 }
{  89,  48, -53, -96,  -5,  89,  16, -83,  12,  75, -51, -43,  81, -14, -75,  67 }
{ 100,   9, -99, -45,  84,  36, -81,   4,  73, -60, -17,  76, -55, -21,  81, -56 }
{  92, -35, -91,  42,  71, -69, -22,  85, -60, -17,  77, -70,  11,  56, -83,  49 }
{  84, -71, -46,  88, -16, -67,  79, -22, -46,  84, -68,  15,  46, -83,  82, -40 }
{  69, -83,   2,  75, -78,  18,  50, -85,  81, -43,  -6,  53, -81,  88, -69,  32 }
{  50, -79,  47,  14, -60,  67, -37,  -6,  49, -80,  98, -99,  90, -71,  48, -19 }
{  51, -89,  73, -17, -45,  89,-103,  97, -83,  71, -56,  44, -31,  22, -11,   5 }
{  44, -95, 124,-126, 108, -81,  54, -30,  16,  -6,   1,   3,  -4,   5,  -3,   2 }
```

Flow chart of the proposed 32-point DCT-2 is shown below, wherein the blue boxes represent matrix multiplies that replace the butterfly used in the 32-point DCT-2 of AV1/AVM.

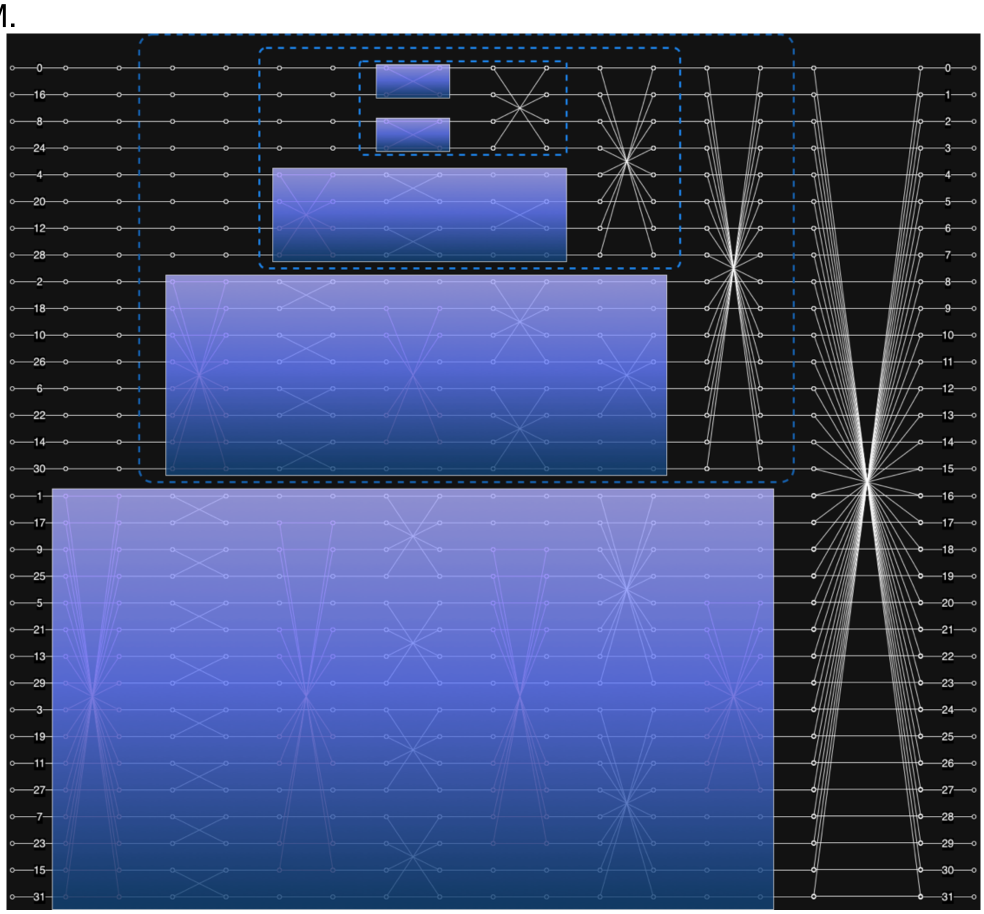

Figure 2: Flow chart of proposed 32-point inverse DCT-2

In v2 of this contribution, an alternative design of 32-point inverse DCT-2 is introduced. This design replaces the 16x16 matrix multiply used in the above proposed 32-point DCT-2 (illustrated in the bottom-left box in Figure 2) with a single stage butterfly operation with right shift operations, followed by two 8x8 matrix multiplications. Each 8x8 matrix multiplication can alternatively be implemented using butterfly operations without right shift operations. The single stage butterfly operation applies the same cosine function look-up table using 12-bit coefficients.

This revised 32-point inverse DCT-2 design significantly reduces the number of multiplication operations, resulting in a considerable reduction in area size, as shown in Table 4. The flow chart of this revised 32-point inverse DCT-2 is shown in Figure 3. Other transforms are kept the same as the v1 proposal.

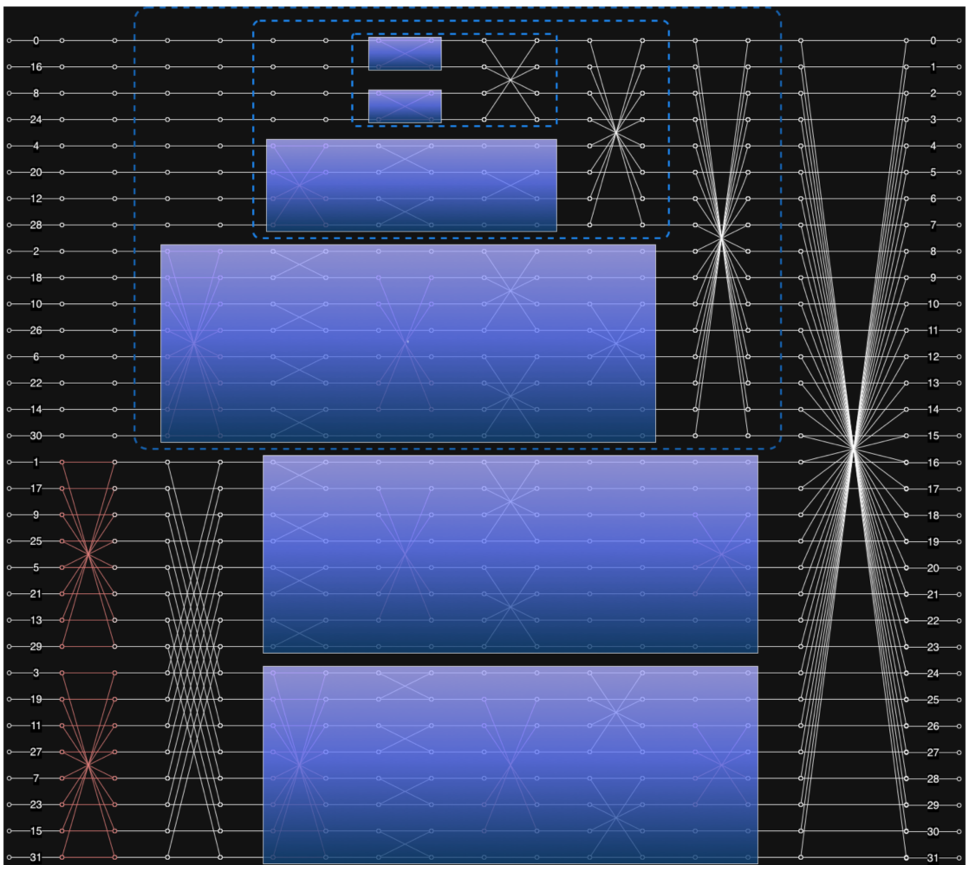

Figure 3: Flow chart of proposed 32-point inverse DCT-2 (V2)

Hardware complexity analysis

Silicon area and latency

A high-level synthesis tool was used to create RTL for each proposed AVM IDCT and its AV1 counterpart. All area numbers were normalized against the 1-D AV1 4-length IDCT at 666 MHz using a 3 nm technology node. The following table introduces terms Latency and Throughput (TP). Latency measures the data pipeline length of the transform hardware, and TP signals the rate at which new rows/columns can be fed into the hardware block. Reduced latency can be crucial for hardware performance as the transform is typically a part of a tight feedback loop. This is increasingly important in AVM that introduces the IST tool adding to the loop delay.

Table 4: Comparison of AV1 and proposed IDCT on area size (normalized) and latency

Module	Frequency	TP	AV1 area	AV1 latency	Proposal area	Proposal latency	Area saving (%)	Latency saving (%)
IDCT 4-Point	666	1	1.2	2	0.7	2	41.7%	0.0%
	666	2	1.15	2	0.64	3	44.3%	-50.0%
	666	4	1	5	0.59	3	41.0%	40.0%
	833	1	1.33	3	0.77	3	42.1%	0.0%
	833	2	1.27	4	0.7	3	44.9%	25.0%
	833	4	1.1	6	0.62	6	43.6%	0.0%
	1000	1	1.43	3	0.83	3	42.0%	0.0%
	1000	2	1.31	4	0.72	3	45.0%	25.0%
	1000	4	1.16	6	0.64	5	44.8%	16.7%
	1250	1	1.62	4	0.98	4	39.5%	0.0%
	1250	2	1.43	5	0.78	5	45.5%	0.0%
	1250	4	1.2	8	0.7	7	41.7%	12.5%
	2000	1	2.2	8	1.42	8	35.5%	0.0%
	2000	2	1.97	8	1.07	8	45.7%	0.0%
	2000	4	1.57	9	0.95	7	39.5%	22.2%
IDCT 8-Point	666	1	3.55	4	2.66	3	25.1%	25.0%
	666	2	3.22	6	2.37	3	26.4%	50.0%
	666	4	2.86	6	2.1	5	26.6%	16.7%
	833	1	4	5	2.88	3	28.0%	40.0%
	833	2	3.48	7	2.57	3	26.1%	57.1%
	833	4	3.12	7	2.24	7	28.2%	0.0%
	1000	1	4.29	6	3.14	4	26.8%	33.3%
	1000	2	3.75	7	2.76	7	26.4%	0.0%
	1000	4	3.29	11	2.36	7	28.3%	36.4%
	1250	1	4.89	8	3.56	5	27.2%	37.5%
	1250	2	4.18	9	3.07	5	26.6%	44.4%
	1250	4	3.55	12	2.54	8	28.5%	33.3%
	2000	1	6.94	11	5.04	9	27.4%	18.2%
	2000	2	5.81	18	4.18	8	28.1%	55.6%
	2000	4	4.6	18	3.28	10	28.7%	44.4%
IDCT 16-Point	666	1	9.76	7	6.71	3	31.3%	57.1%
	666	2	8.82	9	6.22	3	29.5%	66.7%
	666	4	7.5	11	5.87	8	21.7%	27.3%
	833	1	10.83	8	7.41	4	31.6%	50.0%
	833	2	9.65	10	7.07	5	26.7%	50.0%

	833	4	8.39	12	6.25	7	25.5%	41.7%
	1000	1	11.93	10	8.25	4	30.8%	60.0%
	1000	2	10.06	11	7.62	5	24.3%	54.5%
	1000	4	9.53	15	6.7	8	29.7%	46.7%
	1250	1	13.2	11	9.27	5	29.8%	54.5%
	1250	2	11.39	13	8.03	7	29.5%	46.2%
	1250	4	10.96	13	7.11	8	35.1%	38.5%
	2000	1	19.12	18	13.38	8	30.0%	55.6%
	2000	2	16.34	25	11.29	12	30.9%	52.0%
	2000	4	12.71	28	8.65	12	31.9%	57.1%
IDCT 32-Point (V1)	666	1	25.72	8	26.62	3	-3.5%	62.5%
	666	2	22.9	11	25.16	3	-9.9%	72.7%
	666	4	19.82	11	23.83	4	-20.2%	63.6%
	833	1	28.72	11	29.19	3	-1.6%	72.7%
	833	2	25.15	13	27.29	5	-8.5%	61.5%
	833	4	22.44	12	24.26	9	-8.1%	25.0%
	1000	1	31.36	11	32.42	4	-3.4%	63.6%
	1000	2	27.05	14	28.74	6	-6.2%	57.1%
	1000	4	22.95	16	26.09	10	-13.7%	37.5%
	1250	1	35.36	14	36.15	5	-2.2%	64.3%
	1250	2	31.22	20	32.06	7	-2.7%	65.0%
	1250	4	27.18	21	27.28	10	-0.4%	52.4%
	2000	1	49.9	21	52.21	9	-4.6%	57.1%
	2000	2	44.15	24	41.89	12	5.1%	50.0%
	2000	4	33.48	34	32.78	11	2.1%	67.6%
IDCT 32-Point (V2)	666	1	25.72	8	22.87	6	11.1%	25.0%
	666	2	22.9	11	20.44	6	10.7%	45.5%
	666	4	19.82	11	17.36	9	12.4%	18.2%
	833	1	28.72	11	25.93	6	9.7%	45.5%
	833	2	25.15	13	22.54	7	10.4%	46.2%
	833	4	22.44	12	18.81	15	16.2%	-25.0%
	1000	1	31.36	11	28.75	9	8.3%	18.2%
	1000	2	27.05	14	24.38	11	9.9%	21.4%
	1000	4	22.95	16	20.48	13	10.8%	18.8%
	1250	1	35.36	14	32.26	9	8.8%	35.7%
	1250	2	31.22	20	27.08	15	13.3%	25.0%
	1250	4	27.18	21	22.62	21	16.8%	0.0%
	2000	1	49.9	21	46.04	14	7.7%	33.3%
	2000	2	44.15	24	37.53	21	15.0%	12.5%
	2000	4	33.48	34	28.87	26	13.8%	23.5%
IDCT 64-Point	666	1	63.35	10	53.58	3	15.4%	70.0%
	666	2	57.01	13	51.01	3	10.5%	76.9%

	666	4	46.44	16	48.63	6	-4.7%	62.5%
	833	1	70.79	12	59.74	4	15.6%	66.7%
	833	2	62.52	16	55.83	4	10.7%	75.0%
	833	4	56.91	24	50.5	11	11.3%	54.2%
	1000	1	77.32	15	66.17	5	14.4%	66.7%
	1000	2	72.61	19	60.22	6	17.1%	68.4%
	1000	4	58.44	20	54.66	7	6.5%	65.0%
	1250	1	87.54	18	74.8	6	14.6%	66.7%
	1250	2	81.56	26	67.15	9	17.7%	65.4%
	1250	4	67.85	27	55.81	12	17.7%	55.6%
	2000	1	122.24	25	106.02	8	13.3%	68.0%
	2000	2	117.38	26	89.68	13	23.6%	50.0%
	2000	4	81.24	49	74.93	11	7.8%	77.6%

For 4/8/16/64-point inverse transforms, based on the collected data, we can conclude that our proposal reduces per-transform silicon area significantly, therefore providing a huge improvement for hardware implementations over AV1 core transforms and making the needed room for the additional IST in AVM to avoid hardware performance loss over AV1. For 32-point transform, over different clock frequencies and target throughput, there is an average 5.2% area increase for V1 proposal and 11.7% area saving for V2 proposal. The area and latency gains are held over a wide range of implementation target clock frequencies, which proves that the proposal is suitable for all chip categories from data center to battery operated devices. For the largest transform size, the proposed 32-point IDCT in v2 proposal significantly reduces the multiplications required compared to v1 and helps reach good area improvements over AV1 butterfly implementation.

Note that for each configuration in the table, multiple data points were collected via High-Level Synthesis parameter sweeping, and the lowest area was selected into the table.

Arithmetic operations

Table 5: Arithmetic operation counts of AVM 1-D inverse core transforms

Transform type	Transform size	Mult (12 bit coef.)	Add	Shift	Clamp
DCT-2	4-point	8	12	8	4
	8-point	20	36	10	16
	16-point	52	100	26	48
	32-point	132	260	66	128
	64-point	324	644	162	320
ADST/DDT	4-point	16	12	4	4
	8-point	64	56	8	8
	16-point	256	240	16	16

Table 6: Arithmetic operation counts of proposed 1-D inverse core transforms

Transform type	Transform size	Mult (8 bit coef.)	Add	Shift	Clamp

DCT-2	4-point	6	12	4	4
	8-point	22	36	8	8
	16-point	86	68	16	16
	32-point (v1)	342	404	32	32
	32-point (v2)	142	186	48	32
	64-point	854	808	32	64
ADST/DDT	4-point	16	12	4	4
	8-point	64	56	8	8
	16-point	256	240	16	16

For an implementation based on direct matrix multiplication, for a WxH transform block, wherein the top-left wxh coefficients are non-zero, and inverse transform is performed based on the order of row transform first and column transform second, the MACs per coefficient is calculated as follows,

$$\frac{w \cdot h}{H} + h$$

The worst-case scenario is 32x32 and 32x64, which requires 64 MACs per coefficient.

Intermediate bit depth
The proposed method applies 16-bit inputs for the inverse column transforms, and the multiplication operation applies 16-bit and 8-bit operands.

Quantization
No change on quantization

Implementation reference

The proposed changes are implemented on top of AVM-v8.0.0, link is following:
https://gitlab.com/xin_zhao/avm/-/tree/contrib/cwg-e230_core-tx

Results

The experiments are performed using CTC v7.0.0 [3] and anchor is AVM v8.0.0 [4]. Please note that the runtime metrics for the encoder and decoder may not be reliable, as the simulations were conducted on a cluster with heterogeneous nodes. In addition, SIMD has not yet been implemented for the proposed methods.

Table 7: Coding performance using CTC v7.0.0.

All Intra								
	PSNR				SSIM	VMAF	EncT	DecT
	Y	U	V	YUV-Avg	Y	Y	%	%
A1	0.04%	0.07%	0.05%	0.04%	-0.03%	0.05%	148%	114%
A2	0.05%	0.03%	-0.03%	0.05%	0.06%	0.16%	149%	117%
A3	-0.18%	-0.16%	0.17%	-0.17%	-0.23%	-0.35%	140%	101%
A4	0.00%	0.44%	0.53%	0.04%	0.00%	-0.01%	140%	106%
A5	0.00%	1.69%	0.03%	0.07%	-0.09%	0.11%	137%	100%

B1	0.02%	0.07%	0.09%	0.03%	0.05%	0.00%	142%	115%
Avg	**0.00%**	**0.18%**	**0.10%**	**0.01%**	**-0.01%**	**0.02%**	**144%**	**111%**
B2	0.10%	-0.33%	-0.34%	0.07%	-0.05%	0.61%	117%	109%
Random Access								
	PSNR				SSIM	VMAF	EncT	DecT
	Y	U	V	YUV-Avg	Y	Y	%	%
A1	-0.11%	-0.06%	-0.06%	-0.11%	-0.19%	-0.22%	81%	93%
A2	-0.02%	0.24%	-0.18%	-0.01%	-0.11%	-0.25%	126%	103%
A3	-0.12%	-0.08%	0.06%	-0.11%	-0.13%	-0.06%	123%	100%
A4	0.00%	-0.30%	-1.40%	-0.07%	-0.02%	-0.40%	119%	99%
A5	-0.07%	-0.57%	-0.50%	-0.11%	-0.14%	-0.44%	119%	102%
B1	0.02%	0.00%	-0.05%	0.02%	-0.04%	-0.03%	118%	95%
Avg	**-0.04%**	**-0.01%**	**-0.26%**	**-0.05%**	**-0.10%**	**-0.21%**	**115%**	**99%**
B2	-0.20%	-0.81%	-0.03%	-0.22%	0.10%	-0.20%	114%	99%
Low Delay								
	PSNR				SSIM	VMAF	EncT	DecT
	Y	U	V	YUV-Avg	Y	Y	%	%
A2	-0.05%	0.35%	0.68%	0.00%	-0.05%	0.00%	81%	97%
A3	-0.14%	0.10%	-0.17%	-0.14%	-0.30%	-0.21%	143%	111%
A4	-0.02%	0.64%	-0.83%	-0.02%	-0.06%	0.25%	130%	108%
A5	-0.16%	-1.48%	1.76%	-0.14%	-0.31%	0.60%	137%	114%
B1	-0.13%	0.31%	-0.13%	-0.11%	-0.24%	-0.20%	82%	92%
Avg	**-0.09%**	**0.18%**	**0.26%**	**-0.06%**	**-0.16%**	**0.00%**	**99%**	**101%**
B2	-0.04%	-0.18%	-0.14%	-0.05%	0.38%	0.09%	113%	101%

Table 8: Coding performance of alternative design (v2) of 32-point DCT using CTC v7.0.0.

All Intra								
	PSNR				SSIM	VMAF	EncT	DecT
	Y	U	V	YUV-Avg	Y	Y	%	%
A1	0.03%	-0.09%	0.10%	0.03%	-0.01%	0.00%	161%	129%
A2	0.06%	0.11%	-0.02%	0.06%	0.06%	0.07%	157%	124%
A3	0.02%	0.33%	0.44%	0.05%	-0.01%	-0.23%	153%	120%
A4	0.04%	0.22%	0.00%	0.04%	-0.01%	0.35%	150%	120%
A5	0.04%	1.09%	0.33%	0.09%	-0.02%	-0.22%	154%	121%
B1	0.06%	0.05%	0.03%	0.06%	0.07%	0.16%	151%	123%
Avg	**0.05%**	**0.19%**	**0.10%**	**0.05%**	**0.03%**	**0.04%**	**155%**	**123%**
B2	0.08%	-0.67%	-0.16%	0.04%	-0.08%	0.64%	120%	114%
Random Access								
	PSNR				SSIM	VMAF	EncT	DecT
	Y	U	V	YUV-Avg	Y	Y	%	%
A1	-0.07%	-0.13%	0.12%	-0.06%	-0.17%	-0.15%	83%	96%
A2	0.03%	0.18%	0.00%	0.04%	-0.02%	-0.06%	128%	104%
A3	-0.08%	-0.05%	0.18%	-0.07%	-0.26%	-0.21%	122%	96%
A4	-0.09%	-0.43%	0.99%	-0.06%	-0.08%	-0.02%	119%	105%
A5	-0.10%	-0.19%	-0.61%	-0.13%	-0.22%	-0.39%	127%	107%
B1	-0.04%	-0.09%	0.00%	-0.04%	-0.14%	-0.19%	119%	104%
Avg	**-0.03%**	**-0.04%**	**0.11%**	**-0.03%**	**-0.09%**	**-0.14%**	**117%**	**102%**
B2	-0.23%	0.01%	-0.25%	-0.22%	-0.12%	-0.27%	117%	104%
Low Delay								
	PSNR				SSIM	VMAF	EncT	DecT
	Y	U	V	YUV-Avg	Y	Y	%	%
A2	-0.05%	0.38%	0.40%	-0.01%	-0.12%	-0.09%	82%	99%
A3	-0.19%	0.11%	-0.31%	-0.18%	-0.31%	-0.44%	131%	106%
A4	0.07%	-0.53%	0.81%	0.08%	0.10%	0.33%	129%	110%
A5	-0.12%	-0.43%	1.62%	-0.06%	-0.21%	0.30%	129%	102%
B1	-0.11%	0.29%	-0.18%	-0.10%	-0.29%	-0.12%	84%	96%
Avg	**-0.08%**	**0.13%**	**0.31%**	**-0.05%**	**-0.17%**	**-0.07%**	**98%**	**101%**
B2	-0.12%	0.21%	-0.24%	-0.11%	0.03%	-0.80%	110%	99%

A software decoding runtime test was conducted on a standalone PC with transform-related SIMD disabled for both the anchor and the test configuration. The anchor is AVM v8.0.0 with transform-related SIMD disabled. 'V1' represents the proposed core transform from the first version of this proposal, while 'V2' represents the core transform redesigned with the 32-point DCT described in the second version of this proposal.

The runtime comparison results are summarized in Table 9. Without SIMD optimizations related to transform, the proposed core transform achieves average decoding runtime reductions of 13% for AI, 4% for RA, and 9% for LD. Additionally, with the redesigned 32-point DCT (V2), the average decoding runtime savings are 14% for AI, 5% for RA, and 8% for LD.

Table 9: Comparisons on decoding runtime.

	All-Intra		Random-Access		Low-Delay	
	V1	V2	V1	V2	V1	V2
A1	81%	81%	94%	94%	-	-
A2	88%	87%	94%	94%	90%	92%
A3	88%	88%	99%	97%	92%	92%
A4	88%	88%	94%	94%	89%	91%
A5	87%	84%	98%	96%	93%	92%
B1	89%	88%	97%	96%	93%	94%
Avg	87%	86%	96%	95%	91%	92%
B2	89%	89%	95%	94%	92%	94%

Additional results

The following additional experiments are performed using CTC v7.0.0 [3] and anchor is AVM v8.0.0 [4].

In the additional Test #1, the coding performance of the proposed methods, using "column transform first, row transform second" is reported. Note that, in AV1 and the initial proposal (V1), the inverse transform order follows "row transform first, column transform second".

Table 10: Coding performance of proposed methods using column/row transform order.

All Intra						
	PSNR				SSIM	nVMAF
	Y	U	V	YUV-Avg	Y	Y
A1	0.04%	0.02%	0.00%	0.03%	-0.02%	0.04%
A2	0.06%	0.03%	0.23%	0.06%	0.05%	0.10%
A3	0.00%	-0.01%	0.30%	0.01%	-0.04%	-0.03%
A4	0.00%	-0.34%	0.13%	-0.01%	-0.03%	0.23%
A5	0.00%	0.88%	-0.14%	0.03%	-0.04%	0.14%
B1	0.08%	0.23%	0.03%	0.08%	0.09%	0.17%
Avg	0.04%	0.08%	0.13%	0.04%	0.02%	0.10%
B2	0.08%	-0.45%	-0.10%	0.05%	-0.13%	-0.10%
Random Access						
	PSNR				SSIM	nVMAF
	Y	U	V	YUV-Avg	Y	Y
A1	-0.08%	0.03%	0.06%	-0.07%	-0.17%	-0.14%
A2	-0.01%	0.05%	0.22%	0.00%	-0.10%	-0.10%
A3	-0.06%	0.16%	0.52%	-0.03%	-0.07%	0.10%
A4	0.07%	0.27%	0.47%	0.10%	-0.01%	0.30%
A5	0.00%	-1.15%	-0.97%	-0.09%	-0.17%	0.02%
B1	-0.06%	-0.10%	-0.01%	-0.06%	-0.11%	-0.16%
Avg	-0.03%	-0.03%	0.14%	-0.02%	-0.10%	-0.04%
B2	-0.16%	-0.57%	-0.30%	-0.19%	-0.02%	-0.12%
Low Delay						
	PSNR				SSIM	nVMAF

	Y	U	V	YUV-Avg	Y	Y
A2	-0.07%	0.23%	0.25%	-0.04%	-0.12%	-0.04%
A3	-0.15%	0.12%	-1.03%	-0.17%	-0.30%	-0.25%
A4	-0.09%	-0.62%	-0.05%	-0.11%	-0.22%	0.05%
A5	-0.01%	0.03%	0.29%	0.00%	-0.04%	0.25%
B1	-0.10%	0.24%	0.09%	-0.08%	-0.24%	-0.31%
Avg	-0.09%	0.09%	-0.04%	-0.08%	-0.18%	-0.09%
B2	-0.11%	0.22%	0.10%	-0.08%	-0.06%	-0.32%

In the additional Test #2, the coding performance of the proposed methods, excluding changes from "16-bit range clipping between the row and column inverse transforms", is reported. In this test, the AV1 internal bit depth (bitdepth + 8) is used for the clipping between the row and column inverse transforms.

Table 11: Coding performance using the same internal bit depth as AV1.

All Intra						
	PSNR				SSIM	nVMAF
	Y	U	V	YUV-Avg	Y	Y
A1	0.01%	-0.07%	0.01%	0.01%	-0.05%	-0.02%
A2	0.02%	0.04%	0.07%	0.02%	0.02%	0.02%
A3	-0.19%	-0.16%	0.24%	-0.18%	-0.22%	-0.24%
A4	0.00%	0.41%	0.36%	0.03%	-0.01%	0.06%
A5	-0.01%	1.46%	-0.21%	0.05%	-0.08%	0.11%
B1	0.00%	0.25%	-0.09%	0.00%	0.04%	0.03%
Avg	-0.02%	0.18%	0.07%	-0.01%	-0.03%	-0.01%
B2	0.06%	-0.37%	-0.36%	0.02%	-0.06%	0.09%
Random Access						
	PSNR				SSIM	nVMAF
	Y	U	V	YUV-Avg	Y	Y
A1	-0.13%	-0.21%	0.09%	-0.11%	-0.19%	-0.23%
A2	-0.03%	0.16%	-0.03%	-0.02%	-0.12%	-0.15%
A3	-0.12%	-0.07%	-0.14%	-0.13%	-0.10%	-0.05%
A4	0.10%	-0.35%	0.02%	0.09%	-0.04%	-0.13%
A5	-0.07%	-0.65%	-0.57%	-0.11%	-0.19%	-0.36%
B1	-0.05%	0.22%	0.09%	-0.04%	-0.13%	0.00%
Avg	-0.05%	-0.03%	-0.04%	-0.05%	-0.13%	-0.14%
B2	-0.29%	-0.93%	0.00%	-0.31%	0.07%	-0.49%
Low Delay						
	PSNR				SSIM	nVMAF
	Y	U	V	YUV-Avg	Y	Y
A2	-0.08%	0.35%	0.37%	-0.04%	-0.13%	-0.27%
A3	-0.15%	-0.19%	-0.40%	-0.16%	-0.29%	-0.24%
A4	0.12%	0.23%	-0.56%	0.11%	0.04%	0.28%
A5	-0.12%	-1.29%	1.60%	-0.10%	-0.25%	0.36%
B1	-0.10%	0.18%	0.20%	-0.07%	-0.22%	-0.18%
Avg	-0.07%	0.07%	0.19%	-0.06%	-0.16%	-0.12%
B2	-0.03%	-0.21%	-0.11%	-0.04%	0.32%	0.03%

References

[1] A. Segui and J. Sole, "ADST Types Replacement," Alliance for Open Media, Codec Working Group,Document: CWG-D009.

[2] L. Lu and D. Mukherjee, "Data-Driven Transforms for Inter Coding," Alliance for Open Media, Codec Working Group,Document: CWG-E106.

[3] X. Zhao, R. Lei, A. Norkin, T. Daede, A. Tourapis, "AOM Common Test Conditions v7.0," Alliance for Open Media, Codec Working Group, Document: CWG-E083i.

[4] https://gitlab.com/AOMediaCodec/avm/-/tags/research-v8.0.0