

AHG16: COMBINATION OF LFNST WITH TRANSFORM SKIP (JVET-Q0106)



Brian Heng, Tim Hellman, Minhua Zhou, Wade Wan

Broadcom Inc.

LFNST Conditions – JVET-Q0041 Draft Spec Text

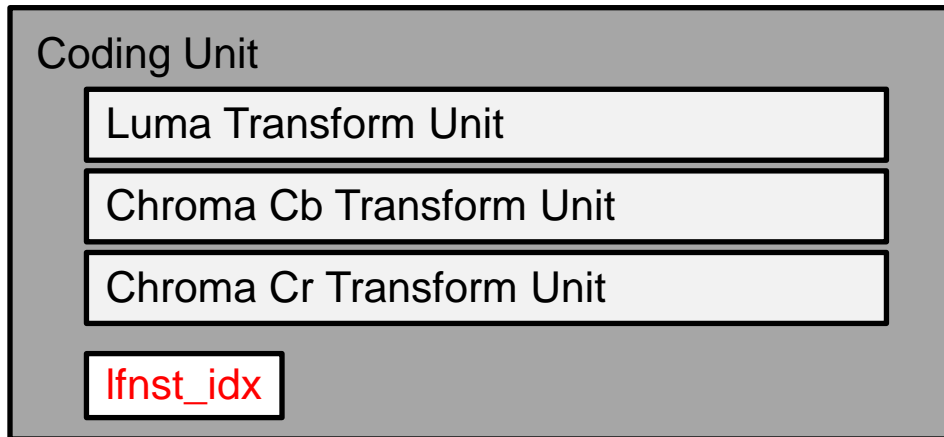
Not allowed with luma transform skip

if(Min(lfnstWidth, lfnstHeight) >= 4 && sps_lfnst_enabled_flag == 1 && CuPredMode[chType][x0][y0] == MODE_INTRA && transform_skip_flag[x0][y0][0] == 0 && (treeType == DUAL_TREE_CHROMA !intra_mip_flag[x0][y0] Min(lfnstWidth, lfnstHeight) >= 16) && Max(cbWidth, cbHeight) <= MaxTbSizeY) {
if((IntraSubPartitionsSplitType != ISP_NO_SPLIT LfnstDcOnly == 0) && LfnstZeroOutSigCoeffFlag == 1)
lfnst_idx
}

- LFNST is not allowed if luma uses transform skip.
- JVET-P0058 introduced transform skip for chroma TU.
- No LFNST requirements added for chroma transform skip.

LFNST Signalling and Zero-Out Conditions

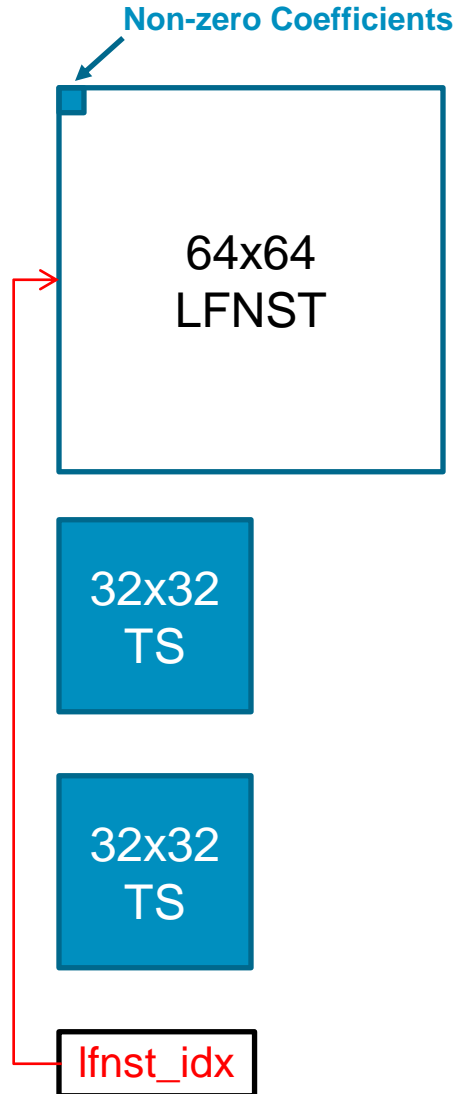
- LfnstZeroOutSigCoeffFlag Condition:
 - At most 16 (4x4) non-zero coefficients allowed in a LFNST transform unit.
- Where is lfnst_idx inserted?
 - Sent at the end of the coding unit (i.e. after transform coefficients)



Problem Statements

- Increased buffering and memory cost.
- Increased latency.
 - Dequantization (first step in pipeline) depends on lfnst_idx at end of CU.
 - JVET-P0365: Disable scaling lists for LFNST.
- Design inconsistency

Mixed LFNST / Transform Skip CUs – Buffering and Latency



- LFNST zero-out conditions don't apply to transform skip TUs.
- Chroma transform skip currently allowed in LFNST CUs.
- Up to 2048 chroma coefficients buffered before determining lfnst_idx for the luma TU.
- Also increases latency for processing all TUs.

Mixed LFNST / Transform Skip CUs – Design Inconsistency

if(Min(lfnstWidth, lfnstHeight) >= 4 && sps_lfnst_enabled_flag == 1 && CuPredMode[chType][x0][y0] == MODE_INTRA && transform_skip_flag[x0][y0][0] == 0 && (treeType == DUAL_TREE_CHROMA !intra_mip_flag[x0][y0] Min(lfnstWidth, lfnstHeight) >= 16) && Max(cbWidth, cbHeight) <= MaxTbSizeY) {
if((IntraSubPartitionsSplitType != ISP_NO_SPLIT LfnstDcOnly == 0) && LfnstZeroOutSigCoeffFlag == 1)
lfnst_idx
}

- Why prevent LFNST with luma transform skip but not chroma?

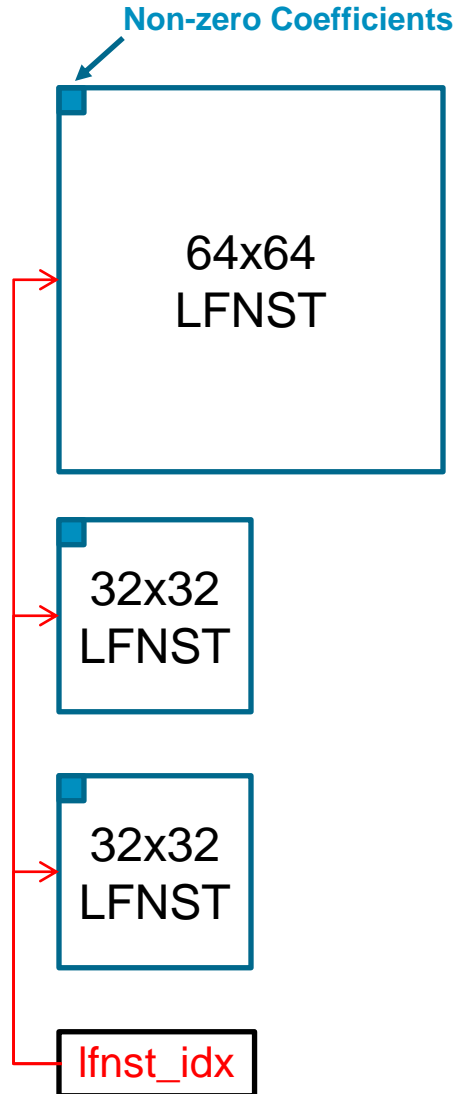
Decide whether combinations of LFNST with transform skip will be allowed or not and make the logic consistent.

Proposed Solution

- Prevent LFNST with Transform Skip
 - Add chroma transform skip conditions to LFNST logic.
 - Fix luma transform skip condition.
 - Only apply to luma condition to CUs that actually contain luma.

```
if( Min( lfnstWidth, lfnstHeight ) >= 4 &&  
    sps_lfnst_enabled_flag == 1 &&  
    CuPredMode[ chType ][ x0 ][ y0 ] == MODE_INTRA &&  
    (treeType == DUAL_TREE_CHROMA || transform_skip_flag[ x0 ][ y0 ][ 0 ] == 0) &&  
    (treeType == DUAL_TREE_LUMA || (transform_skip_flag[ x0 ][ y0 ][ 1 ] == 0 && transform_skip_flag[ x0 ][ y0 ][ 2 ] == 0)) &&  
    ( treeType == DUAL_TREE_CHROMA || !intra_mip_flag[ x0 ][ y0 ] || Min( lfnstWidth, lfnstHeight ) >= 16 ) &&  
    Max( cbWidth, cbHeight ) <= MaxTbSizeY ) {  
  
    if( ( IntraSubPartitionsSplitType != ISP_NO_SPLIT || LfnstDcOnly == 0 ) && LfnstZeroOutSigCoeffFlag == 1 )  
  
        lfnst_idx  
  
}
```

Without Mixed LFNST / Transform Skip CUs



- Zero-out conditions limit total coefficients to 48 (16 per Y/Cb/Cr) in any LFNST CU.
- Any CU with more than 48 coefficients is automatically not LFNST.
- Minimal buffering and delay.
- More consistent design

Experimental Results

- VTM7.0 and CTC

	All Intra Main10				
	Over VTM7.0				
	Y	U	V	EncT	DecT
Class A1	0.00%	-0.03%	0.05%	97%	94%
Class A2	-0.01%	-0.01%	0.01%	105%	106%
Class B	0.00%	0.00%	0.00%	99%	109%
Class C	0.00%	-0.02%	0.05%	93%	99%
Class E	0.00%	0.00%	-0.03%	101%	103%
Overall	0.00%	-0.01%	0.02%	98%	103%
Class D	0.00%	0.03%	-0.01%	95%	107%
Class F (optional)	-0.01%	0.01%	0.00%	99%	91%
	Random Access Main 10				
	Over VTM7.0				
	Y	U	V	EncT	DecT
Class A1	0.00%	-0.01%	0.00%	103%	98%
Class A2	-0.01%	0.11%	0.08%	97%	96%
Class B	0.02%	0.06%	0.04%	94%	89%
Class C	0.02%	0.07%	0.08%	97%	92%
Class E					
Overall	0.01%	0.06%	0.05%	97%	93%
Class D	-0.05%	-0.22%	-0.06%	96%	92%
Class F (optional)	0.01%	0.00%	-0.04%	98%	99%

Summary

- Allowing mixed LFNST / transform skip CUs significantly increases both implementation cost and processing latency, and makes the overall design inconsistent.
- Proposed solution resolves the issue by adding conditions to prevent LFNST when either luma or chroma uses transform skip.
- Experimental results reveal that the suggested changes do not harm coding efficiency.

Thanks to ByteDance for cross-checking