EXHIBIT 20

UNITED STATES PATENT NO. 9,036,701

CLAIM CHART FOR INFRINGEMENT OF CLAIMS 1, 8, AND 15 BY HISENSE ACCUSED PRODUCTS

As demonstrated in the chart below, The Hisense Respondents. ("Hisense") directly and indirectly infringe the asserted claims of US 9,036,701 (the "'701 Patent"). Hisense directly infringes, contributes to the infringement of, and/or induces infringement of the '701 Patent by making, using, selling, offering for sale, and/or importing into the United States the Accused Products that are covered by one or more claims of the '701 Patent. The Accused Products are devices that decode H.265-compliant video. For example, the Hisense 43A7N is a representative product for other Hisense devices that decode H.265-compliant video.

The Hisense 43A7N contains at least one video decoder that helps decode H.265-compliant video.[1] While evidence from the Hisense 43A7N is specifically charted herein, the evidence and contentions charted herein apply equally to the other Hisense Accused Products that decode H.265-compliant video. The example '701 Patent Accused Products are listed in NOK1448_00002051. The '701 Accused Products are also identified in Hisense's Response to Interrogatory No. 1 and all supplements thereto.

No part of this exemplary chart construes, or is intended to construe, the specification, file history, or claims of the '701 Patent. Moreover, this exemplary chart does not limit, and is not intended to limit, Nokia's infringement positions or contentions.

The following infringement chart includes exemplary citations to ITU-T Rec. H.265 (12/2016) High efficiency video coding (available at NOK1448_00002414) (the "H.265 Standard"). The cited functionality has been included in editions of the H.265 Standard since April 2013 and remains in current editions of the H.265 Standard. Any Hisense device that decodes H.265-compliant video practices the claims of the '701 Patent. Thus, the Accused Products each practice the H.265 Standard and are covered by claims of the '701 Patent.

Nokia contends each of the following limitations is met literally, and, to the extent a limitation is not met literally, it is met under the doctrine of equivalents.[2]

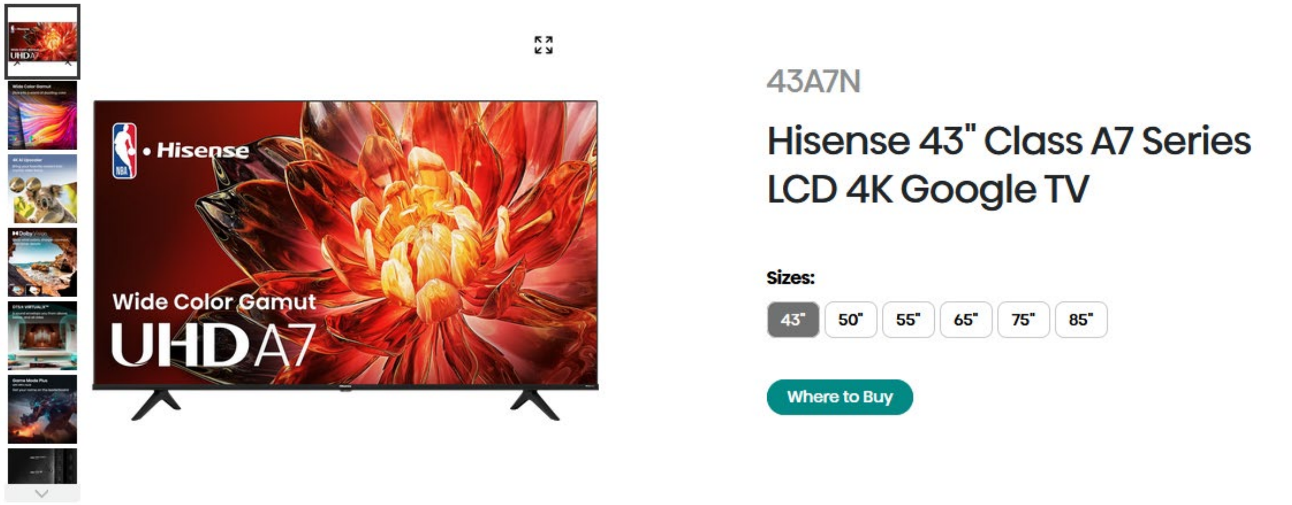
---

[1] *See, e.g.*, Ex. 92, Hisense 43A7N User Manual, available at https://www.hisense-usa.com/televisions/hisense-43-class-a7-series-4k-wide-color-gamut-google-tv-43a7n.

[2] This claim chart is based on the information currently available to Nokia and is intended to be exemplary in nature. Nokia reserves all rights to update and elaborate its infringement positions, including as Nokia obtains additional information during the course of discovery.

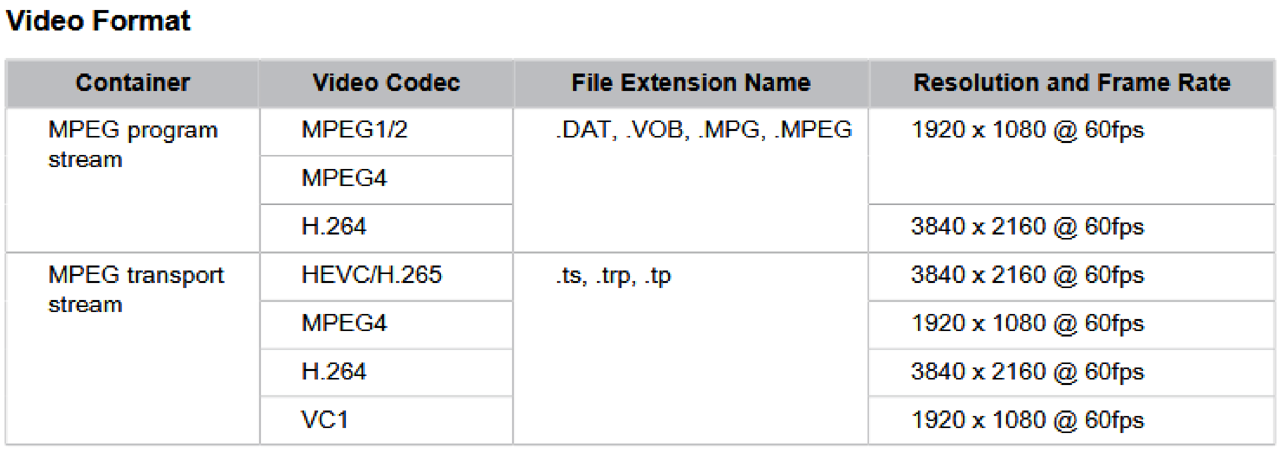
EXHIBIT 20
UNITED STATES PATENT NO. 9,036,701
CLAIM CHART FOR INFRINGEMENT OF CLAIMS 1, 8, AND 15 BY HISENSE ACCUSED PRODUCTS

1. Claim 1

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
1. [Pre] A method comprising:	Each of the Accused Products, such as the Hisense 43A7N, performs a method comprising the limitations below. For example, and without limitation, the Hisense 43A7N uses hardware-accelerated video decoding and includes Mediatek MT9602 Processor.  https://www.hisense-usa.com/televisions/hisense-43-class-a7-series-4k-wide-color-gamut-google-tv-43a7n (last accessed March 29, 2025).

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
	 **Display**: 42.5 in, VA, Direct LED, 3840 x 2160 pixels **Viewing angles (H/V)**: 178 ° / 178 ° **Brightness**: 275 cd/m² **Static contrast**: 4000 : 1 **Refresh rate**: 50 Hz / 60 Hz **Frame interpolation**: 120 MR (Motion Rate) **TV tuner**: Analog (NTSC/PAL/SECAM), ATSC, Clear QAM **SoC**: MediaTek MT9602 **CPU**: ARM Cortex-A53, 1500 MHz, **Cores**: 4 **Dimensions**: 963 x 560 x 74 mm **Weight**: 6.8 kg https://www.displayspecifications.com/en/model/234a3f3b

EXHIBIT 20
UNITED STATES PATENT NO. 9,036,701
CLAIM CHART FOR INFRINGEMENT OF CLAIMS 1, 8, AND 15 BY HISENSE ACCUSED PRODUCTS

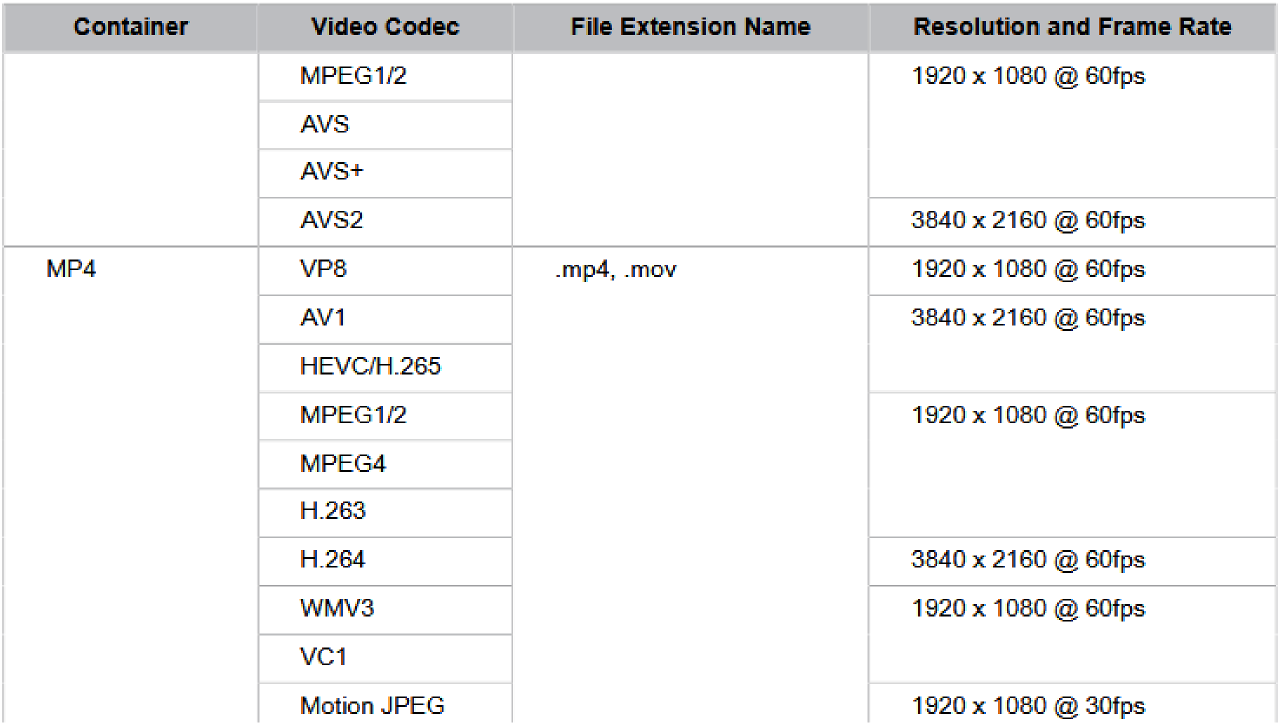
U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
	

Video Format

Container	Video Codec	File Extension Name	Resolution and Frame Rate
MPEG program stream	MPEG1/2	.DAT, .VOB, .MPG, .MPEG	1920 x 1080 @ 60fps
	MPEG4		
	H.264		3840 x 2160 @ 60fps
MPEG transport stream	HEVC/H.265	.ts, .trp, .tp	3840 x 2160 @ 60fps
	MPEG4		1920 x 1080 @ 60fps
	H.264		3840 x 2160 @ 60fps
	VC1		1920 x 1080 @ 60fps

EXHIBIT 20
UNITED STATES PATENT NO. 9,036,701
CLAIM CHART FOR INFRINGEMENT OF CLAIMS 1, 8, AND 15 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS

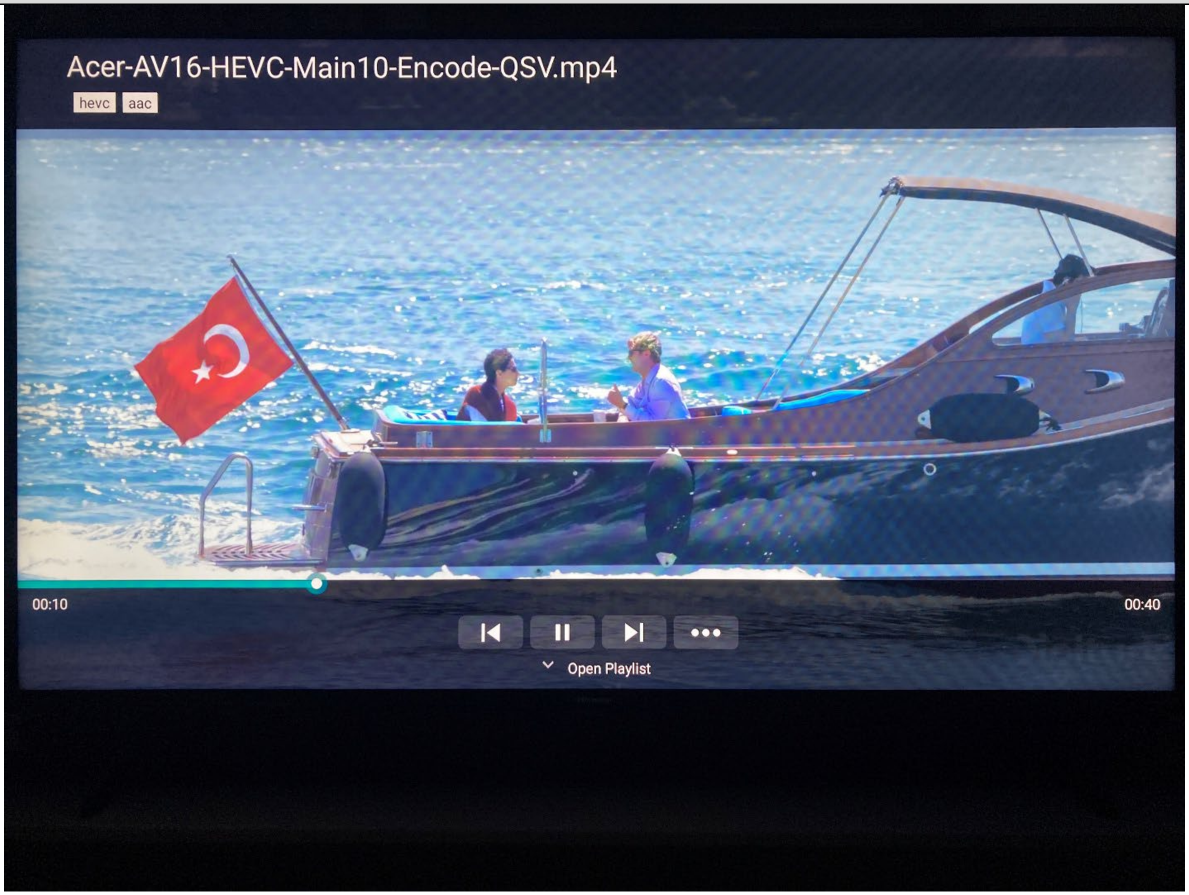


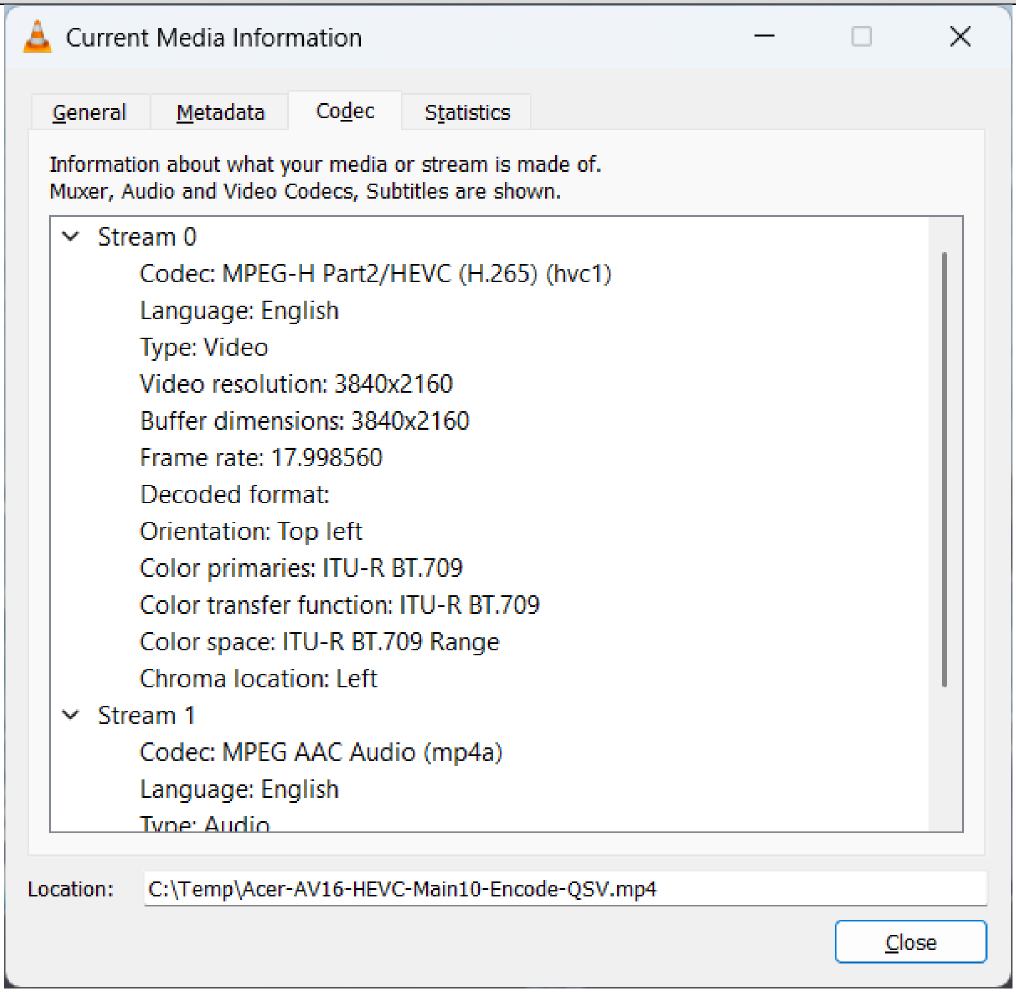
Container	Video Codec	File Extension Name	Resolution and Frame Rate
	MPEG1/2		1920 x 1080 @ 60fps
	AVS		
	AVS+		
	AVS2		3840 x 2160 @ 60fps
MP4	VP8	.mp4, .mov	1920 x 1080 @ 60fps
	AV1		3840 x 2160 @ 60fps
	HEVC/H.265		
	MPEG1/2		1920 x 1080 @ 60fps
	MPEG4		
	H.263		
	H.264		3840 x 2160 @ 60fps
	WMV3		1920 x 1080 @ 60fps
	VC1		
	Motion JPEG		1920 x 1080 @ 30fps

Source: Hisense 43A7N User Manual, at 45-46. Downloaded from https://www.hisense-usa.com/televisions/hisense-43-class-a7-series-4k-wide-color-gamut-google-tv-43a7n (last accessed March 29, 2025).

For example, a Hisense 43A7N was used to playback an H.265-compliant video.

EXHIBIT 20
UNITED STATES PATENT NO. 9,036,701
CLAIM CHART FOR INFRINGEMENT OF CLAIMS 1, 8, AND 15 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
	 Source: Picture of H.265-complaint video playback on Hisense 43A7N

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
	 Current Media Information Information about what your media or stream is made of. Muxer, Audio and Video Codecs, Subtitles are shown. Stream 0 Codec: MPEG-H Part2/HEVC (H.265) (hvc1) Language: English Type: Video Video resolution: 3840x2160 Buffer dimensions: 3840x2160 Frame rate: 17.998560 Decoded format: Orientation: Top left Color primaries: ITU-R BT.709 Color transfer function: ITU-R BT.709 Color space: ITU-R BT.709 Range Chroma location: Left Stream 1 Codec: MPEG AAC Audio (mp4a) Language: English Type: Audio Location: C:\Temp\Acer-AV16-HEVC-Main10-Encode-QSV.mp4 Source: Media Information for video decoded by Hisense 43A7N.

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
	For example, and without limitation, the H.265 Standard specifies the following regarding the decoding process. Each of the Accused Products performs a method comprising the limitations below. **3 Definitions** For the purposes of this Recommendation | International Standard, the following definitions apply. . . . **3.12 bitstream**: A sequence of bits, **. . .** , that forms the representation of *coded pictures* and associated data forming one or more coded video sequences *(CVSs)*. . . . **3.25 coded picture**: A *coded representation* of a *picture* **. . .** . . . **3.44 decoding process**: The process specified in this Specification that reads a *bitstream* and derives *decoded pictures* from it. ITU-T Rec. H.265 (12/2016) High efficiency video coding at pp. 4 – 7.
[a] determining a frequency of occurrence threshold based on an expected frequency of occurrence of syntax elements in a bit stream;	Each of the Accused Products, such as the Hisense 43A7N, performs a method comprising determining a frequency of occurrence threshold based on an expected frequency of occurrence of syntax elements in a bit stream. For example, and without limitation, the H.265 Standard specifies the following regarding the decoding process. Each of the Accused Products performs a method comprising determining a frequency of occurrence threshold based on an expected frequency of occurrence of syntax elements in a bit stream.

Exhibit 20

United States Patent No. 9,036,701

Claim Chart for Infringement of Claims 1, 8, and 15 by HISENSE Accused Products

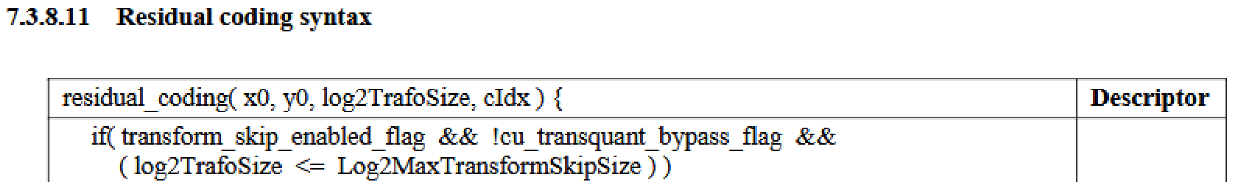
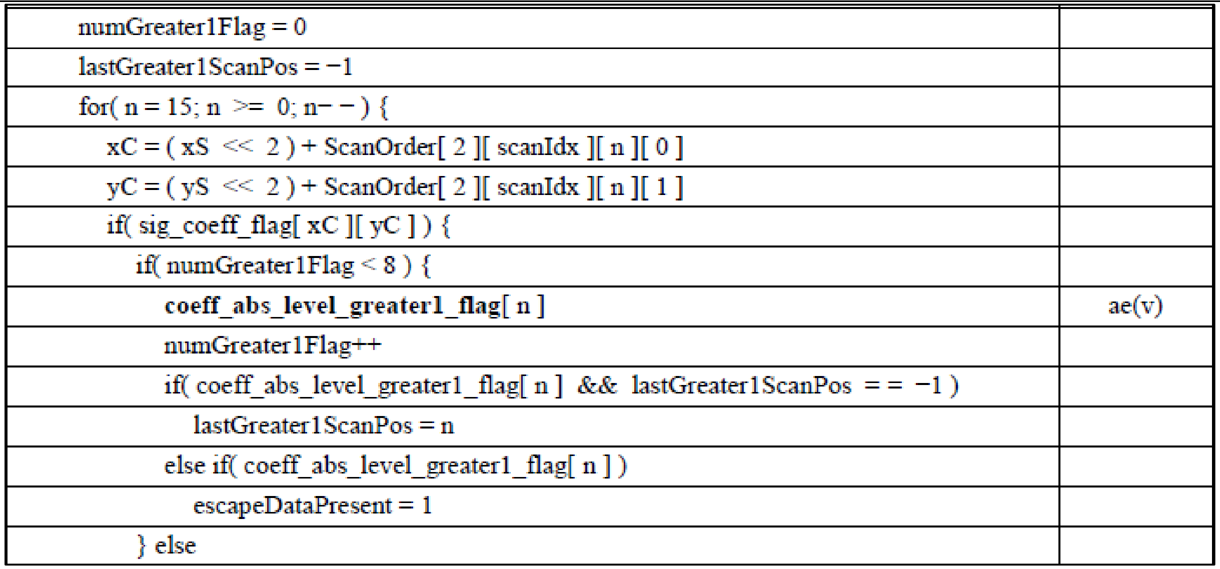
U.S. Patent No. 9,036,701	Hisense Accused Products
	The following specifications provide further evidence of how each of the Accused Products operates: **3 Definitions** For the purposes of this Recommendation | International Standard, the following definitions apply. . . . ***3.153 syntax element****: An element of data represented in the bitstream.* … ***3.166 transform coefficient:*** A Scalar quantity, considered to be in a frequency domain, that is associated with a particular one-dimensional or two-dimensional *frequency index* in a *transform* in the *decoding process*. ***3.167 transform coefficient level:*** *An integer quantity representing the value associated with a particular two-dimensional frequency index in the decoding process prior to scaling for computation of a transform coefficient value.* ITU-T Rec. H.265 (12/2016) High efficiency video coding at p. 12.  **7.3.8.11 Residual coding syntax** residual_coding(x0, y0, log2TrafoSize, cIdx) { — **Descriptor** if(transform_skip_enabled_flag && !cu_transquant_bypass_flag && (log2TrafoSize <= Log2MaxTransformSkipSize)) . . .

EXHIBIT 20

UNITED STATES PATENT NO. 9,036,701

CLAIM CHART FOR INFRINGEMENT OF CLAIMS 1, 8, AND 15 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS



```
numGreater1Flag = 0
lastGreater1ScanPos = −1
for( n = 15; n >= 0; n− − ) {
  xC = ( xS << 2 ) + ScanOrder[ 2 ][ scanIdx ][ n ][ 0 ]
  yC = ( yS << 2 ) + ScanOrder[ 2 ][ scanIdx ][ n ][ 1 ]
  if( sig_coeff_flag[ xC ][ yC ] ) {
    if( numGreater1Flag < 8 ) {
      coeff_abs_level_greater1_flag[ n ]                                  ae(v)
      numGreater1Flag++
      if( coeff_abs_level_greater1_flag[ n ] && lastGreater1ScanPos = = −1 )
        lastGreater1ScanPos = n
      else if( coeff_abs_level_greater1_flag[ n ] )
        escapeDataPresent = 1
    } else
```

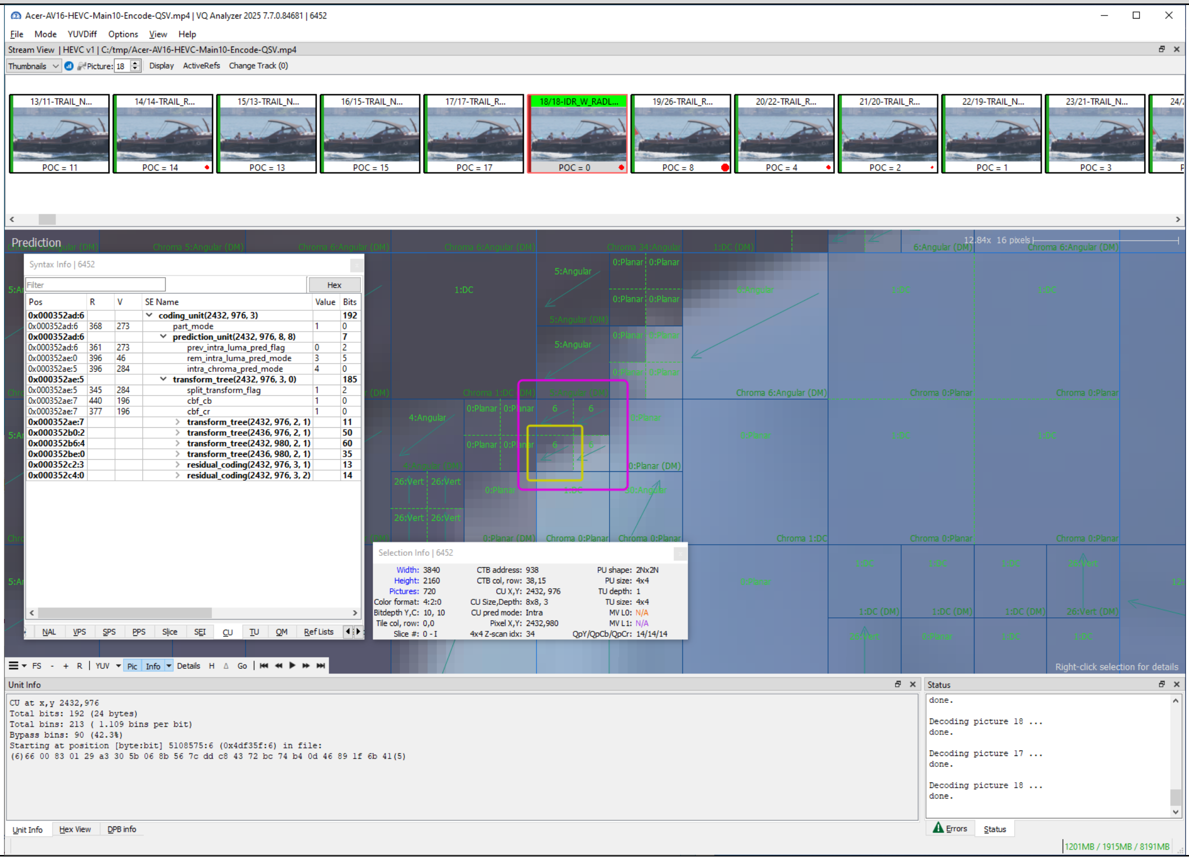
. . .

ITU-T Rec. H.265 (12/2016) High efficiency video coding at pp. 58-59.

Additionally, the video played back on the Hisense 43A7N was analyzed to confirm that the video contained transform units with more than eight non-zero transform coefficients according to the H.265 Standard.

For example, NAL 46, IDR Frame 18, at CTB [38,15], CU [2432,976] is an 8x8 coded block broken up into 4x4 sub-blocks.

EXHIBIT 20
UNITED STATES PATENT NO. 9,036,701
CLAIM CHART FOR INFRINGEMENT OF CLAIMS 1, 8, AND 15 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
	

Acer-AV16-HEVC-Main10-Encode-QSV.mp4 | VQ Analyzer 2025 7.7.0.84681 | 6452

Stream View | HEVC v1 | C:/tmp/Acer-AV16-HEVC-Main10-Encode-QSV.mp4

Picture: 18

Syntax Info | 6452

Pos	R	V	SE Name	Value	Bits
0x000352ad:6			**coding_unit(2432, 976, 3)**		**192**
0x000352ad:6	368	273	part_mode	1	0
0x000352ad:6			**prediction_unit(2432, 976, 8, 8)**		**7**
0x000352ad:6	361	273	prev_intra_luma_pred_flag	0	2
0x000352ae:0	396	46	rem_intra_luma_pred_mode	3	5
0x000352ae:5	396	284	intra_chroma_pred_mode	4	0
0x000352ae:5			**transform_tree(2432, 976, 3, 0)**		**185**
0x000352ae:5	345	284	split_transform_flag	1	2
0x000352ae:7	440	196	cbf_cb	1	0
0x000352ae:7	377	196	cbf_cr	1	0
0x000352ae:7			**transform_tree(2432, 976, 2, 1)**		**11**
0x000352b0:2			**transform_tree(2436, 976, 2, 1)**		**50**
0x000352b6:4			**transform_tree(2432, 980, 2, 1)**		**60**
0x000352be:0			**transform_tree(2436, 980, 2, 1)**		**35**
0x000352c2:3			**residual_coding(2432, 976, 3, 1)**		**13**
0x000352c4:0			**residual_coding(2432, 976, 3, 2)**		**14**

Selection Info | 6452

Width: 3840
Height: 2160
Pictures: 720
Color format: 4:2:0
Bitdepth Y,C: 10, 10
Tile col, row: 0,0
Slice #: 0 - I

CTB address: 938
CTB col, row: 38, 15
CU X,Y: 2432, 976
CU Size,Depth: 8x8, 3
CU pred mode: Intra
Pixel X,Y: 2432,980
4x4 Z-scan idx: 34

PU shape: 2Nx2N
PU size: 4x4
TU depth: 1
TU size: 4x4
MV L0: N/A
MV L1: N/A
QpY/QpCb/QpCr: 14/14/14

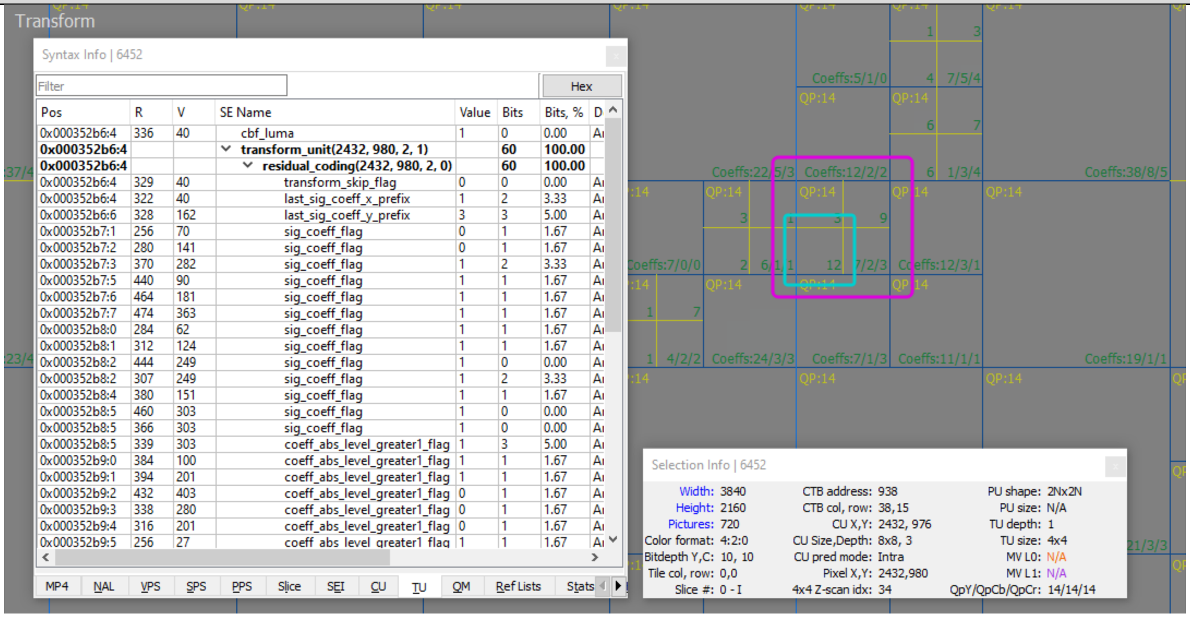
Unit Info

```
CU at x,y 2432,976
Total bits: 192 (24 bytes)
Total bins: 213 ( 1.109 bins per bit)
Bypass bins: 90 (42.3%)
Starting at position [byte:bit] 5108575:6 (0x4df35f:6) in file:
(6)66 00 83 01 29 a3 30 5b 06 8b 56 7c dd c8 43 72 bc 74 b4 0d 46 89 1f 6b 41(5)
```

Status

```
done.

Decoding picture 18 ...
done.

Decoding picture 17 ...
done.

Decoding picture 18 ...
done.
```

EXHIBIT 20
UNITED STATES PATENT NO. 9,036,701
CLAIM CHART FOR INFRINGEMENT OF CLAIMS 1, 8, AND 15 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 9,036,701	**HISENSE ACCUSED PRODUCTS**
	

Syntax Info | 6452

Pos	R	V	SE Name	Value	Bits	Bits, %	D
0x000352b6:4	336	40	cbf_luma	1	0	0.00	Ai
0x000352b6:4			**transform_unit(2432, 980, 2, 1)**		**60**	**100.00**	
0x000352b6:4			**residual_coding(2432, 980, 2, 0)**		**60**	**100.00**	
0x000352b6:4	329	40	transform_skip_flag	0	0	0.00	Ai
0x000352b6:4	322	40	last_sig_coeff_x_prefix	1	2	3.33	Ai
0x000352b6:6	328	162	last_sig_coeff_y_prefix	3	3	5.00	Ai
0x000352b7:1	256	70	sig_coeff_flag	0	1	1.67	Ai
0x000352b7:2	280	141	sig_coeff_flag	0	1	1.67	Ai
0x000352b7:3	370	282	sig_coeff_flag	1	2	3.33	Ai
0x000352b7:5	440	90	sig_coeff_flag	1	1	1.67	Ai
0x000352b7:6	464	181	sig_coeff_flag	1	1	1.67	Ai
0x000352b7:7	474	363	sig_coeff_flag	1	1	1.67	Ai
0x000352b8:0	284	62	sig_coeff_flag	1	1	1.67	Ai
0x000352b8:1	312	124	sig_coeff_flag	1	1	1.67	Ai
0x000352b8:2	444	249	sig_coeff_flag	1	0	0.00	Ai
0x000352b8:2	307	249	sig_coeff_flag	1	2	3.33	Ai
0x000352b8:4	380	151	sig_coeff_flag	1	1	1.67	Ai
0x000352b8:5	460	303	sig_coeff_flag	1	0	0.00	Ai
0x000352b8:5	366	303	sig_coeff_flag	1	0	0.00	Ai
0x000352b8:5	339	303	coeff_abs_level_greater1_flag	1	3	5.00	Ai
0x000352b9:0	384	100	coeff_abs_level_greater1_flag	1	1	1.67	Ai
0x000352b9:1	394	201	coeff_abs_level_greater1_flag	1	1	1.67	Ai
0x000352b9:2	432	403	coeff_abs_level_greater1_flag	0	1	1.67	Ai
0x000352b9:3	338	280	coeff_abs_level_greater1_flag	0	1	1.67	Ai
0x000352b9:4	316	201	coeff_abs_level_greater1_flag	0	1	1.67	Ai
0x000352b9:5	256	27	coeff_abs_level_greater1_flag	1	1	1.67	Ai

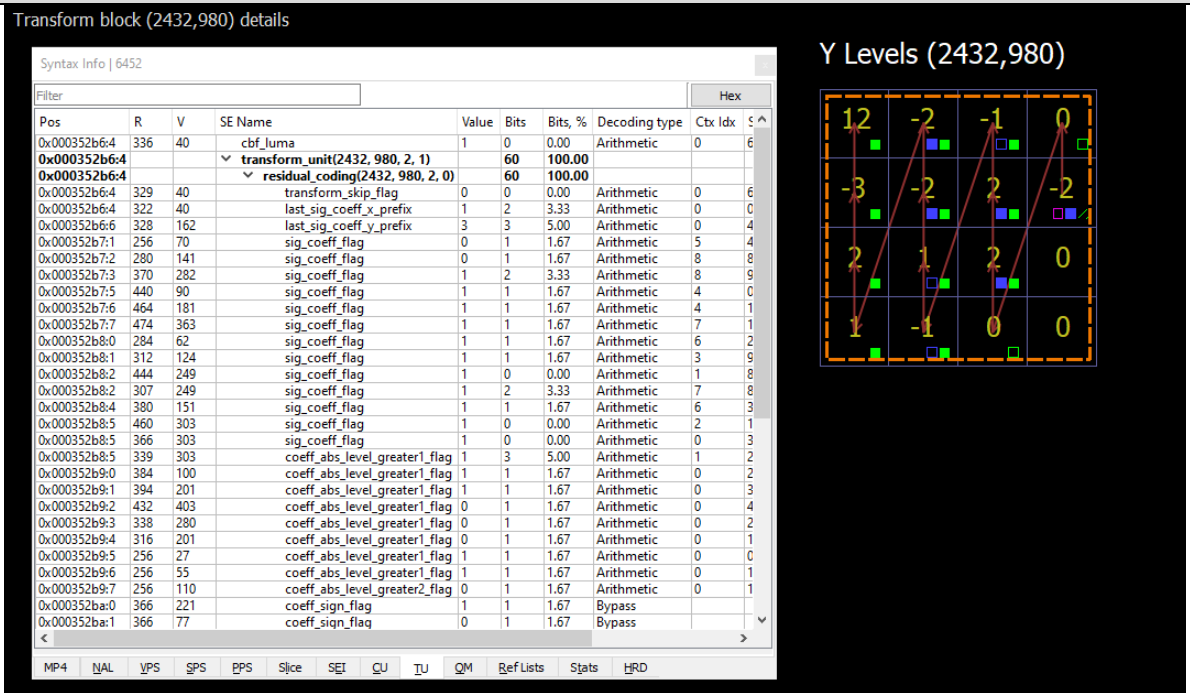
Selection Info | 6452

Width: 3840
Height: 2160
Pictures: 720
Color format: 4:2:0
Bitdepth Y,C: 10, 10
Tile col, row: 0,0
Slice #: 0 - I
CTB address: 938
CTB col, row: 38,15
CU X,Y: 2432, 976
CU Size,Depth: 8x8, 3
CU pred mode: Intra
Pixel X,Y: 2432,980
4x4 Z-scan idx: 34
PU shape: 2Nx2N
PU size: N/A
TU depth: 1
TU size: 4x4
MV L0: N/A
MV L1: N/A
QpY/QpCb/QpCr: 14/14/14

Source: Screenshot of VQ Analyzer software analyzing video played back on Hisense 43A7N.

CU [2432,976] includes transform block [2432,980], which has 12 non-zero transform coefficients.

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS



Transform block (2432,980) details

Syntax Info | 6452

Pos	R	V	SE Name	Value	Bits	Bits, %	Decoding type	Ctx Idx
0x000352b6:4	336	40	cbf_luma	1	0	0.00	Arithmetic	0
0x000352b6:4			**transform_unit(2432, 980, 2, 1)**		**60**	**100.00**		
0x000352b6:4			**residual_coding(2432, 980, 2, 0)**		**60**	**100.00**		
0x000352b6:4	329	40	transform_skip_flag	0	0	0.00	Arithmetic	0
0x000352b6:4	322	40	last_sig_coeff_x_prefix	1	2	3.33	Arithmetic	0
0x000352b6:6	328	162	last_sig_coeff_y_prefix	3	3	5.00	Arithmetic	0
0x000352b7:1	256	70	sig_coeff_flag	0	1	1.67	Arithmetic	5
0x000352b7:2	280	141	sig_coeff_flag	0	1	1.67	Arithmetic	8
0x000352b7:3	370	282	sig_coeff_flag	1	2	3.33	Arithmetic	8
0x000352b7:5	440	90	sig_coeff_flag	1	1	1.67	Arithmetic	4
0x000352b7:6	464	181	sig_coeff_flag	1	1	1.67	Arithmetic	4
0x000352b7:7	474	363	sig_coeff_flag	1	1	1.67	Arithmetic	7
0x000352b8:0	284	62	sig_coeff_flag	1	1	1.67	Arithmetic	6
0x000352b8:1	312	124	sig_coeff_flag	1	1	1.67	Arithmetic	3
0x000352b8:2	444	249	sig_coeff_flag	1	0	0.00	Arithmetic	1
0x000352b8:2	307	249	sig_coeff_flag	1	2	3.33	Arithmetic	7
0x000352b8:4	380	151	sig_coeff_flag	1	1	1.67	Arithmetic	6
0x000352b8:5	460	303	sig_coeff_flag	1	0	0.00	Arithmetic	2
0x000352b8:5	366	303	sig_coeff_flag	1	0	0.00	Arithmetic	0
0x000352b8:5	339	303	coeff_abs_level_greater1_flag	1	3	5.00	Arithmetic	1
0x000352b9:0	384	100	coeff_abs_level_greater1_flag	1	1	1.67	Arithmetic	0
0x000352b9:1	394	201	coeff_abs_level_greater1_flag	1	1	1.67	Arithmetic	0
0x000352b9:2	432	403	coeff_abs_level_greater1_flag	0	1	1.67	Arithmetic	0
0x000352b9:3	338	280	coeff_abs_level_greater1_flag	0	1	1.67	Arithmetic	0
0x000352b9:4	316	201	coeff_abs_level_greater1_flag	0	1	1.67	Arithmetic	0
0x000352b9:5	256	27	coeff_abs_level_greater1_flag	1	1	1.67	Arithmetic	0
0x000352b9:6	256	55	coeff_abs_level_greater1_flag	1	1	1.67	Arithmetic	0
0x000352b9:7	256	110	coeff_abs_level_greater2_flag	0	1	1.67	Arithmetic	0
0x000352ba:0	366	221	coeff_sign_flag	1	1	1.67	Bypass	
0x000352ba:1	366	77	coeff_sign_flag	0	1	1.67	Bypass	

Y Levels (2432,980)

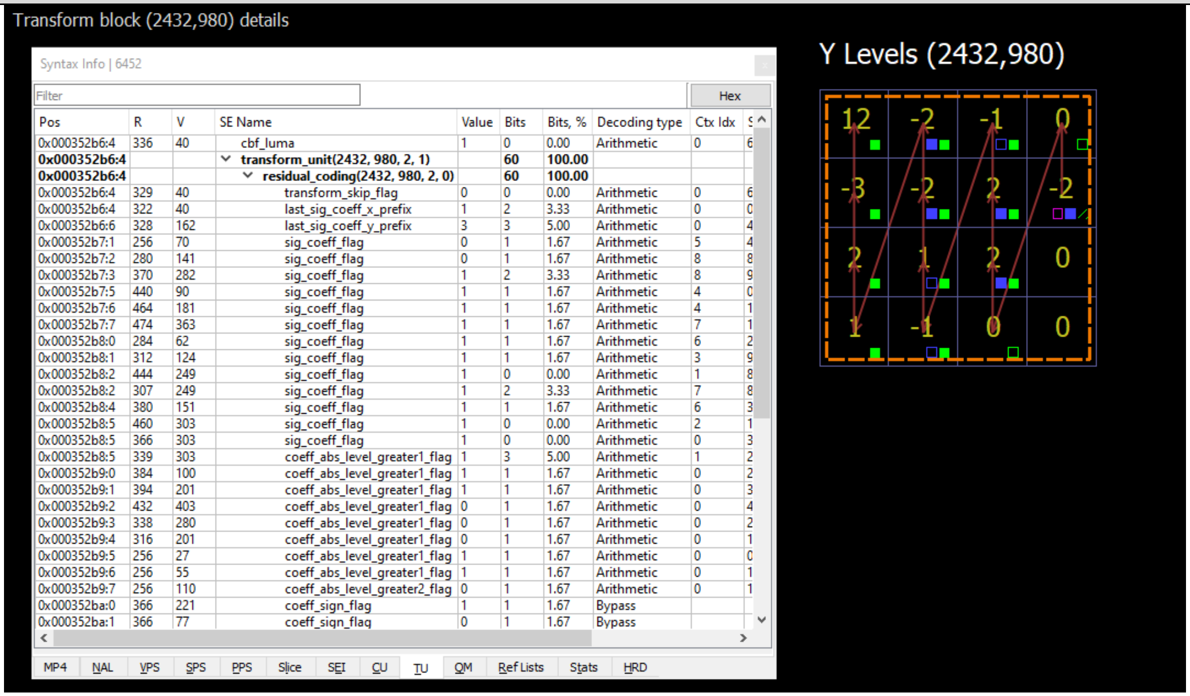
12	-2	-1	0
-3	-2	2	-2
2	1	2	0
1	-1	0	0

Source: Screenshot of VQ Analyzer software analyzing video played back on Hisense 43A7N.

The first 8 non-zero transform coefficient values are coded using the coeff_abs_level_greater1_flag, which is arithmetic coded using context updating. The remaining non-zero transform coefficient values are coded suing coeff_abs_level_remaining, which is bypass coded.

EXHIBIT 20
UNITED STATES PATENT NO. 9,036,701
CLAIM CHART FOR INFRINGEMENT OF CLAIMS 1, 8, AND 15 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS



Transform block (2432,980) details

Syntax Info | 6452

Pos	R	V	SE Name	Value	Bits	Bits, %	Decoding type	Ctx Idx
0x000352b6:4	336	40	cbf_luma	1	0	0.00	Arithmetic	0
0x000352b6:4			**transform_unit(2432, 980, 2, 1)**		**60**	**100.00**		
0x000352b6:4			**residual_coding(2432, 980, 2, 0)**		**60**	**100.00**		
0x000352b6:4	329	40	transform_skip_flag	0	0	0.00	Arithmetic	0
0x000352b6:4	322	40	last_sig_coeff_x_prefix	1	2	3.33	Arithmetic	0
0x000352b6:6	328	162	last_sig_coeff_y_prefix	3	3	5.00	Arithmetic	0
0x000352b7:1	256	70	sig_coeff_flag	0	1	1.67	Arithmetic	5
0x000352b7:2	280	141	sig_coeff_flag	0	1	1.67	Arithmetic	8
0x000352b7:3	370	282	sig_coeff_flag	1	2	3.33	Arithmetic	8
0x000352b7:5	440	90	sig_coeff_flag	1	1	1.67	Arithmetic	4
0x000352b7:6	464	181	sig_coeff_flag	1	1	1.67	Arithmetic	4
0x000352b7:7	474	363	sig_coeff_flag	1	1	1.67	Arithmetic	7
0x000352b8:0	284	62	sig_coeff_flag	1	1	1.67	Arithmetic	6
0x000352b8:1	312	124	sig_coeff_flag	1	1	1.67	Arithmetic	3
0x000352b8:2	444	249	sig_coeff_flag	1	0	0.00	Arithmetic	1
0x000352b8:2	307	249	sig_coeff_flag	1	2	3.33	Arithmetic	7
0x000352b8:4	380	151	sig_coeff_flag	1	1	1.67	Arithmetic	6
0x000352b8:5	460	303	sig_coeff_flag	1	0	0.00	Arithmetic	2
0x000352b8:5	366	303	sig_coeff_flag	1	0	0.00	Arithmetic	0
0x000352b8:5	339	303	coeff_abs_level_greater1_flag	1	3	5.00	Arithmetic	1
0x000352b9:0	384	100	coeff_abs_level_greater1_flag	1	1	1.67	Arithmetic	0
0x000352b9:1	394	201	coeff_abs_level_greater1_flag	1	1	1.67	Arithmetic	0
0x000352b9:2	432	403	coeff_abs_level_greater1_flag	0	1	1.67	Arithmetic	0
0x000352b9:3	338	280	coeff_abs_level_greater1_flag	0	1	1.67	Arithmetic	0
0x000352b9:4	316	201	coeff_abs_level_greater1_flag	0	1	1.67	Arithmetic	0
0x000352b9:5	256	27	coeff_abs_level_greater1_flag	1	1	1.67	Arithmetic	0
0x000352b9:6	256	55	coeff_abs_level_greater1_flag	1	1	1.67	Arithmetic	0
0x000352b9:7	256	110	coeff_abs_level_greater2_flag	0	1	1.67	Arithmetic	0
0x000352ba:0	366	221	coeff_sign_flag	1	1	1.67	Bypass	
0x000352ba:1	366	77	coeff_sign_flag	0	1	1.67	Bypass	

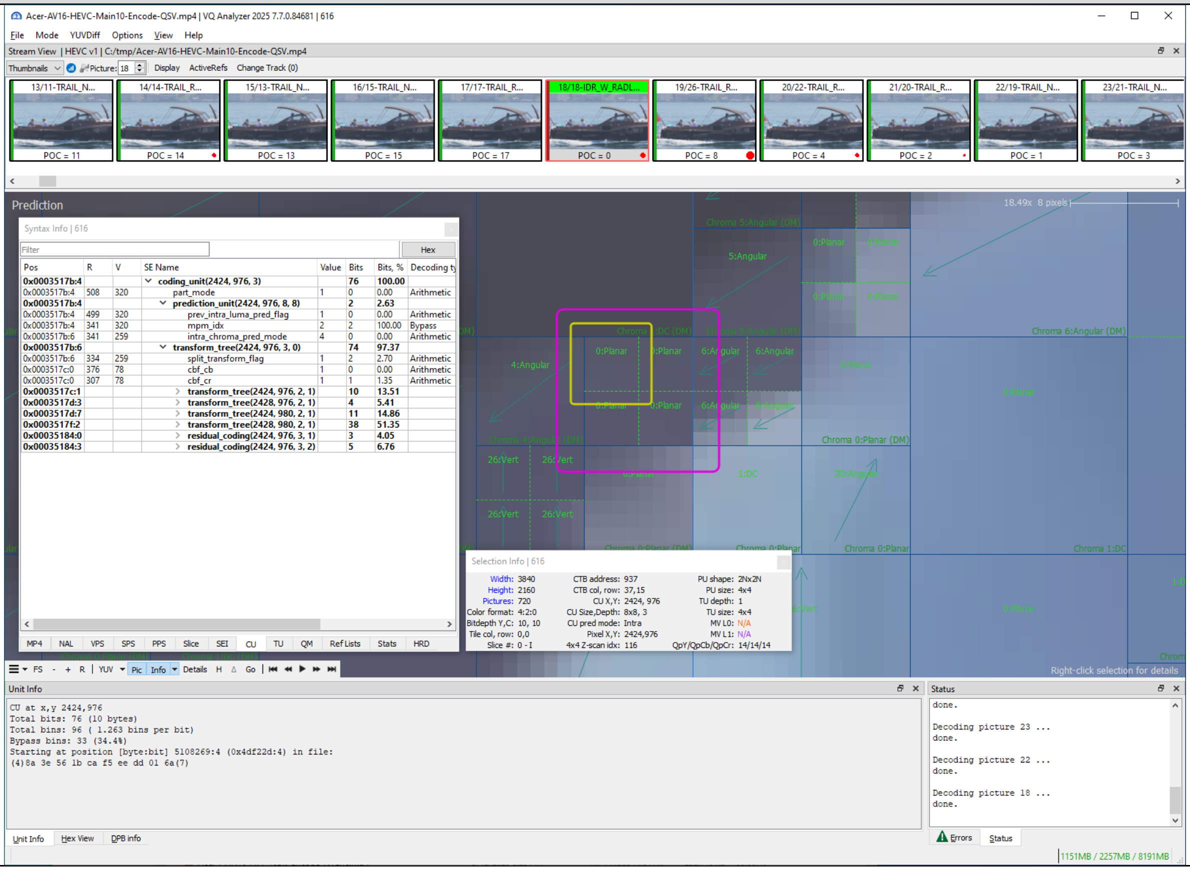
Y Levels (2432,980)

12	-2	-1	0
-3	-2	2	-2
2	1	2	0
1	-1	0	0

Source: Screenshot of VQ Analyzer software analyzing video played back on Hisense 43A7N.

As another example, NAL 46, IDR Frame 18, CTB [37,15] CU[2424,976] is an 8x8 intra-coded block broken up into 4x4 sub-blocks.

EXHIBIT 20
UNITED STATES PATENT NO. 9,036,701
CLAIM CHART FOR INFRINGEMENT OF CLAIMS 1, 8, AND 15 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
	

Pos	R	V	SE Name	Value	Bits	Bits, %	Decoding ty
0x0003517b:4			**coding_unit(2424, 976, 3)**		**76**	**100.00**	
0x0003517b:4	508	320	part_mode	1	0	0.00	Arithmetic
0x0003517b:4			**prediction_unit(2424, 976, 8, 8)**		**2**	**2.63**	
0x0003517b:4	499	320	prev_intra_luma_pred_flag	1	0	0.00	Arithmetic
0x0003517b:4	341	320	mpm_idx	2	2	100.00	Bypass
0x0003517b:6	341	259	intra_chroma_pred_mode	4	0	0.00	Arithmetic
0x0003517b:6			**transform_tree(2424, 976, 3, 0)**		**74**	**97.37**	
0x0003517b:6	334	259	split_transform_flag	1	2	2.70	Arithmetic
0x0003517c:0	376	78	cbf_cb	1	0	0.00	Arithmetic
0x0003517c:0	307	78	cbf_cr	1	1	1.35	Arithmetic
0x0003517c:1			**transform_tree(2424, 976, 2, 1)**		**10**	**13.51**	
0x0003517d:3			**transform_tree(2428, 976, 2, 1)**		**4**	**5.41**	
0x0003517d:7			**transform_tree(2424, 980, 2, 1)**		**11**	**14.86**	
0x0003517f:2			**transform_tree(2428, 980, 2, 1)**		**38**	**51.35**	
0x00035184:0			**residual_coding(2424, 976, 3, 1)**		**3**	**4.05**	
0x00035184:3			**residual_coding(2424, 976, 3, 2)**		**5**	**6.76**	

Width: 3840
Height: 2160
Pictures: 720
Color format: 4:2:0
Bitdepth Y,C: 10, 10
Tile col, row: 0,0
Slice #: 0 - I

CTB address: 937
CTB col, row: 37,15
CU X,Y: 2424, 976
CU Size,Depth: 8x8, 3
CU pred mode: Intra
Pixel X,Y: 2424,976
4x4 Z-scan idx: 116

PU shape: 2Nx2N
PU size: 4x4
TU depth: 1
TU size: 4x4
MV L0: N/A
MV L1: N/A
QpY/QpCb/QpCr: 14/14/14

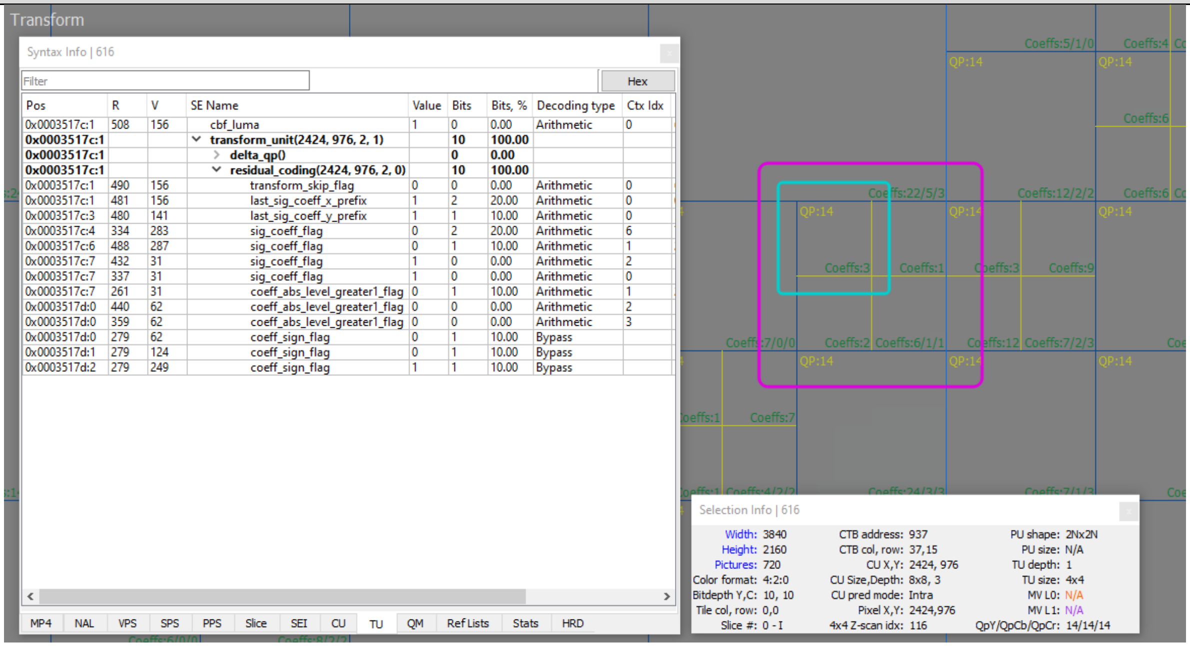
```
CU at x,y 2424,976
Total bits: 76 (10 bytes)
Total bins: 96 ( 1.263 bins per bit)
Bypass bins: 33 (34.4%)
Starting at position [byte:bit] 5108269:4 (0x4df22d:4) in file:
(4)8a 3e 56 1b ca f5 ee dd 01 6a(7)
```

```
done.

Decoding picture 23 ...
done.

Decoding picture 22 ...
done.

Decoding picture 18 ...
done.
```

EXHIBIT 20

UNITED STATES PATENT NO. 9,036,701

CLAIM CHART FOR INFRINGEMENT OF CLAIMS 1, 8, AND 15 BY HISENSE ACCUSED PRODUCTS

<table>
<tr><th>U.S. PATENT NO. 9,036,701</th><th>HISENSE ACCUSED PRODUCTS</th></tr>
<tr><td></td><td>


Syntax Info | 616

Filter

<table>
<tr><th>Pos</th><th>R</th><th>V</th><th>SE Name</th><th>Value</th><th>Bits</th><th>Bits, %</th><th>Decoding type</th><th>Ctx Idx</th></tr>
<tr><td>0x0003517c:1</td><td>508</td><td>156</td><td>cbf_luma</td><td>1</td><td>0</td><td>0.00</td><td>Arithmetic</td><td>0</td></tr>
<tr><td>0x0003517c:1</td><td></td><td></td><td>transform_unit(2424, 976, 2, 1)</td><td></td><td>10</td><td>100.00</td><td></td><td></td></tr>
<tr><td>0x0003517c:1</td><td></td><td></td><td>delta_qp()</td><td></td><td>0</td><td>0.00</td><td></td><td></td></tr>
<tr><td>0x0003517c:1</td><td></td><td></td><td>residual_coding(2424, 976, 2, 0)</td><td></td><td>10</td><td>100.00</td><td></td><td></td></tr>
<tr><td>0x0003517c:1</td><td>490</td><td>156</td><td>transform_skip_flag</td><td>0</td><td>0</td><td>0.00</td><td>Arithmetic</td><td>0</td></tr>
<tr><td>0x0003517c:1</td><td>481</td><td>156</td><td>last_sig_coeff_x_prefix</td><td>1</td><td>2</td><td>20.00</td><td>Arithmetic</td><td>0</td></tr>
<tr><td>0x0003517c:3</td><td>480</td><td>141</td><td>last_sig_coeff_y_prefix</td><td>1</td><td>1</td><td>10.00</td><td>Arithmetic</td><td>0</td></tr>
<tr><td>0x0003517c:4</td><td>334</td><td>283</td><td>sig_coeff_flag</td><td>0</td><td>2</td><td>20.00</td><td>Arithmetic</td><td>6</td></tr>
<tr><td>0x0003517c:6</td><td>488</td><td>287</td><td>sig_coeff_flag</td><td>0</td><td>1</td><td>10.00</td><td>Arithmetic</td><td>1</td></tr>
<tr><td>0x0003517c:7</td><td>432</td><td>31</td><td>sig_coeff_flag</td><td>1</td><td>0</td><td>0.00</td><td>Arithmetic</td><td>2</td></tr>
<tr><td>0x0003517c:7</td><td>337</td><td>31</td><td>sig_coeff_flag</td><td>1</td><td>0</td><td>0.00</td><td>Arithmetic</td><td>0</td></tr>
<tr><td>0x0003517c:7</td><td>261</td><td>31</td><td>coeff_abs_level_greater1_flag</td><td>0</td><td>1</td><td>10.00</td><td>Arithmetic</td><td>1</td></tr>
<tr><td>0x0003517d:0</td><td>440</td><td>62</td><td>coeff_abs_level_greater1_flag</td><td>0</td><td>0</td><td>0.00</td><td>Arithmetic</td><td>2</td></tr>
<tr><td>0x0003517d:0</td><td>359</td><td>62</td><td>coeff_abs_level_greater1_flag</td><td>0</td><td>0</td><td>0.00</td><td>Arithmetic</td><td>3</td></tr>
<tr><td>0x0003517d:0</td><td>279</td><td>62</td><td>coeff_sign_flag</td><td>0</td><td>1</td><td>10.00</td><td>Bypass</td><td></td></tr>
<tr><td>0x0003517d:1</td><td>279</td><td>124</td><td>coeff_sign_flag</td><td>0</td><td>1</td><td>10.00</td><td>Bypass</td><td></td></tr>
<tr><td>0x0003517d:2</td><td>279</td><td>249</td><td>coeff_sign_flag</td><td>1</td><td>1</td><td>10.00</td><td>Bypass</td><td></td></tr>
</table>

Selection Info | 616

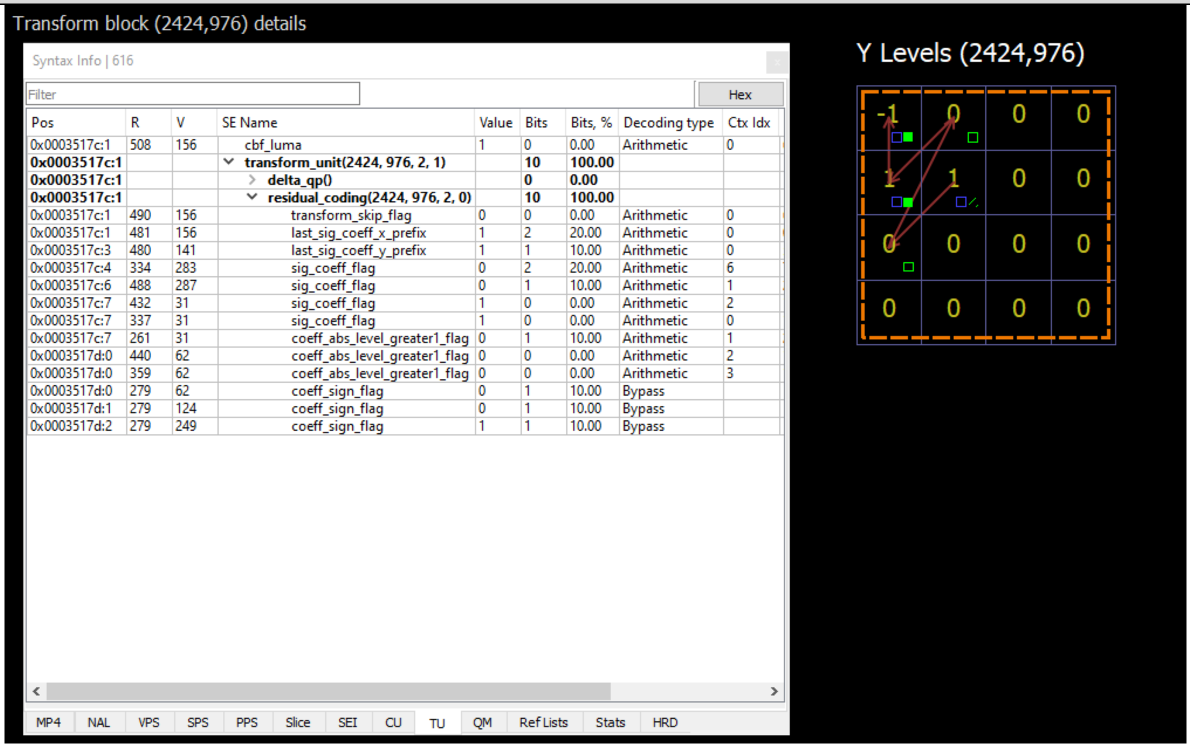
Width: 3840
Height: 2160
Pictures: 720
Color format: 4:2:0
Bitdepth Y,C: 10, 10
Tile col, row: 0,0
Slice #: 0 - I

CTB address: 937
CTB col, row: 37,15
CU X,Y: 2424, 976
CU Size,Depth: 8x8, 3
CU pred mode: Intra
Pixel X,Y: 2424,976
4x4 Z-scan idx: 116

PU shape: 2Nx2N
PU size: N/A
TU depth: 1
TU size: 4x4
MV L0: N/A
MV L1: N/A
QpY/QpCb/QpCr: 14/14/14

Source: Screenshot of VQ Analyzer software analyzing video played back on Hisense 43A7N.

CU [2424,976] includes transform block [2424,976], which has 3 non-zero transform coefficients.
</td></tr>
</table>

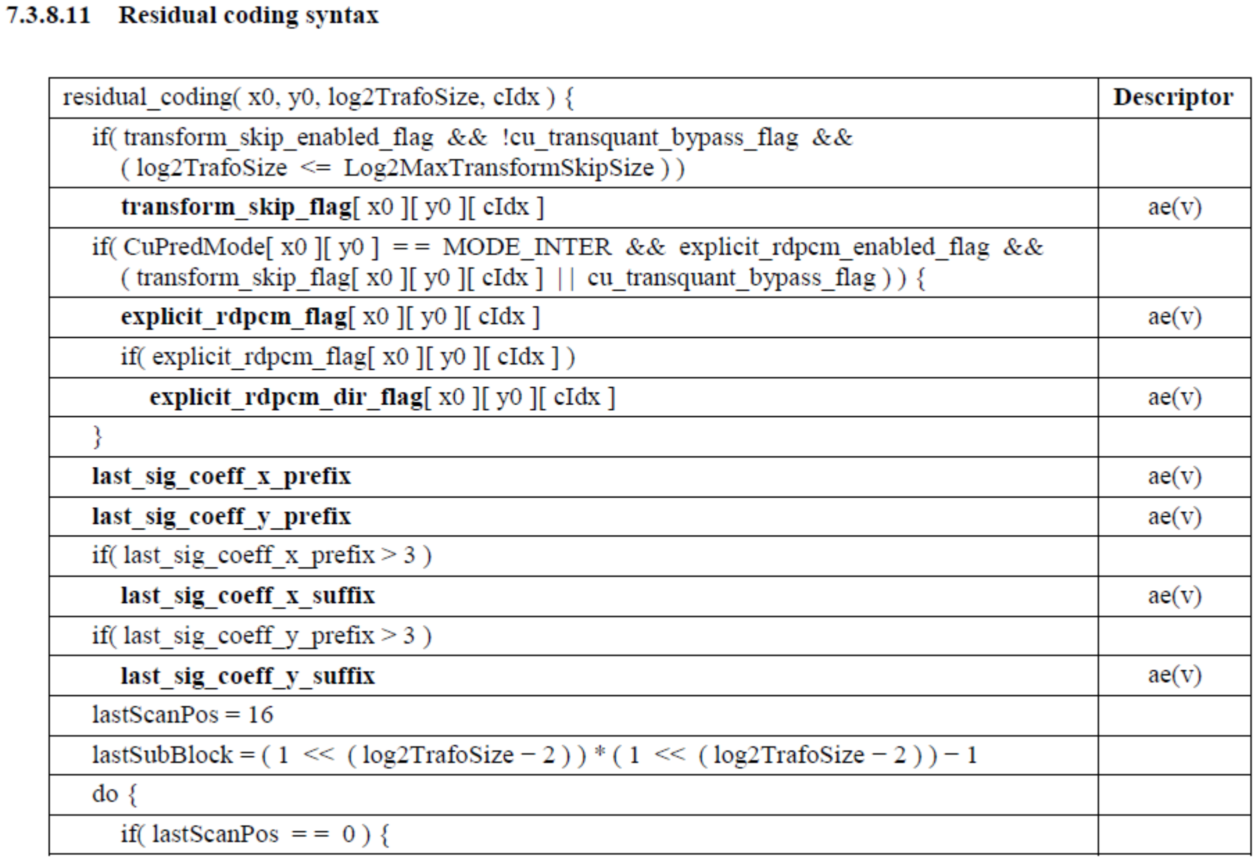
<table>
<tr><th>U.S. PATENT NO. 9,036,701</th><th>HISENSE ACCUSED PRODUCTS</th></tr>
<tr><td></td><td>

Transform block (2424,976) details

<table>
<tr><th>Pos</th><th>R</th><th>V</th><th>SE Name</th><th>Value</th><th>Bits</th><th>Bits, %</th><th>Decoding type</th><th>Ctx Idx</th></tr>
<tr><td>0x0003517c:1</td><td>508</td><td>156</td><td>cbf_luma</td><td>1</td><td>0</td><td>0.00</td><td>Arithmetic</td><td>0</td></tr>
<tr><td>**0x0003517c:1**</td><td></td><td></td><td>**transform_unit(2424, 976, 2, 1)**</td><td></td><td>**10**</td><td>**100.00**</td><td></td><td></td></tr>
<tr><td>**0x0003517c:1**</td><td></td><td></td><td>**delta_qp()**</td><td></td><td>**0**</td><td>**0.00**</td><td></td><td></td></tr>
<tr><td>**0x0003517c:1**</td><td></td><td></td><td>**residual_coding(2424, 976, 2, 0)**</td><td></td><td>**10**</td><td>**100.00**</td><td></td><td></td></tr>
<tr><td>0x0003517c:1</td><td>490</td><td>156</td><td>transform_skip_flag</td><td>0</td><td>0</td><td>0.00</td><td>Arithmetic</td><td>0</td></tr>
<tr><td>0x0003517c:1</td><td>481</td><td>156</td><td>last_sig_coeff_x_prefix</td><td>1</td><td>2</td><td>20.00</td><td>Arithmetic</td><td>0</td></tr>
<tr><td>0x0003517c:3</td><td>480</td><td>141</td><td>last_sig_coeff_y_prefix</td><td>1</td><td>1</td><td>10.00</td><td>Arithmetic</td><td>0</td></tr>
<tr><td>0x0003517c:4</td><td>334</td><td>283</td><td>sig_coeff_flag</td><td>0</td><td>2</td><td>20.00</td><td>Arithmetic</td><td>6</td></tr>
<tr><td>0x0003517c:6</td><td>488</td><td>287</td><td>sig_coeff_flag</td><td>0</td><td>1</td><td>10.00</td><td>Arithmetic</td><td>1</td></tr>
<tr><td>0x0003517c:7</td><td>432</td><td>31</td><td>sig_coeff_flag</td><td>1</td><td>0</td><td>0.00</td><td>Arithmetic</td><td>2</td></tr>
<tr><td>0x0003517c:7</td><td>337</td><td>31</td><td>sig_coeff_flag</td><td>1</td><td>0</td><td>0.00</td><td>Arithmetic</td><td>0</td></tr>
<tr><td>0x0003517c:7</td><td>261</td><td>31</td><td>coeff_abs_level_greater1_flag</td><td>0</td><td>1</td><td>10.00</td><td>Arithmetic</td><td>1</td></tr>
<tr><td>0x0003517d:0</td><td>440</td><td>62</td><td>coeff_abs_level_greater1_flag</td><td>0</td><td>0</td><td>0.00</td><td>Arithmetic</td><td>2</td></tr>
<tr><td>0x0003517d:0</td><td>359</td><td>62</td><td>coeff_abs_level_greater1_flag</td><td>0</td><td>0</td><td>0.00</td><td>Arithmetic</td><td>3</td></tr>
<tr><td>0x0003517d:0</td><td>279</td><td>62</td><td>coeff_sign_flag</td><td>0</td><td>1</td><td>10.00</td><td>Bypass</td><td></td></tr>
<tr><td>0x0003517d:1</td><td>279</td><td>124</td><td>coeff_sign_flag</td><td>0</td><td>1</td><td>10.00</td><td>Bypass</td><td></td></tr>
<tr><td>0x0003517d:2</td><td>279</td><td>249</td><td>coeff_sign_flag</td><td>1</td><td>1</td><td>10.00</td><td>Bypass</td><td></td></tr>
</table>

Y Levels (2424,976)
<table>
<tr><td>-1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
Source: Screenshot of VQ Analyzer software analyzing video played back on Hisense 43A7N.
</td></tr>
<tr><td>**[b]** categorizing a plurality of syntax elements of video content into first and second categories based on the frequency of occurrence threshold, wherein syntax elements which occur greater</td><td>Each of the Accused Products, such as the Hisense 43A7N, performs a method comprising categorizing a plurality of syntax elements of video content into first and second categories based on the frequency of occurrence threshold, wherein syntax elements which occur greater than the frequency of occurrence threshold are categorized into the second category and syntax elements which occur less than the frequency of occurrence are categorized into the first category.

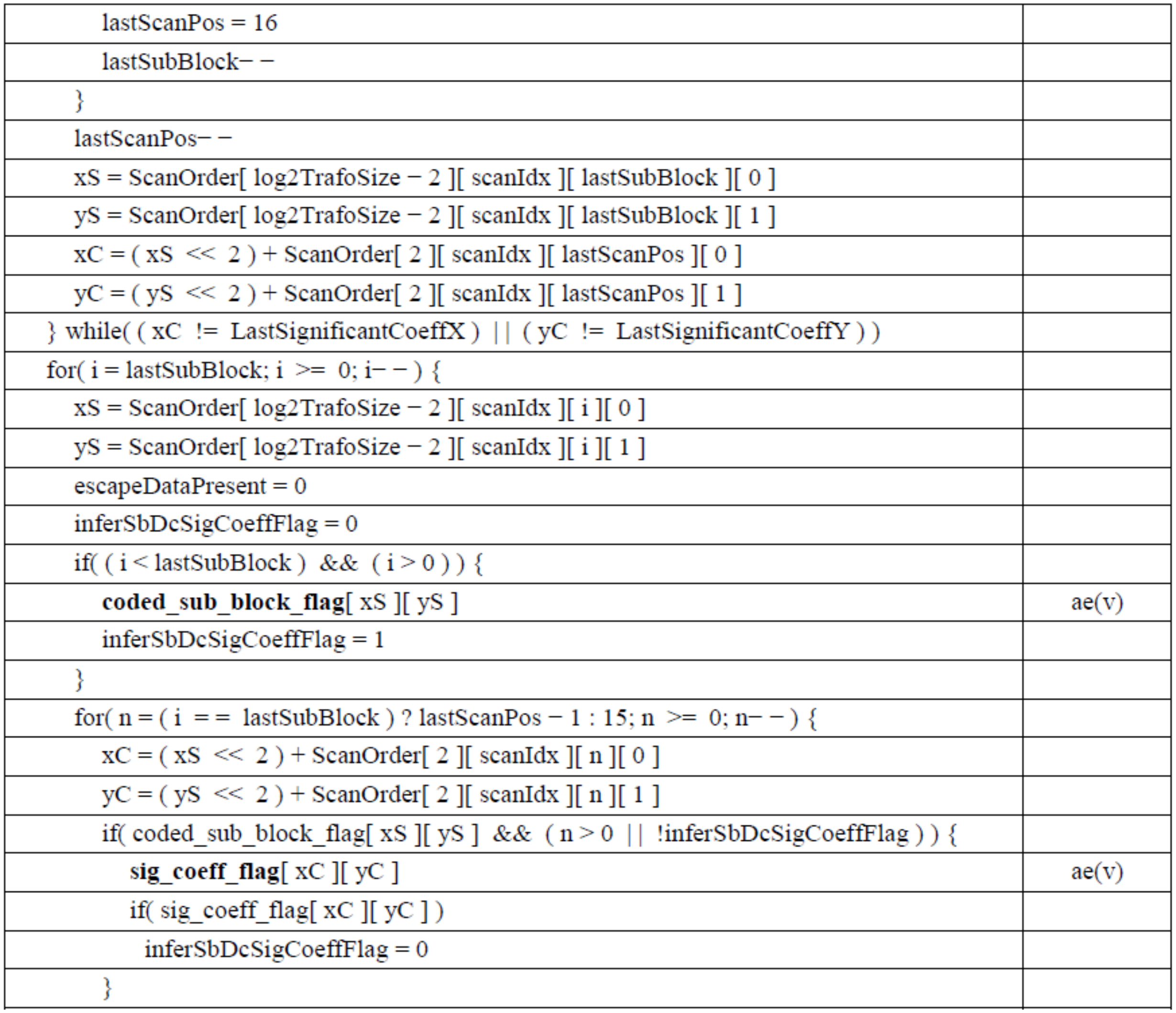
For example, and without limitation, the H.265 Standard specifies the following regarding the decoding process. Each of the Accused Products performs a method comprising</td></tr>
</table>

U.S. PATENT NO. 9,036,701	**HISENSE ACCUSED PRODUCTS**
than the frequency of occurrence threshold are categorized into the second category and syntax elements which occur less than the frequency of occurrence are categorized into the first category;	categorizing a plurality of syntax elements of video content into first and second categories based on the frequency of occurrence threshold, wherein syntax elements which occur greater than the frequency of occurrence threshold are categorized into the second category and syntax elements which occur less than the frequency of occurrence are categorized into the first category. The following specifications provide further evidence of how each of the Accused Products operates:

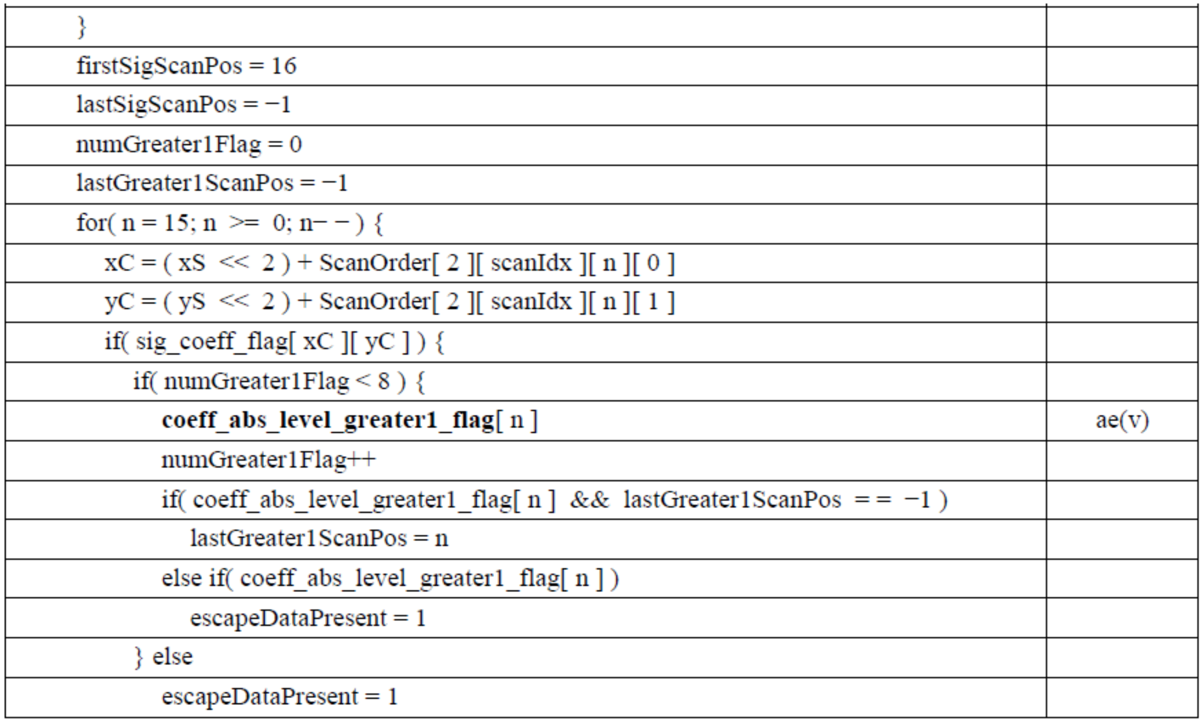
U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
	

7.3.8.11 Residual coding syntax

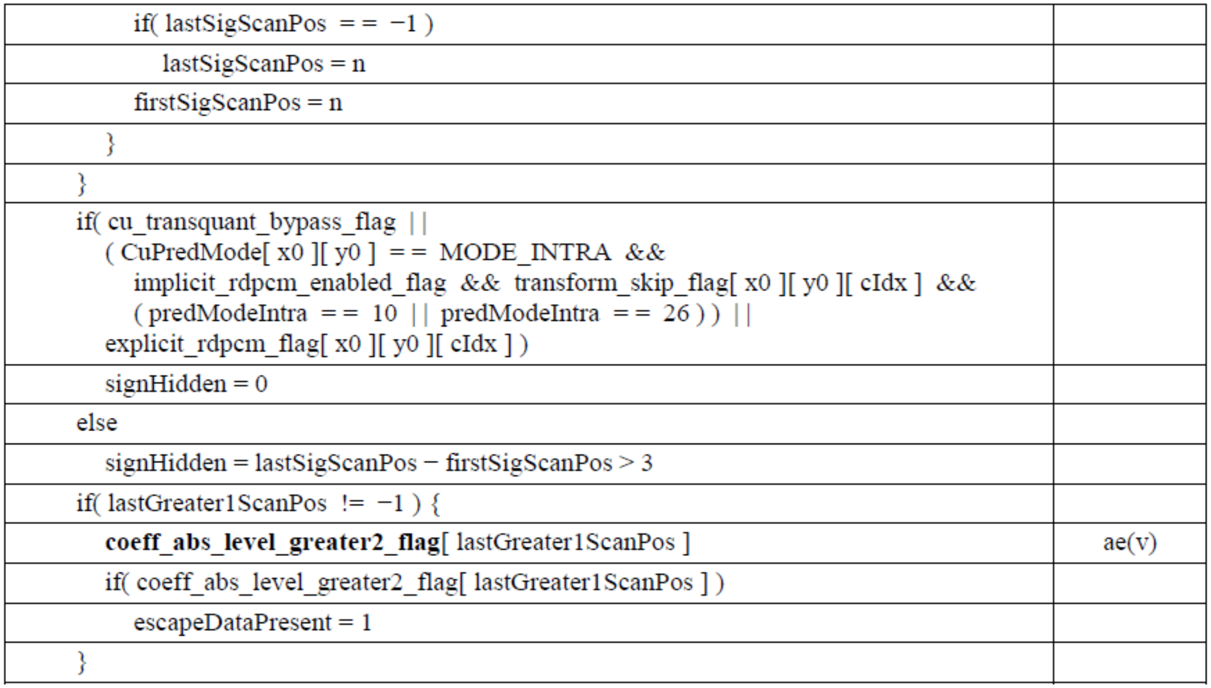
residual_coding(x0, y0, log2TrafoSize, cIdx) {	**Descriptor**
if(transform_skip_enabled_flag && !cu_transquant_bypass_flag && (log2TrafoSize <= Log2MaxTransformSkipSize))	
transform_skip_flag[x0][y0][cIdx]	ae(v)
if(CuPredMode[x0][y0] = = MODE_INTER && explicit_rdpcm_enabled_flag && (transform_skip_flag[x0][y0][cIdx] || cu_transquant_bypass_flag)) {	
explicit_rdpcm_flag[x0][y0][cIdx]	ae(v)
if(explicit_rdpcm_flag[x0][y0][cIdx])	
explicit_rdpcm_dir_flag[x0][y0][cIdx]	ae(v)
}	
last_sig_coeff_x_prefix	ae(v)
last_sig_coeff_y_prefix	ae(v)
if(last_sig_coeff_x_prefix > 3)	
last_sig_coeff_x_suffix	ae(v)
if(last_sig_coeff_y_prefix > 3)	
last_sig_coeff_y_suffix	ae(v)
lastScanPos = 16	
lastSubBlock = (1 << (log2TrafoSize − 2)) * (1 << (log2TrafoSize − 2)) − 1	
do {	
if(lastScanPos = = 0) {	

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
	

lastScanPos = 16	
lastSubBlock− −	
}	
lastScanPos− −	
xS = ScanOrder[log2TrafoSize − 2][scanIdx][lastSubBlock][0]	
yS = ScanOrder[log2TrafoSize − 2][scanIdx][lastSubBlock][1]	
xC = (xS << 2) + ScanOrder[2][scanIdx][lastScanPos][0]	
yC = (yS << 2) + ScanOrder[2][scanIdx][lastScanPos][1]	
} while((xC != LastSignificantCoeffX) || (yC != LastSignificantCoeffY))	
for(i = lastSubBlock; i >= 0; i− −) {	
xS = ScanOrder[log2TrafoSize − 2][scanIdx][i][0]	
yS = ScanOrder[log2TrafoSize − 2][scanIdx][i][1]	
escapeDataPresent = 0	
inferSbDcSigCoeffFlag = 0	
if((i < lastSubBlock) && (i > 0)) {	
coded_sub_block_flag[xS][yS]	ae(v)
inferSbDcSigCoeffFlag = 1	
}	
for(n = (i = = lastSubBlock) ? lastScanPos − 1 : 15; n >= 0; n− −) {	
xC = (xS << 2) + ScanOrder[2][scanIdx][n][0]	
yC = (yS << 2) + ScanOrder[2][scanIdx][n][1]	
if(coded_sub_block_flag[xS][yS] && (n > 0 || !inferSbDcSigCoeffFlag)) {	
sig_coeff_flag[xC][yC]	ae(v)
if(sig_coeff_flag[xC][yC])	
inferSbDcSigCoeffFlag = 0	
}	

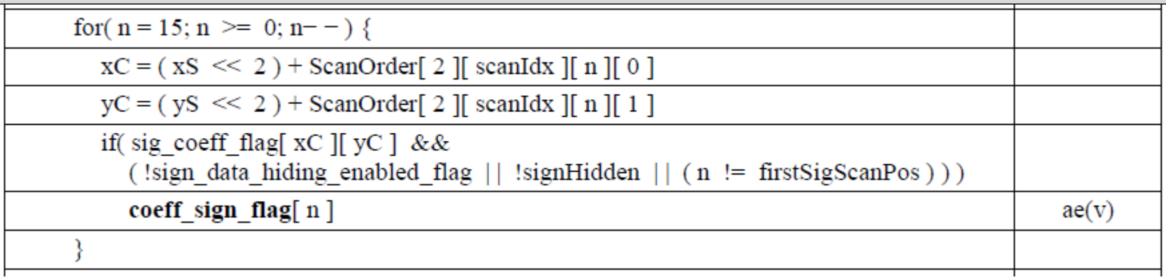
U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
	

```
  }
  firstSigScanPos = 16
  lastSigScanPos = −1
  numGreater1Flag = 0
  lastGreater1ScanPos = −1
  for( n = 15; n >= 0; n− − ) {
    xC = ( xS << 2 ) + ScanOrder[ 2 ][ scanIdx ][ n ][ 0 ]
    yC = ( yS << 2 ) + ScanOrder[ 2 ][ scanIdx ][ n ][ 1 ]
    if( sig_coeff_flag[ xC ][ yC ] ) {
      if( numGreater1Flag < 8 ) {
        coeff_abs_level_greater1_flag[ n ]                              ae(v)
        numGreater1Flag++
        if( coeff_abs_level_greater1_flag[ n ] && lastGreater1ScanPos = = −1 )
          lastGreater1ScanPos = n
        else if( coeff_abs_level_greater1_flag[ n ] )
          escapeDataPresent = 1
      } else
        escapeDataPresent = 1
```

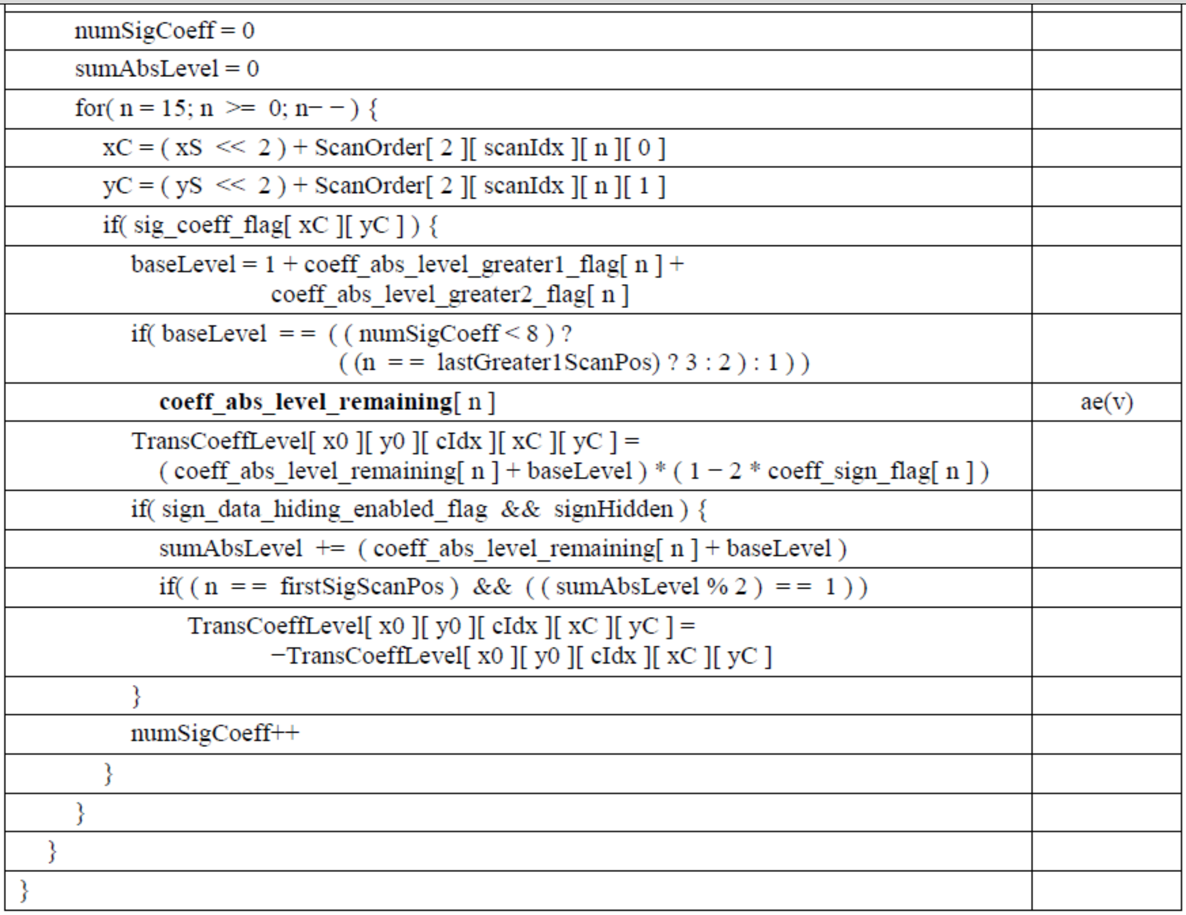
U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
	

if(lastSigScanPos = = −1)	
lastSigScanPos = n	
firstSigScanPos = n	
}	
}	
if(cu_transquant_bypass_flag || (CuPredMode[x0][y0] = = MODE_INTRA && implicit_rdpcm_enabled_flag && transform_skip_flag[x0][y0][cIdx] && (predModeIntra = = 10 || predModeIntra = = 26)) || explicit_rdpcm_flag[x0][y0][cIdx])	
signHidden = 0	
else	
signHidden = lastSigScanPos − firstSigScanPos > 3	
if(lastGreater1ScanPos != −1) {	
coeff_abs_level_greater2_flag[lastGreater1ScanPos]	ae(v)
if(coeff_abs_level_greater2_flag[lastGreater1ScanPos])	
escapeDataPresent = 1	
}	

EXHIBIT 20
UNITED STATES PATENT NO. 9,036,701
CLAIM CHART FOR INFRINGEMENT OF CLAIMS 1, 8, AND 15 BY HISENSE ACCUSED PRODUCTS

<table>
<tr><th>U.S. PATENT NO. 9,036,701</th><th>HISENSE ACCUSED PRODUCTS</th></tr>
<tr><td></td><td>
<table>
<tr><td>for(n = 15; n >= 0; n− −) {</td><td></td></tr>
<tr><td>xC = (xS << 2) + ScanOrder[2][scanIdx][n][0]</td><td></td></tr>
<tr><td>yC = (yS << 2) + ScanOrder[2][scanIdx][n][1]</td><td></td></tr>
<tr><td>if(sig_coeff_flag[xC][yC] &&
(!sign_data_hiding_enabled_flag || !signHidden || (n != firstSigScanPos)))</td><td></td></tr>
<tr><td>coeff_sign_flag[n]</td><td>ae(v)</td></tr>
<tr><td>}</td><td></td></tr>
</table>
</td></tr>
</table>

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS

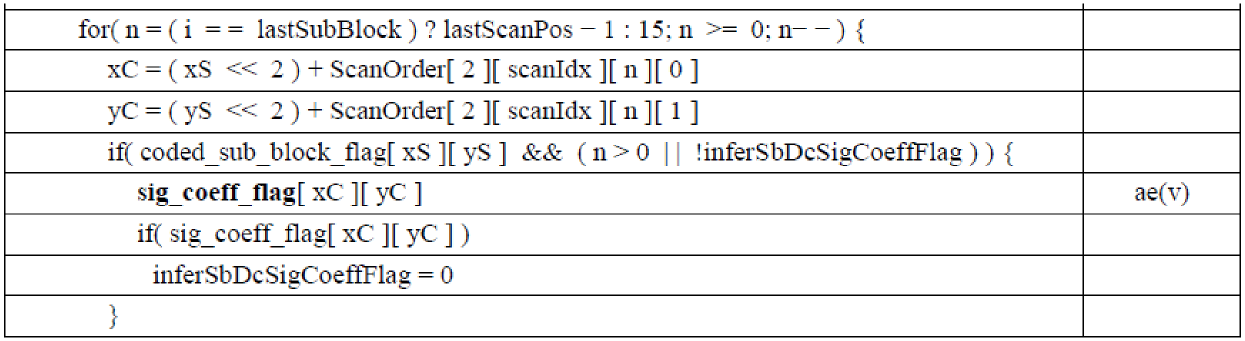


```
      numSigCoeff = 0
      sumAbsLevel = 0
      for( n = 15; n >= 0; n− − ) {
        xC = ( xS << 2 ) + ScanOrder[ 2 ][ scanIdx ][ n ][ 0 ]
        yC = ( yS << 2 ) + ScanOrder[ 2 ][ scanIdx ][ n ][ 1 ]
        if( sig_coeff_flag[ xC ][ yC ] ) {
          baseLevel = 1 + coeff_abs_level_greater1_flag[ n ] +
                      coeff_abs_level_greater2_flag[ n ]
          if( baseLevel = = ( ( numSigCoeff < 8 ) ?
                      ( (n = = lastGreater1ScanPos) ? 3 : 2 ) : 1 ) )
            coeff_abs_level_remaining[ n ]                                ae(v)
          TransCoeffLevel[ x0 ][ y0 ][ cIdx ][ xC ][ yC ] =
            ( coeff_abs_level_remaining[ n ] + baseLevel ) * ( 1 − 2 * coeff_sign_flag[ n ] )
          if( sign_data_hiding_enabled_flag && signHidden ) {
            sumAbsLevel += ( coeff_abs_level_remaining[ n ] + baseLevel )
            if( ( n = = firstSigScanPos ) && ( ( sumAbsLevel % 2 ) = = 1 ) )
              TransCoeffLevel[ x0 ][ y0 ][ cIdx ][ xC ][ yC ] =
                    −TransCoeffLevel[ x0 ][ y0 ][ cIdx ][ xC ][ yC ]
          }
          numSigCoeff++
        }
      }
    }
  }
```

ITU-T Rec. H.265 (12/2016) High efficiency video coding at pp. 58-60.

As set out in the Residual Coding Syntax in section 7.3.8.11, an H.265 compliant decoder parses *sig_coeff_flags* from the bit stream according to the logic reproduced below.

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS



for(n = (i = = lastSubBlock) ? lastScanPos − 1 : 15; n >= 0; n− −) {	
xC = (xS << 2) + ScanOrder[2][scanIdx][n][0]	
yC = (yS << 2) + ScanOrder[2][scanIdx][n][1]	
if(coded_sub_block_flag[xS][yS] && (n > 0 || !inferSbDcSigCoeffFlag)) {	
sig_coeff_flag[xC][yC]	ae(v)
if(sig_coeff_flag[xC][yC])	
inferSbDcSigCoeffFlag = 0	
}	

ITU-T Rec. H.265 (12/2016) High efficiency video coding at p. 59.

7.4.9.11 Residual coding semantics

. . .

sig_coeff_flag[xC][yC] specifies for the transform coefficient location (xC, yC) within the current transform block whether the corresponding transform coefficient level at the location (xC, yC) is non-zero as follows:

– If sig_coeff_flag[xC][yC] is equal to 0, the transform coefficient level at the location (xC, yC) is set equal to 0.

– Otherwise (sig_coeff_flag[xC][yC] is equal to 1), the transform coefficient level at the location (xC, yC) has a non-zero value.

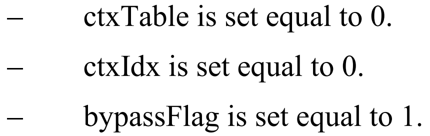
When sig_coeff_flag[xC][yC] is not present, it is inferred as follows:

– If (xC, yC) is the last significant location (LastSignificantCoeffX, LastSignificantCoeffY) in scan order or all of the following conditions are true, sig_coeff_flag[xC][yC] is inferred to be equal to 1:

  – (xC & 3, yC & 3) is equal to (0, 0).

  – inferSbDcSigCoeffFlag is equal to 1.

U.S. PATENT NO. 9,036,701	**HISENSE ACCUSED PRODUCTS**
	– coded_sub_block_flag[xS][yS] is equal to 1. – Otherwise, sig_coeff_flag[xC][yC] is inferred to be equal to 0. . . . ITU-T Rec. H.265 (12/2016) High efficiency video coding at p. 111. **9.3.4.2 Derivation process for ctxTable, ctxIdx and bypassFlag** **9.3.4.2.1 General** Input to this process is the position of the current bin within the bin string, binIdx. Outputs of this process are ctxTable, ctxIdx and bypassFlag. The values of ctxTable, ctxIdx and bypassFlag are derived as follows based on the entries for binIdx of the corresponding syntax element in Table 9-48: – If the entry in Table 9-48 is not equal to "bypass", "terminate" or "na", the values of binIdx are decoded by invoking the DecodeDecision process as specified in clause 9.3.4.3.2 and the following applies: – ctxTable is specified in Table 9-4. – The variable ctxInc is specified by the corresponding entry in Table 9-48 and when more than one value is listed in Table 9-48 for a binIdx, the assignment process for ctxInc for that binIdx is further specified in the clauses given in parenthesis. – The variable ctxIdxOffset is specified by the lowest value of ctxIdx in Table 9-4 depending on the current value of initType. – ctxIdx is set equal to the sum of ctxInc and ctxIdxOffset. – bypassFlag is set equal to 0. – Otherwise, if the entry in Table 9-48 is equal to "bypass", the values of binIdx are decoded by invoking the DecodeBypass process as specified in clause 9.3.4.3.4 and the following applies:

<table>
<tr><th>U.S. PATENT NO. 9,036,701</th><th>HISENSE ACCUSED PRODUCTS</th></tr>
<tr><td></td><td>

– ctxTable is set equal to 0.
– ctxIdx is set equal to 0.
– bypassFlag is set equal to 1.

ITU-T Rec. H.265 (12/2016) High efficiency video coding at p. 225.

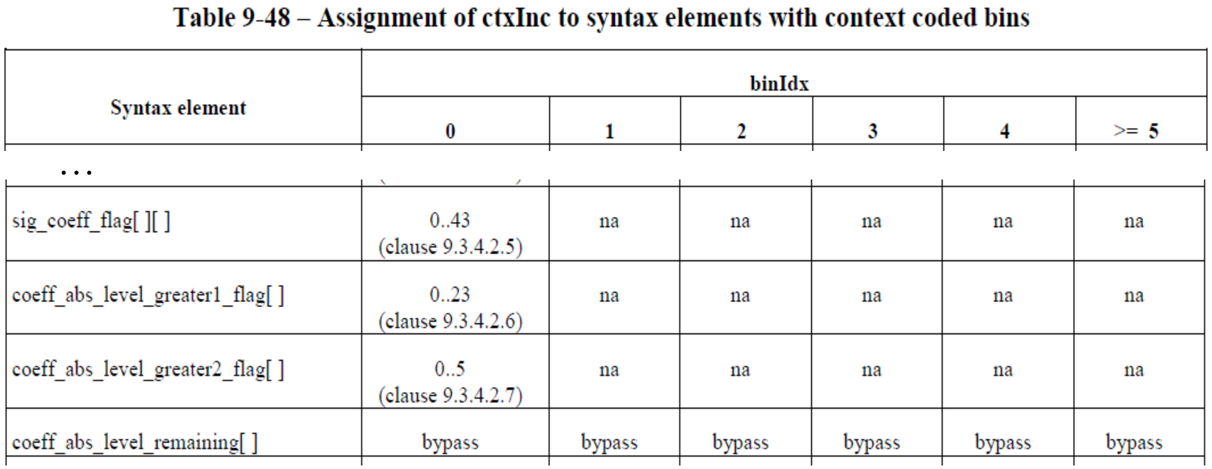


Table 9-48 – Assignment of ctxInc to syntax elements with context coded bins
<table>
<tr><th rowspan="2">Syntax element</th><th colspan="6">binIdx</th></tr>
<tr><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>>= 5</th></tr>
<tr><td>. . .</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>sig_coeff_flag[ ][ ]</td><td>0..43
(clause 9.3.4.2.5)</td><td>na</td><td>na</td><td>na</td><td>na</td><td>na</td></tr>
<tr><td>coeff_abs_level_greater1_flag[ ]</td><td>0..23
(clause 9.3.4.2.6)</td><td>na</td><td>na</td><td>na</td><td>na</td><td>na</td></tr>
<tr><td>coeff_abs_level_greater2_flag[ ]</td><td>0..5
(clause 9.3.4.2.7)</td><td>na</td><td>na</td><td>na</td><td>na</td><td>na</td></tr>
<tr><td>coeff_abs_level_remaining[ ]</td><td>bypass</td><td>bypass</td><td>bypass</td><td>bypass</td><td>bypass</td><td>bypass</td></tr>
</table>
ITU-T Rec. H.265 (12/2016) High efficiency video coding at pp. 225-227.
</td></tr>
<tr><td>[c] entropy coding symbols that correspond to the first category of syntax elements and that have been subjected to a context update; and</td><td>Each of the Accused Products, such as the Hisense 43A7N, performs a method comprising entropy coding symbols that correspond to the first category of syntax elements and that have been subjected to a context update.</td></tr>
</table>

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
	For example, and without limitation, the H.265 Standard specifies the following regarding the decoding process. Each of the Accused Products performs a method comprising entropy coding symbols that correspond to the first category of syntax elements and that have been subjected to a context update. The following specifications provide further evidence of how each of the Accused Products operates: **9.3.4.2 Derivation process for ctxTable, ctxIdx and bypassFlag** **9.3.4.2.1 General** Input to this process is the position of the current bin within the bin string, binIdx. Outputs of this process are ctxTable, ctxIdx and bypassFlag. The values of ctxTable, ctxIdx and bypassFlag are derived as follows based on the entries for binIdx of the corresponding syntax element in Table 9-48: – If the entry in Table 9-48 is not equal to "bypass", "terminate" or "na", the values of binIdx are decoded by invoking the DecodeDecision process as specified in clause 9.3.4.3.2 and the following applies: – ctxTable is specified in Table 9-4. – The variable ctxInc is specified by the corresponding entry in Table 9-48 and when more than one value is listed in Table 9-48 for a binIdx, the assignment process for ctxInc for that binIdx is further specified in the clauses given in parenthesis. – The variable ctxIdxOffset is specified by the lowest value of ctxIdx in Table 9-4 depending on the current value of initType. – ctxIdx is set equal to the sum of ctxInc and ctxIdxOffset. – bypassFlag is set equal to 0. . . . ITU-T Rec. H.265 (12/2016) High efficiency video coding at p. 225.

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
	9.3.4.3 Arithmetic decoding process **9.3.4.3.1 General** Inputs to this process are ctxTable, ctxIdx and bypassFlag, as derived in clause 9.3.4.2, and the state variables ivlCurrRange and ivlOffset of the arithmetic decoding engine. Output of this process is the value of the bin. Figure 9-5 illustrates the whole arithmetic decoding process for a single bin. For decoding the value of a bin, the context index table ctxTable and the ctxIdx are passed to the arithmetic decoding process DecodeBin(ctxTable, ctxIdx), which is specified as follows: – If bypassFlag is equal to 1, DecodeBypass( ) as specified in clause 9.3.4.3.4 is invoked. – Otherwise, if bypassFlag is equal to 0, ctxTable is equal to 0 and ctxIdx is equal to 0, DecodeTerminate( ) as specified in clause 9.3.4.3.5 is invoked. – Otherwise (bypassFlag is equal to 0 and ctxTable is not equal to 0), DecodeDecision( ) as specified in clause 9.3.4.3.2 is invoked. . . . ITU-T Rec. H.265 (12/2016) High efficiency video coding at p. 232. **9.3.4.2.6 Derivation process of ctxInc for the syntax element coeff_abs_level_greater1_flag** Inputs to this process are the colour component index cIdx, the current sub-block scan index i and the current coefficient scan index n within the current sub-block. Output of this process is the variable ctxInc. . . . ITU-T Rec. H.265 (12/2016) High efficiency video coding at p 231. **9.3.4.2.7 Derivation process of ctxInc for the syntax element coeff_abs_level_greater2_flag**

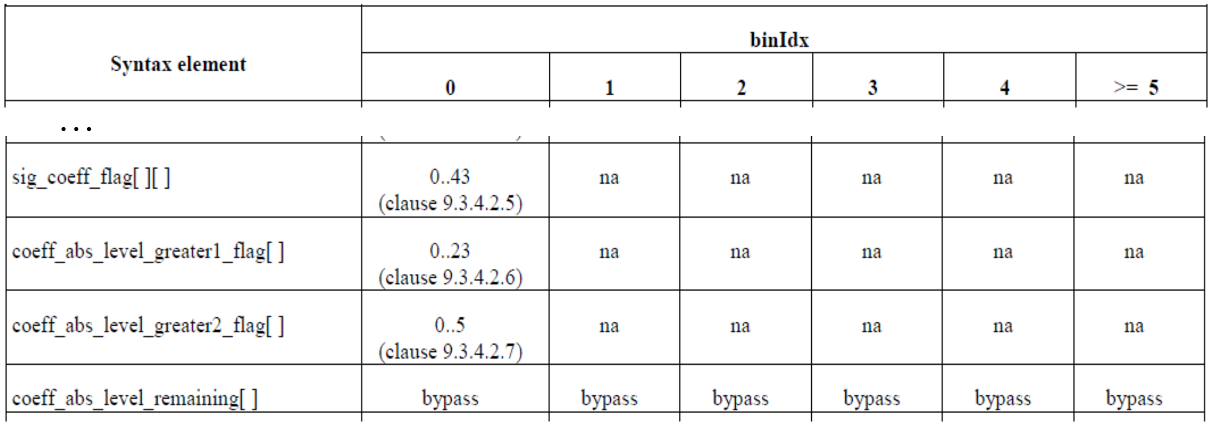
U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS

> Inputs to this process are the colour component index cIdx, the current sub-block scan index i and the current coefficient scan index n within the current sub-block.
>
> Output of this process is the variable ctxInc.
>
> . . .

ITU-T Rec. H.265 (12/2016) High efficiency video coding at p. 232.

For example, the symbols that correspond to the first category of syntax elements and that have been subjected to a context update are the **coeff_abs_level_greater1_flag**, and **coeff_abs_level_greater2_flag** that are used to represent, at least in part, the *non-zero transform coefficient levels* in the first category, as shown in Table 9-48 and described in section 9.3.2.5 of the H.265 Standard reproduced below, in which the process for calculating the context for each of **coeff_abs_level_greater1_flag** and **coeff_abs_level_greater2_flag** is provided.

Table 9-48 – Assignment of ctxInc to syntax elements with context coded bins



Syntax element	binIdx					
	0	1	2	3	4	>= 5
. . .						
sig_coeff_flag[ ][ ]	0..43 (clause 9.3.4.2.5)	na	na	na	na	na
coeff_abs_level_greater1_flag[ ]	0..23 (clause 9.3.4.2.6)	na	na	na	na	na
coeff_abs_level_greater2_flag[ ]	0..5 (clause 9.3.4.2.7)	na	na	na	na	na
coeff_abs_level_remaining[ ]	bypass	bypass	bypass	bypass	bypass	bypass

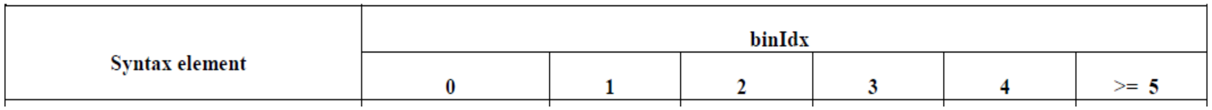
ITU-T Rec. H.265 (12/2016) High efficiency video coding at pp. 225-227.

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
	9.3.2.5 Synchronization process for context variables, Rice parameter initialization states, and palette predictor variables Inputs to this process are: – The variables tableStateSync and tableMPSSync containing the values of the variables pStateIdx and valMps used in the storage process of context variables that are assigned to all syntax elements in clauses 7.3.8.1 through 7.3.8.12, except end_of_slice_segment_flag, end_of_subset_one_bit and pcm_flag. – The variable tableStatCoeffSync containing the values of the variables StatCoeff[k] used in the storage process of context variables and Rice parameter initialization states. – The variables PredictorPaletteSizeSync and tablePredictorPaletteEntriesSync containing the values used in the storage process of palette predictor variables. Outputs of this process are: – The initialized CABAC context variables indexed by ctxTable and ctxIdx. – The initialized Rice parameter initialization states StatCoeff indexed by k. – The palette predictor variables, PredictorPaletteSize and PredictorPaletteEntries. . . . ITU-T Rec. H.265 (12/2016) High efficiency video coding at p. 215.
[d] entropy coding symbols that correspond to the second category of syntax elements and that have bypassed context updating.	Each of the Accused Products, such as the Hisense 43A7N, performs a method comprising entropy coding symbols that correspond to the second category of syntax elements and that have bypassed context updating. For example, and without limitation, the H.265 Standard specifies the following regarding the decoding process. Each of the Accused Products performs a method comprising entropy coding symbols that correspond to the second category of syntax elements and that have bypassed context updating.

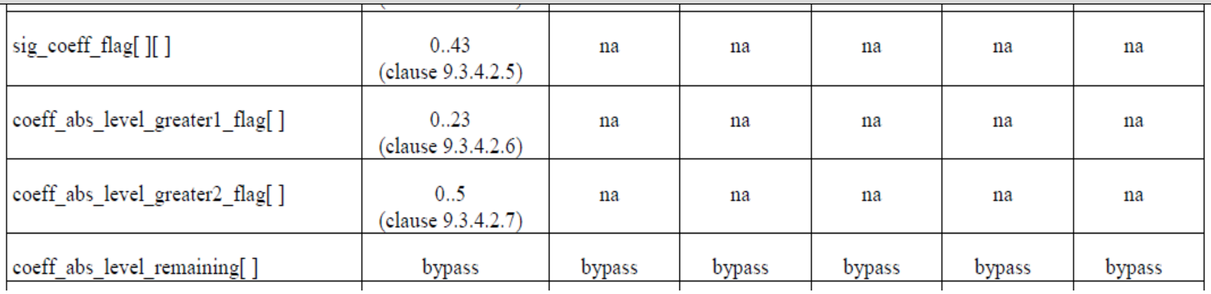
EXHIBIT 20

UNITED STATES PATENT NO. 9,036,701

CLAIM CHART FOR INFRINGEMENT OF CLAIMS 1, 8, AND 15 BY HISENSE ACCUSED PRODUCTS

<table>
<tr><th>U.S. PATENT NO. 9,036,701</th><th>HISENSE ACCUSED PRODUCTS</th></tr>
<tr><td></td><td>
The following specifications provide further evidence of how each of the Accused Products operates:
9.3.4.2 Derivation process for ctxTable, ctxIdx and bypassFlag
9.3.4.2.1 General
Input to this process is the position of the current bin within the bin string, binIdx.
Outputs of this process are ctxTable, ctxIdx and bypassFlag.
The values of ctxTable, ctxIdx and bypassFlag are derived as follows based on the entries for binIdx of the corresponding syntax element in Table 9-48:
. . .
– Otherwise, if the entry in Table 9-48 is equal to "bypass", the values of binIdx are decoded by invoking the DecodeBypass process as specified in clause 9.3.4.3.4 and the following applies:
  – ctxTable is set equal to 0.
  – ctxIdx is set equal to 0.
  – bypassFlag is set equal to 1.
. . .
Table 9-48 – Assignment of ctxInc to syntax elements with context coded bins

<table>
<tr><th rowspan="2">Syntax element</th><th colspan="6">binIdx</th></tr>
<tr><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>>= 5</th></tr>
</table>
. . .
</td></tr>
</table>

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS



sig_coeff_flag[ ][ ]	0..43 (clause 9.3.4.2.5)	na	na	na	na	na
coeff_abs_level_greater1_flag[ ]	0..23 (clause 9.3.4.2.6)	na	na	na	na	na
coeff_abs_level_greater2_flag[ ]	0..5 (clause 9.3.4.2.7)	na	na	na	na	na
coeff_abs_level_remaining[ ]	bypass	bypass	bypass	bypass	bypass	bypass

ITU-T Rec. H.265 (12/2016) High efficiency video coding at pp. 225-227.

> ### 9.3.4.3 Arithmetic decoding process
>
> #### 9.3.4.3.1 General
>
> Inputs to this process are ctxTable, ctxIdx and bypassFlag, as derived in clause 9.3.4.2, and the state variables ivlCurrRange and ivlOffset of the arithmetic decoding engine.
>
> Output of this process is the value of the bin.
>
> Figure 9-5 illustrates the whole arithmetic decoding process for a single bin. For decoding the value of a bin, the context index table ctxTable and the ctxIdx are passed to the arithmetic decoding process DecodeBin(ctxTable, ctxIdx), which is specified as follows:
>
> – If bypassFlag is equal to 1, DecodeBypass( ) as specified in clause 9.3.4.3.4 is invoked.
>
> – Otherwise, if bypassFlag is equal to 0, ctxTable is equal to 0 and ctxIdx is equal to 0, DecodeTerminate( ) as specified in clause 9.3.4.3.5 is invoked.
>
> – Otherwise (bypassFlag is equal to 0 and ctxTable is not equal to 0), DecodeDecision( ) as specified in clause 9.3.4.3.2 is invoked.
>
> . . .

ITU-T Rec. H.265 (12/2016) High efficiency video coding at p. 232.

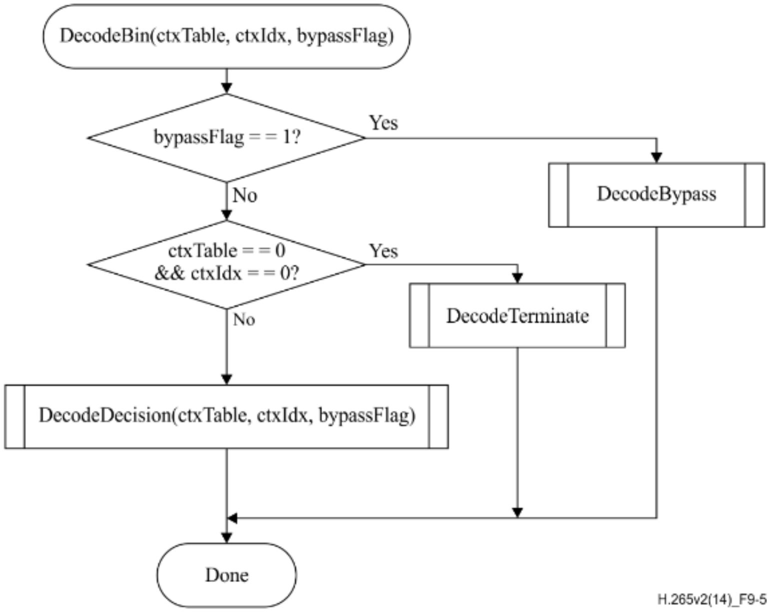
U.S. PATENT NO. 9,036,701	**HISENSE ACCUSED PRODUCTS**
	 **Figure 9-5 – Overview of the arithmetic decoding process for a single bin (informative)** NOTE – Arithmetic coding is based on the principle of recursive interval subdivision. Given a probability estimation p(0) and p(1) = 1 − p(0) of a binary decision (0, 1), an initially given code sub-interval with the range ivlCurrRange will be subdivided into two sub-intervals having range p(0) * ivlCurrRange and ivlCurrRange − p(0) * ivlCurrRange, respectively. Depending on the decision, which has been observed, the corresponding sub-interval will be chosen as the new code interval, and a binary code string pointing into that interval will represent the sequence of observed binary decisions. It is useful to distinguish between the most probable symbol (MPS) and the least probable symbol (LPS), so that binary decisions have to be identified as either MPS or LPS, rather than 0 or 1. Given this terminology, each context is specified by the probability pLPS of the LPS and the value of MPS (valMps), which is either 0 or 1. The arithmetic core engine in this Specification has three distinct properties:

EXHIBIT 20
UNITED STATES PATENT NO. 9,036,701
CLAIM CHART FOR INFRINGEMENT OF CLAIMS 1, 8, AND 15 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
	– The probability estimation is performed by means of a finite-state machine with a table-based transition process between 64 different representative probability states { pLPS(pStateIdx) | 0 <= pStateIdx < 64 } for the LPS probability pLPS. The numbering of the states is arranged in such a way that the probability state with index pStateIdx = 0 corresponds to an LPS probability value of 0.5, with decreasing LPS probability towards higher state indices. – The range ivlCurrRange representing the state of the coding engine is quantized to a small set {Q1,...,Q4} of pre-set quantization values prior to the calculation of the new interval range. Storing a table containing all 64x4 pre-computed product values of Qi * pLPS(pStateIdx) allows a multiplication-free approximation of the product ivlCurrRange * pLPS(pStateIdx). – For syntax elements or parts thereof for which an approximately uniform probability distribution is assumed to be given a separate simplified encoding and decoding bypass process is used. ITU-T Rec. H.265 (12/2016) High efficiency video coding at p. 233.

2. **Claim 3**

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
3. Pre A method according to claim 1 further comprising	Each of the Accused Products, such as the Hisense 43A7N, performs a method according to claim 1. The evidence cited for Claim 1 applies to this claim limitation.
[a] performing binarization of the symbols that correspond to the second category of syntax elements prior to bypassing the context updating	Each of the Accused Products, such as the Hisense 43A7N, performs a method according to claim 1 further comprising performing binarization of the symbols that correspond to the second category of syntax elements prior to bypassing the context updating. The evidence cited for Claim 1 applies to this claim limitation.

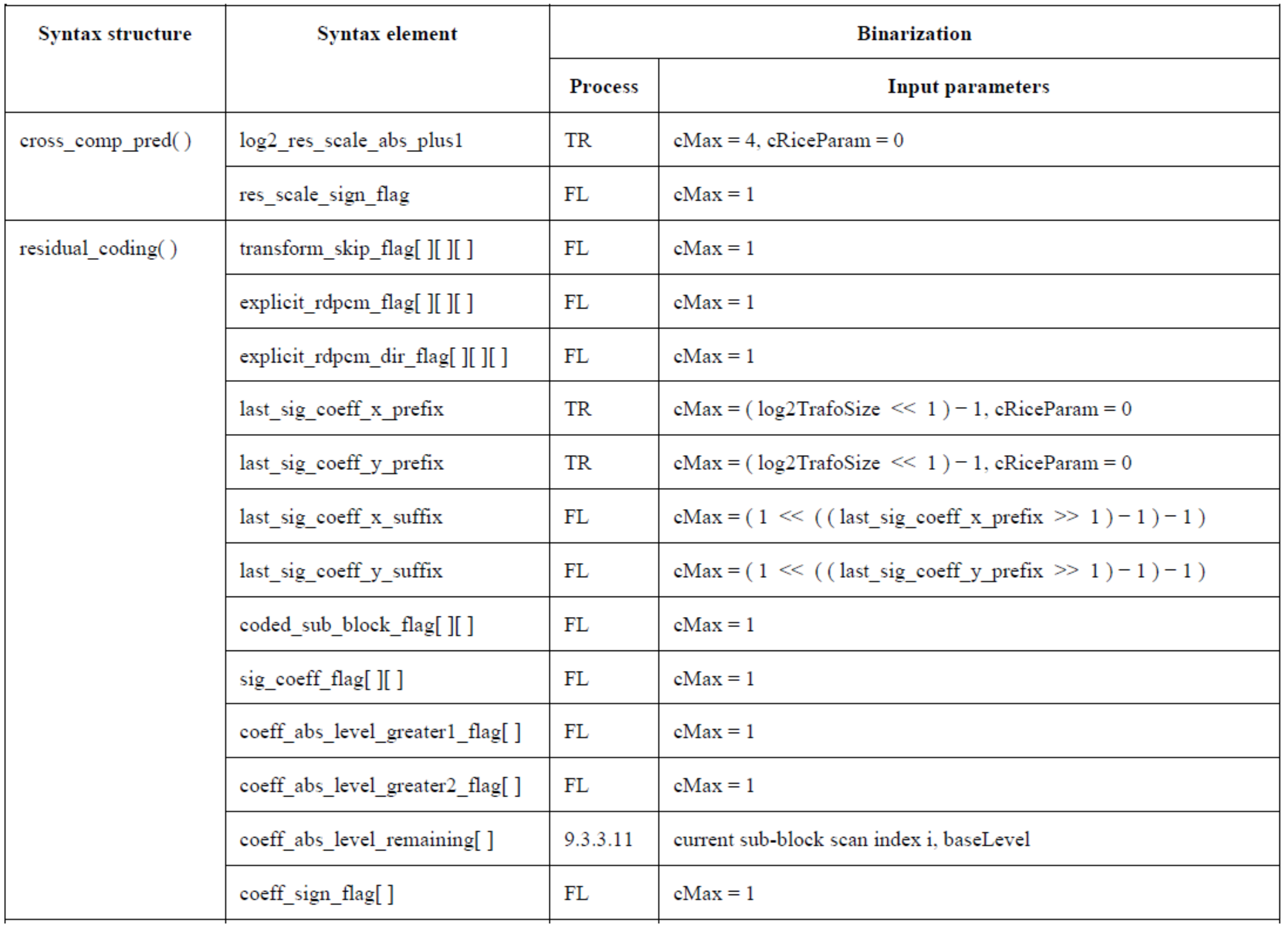
U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
	For example, and without limitation, the H.265 Standard specifies the following regarding the decoding process. Each of the Accused Products performs a method according to claim 1 further comprising performing binarization of the symbols that correspond to the second category of syntax elements prior to bypassing the context updating. The following specifications provide further evidence of how each of the Accused Products operates: **9.3 CABAC parsing for slice segment data** **9.3.1 General** This process is invoked when parsing syntax elements with descriptor ae(v) in clauses 7.3.8.1 through 7.3.8.12. Inputs to this process are a request for a value of a syntax element and value of prior parsed syntax elements. Outputs of this process is the value of the syntax element . . . For each requested value of a syntax element a binarization is derived as specified in clause 9.3.3. The binarization for the syntax element and the sequence of parsed bins determines the decoding process flow as described in clause 9.3.4. ITU-T Rec. H.265 (12/2016) High efficiency video coding at p. 201. **9.3.3 Binarization process** **9.3.3.1 General** Inputs to this process is a request for syntax element. Output of this process is the binarization of the syntax element.

U.S. PATENT NO. 9,036,701	**HISENSE ACCUSED PRODUCTS**
	Table 9-43 specifies the type of binarization process associated with each syntax element and corresponding inputs. The specification of the truncated Rice (TR) binarization process, the k-th order Exp-Golomb (EGk) binarization process, limited k-th order Exp-Golomb (EGk) binarization process, the fixed-length (FL) binarization process, and the truncated binary binarization process are given in clauses 9.3.3.2 through 9.3.3.6, respectively. Other binarizations are specified in clauses 9.3.3.7 through 9.3.3.14. . . .

EXHIBIT 20
UNITED STATES PATENT NO. 9,036,701
CLAIM CHART FOR INFRINGEMENT OF CLAIMS 1, 8, AND 15 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS

Table 9-43 – Syntax elements and associated binarizations



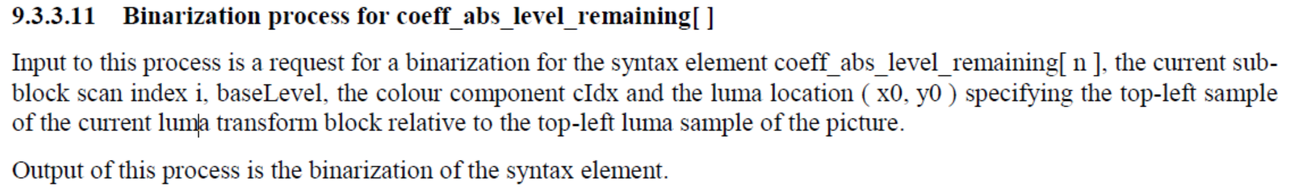
Syntax structure	Syntax element	Binarization	
		Process	Input parameters
cross_comp_pred( )	log2_res_scale_abs_plus1	TR	cMax = 4, cRiceParam = 0
	res_scale_sign_flag	FL	cMax = 1
residual_coding( )	transform_skip_flag[ ][ ][ ]	FL	cMax = 1
	explicit_rdpcm_flag[ ][ ][ ]	FL	cMax = 1
	explicit_rdpcm_dir_flag[ ][ ][ ]	FL	cMax = 1
	last_sig_coeff_x_prefix	TR	cMax = (log2TrafoSize << 1) − 1, cRiceParam = 0
	last_sig_coeff_y_prefix	TR	cMax = (log2TrafoSize << 1) − 1, cRiceParam = 0
	last_sig_coeff_x_suffix	FL	cMax = (1 << ((last_sig_coeff_x_prefix >> 1) − 1) − 1)
	last_sig_coeff_y_suffix	FL	cMax = (1 << ((last_sig_coeff_y_prefix >> 1) − 1) − 1)
	coded_sub_block_flag[ ][ ]	FL	cMax = 1
	sig_coeff_flag[ ][ ]	FL	cMax = 1
	coeff_abs_level_greater1_flag[ ]	FL	cMax = 1
	coeff_abs_level_greater2_flag[ ]	FL	cMax = 1
	coeff_abs_level_remaining[ ]	9.3.3.11	current sub-block scan index i, baseLevel
	coeff_sign_flag[ ]	FL	cMax = 1

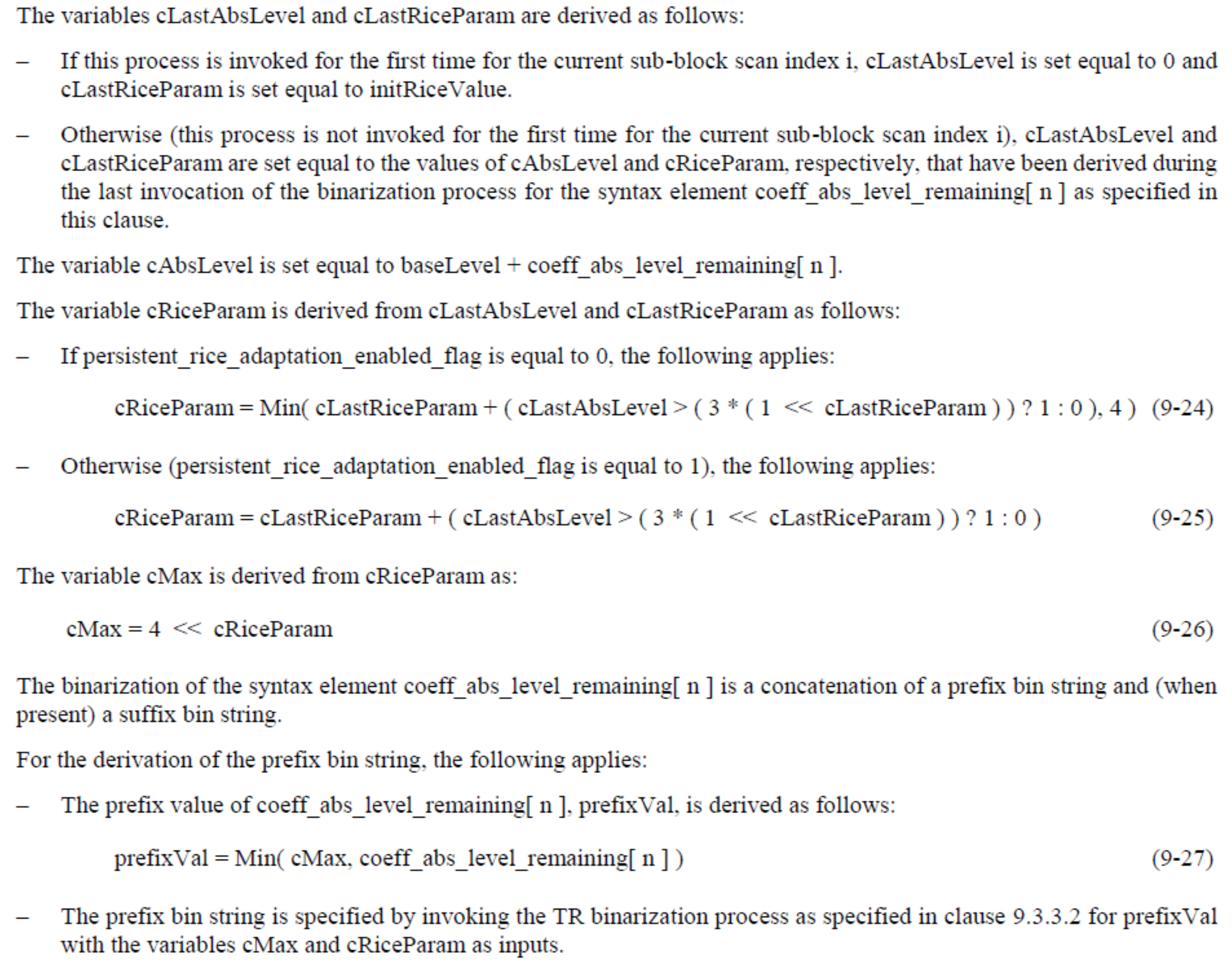
ITU-T Rec. H.265 (12/2016) High efficiency video coding at pp. 216-218.

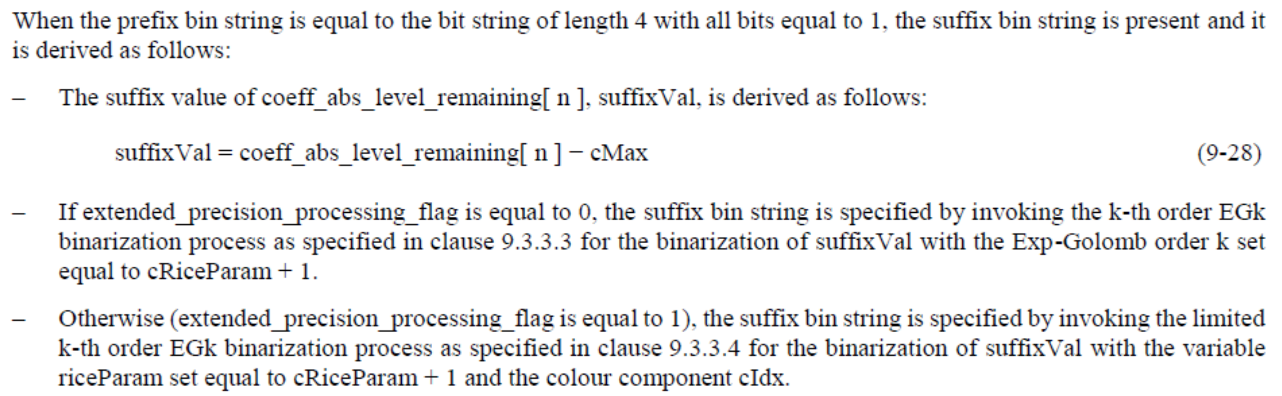
U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
	9.3.4 Decoding process flow **9.3.4.1 General** Inputs to this process are all bin strings of the binarization of the requested syntax element as specified in clause 9.3.3. Output of this process is the value of the syntax element. ITU-T Rec. H.265 (12/2016) High efficiency video coding at p. 225.

3. Claim 4

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
4. Pre A method according to claim 3	Each of the Accused Products, such as the Hisense 43A7N, performs a method according to claim 3. The evidence cited for claims 1 and 3 applies to this limitation.
[a] wherein performing binarization comprises performing binarization so as to target a predetermined probability.	Each of the Accused Products, such as the Hisense 43A7N, performs a method according to claim 3 wherein performing binarization comprises performing binarization so as to target a predetermined probability. The evidence cited for Claim 1 applies to this claim limitation. For example, and without limitation, the H.265 Standard specifies the following regarding the decoding process. Each of the Accused Products performs a method according to claim 3 wherein performing binarization comprises performing binarization so as to target a predetermined probability. The following specifications provide further evidence of how each of the Accused Products operates:

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
	 **9.3.3.11 Binarization process for coeff_abs_level_remaining[ ]** Input to this process is a request for a binarization for the syntax element coeff_abs_level_remaining[n], the current sub-block scan index i, baseLevel, the colour component cIdx and the luma location (x0, y0) specifying the top-left sample of the current luma transform block relative to the top-left luma sample of the picture. Output of this process is the binarization of the syntax element.

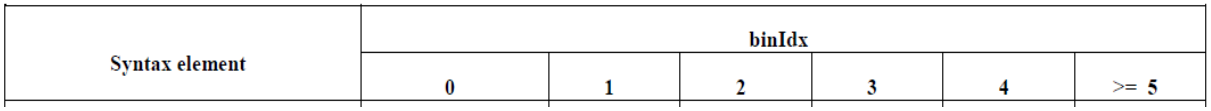
U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
	 The variables cLastAbsLevel and cLastRiceParam are derived as follows: – If this process is invoked for the first time for the current sub-block scan index i, cLastAbsLevel is set equal to 0 and cLastRiceParam is set equal to initRiceValue. – Otherwise (this process is not invoked for the first time for the current sub-block scan index i), cLastAbsLevel and cLastRiceParam are set equal to the values of cAbsLevel and cRiceParam, respectively, that have been derived during the last invocation of the binarization process for the syntax element coeff_abs_level_remaining[n] as specified in this clause. The variable cAbsLevel is set equal to baseLevel + coeff_abs_level_remaining[n]. The variable cRiceParam is derived from cLastAbsLevel and cLastRiceParam as follows: – If persistent_rice_adaptation_enabled_flag is equal to 0, the following applies: $\text{cRiceParam} = \text{Min}(\text{cLastRiceParam} + (\text{cLastAbsLevel} > (3 * (1 \ll \text{cLastRiceParam})) ? 1 : 0), 4)$ (9-24) – Otherwise (persistent_rice_adaptation_enabled_flag is equal to 1), the following applies: $\text{cRiceParam} = \text{cLastRiceParam} + (\text{cLastAbsLevel} > (3 * (1 \ll \text{cLastRiceParam})) ? 1 : 0)$ (9-25) The variable cMax is derived from cRiceParam as: $\text{cMax} = 4 \ll \text{cRiceParam}$ (9-26) The binarization of the syntax element coeff_abs_level_remaining[n] is a concatenation of a prefix bin string and (when present) a suffix bin string. For the derivation of the prefix bin string, the following applies: – The prefix value of coeff_abs_level_remaining[n], prefixVal, is derived as follows: $\text{prefixVal} = \text{Min}(\text{cMax}, \text{coeff\_abs\_level\_remaining}[n])$ (9-27) – The prefix bin string is specified by invoking the TR binarization process as specified in clause 9.3.3.2 for prefixVal with the variables cMax and cRiceParam as inputs.

U.S. PATENT NO. 9,036,701	**HISENSE ACCUSED PRODUCTS**
	 When the prefix bin string is equal to the bit string of length 4 with all bits equal to 1, the suffix bin string is present and it is derived as follows: – The suffix value of coeff_abs_level_remaining[n], suffixVal, is derived as follows: $\text{suffixVal} = \text{coeff\_abs\_level\_remaining}[\ n\ ] - \text{cMax} \quad (9\text{-}28)$ – If extended_precision_processing_flag is equal to 0, the suffix bin string is specified by invoking the k-th order EGk binarization process as specified in clause 9.3.3.3 for the binarization of suffixVal with the Exp-Golomb order k set equal to cRiceParam + 1. – Otherwise (extended_precision_processing_flag is equal to 1), the suffix bin string is specified by invoking the limited k-th order EGk binarization process as specified in clause 9.3.3.4 for the binarization of suffixVal with the variable riceParam set equal to cRiceParam + 1 and the colour component cIdx. ITU-T Rec. H.265 (12/2016) High efficiency video coding at pp. 222-224. **9.3.4.2 Derivation process for ctxTable, ctxIdx and bypassFlag** **9.3.4.2.1 General** Input to this process is the position of the current bin within the bin string, binIdx. Outputs of this process are ctxTable, ctxIdx and bypassFlag. The values of ctxTable, ctxIdx and bypassFlag are derived as follows based on the entries for binIdx of the corresponding syntax element in Table 9-48: . . . – Otherwise, if the entry in Table 9-48 is equal to "bypass", the values of binIdx are decoded by invoking the DecodeBypass process as specified in clause 9.3.4.3.4 and the following applies: – ctxTable is set equal to 0. – ctxIdx is set equal to 0. – bypassFlag is set equal to 1.

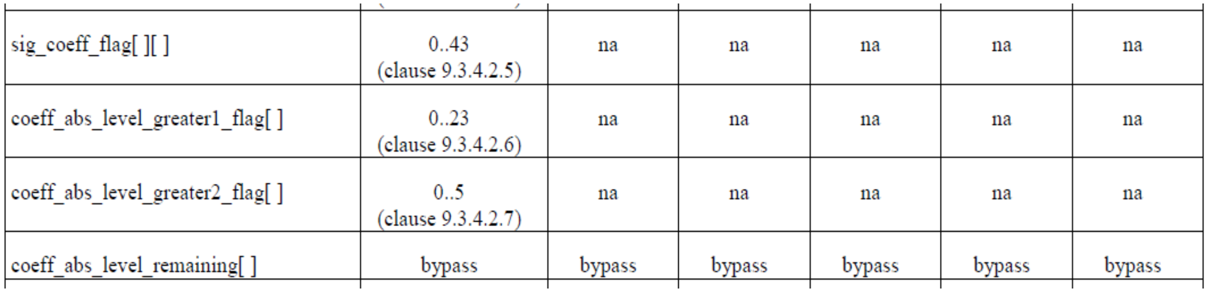
U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS

. . .

Table 9-48 – Assignment of ctxInc to syntax elements with context coded bins



. . .



Syntax element	binIdx					
	0	1	2	3	4	>= 5
sig_coeff_flag[ ][ ]	0..43 (clause 9.3.4.2.5)	na	na	na	na	na
coeff_abs_level_greater1_flag[ ]	0..23 (clause 9.3.4.2.6)	na	na	na	na	na
coeff_abs_level_greater2_flag[ ]	0..5 (clause 9.3.4.2.7)	na	na	na	na	na
coeff_abs_level_remaining[ ]	bypass	bypass	bypass	bypass	bypass	bypass

ITU-T Rec. H.265 (12/2016) High efficiency video coding at pp. 225-227.

9.3.4.3 Arithmetic decoding process

9.3.4.3.1 General

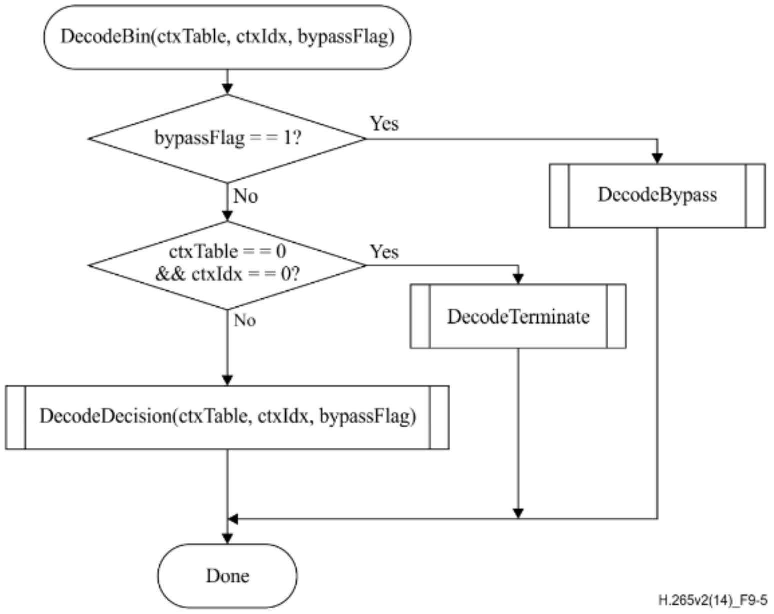
Inputs to this process are ctxTable, ctxIdx and bypassFlag, as derived in clause 9.3.4.2, and the state variables ivlCurrRange and ivlOffset of the arithmetic decoding engine.

Output of this process is the value of the bin.

Figure 9-5 illustrates the whole arithmetic decoding process for a single bin. For decoding the value of a bin, the context index table ctxTable and the ctxIdx are passed to the arithmetic decoding process DecodeBin(ctxTable, ctxIdx), which is specified as follows:

– If bypassFlag is equal to 1, DecodeBypass( ) as specified in clause 9.3.4.3.4 is invoked.

– Otherwise, if bypassFlag is equal to 0, ctxTable is equal to 0 and ctxIdx is equal to 0,

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
	DecodeTerminate( ) as specified in clause 9.3.4.3.5 is invoked. – Otherwise (bypassFlag is equal to 0 and ctxTable is not equal to 0), DecodeDecision( ) as specified in clause 9.3.4.3.2 is invoked. . . . ITU-T Rec. H.265 (12/2016) High efficiency video coding at p. 232.  **Figure 9-5 – Overview of the arithmetic decoding process for a single bin (informative)** NOTE – Arithmetic coding is based on the principle of recursive interval subdivision. Given a probability estimation p(0) and p(1) = 1 − p(0) of a binary decision (0, 1), an initially given code sub-interval with

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
	the range ivlCurrRange will be subdivided into two sub-intervals having range p(0) * ivlCurrRange and ivlCurrRange − p(0) * ivlCurrRange, respectively. Depending on the decision, which has been observed, the corresponding sub-interval will be chosen as the new code interval, and a binary code string pointing into that interval will represent the sequence of observed binary decisions. It is useful to distinguish between the most probable symbol (MPS) and the least probable symbol (LPS), so that binary decisions have to be identified as either MPS or LPS, rather than 0 or 1. Given this terminology, each context is specified by the probability pLPS of the LPS and the value of MPS (valMps), which is either 0 or 1. The arithmetic core engine in this Specification has three distinct properties: – The probability estimation is performed by means of a finite-state machine with a table-based transition process between 64 different representative probability states { pLPS(pStateIdx) | 0 <= pStateIdx < 64 } for the LPS probability pLPS. The numbering of the states is arranged in such a way that the probability state with index pStateIdx = 0 corresponds to an LPS probability value of 0.5, with decreasing LPS probability towards higher state indices. – The range ivlCurrRange representing the state of the coding engine is quantized to a small set {Q1,...,Q4} of pre-set quantization values prior to the calculation of the new interval range. Storing a table containing all 64x4 pre-computed product values of Qi * pLPS(pStateIdx) allows a multiplication-free approximation of the product ivlCurrRange * pLPS(pStateIdx). – For syntax elements or parts thereof for which an approximately uniform probability distribution is assumed to be given a separate simplified encoding and decoding bypass process is used. ITU-T Rec. H.265 (12/2016) High efficiency video coding at p. 233.

9.3.4.3.4 Bypass decoding process for binary decisions

Inputs to this process are bits from slice segment data and the variables ivlCurrRange and ivlOffset.

Outputs of this process are the updated variable ivlOffset and the decoded value binVal.

The bypass decoding process is invoked when bypassFlag is equal to 1. Figure 9-8 shows a flowchart of the corresponding process.

First, the value of ivlOffset is doubled, i.e., left-shifted by 1 and a single bit is shifted into ivlOffset by using read_bits(1). Then, the value of ivlOffset is compared to the value of ivlCurrRange and then the following applies:

– If ivlOffset is greater than or equal to ivlCurrRange, the variable binVal is set equal to 1 and ivlOffset is decremented by ivlCurrRange.

– Otherwise (ivlOffset is less than ivlCurrRange), the variable binVal is set equal to 0.

The bitstream shall not contain data that result in a value of ivlOffset being greater than or equal to ivlCurrRange upon completion of this process.

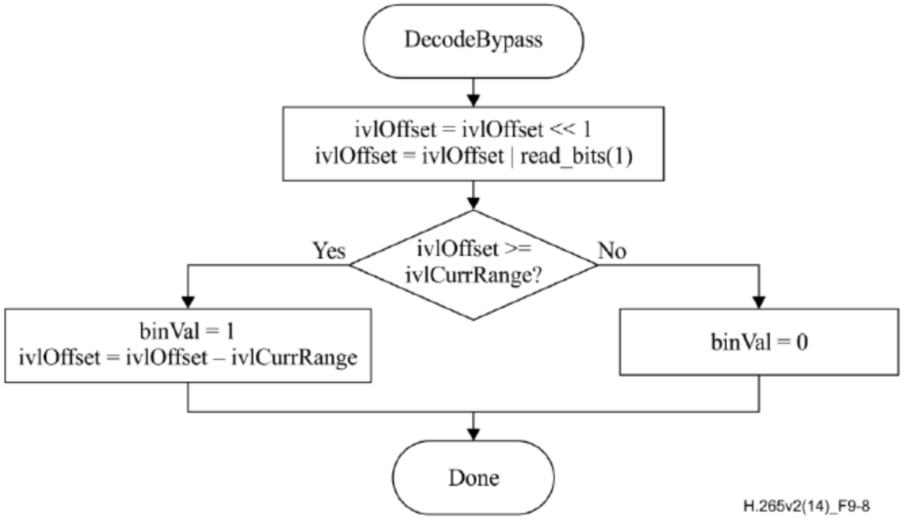
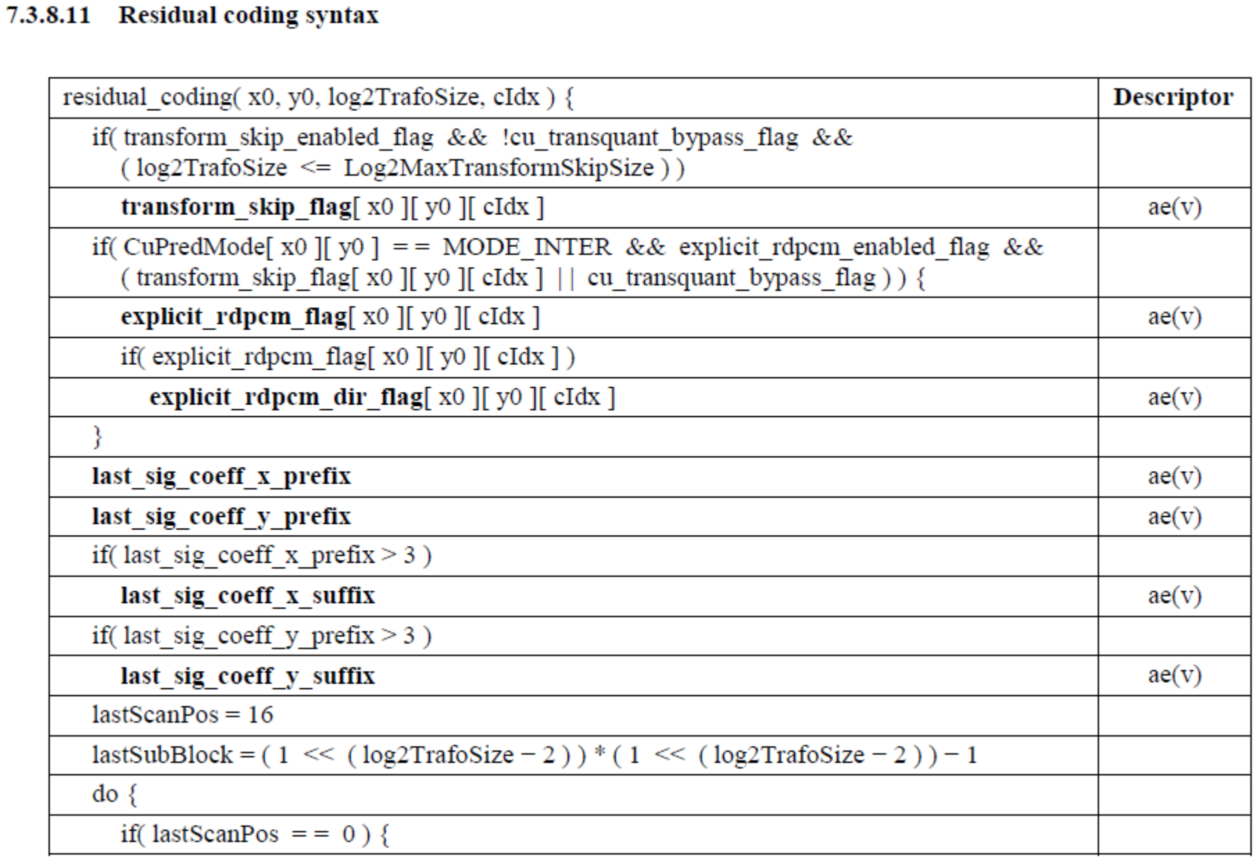


Figure 9-8 – Flowchart of bypass decoding process

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
	ITU-T Rec. H.265 (12/2016) High efficiency video coding at p. 237.

4. Claim 6

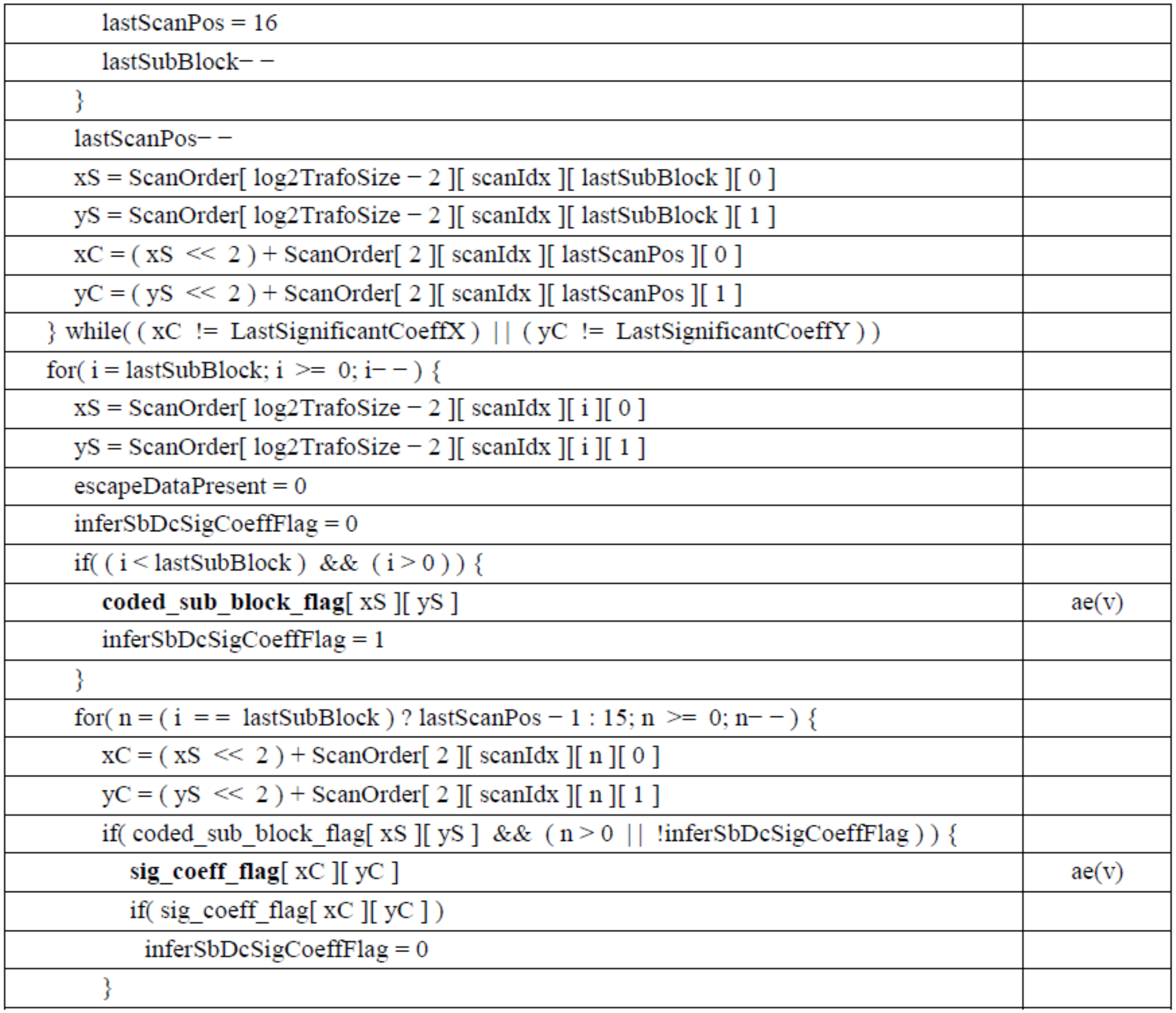
U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
6. Pre A method according to claim 1	Each of the Accused Products, such as the Hisense 43A7N, performs a method according to claim 1 The evidence cited for Claim 1 applies to this claim limitation.
wherein categorizing the plurality of syntax elements comprises categorizing the plurality of syntax elements based on a relationship of the frequency of occurrence of the syntax elements to a predefined threshold.	Each of the Accused Products, such as the Hisense 43A7N, performs a method according to claim 1 wherein categorizing the plurality of syntax elements comprises categorizing the plurality of syntax elements based on a relationship of the frequency of occurrence of the syntax elements to a predefined threshold. The evidence cited for Claim 1 applies to this claim limitation. For example, and without limitation, the H.265 Standard specifies the following regarding the decoding process. Each of the Accused Products performs a method according to claim 1 wherein categorizing the plurality of syntax elements comprises categorizing the plurality of syntax elements based on a relationship of the frequency of occurrence of the syntax elements to a predefined threshold. The following specifications provide further evidence of how each of the Accused Products operates:

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
	

7.3.8.11 Residual coding syntax

residual_coding(x0, y0, log2TrafoSize, cIdx) {	**Descriptor**
if(transform_skip_enabled_flag && !cu_transquant_bypass_flag && (log2TrafoSize <= Log2MaxTransformSkipSize))	
transform_skip_flag[x0][y0][cIdx]	ae(v)
if(CuPredMode[x0][y0] = = MODE_INTER && explicit_rdpcm_enabled_flag && (transform_skip_flag[x0][y0][cIdx] || cu_transquant_bypass_flag)) {	
explicit_rdpcm_flag[x0][y0][cIdx]	ae(v)
if(explicit_rdpcm_flag[x0][y0][cIdx])	
explicit_rdpcm_dir_flag[x0][y0][cIdx]	ae(v)
}	
last_sig_coeff_x_prefix	ae(v)
last_sig_coeff_y_prefix	ae(v)
if(last_sig_coeff_x_prefix > 3)	
last_sig_coeff_x_suffix	ae(v)
if(last_sig_coeff_y_prefix > 3)	
last_sig_coeff_y_suffix	ae(v)
lastScanPos = 16	
lastSubBlock = (1 << (log2TrafoSize − 2)) * (1 << (log2TrafoSize − 2)) − 1	
do {	
if(lastScanPos = = 0) {	

EXHIBIT 20
UNITED STATES PATENT NO. 9,036,701
CLAIM CHART FOR INFRINGEMENT OF CLAIMS 1, 8, AND 15 BY HISENSE ACCUSED PRODUCTS

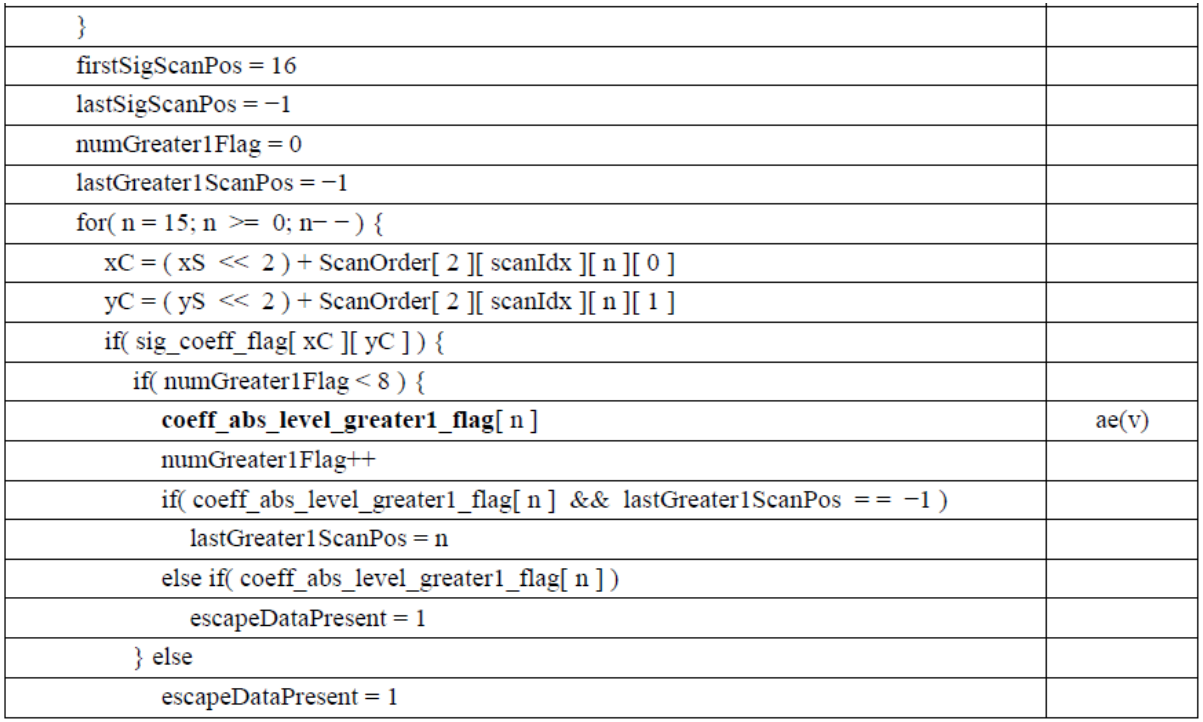
U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
	

```
        lastScanPos = 16
        lastSubBlock− −
      }
      lastScanPos− −
      xS = ScanOrder[ log2TrafoSize − 2 ][ scanIdx ][ lastSubBlock ][ 0 ]
      yS = ScanOrder[ log2TrafoSize − 2 ][ scanIdx ][ lastSubBlock ][ 1 ]
      xC = ( xS  <<  2 ) + ScanOrder[ 2 ][ scanIdx ][ lastScanPos ][ 0 ]
      yC = ( yS  <<  2 ) + ScanOrder[ 2 ][ scanIdx ][ lastScanPos ][ 1 ]
    } while( ( xC  !=  LastSignificantCoeffX ) || ( yC  !=  LastSignificantCoeffY ) )
    for( i = lastSubBlock; i  >=  0; i− − ) {
      xS = ScanOrder[ log2TrafoSize − 2 ][ scanIdx ][ i ][ 0 ]
      yS = ScanOrder[ log2TrafoSize − 2 ][ scanIdx ][ i ][ 1 ]
      escapeDataPresent = 0
      inferSbDcSigCoeffFlag = 0
      if( ( i < lastSubBlock )  &&  ( i > 0 ) ) {
        coded_sub_block_flag[ xS ][ yS ]                                          ae(v)
        inferSbDcSigCoeffFlag = 1
      }
      for( n = ( i  = =  lastSubBlock ) ? lastScanPos − 1 : 15; n  >=  0; n− − ) {
        xC = ( xS  <<  2 ) + ScanOrder[ 2 ][ scanIdx ][ n ][ 0 ]
        yC = ( yS  <<  2 ) + ScanOrder[ 2 ][ scanIdx ][ n ][ 1 ]
        if( coded_sub_block_flag[ xS ][ yS ]  &&  ( n > 0  ||  !inferSbDcSigCoeffFlag ) ) {
          sig_coeff_flag[ xC ][ yC ]                                              ae(v)
          if( sig_coeff_flag[ xC ][ yC ] )
            inferSbDcSigCoeffFlag = 0
        }
```

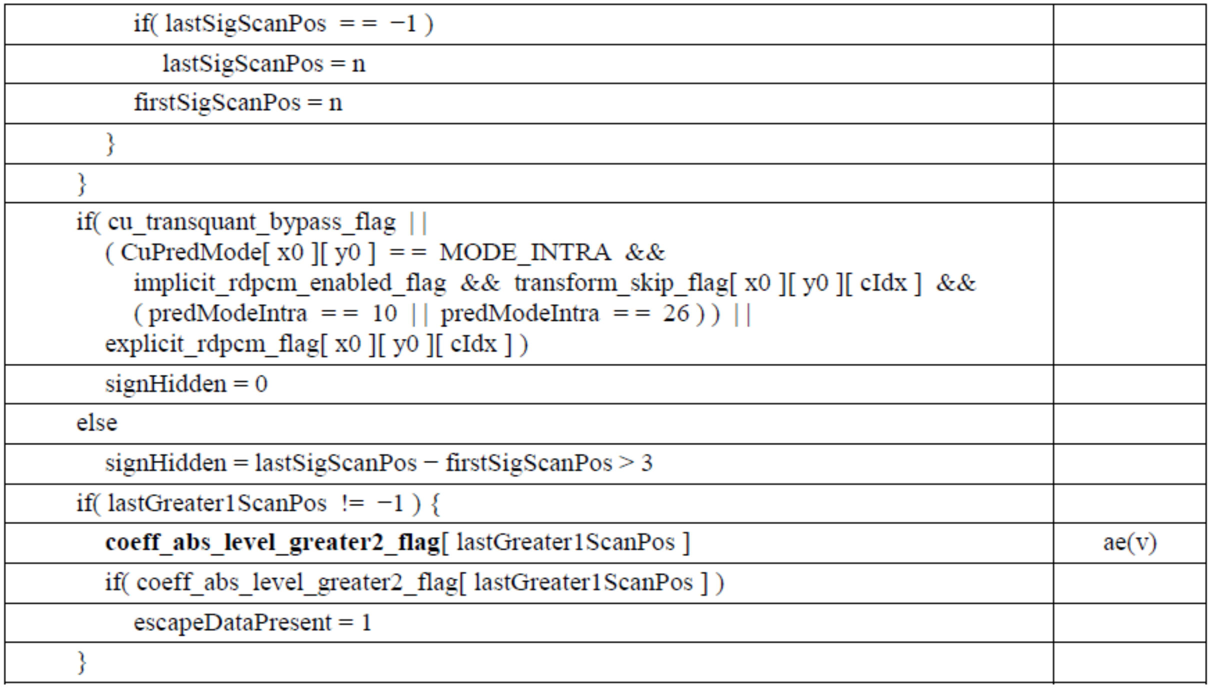
EXHIBIT 20

UNITED STATES PATENT NO. 9,036,701

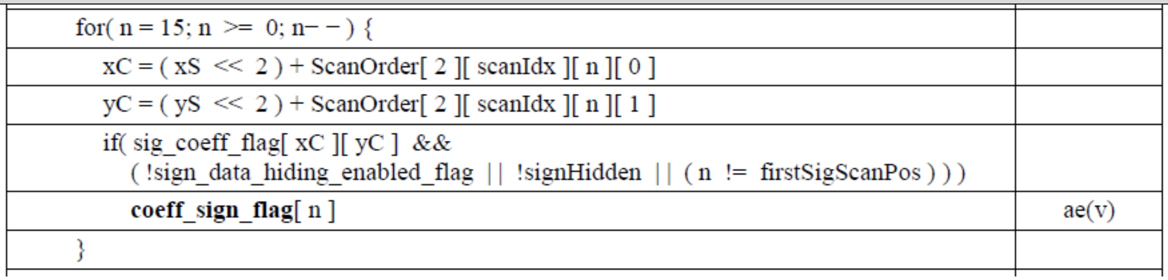
CLAIM CHART FOR INFRINGEMENT OF CLAIMS 1, 8, AND 15 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
	

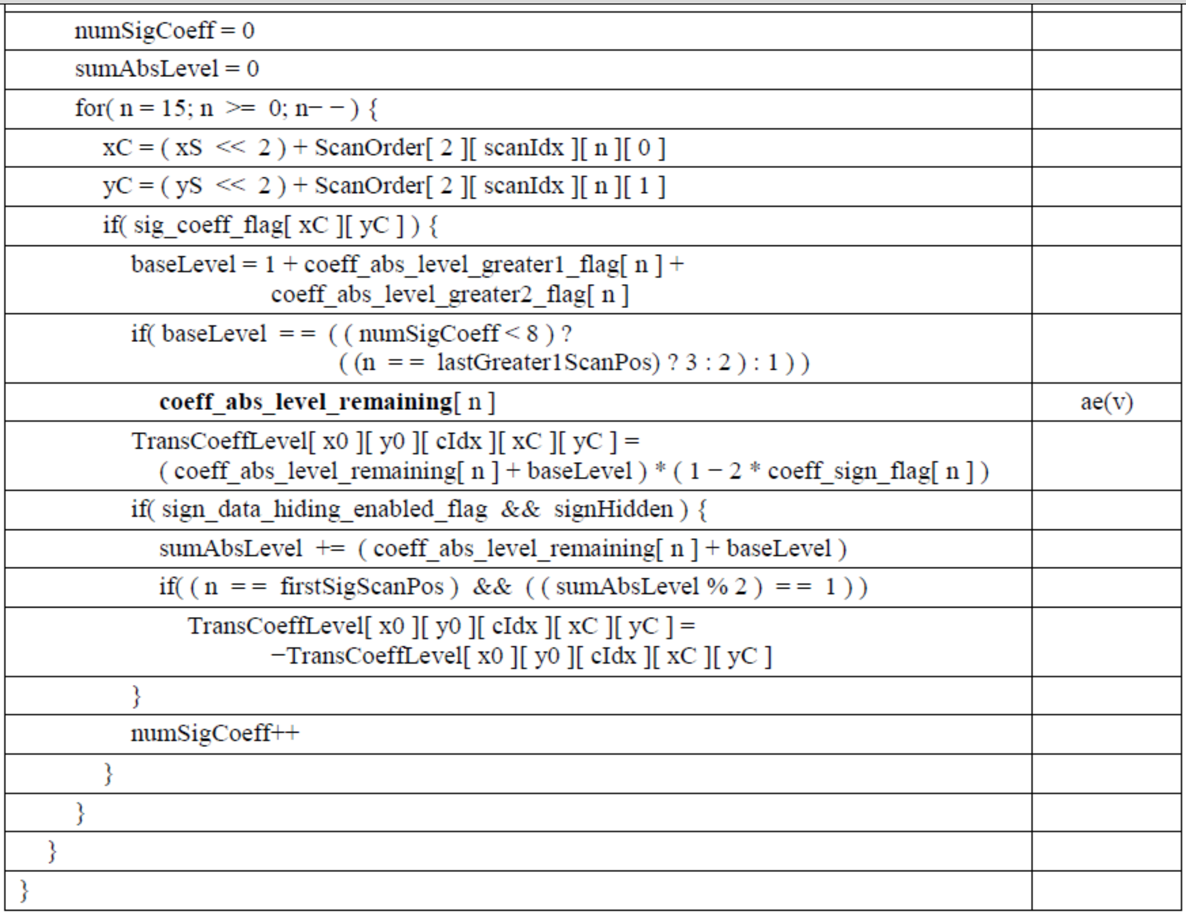
```
    }
    firstSigScanPos = 16
    lastSigScanPos = −1
    numGreater1Flag = 0
    lastGreater1ScanPos = −1
    for( n = 15; n  >=  0; n− − ) {
      xC = ( xS  <<  2 ) + ScanOrder[ 2 ][ scanIdx ][ n ][ 0 ]
      yC = ( yS  <<  2 ) + ScanOrder[ 2 ][ scanIdx ][ n ][ 1 ]
      if( sig_coeff_flag[ xC ][ yC ] ) {
        if( numGreater1Flag < 8 ) {
          coeff_abs_level_greater1_flag[ n ]                              ae(v)
          numGreater1Flag++
          if( coeff_abs_level_greater1_flag[ n ]  &&  lastGreater1ScanPos  = =  −1 )
            lastGreater1ScanPos = n
          else if( coeff_abs_level_greater1_flag[ n ] )
            escapeDataPresent = 1
        } else
          escapeDataPresent = 1
```

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
	

if(lastSigScanPos = = −1)	
lastSigScanPos = n	
firstSigScanPos = n	
}	
}	
if(cu_transquant_bypass_flag || (CuPredMode[x0][y0] = = MODE_INTRA && implicit_rdpcm_enabled_flag && transform_skip_flag[x0][y0][cIdx] && (predModeIntra = = 10 || predModeIntra = = 26)) || explicit_rdpcm_flag[x0][y0][cIdx])	
signHidden = 0	
else	
signHidden = lastSigScanPos − firstSigScanPos > 3	
if(lastGreater1ScanPos != −1) {	
coeff_abs_level_greater2_flag[lastGreater1ScanPos]	ae(v)
if(coeff_abs_level_greater2_flag[lastGreater1ScanPos])	
escapeDataPresent = 1	
}	

<table>
<tr><th>U.S. PATENT NO. 9,036,701</th><th>HISENSE ACCUSED PRODUCTS</th></tr>
<tr><td></td><td>
<table>
<tr><td>for(n = 15; n >= 0; n− −) {</td><td></td></tr>
<tr><td>xC = (xS << 2) + ScanOrder[2][scanIdx][n][0]</td><td></td></tr>
<tr><td>yC = (yS << 2) + ScanOrder[2][scanIdx][n][1]</td><td></td></tr>
<tr><td>if(sig_coeff_flag[xC][yC] &&
(!sign_data_hiding_enabled_flag || !signHidden || (n != firstSigScanPos)))</td><td></td></tr>
<tr><td>**coeff_sign_flag**[n]</td><td>ae(v)</td></tr>
<tr><td>}</td><td></td></tr>
</table></td></tr>
</table>

U.S. PATENT NO. 9,036,701	**HISENSE ACCUSED PRODUCTS**

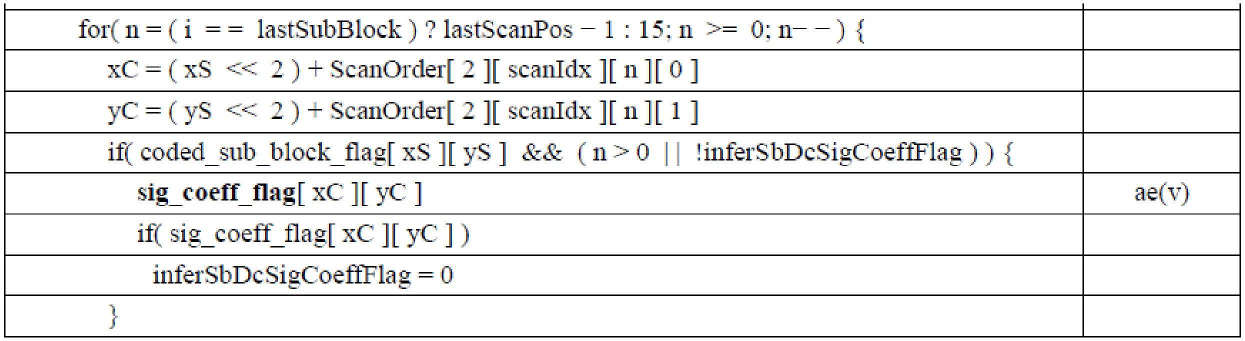


numSigCoeff = 0	
sumAbsLevel = 0	
for(n = 15; n >= 0; n− −) {	
xC = (xS << 2) + ScanOrder[2][scanIdx][n][0]	
yC = (yS << 2) + ScanOrder[2][scanIdx][n][1]	
if(sig_coeff_flag[xC][yC]) {	
baseLevel = 1 + coeff_abs_level_greater1_flag[n] + coeff_abs_level_greater2_flag[n]	
if(baseLevel = = ((numSigCoeff < 8) ? ((n = = lastGreater1ScanPos) ? 3 : 2) : 1))	
coeff_abs_level_remaining[n]	ae(v)
TransCoeffLevel[x0][y0][cIdx][xC][yC] = (coeff_abs_level_remaining[n] + baseLevel) * (1 − 2 * coeff_sign_flag[n])	
if(sign_data_hiding_enabled_flag && signHidden) {	
sumAbsLevel += (coeff_abs_level_remaining[n] + baseLevel)	
if((n = = firstSigScanPos) && ((sumAbsLevel % 2) = = 1))	
TransCoeffLevel[x0][y0][cIdx][xC][yC] = −TransCoeffLevel[x0][y0][cIdx][xC][yC]	
}	
numSigCoeff++	
}	
}	
}	
}	

ITU-T Rec. H.265 (12/2016) High efficiency video coding at pp. 58-60.

As set out in the Residual Coding Syntax in section 7.3.8.11, an H.265 compliant decoder parses *sig_coeff_flags* from the bit stream according to the logic reproduced below.

U.S. Patent No. 9,036,701	Hisense Accused Products



for(n = (i = = lastSubBlock) ? lastScanPos − 1 : 15; n >= 0; n− −) {	
xC = (xS << 2) + ScanOrder[2][scanIdx][n][0]	
yC = (yS << 2) + ScanOrder[2][scanIdx][n][1]	
if(coded_sub_block_flag[xS][yS] && (n > 0 || !inferSbDcSigCoeffFlag)) {	
sig_coeff_flag[xC][yC]	ae(v)
if(sig_coeff_flag[xC][yC])	
inferSbDcSigCoeffFlag = 0	
}	

ITU-T Rec. H.265 (12/2016) High efficiency video coding at p. 59.

7.4.9.11 Residual coding semantics

. . .

sig_coeff_flag[xC][yC] specifies for the transform coefficient location (xC, yC) within the current transform block whether the corresponding transform coefficient level at the location (xC, yC) is non-zero as follows:

– If sig_coeff_flag[xC][yC] is equal to 0, the transform coefficient level at the location (xC, yC) is set equal to 0.

– Otherwise (sig_coeff_flag[xC][yC] is equal to 1), the transform coefficient level at the location (xC, yC) has a non-zero value.

When sig_coeff_flag[xC][yC] is not present, it is inferred as follows:

– If (xC, yC) is the last significant location (LastSignificantCoeffX, LastSignificantCoeffY) in scan order or all of the following conditions are true, sig_coeff_flag[xC][yC] is inferred to be equal to 1:

  – (xC & 3, yC & 3) is equal to (0, 0).

  – inferSbDcSigCoeffFlag is equal to 1.

U.S. PATENT NO. 9,036,701	**HISENSE ACCUSED PRODUCTS**
	– coded_sub_block_flag[xS][yS] is equal to 1. – Otherwise, sig_coeff_flag[xC][yC] is inferred to be equal to 0. . . . ITU-T Rec. H.265 (12/2016) High efficiency video coding at p. 111. **9.3.4.2 Derivation process for ctxTable, ctxIdx and bypassFlag** **9.3.4.2.1 General** Input to this process is the position of the current bin within the bin string, binIdx. Outputs of this process are ctxTable, ctxIdx and bypassFlag. The values of ctxTable, ctxIdx and bypassFlag are derived as follows based on the entries for binIdx of the corresponding syntax element in Table 9-48: – If the entry in Table 9-48 is not equal to "bypass", "terminate" or "na", the values of binIdx are decoded by invoking the DecodeDecision process as specified in clause 9.3.4.3.2 and the following applies: – ctxTable is specified in Table 9-4. – The variable ctxInc is specified by the corresponding entry in Table 9-48 and when more than one value is listed in Table 9-48 for a binIdx, the assignment process for ctxInc for that binIdx is further specified in the clauses given in parenthesis. – The variable ctxIdxOffset is specified by the lowest value of ctxIdx in Table 9-4 depending on the current value of initType. – ctxIdx is set equal to the sum of ctxInc and ctxIdxOffset. – bypassFlag is set equal to 0. – Otherwise, if the entry in Table 9-48 is equal to "bypass", the values of binIdx are decoded by invoking the DecodeBypass process as specified in clause 9.3.4.3.4 and the following applies:

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
	– ctxTable is set equal to 0. – ctxIdx is set equal to 0. – bypassFlag is set equal to 1. ITU-T Rec. H.265 (12/2016) High efficiency video coding at p. 225.

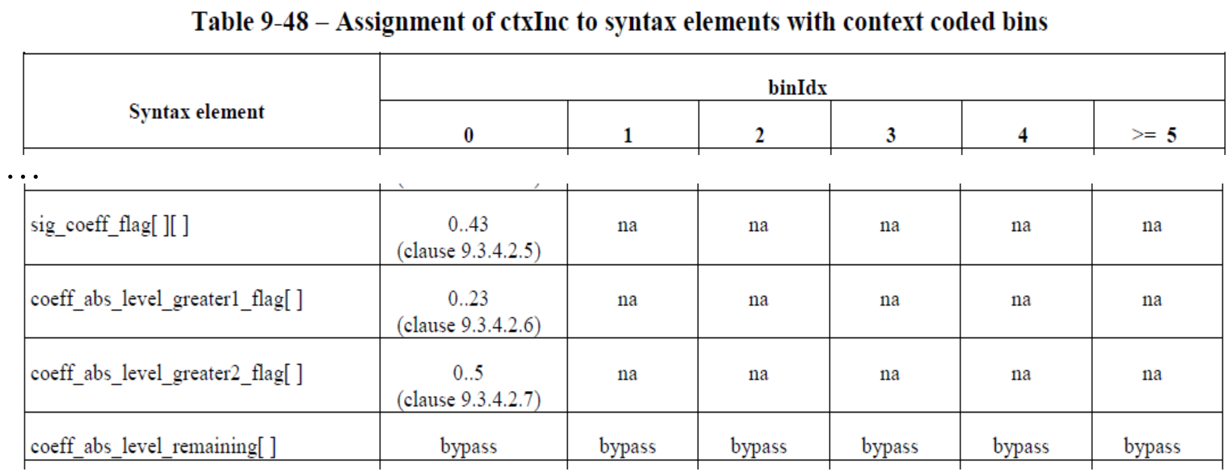


Table 9-48 – Assignment of ctxInc to syntax elements with context coded bins

Syntax element	binIdx					
	0	1	2	3	4	>= 5
. . .						
sig_coeff_flag[ ][ ]	0..43 (clause 9.3.4.2.5)	na	na	na	na	na
coeff_abs_level_greater1_flag[ ]	0..23 (clause 9.3.4.2.6)	na	na	na	na	na
coeff_abs_level_greater2_flag[ ]	0..5 (clause 9.3.4.2.7)	na	na	na	na	na
coeff_abs_level_remaining[ ]	bypass	bypass	bypass	bypass	bypass	bypass

ITU-T Rec. H.265 (12/2016) High efficiency video coding at pp. 225-227.

5. Claim 7

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
7. Pre A method according to claim 1	Each of the Accused Products, such as the Hisense 43A7N, performs a method according to claim 1. The evidence cited for Claim 1 applies to this claim limitation.
[a] further comprising performing binarization of the symbols that correspond to the first category of syntax elements prior to entropy coding the symbols that correspond to the first category of syntax elements.	Each of the Accused Products, such as the Hisense 43A7N, performs a method according to claim 1 further comprising performing binarization of the symbols that correspond to the first category of syntax elements prior to entropy coding the symbols that correspond to the first category of syntax elements. The evidence cited for Claim 1 applies to this claim limitation. For example, and without limitation, the H.265 Standard specifies the following regarding the decoding process. Each of the Accused Products performs a method according to claim 1 further comprising performing binarization of the symbols that correspond to the first category of syntax elements prior to entropy coding the symbols that correspond to the first category of syntax elements. The following specifications provide further evidence of how each of the Accused Products operates: **9.3 CABAC parsing for slice segment data** **9.3.1 General** This process is invoked when parsing syntax elements with descriptor ae(v) in clauses 7.3.8.1 through 7.3.8.12. Inputs to this process are a request for a value of a syntax element and value of prior parsed syntax elements. Outputs of this process is the value of the syntax element . . . For each requested value of a syntax element a binarization is derived as specified in clause 9.3.3.

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
	The binarization for the syntax element and the sequence of parsed bins determines the decoding process flow as described in clause 9.3.4. ITU-T Rec. H.265 (12/2016) High efficiency video coding at p. 201. **9.3.3 Binarization process** **9.3.3.1 General** Inputs to this process is a request for syntax element. Output of this process is the binarization of the syntax element. Table 9-43 specifies the type of binarization process associated with each syntax element and corresponding inputs. The specification of the truncated Rice (TR) binarization process, the k-th order Exp-Golomb (EGk) binarization process, limited k-th order Exp-Golomb (EGk) binarization process, the fixed-length (FL) binarization process, and the truncated binary binarization process are given in clauses 9.3.3.2 through 9.3.3.6, respectively. Other binarizations are specified in clauses 9.3.3.7 through 9.3.3.14. . . .

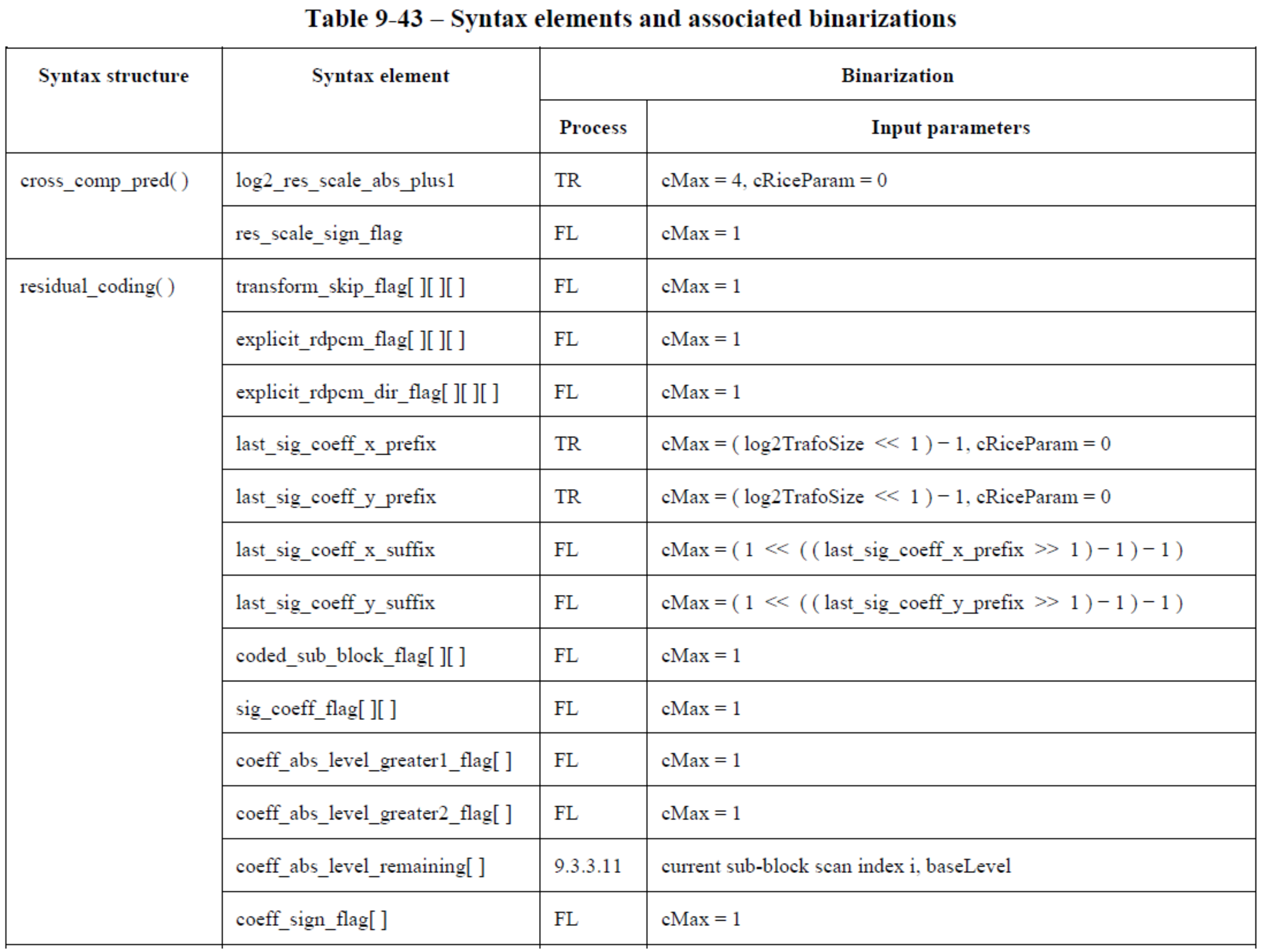
<table>
<tr><th>U.S. PATENT NO. 9,036,701</th><th>HISENSE ACCUSED PRODUCTS</th></tr>
<tr><td></td><td>


Table 9-43 – Syntax elements and associated binarizations

<table>
<tr><th rowspan="2">Syntax structure</th><th rowspan="2">Syntax element</th><th colspan="2">Binarization</th></tr>
<tr><th>Process</th><th>Input parameters</th></tr>
<tr><td>cross_comp_pred( )</td><td>log2_res_scale_abs_plus1</td><td>TR</td><td>cMax = 4, cRiceParam = 0</td></tr>
<tr><td></td><td>res_scale_sign_flag</td><td>FL</td><td>cMax = 1</td></tr>
<tr><td>residual_coding( )</td><td>transform_skip_flag[ ][ ][ ]</td><td>FL</td><td>cMax = 1</td></tr>
<tr><td></td><td>explicit_rdpcm_flag[ ][ ][ ]</td><td>FL</td><td>cMax = 1</td></tr>
<tr><td></td><td>explicit_rdpcm_dir_flag[ ][ ][ ]</td><td>FL</td><td>cMax = 1</td></tr>
<tr><td></td><td>last_sig_coeff_x_prefix</td><td>TR</td><td>cMax = (log2TrafoSize << 1) − 1, cRiceParam = 0</td></tr>
<tr><td></td><td>last_sig_coeff_y_prefix</td><td>TR</td><td>cMax = (log2TrafoSize << 1) − 1, cRiceParam = 0</td></tr>
<tr><td></td><td>last_sig_coeff_x_suffix</td><td>FL</td><td>cMax = (1 << ((last_sig_coeff_x_prefix >> 1) − 1) − 1)</td></tr>
<tr><td></td><td>last_sig_coeff_y_suffix</td><td>FL</td><td>cMax = (1 << ((last_sig_coeff_y_prefix >> 1) − 1) − 1)</td></tr>
<tr><td></td><td>coded_sub_block_flag[ ][ ]</td><td>FL</td><td>cMax = 1</td></tr>
<tr><td></td><td>sig_coeff_flag[ ][ ]</td><td>FL</td><td>cMax = 1</td></tr>
<tr><td></td><td>coeff_abs_level_greater1_flag[ ]</td><td>FL</td><td>cMax = 1</td></tr>
<tr><td></td><td>coeff_abs_level_greater2_flag[ ]</td><td>FL</td><td>cMax = 1</td></tr>
<tr><td></td><td>coeff_abs_level_remaining[ ]</td><td>9.3.3.11</td><td>current sub-block scan index i, baseLevel</td></tr>
<tr><td></td><td>coeff_sign_flag[ ]</td><td>FL</td><td>cMax = 1</td></tr>
</table>

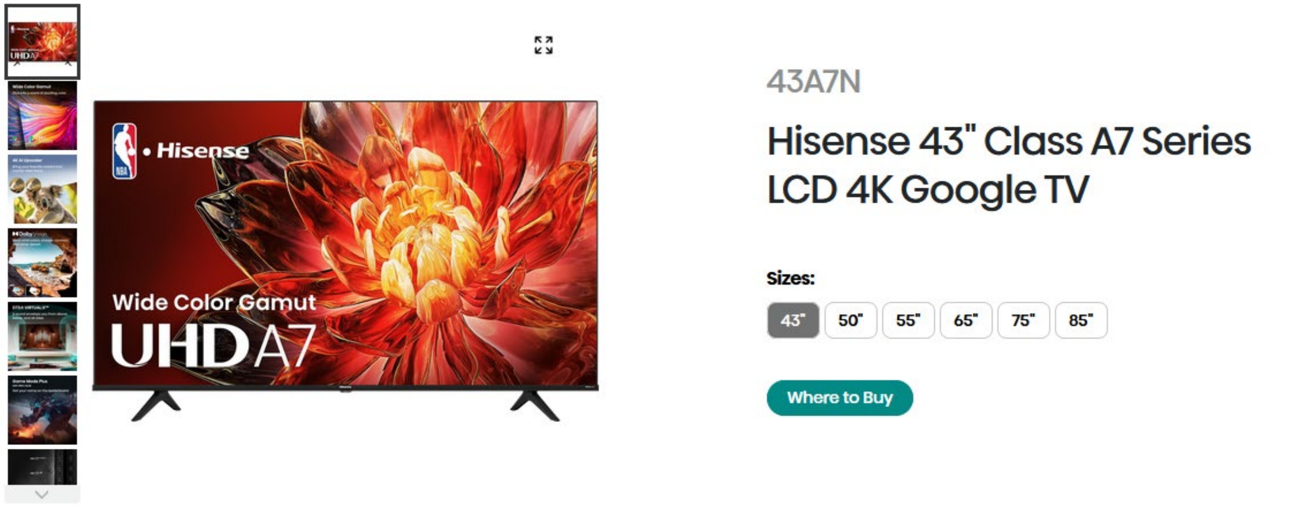
ITU-T Rec. H.265 (12/2016) High efficiency video coding at pp. 216-218.
</td></tr>
</table>

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
	9.3.4 Decoding process flow **9.3.4.1 General** Inputs to this process are all bin strings of the binarization of the requested syntax element as specified in clause 9.3.3. Output of this process is the value of the syntax element. ITU-T Rec. H.265 (12/2016) High efficiency video coding at p. 225.

6. Claim 8

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
8. [Pre] An apparatus comprising at least one processor and at least one memory including computer program code, the at least one memory and the computer program code configured to, with the processor, cause the apparatus to at least:	Each of the Accused Products, such as the Hisense 43A7N, comprises an apparatus comprising at least one processor and at least one memory including computer program code, the at least one memory and the computer program code configured to, with the processor, cause the apparatus to satisfy the limitations below. For example, and without limitation, the Hisense 43A7N uses hardware-accelerated video decoding and includes Mediatek MT9602 Processor with onboard memory.

EXHIBIT 20
UNITED STATES PATENT NO. 9,036,701
CLAIM CHART FOR INFRINGEMENT OF CLAIMS 1, 8, AND 15 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
	 43A7N Hisense 43" Class A7 Series LCD 4K Google TV https://www.hisense-usa.com/televisions/hisense-43-class-a7-series-4k-wide-color-gamut-google-tv-43a7n (last accessed March 29, 2025).

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
	 **Display**: 42.5 in, VA, Direct LED, 3840 x 2160 pixels **Viewing angles (H/V)**: 178 ° / 178 ° **Brightness**: 275 cd/m² **Static contrast**: 4000 : 1 **Refresh rate**: 50 Hz / 60 Hz **Frame interpolation**: 120 MR (Motion Rate) **TV tuner**: Analog (NTSC/PAL/SECAM), ATSC, Clear QAM **SoC**: MediaTek MT9602 **CPU**: ARM Cortex-A53, 1500 MHz, **Cores**: 4 **Dimensions**: 963 x 560 x 74 mm **Weight**: 6.8 kg https://www.displayspecifications.com/en/model/234a3f3b

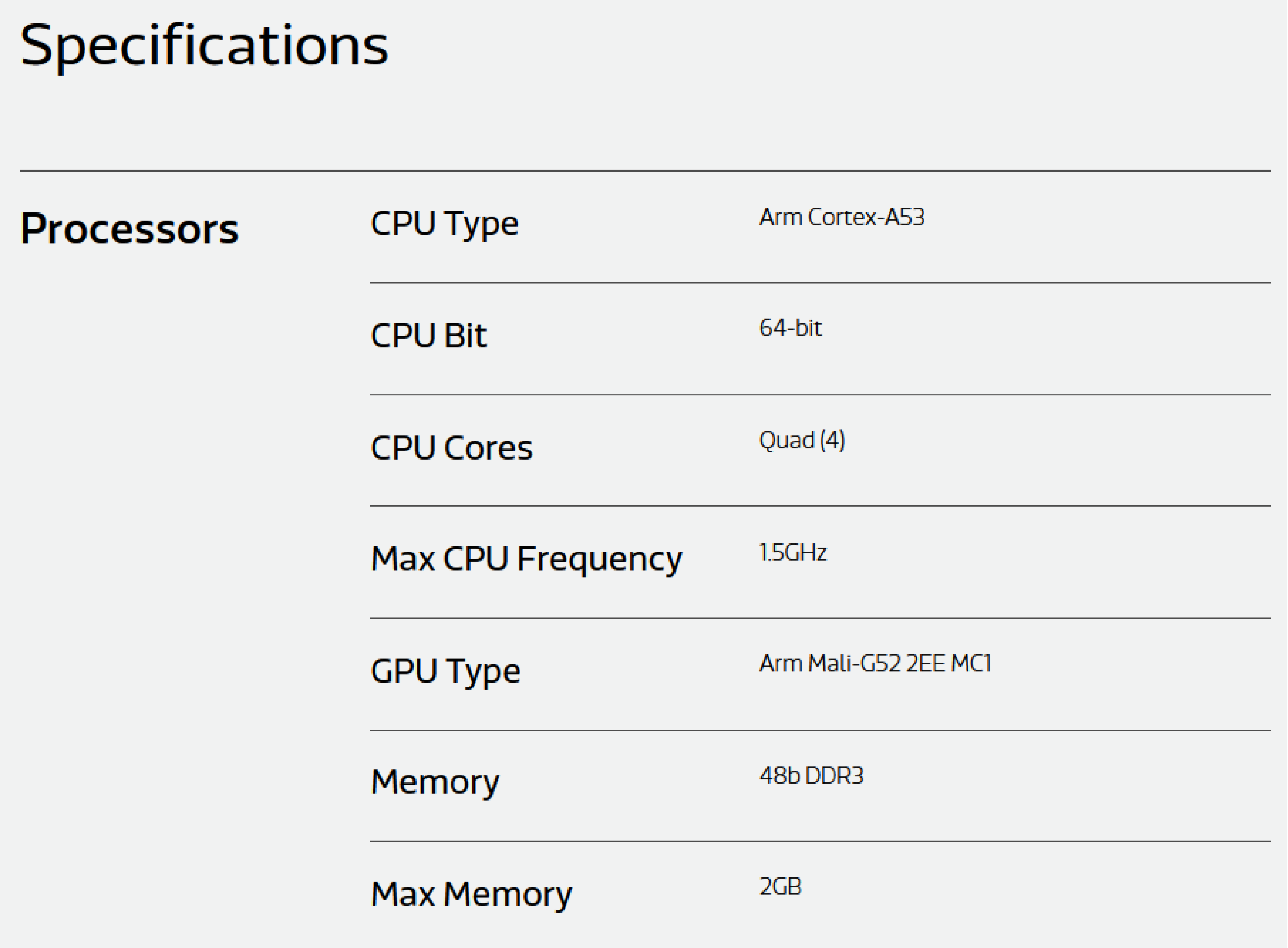
U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
	 Specifications **Processors** CPU Type: Arm Cortex-A53 CPU Bit: 64-bit CPU Cores: Quad (4) Max CPU Frequency: 1.5GHz GPU Type: Arm Mali-G52 2EE MC1 Memory: 48b DDR3 Max Memory: 2GB Source: https://www.mediatek.com/products/pentonic/mt9602. (Last accessed March 29, 2025).

EXHIBIT 20

UNITED STATES PATENT NO. 9,036,701

CLAIM CHART FOR INFRINGEMENT OF CLAIMS 1, 8, AND 15 BY HISENSE ACCUSED PRODUCTS

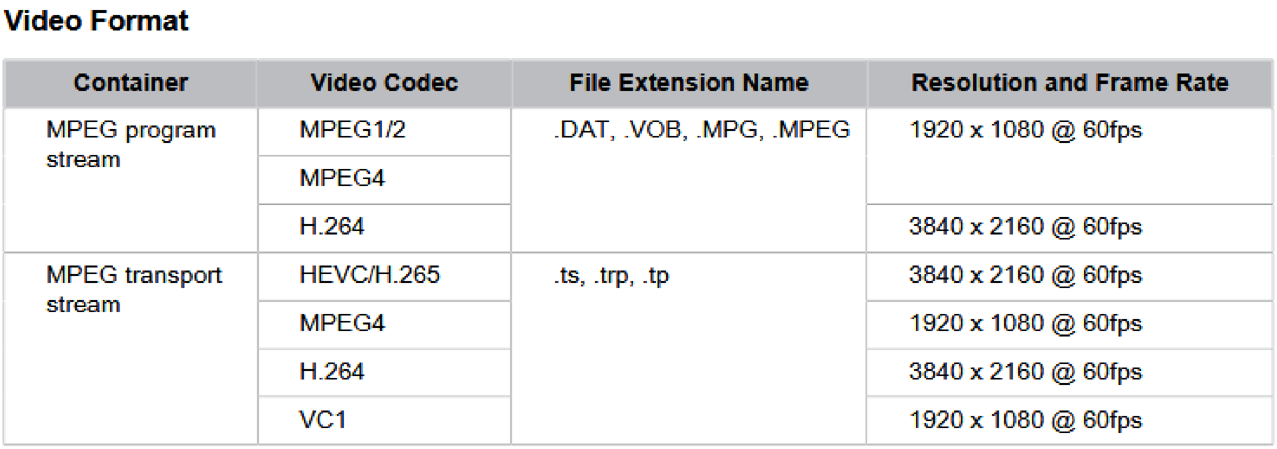
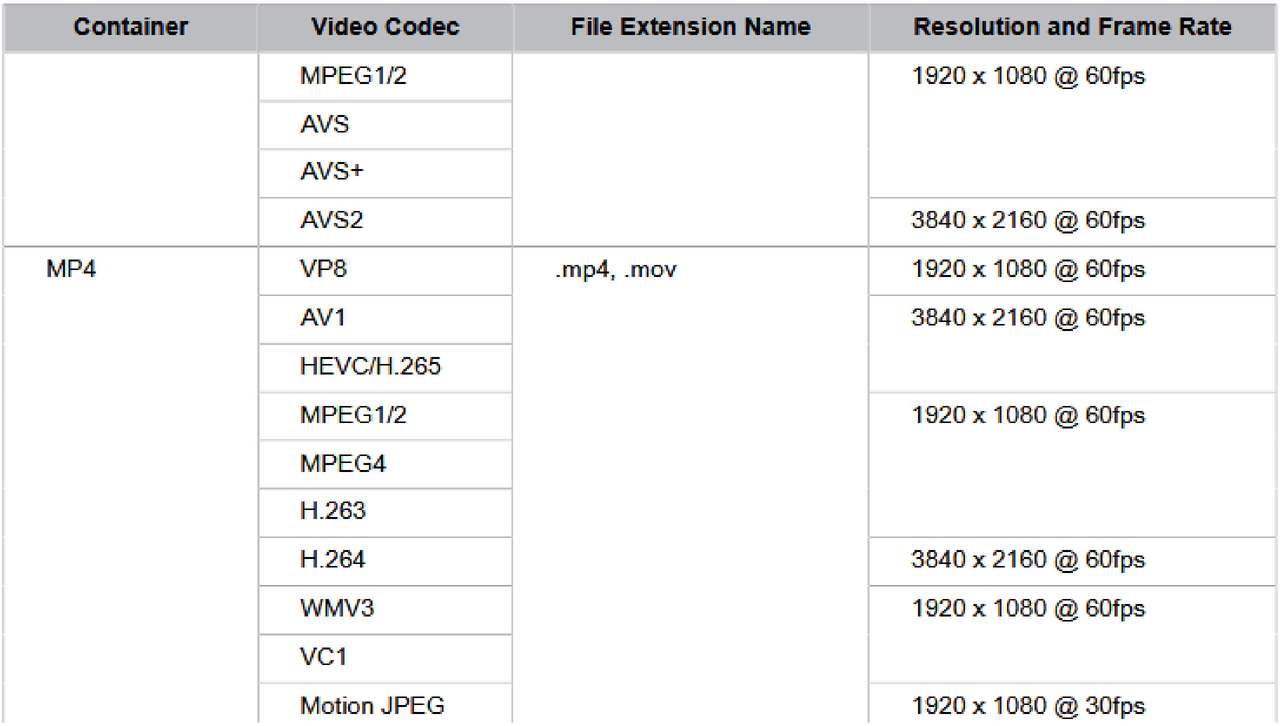
<table>
<tr><th>U.S. PATENT NO. 9,036,701</th><th>HISENSE ACCUSED PRODUCTS</th></tr>
<tr><td></td><td>

Video Format
<table>
<tr><th>Container</th><th>Video Codec</th><th>File Extension Name</th><th>Resolution and Frame Rate</th></tr>
<tr><td rowspan="3">MPEG program stream</td><td>MPEG1/2</td><td rowspan="3">.DAT, .VOB, .MPG, .MPEG</td><td rowspan="2">1920 x 1080 @ 60fps</td></tr>
<tr><td>MPEG4</td></tr>
<tr><td>H.264</td><td>3840 x 2160 @ 60fps</td></tr>
<tr><td rowspan="4">MPEG transport stream</td><td>HEVC/H.265</td><td rowspan="4">.ts, .trp, .tp</td><td>3840 x 2160 @ 60fps</td></tr>
<tr><td>MPEG4</td><td>1920 x 1080 @ 60fps</td></tr>
<tr><td>H.264</td><td>3840 x 2160 @ 60fps</td></tr>
<tr><td>VC1</td><td>1920 x 1080 @ 60fps</td></tr>
</table>
</td></tr>
</table>

EXHIBIT 20
UNITED STATES PATENT NO. 9,036,701
CLAIM CHART FOR INFRINGEMENT OF CLAIMS 1, 8, AND 15 BY HISENSE ACCUSED PRODUCTS

<table>
<tr><th>U.S. PATENT NO. 9,036,701</th><th>HISENSE ACCUSED PRODUCTS</th></tr>
<tr><td></td><td>

<table>
<tr><th>Container</th><th>Video Codec</th><th>File Extension Name</th><th>Resolution and Frame Rate</th></tr>
<tr><td></td><td>MPEG1/2</td><td></td><td>1920 x 1080 @ 60fps</td></tr>
<tr><td></td><td>AVS</td><td></td><td></td></tr>
<tr><td></td><td>AVS+</td><td></td><td></td></tr>
<tr><td></td><td>AVS2</td><td></td><td>3840 x 2160 @ 60fps</td></tr>
<tr><td>MP4</td><td>VP8</td><td>.mp4, .mov</td><td>1920 x 1080 @ 60fps</td></tr>
<tr><td></td><td>AV1</td><td></td><td>3840 x 2160 @ 60fps</td></tr>
<tr><td></td><td>HEVC/H.265</td><td></td><td></td></tr>
<tr><td></td><td>MPEG1/2</td><td></td><td>1920 x 1080 @ 60fps</td></tr>
<tr><td></td><td>MPEG4</td><td></td><td></td></tr>
<tr><td></td><td>H.263</td><td></td><td></td></tr>
<tr><td></td><td>H.264</td><td></td><td>3840 x 2160 @ 60fps</td></tr>
<tr><td></td><td>WMV3</td><td></td><td>1920 x 1080 @ 60fps</td></tr>
<tr><td></td><td>VC1</td><td></td><td></td></tr>
<tr><td></td><td>Motion JPEG</td><td></td><td>1920 x 1080 @ 30fps</td></tr>
</table>
Source: Hisense 43A7N User Manual, at 45-46. Downloaded from https://www.hisense-usa.com/televisions/hisense-43-class-a7-series-4k-wide-color-gamut-google-tv-43a7n (last accessed March 29, 2025).

The evidence cited for the claim limitation of claim 1[A] applies to this claim limitation.
</td></tr>
<tr><td>[a] determine a frequency occurrence threshold based on an expected frequency of</td><td>Each of the Accused Products, such as the Hisense 43A7N, comprises an apparatus comprising at least one processor and at least one memory including computer program code, the at least one memory and the computer program code configured to, with the processor, determine a frequency occurrence threshold based on an expected frequency of occurrence of syntax elements in a bit stream.</td></tr>
</table>

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
occurrence of syntax elements in a bit stream;	The evidence cited for the claim limitations of claim 1[a] applies to this claim limitation.
[b] categorize a plurality of syntax elements of video content into first and second categories based on the frequency of occurrence threshold, wherein syntax elements which occur greater than the frequency of occurrence threshold are categorized into the second category and syntax elements which occur less than the frequency of occurrence are categorized into the first category;	Each of the Accused Products, such as the Hisense 43A7N, comprises an apparatus comprising at least one processor and at least one memory including computer program code, the at least one memory and the computer program code configured to, with the processor, categorize a plurality of syntax elements of video content into first and second categories based on the frequency of occurrence threshold, wherein syntax elements which occur greater than the frequency of occurrence threshold are categorized into the second category and syntax elements which occur less than the frequency of occurrence are categorized into the first category. The evidence cited for the claim limitations of claim 1[b] applies to this claim limitation.
[c] entropy code symbols that correspond to the first category of syntax elements and that have been subjected to a context update; and	Each of the Accused Products, such as the Hisense 43A7N, comprises an apparatus comprising at least one processor and at least one memory including computer program code, the at least one memory and the computer program code configured to, with the processor, entropy code symbols that correspond to the first category of syntax elements and that have been subjected to a context update. The evidence cited for the claim limitations of claim 1[c] applies to this claim limitation.
[d] entropy code symbols that correspond to the second category of syntax elements	Each of the Accused Products, such as the Hisense 43A7N, comprises an apparatus comprising at least one processor and at least one memory including computer program code, the at least one memory and the computer program code configured to, with the processor, entropy code symbols that correspond to the second category of syntax elements and that have bypassed context updating.

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
and that have bypassed context updating.	The evidence cited for the claim limitations of claim 1[d] applies to this claim limitation.

7. Claim 10

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
10. Pre An apparatus according to claim 8	Each of the Accused Products, such as the Hisense 43A7N, comprise an apparatus according to claim 8. The evidence cited for Claim 8 applies to this claim limitation. For example, and without limitation, the H.265 Standard specifies the following regarding the decoding process. Each of the Accused Products performs a method comprising entropy coding symbols that correspond to the second category of syntax elements and that have bypassed context updating.
[a] wherein the at least one memory and the computer program code are further configured to, with the processor, cause the apparatus to perform binarization of the symbols that correspond to the second category of syntax elements prior to bypassing the context updating.	Each of the Accused Products, such as the Hisense 43A7N, comprise an apparatus according to claim 8 wherein the at least one memory and the computer program code are further configured to, with the processor, cause the apparatus to perform binarization of the symbols that correspond to the second category of syntax elements prior to bypassing the context updating. The evidence cited for Claims 3 and 8 applies to this claim limitation.

8. Claim 11

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
11. Pre An apparatus according to claim 10	Each of the Accused Products, such as the Hisense 43A7N, comprise an apparatus according to claim 10. The evidence cited for Claim 10 applies to this claim limitation.
wherein the at least one memory and the computer program code are configured to, with the processor, cause the apparatus to perform binarization by performing binarization so as to target a predetermined probability.	Each of the Accused Products, such as the Hisense 43A7N, comprise an apparatus according to claim 10 wherein the at least one memory and the computer program code are configured to, with the processor, cause the apparatus to perform binarization by performing binarization so as to target a predetermined probability. The evidence cited for Claims 4 and 10 applies to this claim limitation.

9. Claim 13

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
13. Pre An apparatus according to claim 8	Each of the Accused Products, such as the Hisense 43A7N, comprise an apparatus according to claim 8. The evidence cited for Claim 8 applies to this claim limitation.

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
[a] wherein the at least one memory and the computer program code are configured to, with the processor, cause the apparatus to categorize the plurality of syntax elements by categorizing the plurality of syntax elements based on a relationship of the frequency of occurrence of the syntax elements to a predefined threshold.	Each of the Accused Products, such as the Hisense 43A7N, comprise an apparatus according to claim 8 wherein the at least one memory and the computer program code are configured to, with the processor, cause the apparatus to categorize the plurality of syntax elements by categorizing the plurality of syntax elements based on a relationship of the frequency of occurrence of the syntax elements to a predefined threshold. The evidence cited for Claims 6 and 8 applies to this claim limitation.

10. Claim 14

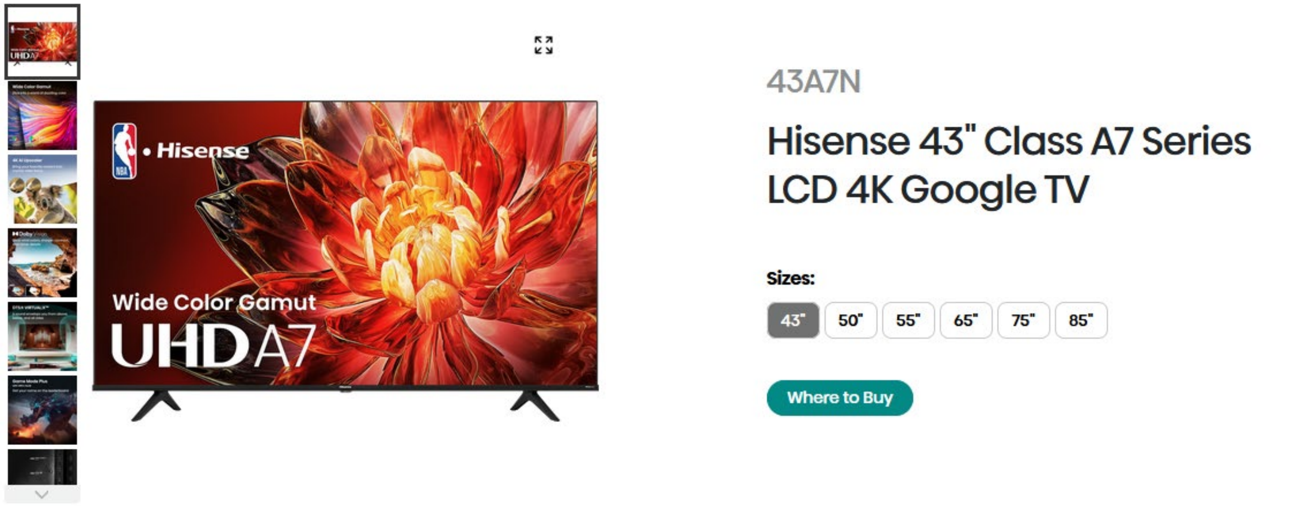
U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
14. Pre An apparatus according to claim 8	Each of the Accused Products, such as the Hisense 43A7N, comprise an apparatus according to claim 8. The evidence cited for Claim 8 applies to this claim limitation.
[a] wherein the at least one memory and the computer program code are further configured to, with the processor, cause the	Each of the Accused Products, such as the Hisense 43A7N, comprise an apparatus according to claim 8wherein the at least one memory and the computer program code are further configured to, with the processor, cause the apparatus to perform binarization of the symbols that correspond to the first category of syntax elements prior to entropy coding the symbols that correspond to the first category of syntax elements.

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
apparatus to perform binarization of the symbols that correspond to the first category of syntax elements prior to entropy coding the symbols that correspond to the first category of syntax elements.	The evidence cited for Claims 7 and 8 applies to this claim limitation.

11. Claim 15

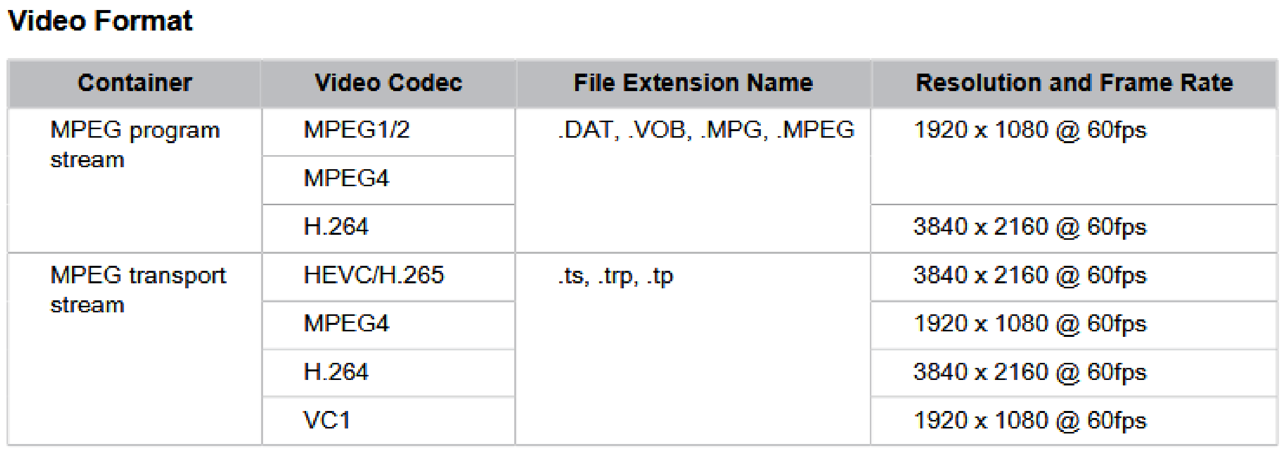
U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
15. [Pre] A computer program product comprising at least one non-transitory computer-readable storage medium having computer-executable program code instructions stored therein, the computer-executable program code instructions comprising program code instructions to:	Each of the Accused Products, such as the Hisense 43A7N, comprises a computer program product comprising at least one non-transitory computer-readable storage medium having computer-executable program code instructions stored therein, the computer-executable program code instructions comprising program code instructions to satisfy the limitations below. For example, and without limitation, the Hisense 43A7N uses hardware-accelerated video decoding and includes Mediatek MT9602 Processor.

EXHIBIT 20
UNITED STATES PATENT NO. 9,036,701
CLAIM CHART FOR INFRINGEMENT OF CLAIMS 1, 8, AND 15 BY HISENSE ACCUSED PRODUCTS

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
	 43A7N Hisense 43" Class A7 Series LCD 4K Google TV Sizes: https://www.hisense-usa.com/televisions/hisense-43-class-a7-series-4k-wide-color-gamut-google-tv-43a7n (last accessed March 29, 2025).

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
	 **Display**: 42.5 in, VA, Direct LED, 3840 x 2160 pixels **Viewing angles (H/V)**: 178 ° / 178 ° **Brightness**: 275 cd/m² **Static contrast**: 4000 : 1 **Refresh rate**: 50 Hz / 60 Hz **Frame interpolation**: 120 MR (Motion Rate) **TV tuner**: Analog (NTSC/PAL/SECAM), ATSC, Clear QAM **SoC**: MediaTek MT9602 **CPU**: ARM Cortex-A53, 1500 MHz, **Cores**: 4 **Dimensions**: 963 x 560 x 74 mm **Weight**: 6.8 kg https://www.displayspecifications.com/en/model/234a3f3b

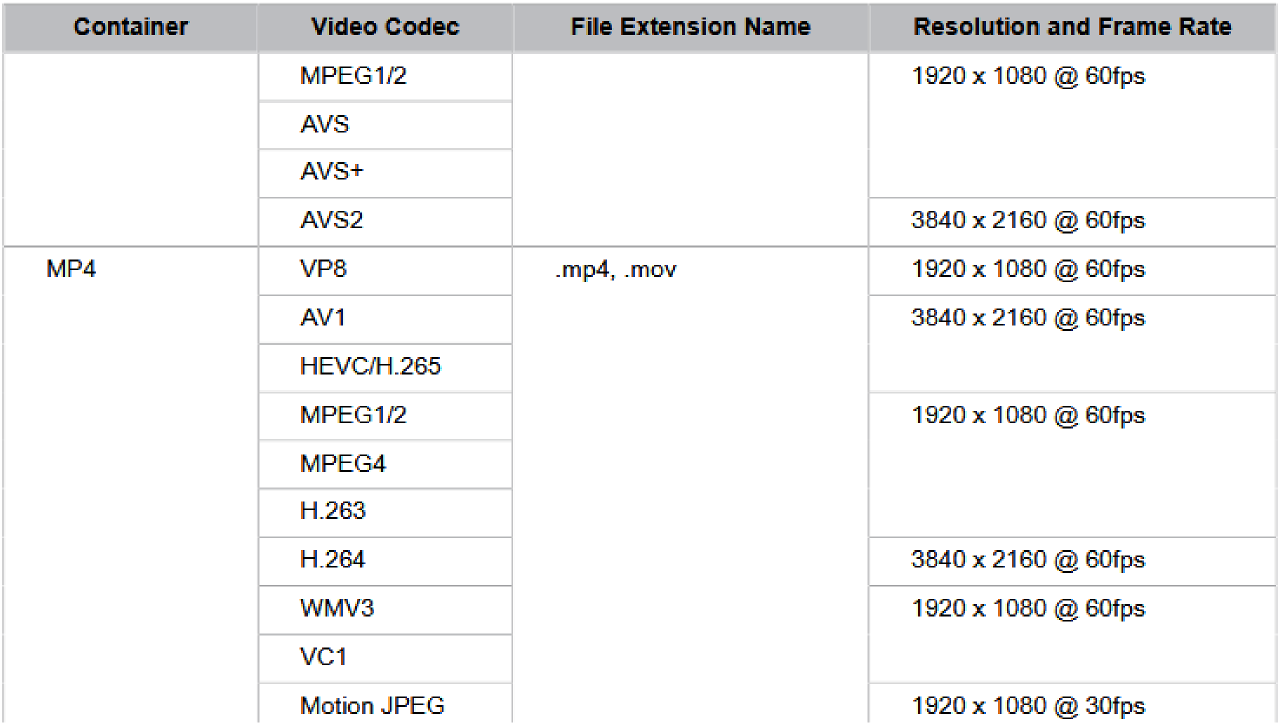
U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS



Video Format

Container	Video Codec	File Extension Name	Resolution and Frame Rate
MPEG program stream	MPEG1/2	.DAT, .VOB, .MPG, .MPEG	1920 x 1080 @ 60fps
	MPEG4		
	H.264		3840 x 2160 @ 60fps
MPEG transport stream	HEVC/H.265	.ts, .trp, .tp	3840 x 2160 @ 60fps
	MPEG4		1920 x 1080 @ 60fps
	H.264		3840 x 2160 @ 60fps
	VC1		1920 x 1080 @ 60fps

EXHIBIT 20
UNITED STATES PATENT NO. 9,036,701
CLAIM CHART FOR INFRINGEMENT OF CLAIMS 1, 8, AND 15 BY HISENSE ACCUSED PRODUCTS

<table>
<tr><th>U.S. PATENT NO. 9,036,701</th><th>HISENSE ACCUSED PRODUCTS</th></tr>
<tr><td></td><td>

<table>
<tr><th>Container</th><th>Video Codec</th><th>File Extension Name</th><th>Resolution and Frame Rate</th></tr>
<tr><td></td><td>MPEG1/2</td><td></td><td>1920 x 1080 @ 60fps</td></tr>
<tr><td></td><td>AVS</td><td></td><td></td></tr>
<tr><td></td><td>AVS+</td><td></td><td></td></tr>
<tr><td></td><td>AVS2</td><td></td><td>3840 x 2160 @ 60fps</td></tr>
<tr><td>MP4</td><td>VP8</td><td>.mp4, .mov</td><td>1920 x 1080 @ 60fps</td></tr>
<tr><td></td><td>AV1</td><td></td><td>3840 x 2160 @ 60fps</td></tr>
<tr><td></td><td>HEVC/H.265</td><td></td><td></td></tr>
<tr><td></td><td>MPEG1/2</td><td></td><td>1920 x 1080 @ 60fps</td></tr>
<tr><td></td><td>MPEG4</td><td></td><td></td></tr>
<tr><td></td><td>H.263</td><td></td><td></td></tr>
<tr><td></td><td>H.264</td><td></td><td>3840 x 2160 @ 60fps</td></tr>
<tr><td></td><td>WMV3</td><td></td><td>1920 x 1080 @ 60fps</td></tr>
<tr><td></td><td>VC1</td><td></td><td></td></tr>
<tr><td></td><td>Motion JPEG</td><td></td><td>1920 x 1080 @ 30fps</td></tr>
</table>
Source: Hisense 43A7N User Manual, at 45-46. Downloaded from https://www.hisense-usa.com/televisions/hisense-43-class-a7-series-4k-wide-color-gamut-google-tv-43a7n (last accessed March 29, 2025).

The evidence cited for the claim limitation of claim 1[A] applies to this claim limitation.
</td></tr>
<tr><td>[a] determine a frequency of occurrence threshold based on an expected frequency of</td><td>Each of the Accused Products, such as the Hisense 43A7N, comprises a computer program product comprising at least one non-transitory computer-readable storage medium having computer-executable program code instructions stored therein, the computer-executable program code instructions comprising</td></tr>
</table>

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
occurrence of syntax elements in a bit stream;	program code instructions to determine a frequency of occurrence threshold based on an expected frequency of occurrence of syntax elements in a bit stream. The evidence cited for the claim limitations of claim 1[a] applies to this claim limitation.
[b] categorize a plurality of syntax elements of video content into first and second categories based on the frequency of occurrence threshold, wherein syntax elements which occur greater than the frequency of occurrence threshold are categorized into the second category and syntax elements which occur less than the frequency of occurrence are categorized into the first category;	Each of the Accused Products, such as the Hisense 43A7N, comprises a computer program product comprising at least one non-transitory computer-readable storage medium having computer-executable program code instructions stored therein, the computer-executable program code instructions comprising program code instructions to categorize a plurality of syntax elements of video content into first and second categories based on the frequency of occurrence threshold, wherein syntax elements which occur greater than the frequency of occurrence threshold are categorized into the second category and syntax elements which occur less than the frequency of occurrence are categorized into the first category. The evidence cited for the claim limitations of claim 1[b] applies to this claim limitation.
[c] entropy code symbols that correspond to the first category of syntax elements and that have been subjected to a context update; and	Each of the Accused Products, such as the Hisense 43A7N, comprises a computer program product comprising at least one non-transitory computer-readable storage medium having computer-executable program code instructions stored therein, the computer-executable program code instructions comprising program code instructions to entropy code symbols that correspond to the first category of syntax elements and that have been subjected to a context update. The evidence cited for the claim limitations of claim 1[c] applies to this claim limitation.

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
[d] entropy code symbols that correspond to the second category of syntax elements and that have bypassed context updating.	Each of the Accused Products, such as the Hisense 43A7N, comprises a computer program product comprising at least one non-transitory computer-readable storage medium having computer-executable program code instructions stored therein, the computer-executable program code instructions comprising program code instructions to entropy code symbols that correspond to the second category of syntax elements and that have bypassed context updating. The evidence cited for the claim limitations of claim 1[d] applies to this claim limitation.

12. Claim 17

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
17. Pre A computer program product according to claim 15	Each of the Accused Products, such as the Hisense 43A7N, comprise a computer program product according to claim 15. The evidence cited for Claim 15 applies to this claim limitation.
[a] wherein the computer-executable program code instructions further comprise program code instructions to perform binarization of the symbols that correspond to the second category of syntax elements prior to bypassing the context updating	Each of the Accused Products, such as the Hisense 43A7N, comprise a computer program product according to claim 15 wherein the computer-executable program code instructions further comprise program code instructions to perform binarization of the symbols that correspond to the second category of syntax elements prior to bypassing the context updating. The evidence cited for Claims 3 and 15 applies to this claim limitation.

13. Claim 18

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
3. A computer program product according to claim 17	Each of the Accused Products, such as the Hisense 43A7N, comprise a computer program product according to claim 17. The evidence cited for Claim 17 applies to this claim limitation.
[a] wherein the comprise program code instructions to perform binarization comprise program code instructions to perform binarization so as to target a predetermined probability.	Each of the Accused Products, such as the Hisense 43A7N, comprise a computer program product according to claim 17 wherein the comprise program code instructions to perform binarization comprise program code instructions to perform binarization so as to target a predetermined probability. The evidence cited for Claims 4 and 17 applies to this claim limitation.

14. Claim 20

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
20. Pre A computer program product according to claim 15	Each of the Accused Products, such as the Hisense 43A7N, comprise a computer program product according to claim 15. The evidence cited for Claim 15 applies to this claim limitation.
[a] comprise the program code instructions to categorize the plurality of	Each of the Accused Products, such as the Hisense 43A7N, comprise a computer program product according to claim 15 comprise the program code instructions to categorize the plurality of syntax

U.S. PATENT NO. 9,036,701	HISENSE ACCUSED PRODUCTS
syntax elements comprise program code instructions to categorize the plurality of syntax elements based on a relationship of the frequency of occurrence of the syntax elements to a predefined threshold.	elements comprise program code instructions to categorize the plurality of syntax elements based on a relationship of the frequency of occurrence of the syntax elements to a predefined threshold. The evidence cited for Claims 6 and 15 applies to this claim limitation.