|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11**  8th Meeting: San José, CA, USA, 1–10 February, 2012 | Document: JCTVC-H0445 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **Non-CE11: Low complexity parallel significance map coding** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Joel Sole, Vadim Seregin, Marta Karczewicz  5775 Morehouse Drive San Diego, CA 92121, USA  Takeshi Tsukuba, Ikai Tomohiro  1-9-2 Nakase, Mihama-ku, Chiba-shi, Chiba 261-8520  JAPAN | Email: | [joels@qualcomm.com](mailto:joels@qualcomm.com)  [ikai.tomohiro@sharp.co.jp](mailto:ikai.tomohiro@sharp.co.jp) |
| *Source:* | Qualcomm, SHARP Corporation | | |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Abstract

In HM5.0, for large transforms, significance map context calculation is dependent on the number of non-zero coefficients in a neighborhood support. Up to 10 conditions have to be checked to establish the right neighborhood. This is due to the border conditions on the block and to allow for 2-bin parallel processing. This contribution reduces the number of conditions to check while allowing for a 3-bin parallel processing. The average BD-rate of the method is 0.1%.

# Introduction

In current test model (HM), a five-point support is used to define a context model to code significance map of the transform coefficients (Fig 1). The significance flags of the neighbours from the support are added. However, the neighbours can be located outside the current block. Also, some neighbours are not considered in some cases to allow for a 2-bin parallel processing. Therefore, up to 10 conditions might be required to add the appropriate neighboring coefficients. For a 32×32 TU, this is up to around 10000 checks.



Fig. 1. Dependency in significance context calculation within support.

# Technical Description

This proposal combines the methods of [1] and [2] to achieve a lower complexity significance map for large transforms and allowing for processing 3-bins in parallel.

JCTVC-H0095 [1] uses a fix context for high-frequency coefficients, i.e., those with (xC,yC) such that

(xC>>2) + (yC>>2) >= 3\*(TU size >> 4)

This is around 67% of the coefficients in a 32×32 TU and includes the sub-blocks at the bottom row and right column.

JCTVC-H0427 [2] uses a modified support (Fig. 2 and Fig. 3) that doesn’t include the coefficient located below the current coefficient in any case. To achieve a 3-bin parallelization, additional checks are required for the coefficient at position (0, 0) (top-left corner of the sub-block) and the coefficient at position (2, 3) as shown by green colour on Fig. 3. This is the same number of additional checks as in HM5.0. Then, the total number of conditions is 5 for the low frequency coefficients, since there is no need to check whether coefficients are out of boundaries because of [1].

Considering current HM5.0 implementation (which might not be optimal in this part), this implies that the number of conditions for the proposed method for a 32×32 TU can be up to 32\*32\*(0.67 + 0.33\*5) = 2375 conditions, as opposed to 10000 of the current method.

This method adds one additional context for the high frequency area for luma and for chroma. In [1], the method was also modified in order to not add this additional context. In this case, high frequencies share context with those that all neighbors are zero (cxtCnt = 0). This variant of the method is also tested and results provided.



Fig. 2. Modified support for the parallel processing.



Fig. 3. Modified support and conditions for the three bins parallel processing.

# Simulation results

The method is implemented on top of HM5.0. Results for both variants are provided using common test conditions and low QP (12, 17, 22, 37) and they are summarised in the tables below. More detailed test data can be found in the accompanied excel sheets.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **All Intra HE** | | | **All Intra LC** | | | **All Intra HE-10** | | |
|  | Y | U | V | Y | U | V | Y | U | V |
| Class A (8bit) | 0.08% | 0.21% | 0.32% | 0.02% | 0.10% | 0.15% |  |  |  |
| Class B | 0.12% | 0.38% | 0.40% | 0.04% | 0.12% | 0.13% |  |  |  |
| Class C | 0.09% | 0.12% | 0.14% | 0.04% | 0.15% | 0.15% |  |  |  |
| Class D | 0.08% | 0.19% | 0.15% | 0.04% | 0.14% | 0.07% |  |  |  |
| Class E | 0.23% | 0.38% | 0.43% | 0.10% | 0.48% | 0.41% |  |  |  |
| **Overall** | 0.12% | 0.26% | 0.28% | 0.05% | 0.19% | 0.17% |  |  |  |
|  | 0.12% | 0.27% | 0.29% | 0.05% | 0.19% | 0.17% |  |  |  |
| Class F | 0.09% | 0.17% | 0.05% | 0.04% | 0.14% | 0.08% |  |  |  |
| Enc Time[%] | 100% | | | 98% | | |  | | |
| Dec Time[%] | 100% | | | 101% | | |  | | |
|  |  |  |  |  |  |  |  |  |  |
|  | **Random Access HE** | | | **Random Access LC** | | | **Random Access HE-10** | | |
|  | Y | U | V | Y | U | V | Y | U | V |
| Class A (8bit) | 0.06% | 0.11% | 0.01% | 0.08% | 0.19% | -0.18% | 0.19% | 0.30% | 0.48% |
| Class B | 0.13% | 0.30% | 0.27% | 0.10% | 0.28% | 0.16% | 0.18% | 0.14% | 0.32% |
| Class C | 0.08% | 0.00% | 0.14% | 0.09% | 0.20% | 0.03% |  |  |  |
| Class D | 0.09% | 0.04% | 0.33% | 0.04% | 0.15% | -0.12% |  |  |  |
| Class E |  |  |  |  |  |  |  |  |  |
| **Overall** | 0.10% | 0.13% | 0.21% | 0.08% | 0.21% | 0.00% | 0.18% | 0.21% | 0.39% |
|  | 0.10% | 0.11% | 0.24% | 0.08% | 0.19% | -0.05% | 0.19% | 0.21% | 0.34% |
| Class F | 0.05% | 0.08% | 0.06% | 0.10% | 0.01% | 0.11% |  |  |  |
| Enc Time[%] | 99% | | | 99% | | | 98% | | |
| Dec Time[%] | 101% | | | 100% | | | 102% | | |
|  |  |  |  |  |  |  |  |  |  |
|  | **Low delay B HE** | | | **Low delay B LC** | | | **Low delay B HE-10** | | |
|  | Y | U | V | Y | U | V | Y | U | V |
| Class A |  |  |  |  |  |  |  |  |  |
| Class B | 0.13% | 0.35% | 0.41% | 0.05% | -0.35% | 0.14% |  |  |  |
| Class C | 0.11% | 0.07% | 0.16% | 0.07% | 0.32% | 0.02% |  |  |  |
| Class D | 0.09% | -0.16% | 0.09% | 0.00% | 0.53% | 0.00% |  |  |  |
| Class E | 0.09% | -0.14% | -0.67% | 0.13% | -0.20% | 0.29% |  |  |  |
| **Overall** | 0.11% | 0.06% | 0.07% | 0.06% | 0.07% | 0.10% |  |  |  |
|  | 0.10% | 0.12% | 0.04% | 0.06% | 0.03% | 0.12% |  |  |  |
| Class F | 0.02% | 0.60% | -0.19% | 0.03% | -1.08% | -0.30% |  |  |  |
| Enc Time[%] | 100% | | | 99% | | |  | | |
| Dec Time[%] | 101% | | | 101% | | |  | | |

*Table 1. BD-rates for common conditions (method with high-frequency context)*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **All Intra HE** | | | **All Intra LC** | | | **All Intra HE-10** | | |
|  | Y | U | V | Y | U | V | Y | U | V |
| Class A (8bit) | 0.00% | 0.04% | 0.05% | -0.06% | -0.13% | -0.14% |  |  |  |
| Class B | -0.03% | 0.10% | 0.18% | -0.05% | -0.10% | -0.09% |  |  |  |
| Class C | 0.05% | 0.06% | 0.06% | 0.02% | 0.09% | 0.10% |  |  |  |
| Class D | 0.06% | 0.08% | 0.06% | 0.03% | 0.08% | 0.08% |  |  |  |
| Class E | 0.07% | 0.16% | 0.21% | 0.00% | 0.04% | 0.06% |  |  |  |
| **Overall** | 0.03% | 0.09% | 0.12% | -0.01% | 0.00% | 0.01% |  |  |  |
|  | 0.01% | 0.09% | 0.12% | -0.01% | 0.00% | 0.01% |  |  |  |
| Class F | 0.04% | 0.06% | 0.03% | 0.03% | 0.04% | 0.07% |  |  |  |
| Enc Time[%] | 99% | | | 99% | | |  | | |
| Dec Time[%] | 100% | | | 101% | | |  | | |
|  |  |  |  |  |  |  |  |  |  |
|  | **Random Access HE** | | | **Random Access LC** | | | **Random Access HE-10** | | |
|  | Y | U | V | Y | U | V | Y | U | V |
| Class A (8bit) | 0.06% | 0.03% | 0.04% | -0.01% | 0.05% | 0.02% | 0.02% | 0.25% | 0.22% |
| Class B | 0.04% | 0.14% | 0.20% | 0.01% | 0.00% | 0.04% | 0.06% | 0.12% | 0.14% |
| Class C | 0.06% | 0.05% | 0.03% | 0.05% | 0.10% | 0.03% |  |  |  |
| Class D | 0.06% | 0.02% | 0.10% | 0.02% | 0.00% | 0.01% |  |  |  |
| Class E |  |  |  |  |  |  |  |  |  |
| **Overall** | 0.05% | 0.07% | 0.11% | 0.02% | 0.03% | 0.02% | 0.04% | 0.18% | 0.18% |
|  | 0.05% | 0.06% | 0.10% | 0.02% | 0.03% | 0.02% | 0.03% | 0.21% | 0.22% |
| Class F | -0.04% | 0.00% | 0.00% | 0.09% | -0.05% | 0.07% |  |  |  |
| Enc Time[%] | 99% | | | 100% | | | 98% | | |
| Dec Time[%] | 99% | | | 99% | | | 99% | | |
|  |  |  |  |  |  |  |  |  |  |
|  | **Low delay B HE** | | | **Low delay B LC** | | | **Low delay B HE-10** | | |
|  | Y | U | V | Y | U | V | Y | U | V |
| Class A |  |  |  |  |  |  |  |  |  |
| Class B | 0.04% | 0.12% | 0.17% | 0.00% | -0.02% | -0.03% |  |  |  |
| Class C | 0.04% | 0.04% | 0.11% | 0.05% | 0.03% | 0.06% |  |  |  |
| Class D | 0.07% | 0.04% | 0.10% | 0.03% | 0.06% | -0.03% |  |  |  |
| Class E | 0.07% | -0.38% | 0.00% | 0.06% | 0.32% | 0.49% |  |  |  |
| **Overall** | 0.05% | -0.01% | 0.11% | 0.03% | 0.08% | 0.09% |  |  |  |
|  | 0.05% | 0.00% | 0.12% | 0.03% | 0.08% | 0.10% |  |  |  |
| Class F | 0.08% | 0.06% | 0.07% | 0.32% | 0.35% | 0.13% |  |  |  |
| Enc Time[%] | 100% | | | 99% | | |  | | |
| Dec Time[%] | 99% | | | 101% | | |  | | |

*Table 2. BD-rates for low QPs (method with high-frequency context)*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **All Intra HE** | | | **All Intra LC** | | | **All Intra HE-10** | | |
|  | Y | U | V | Y | U | V | Y | U | V |
| Class A (8bit) | 0.09% | 0.21% | 0.31% | 0.04% | 0.13% | 0.20% |  |  |  |
| Class B | 0.14% | 0.40% | 0.43% | 0.05% | 0.19% | 0.18% |  |  |  |
| Class C | 0.12% | 0.15% | 0.20% | 0.05% | 0.18% | 0.18% |  |  |  |
| Class D | 0.10% | 0.25% | 0.21% | 0.04% | 0.13% | 0.09% |  |  |  |
| Class E | 0.24% | 0.47% | 0.50% | 0.10% | 0.44% | 0.42% |  |  |  |
| **Overall** | 0.14% | 0.30% | 0.33% | 0.06% | 0.21% | 0.20% |  |  |  |
|  | 0.13% | 0.30% | 0.33% | 0.05% | 0.21% | 0.20% |  |  |  |
| Class F | 0.09% | 0.18% | 0.13% | 0.05% | 0.16% | 0.04% |  |  |  |
| Enc Time[%] | 99% | | | 97% | | |  | | |
| Dec Time[%] | 98% | | | 99% | | |  | | |
|  |  |  |  |  |  |  |  |  |  |
|  | **Random Access HE** | | | **Random Access LC** | | | **Random Access HE-10** | | |
|  | Y | U | V | Y | U | V | Y | U | V |
| Class A (8bit) | 0.09% | 0.08% | -0.07% | 0.06% | 0.28% | 0.04% | 0.16% | 0.26% | 0.86% |
| Class B | 0.13% | 0.34% | 0.24% | 0.12% | 0.33% | 0.18% | 0.16% | 0.26% | 0.23% |
| Class C | 0.10% | 0.07% | 0.22% | 0.09% | 0.01% | 0.00% |  |  |  |
| Class D | 0.12% | -0.09% | 0.16% | 0.03% | 0.16% | -0.16% |  |  |  |
| Class E |  |  |  |  |  |  |  |  |  |
| **Overall** | 0.12% | 0.12% | 0.17% | 0.08% | 0.19% | 0.02% | 0.16% | 0.26% | 0.51% |
|  | 0.12% | 0.08% | 0.18% | 0.08% | 0.17% | 0.00% | 0.15% | 0.29% | 0.49% |
| Class F | -0.02% | 0.14% | -0.03% | 0.06% | -0.01% | 0.12% |  |  |  |
| Enc Time[%] | 99% | | | 100% | | | 99% | | |
| Dec Time[%] | 100% | | | 100% | | | 98% | | |
|  |  |  |  |  |  |  |  |  |  |
|  | **Low delay B HE** | | | **Low delay B LC** | | | **Low delay B HE-10** | | |
|  | Y | U | V | Y | U | V | Y | U | V |
| Class A |  |  |  |  |  |  |  |  |  |
| Class B | 0.18% | 0.14% | 0.54% | 0.06% | -0.11% | 0.20% |  |  |  |
| Class C | 0.09% | -0.05% | 0.04% | 0.12% | 0.20% | 0.07% |  |  |  |
| Class D | 0.14% | -0.52% | -0.37% | -0.01% | 0.62% | -0.17% |  |  |  |
| Class E | 0.17% | 1.11% | -0.44% | 0.11% | -1.04% | -0.44% |  |  |  |
| **Overall** | 0.15% | 0.11% | 0.00% | 0.07% | -0.03% | -0.04% |  |  |  |
|  | 0.14% | 0.08% | -0.02% | 0.07% | -0.03% | 0.02% |  |  |  |
| Class F | 0.10% | 0.22% | -0.14% | -0.25% | -1.08% | -0.86% |  |  |  |
| Enc Time[%] | 99% | | | 100% | | |  | | |
| Dec Time[%] | 99% | | | 100% | | |  | | |

*Table 3. BD-rates for common conditions (no additional contexts)*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **All Intra HE** | | | **All Intra LC** | | | **All Intra HE-10** | | |
|  | Y | U | V | Y | U | V | Y | U | V |
| Class A (8bit) | 0.00% | 0.03% | 0.04% | -0.03% | -0.10% | -0.10% |  |  |  |
| Class B | -0.01% | 0.09% | 0.18% | -0.04% | -0.08% | -0.07% |  |  |  |
| Class C | 0.06% | 0.09% | 0.12% | 0.02% | 0.08% | 0.09% |  |  |  |
| Class D | 0.09% | 0.10% | 0.09% | 0.03% | 0.08% | 0.08% |  |  |  |
| Class E | 0.08% | 0.19% | 0.24% | 0.01% | 0.05% | 0.08% |  |  |  |
| **Overall** | 0.04% | 0.10% | 0.14% | 0.00% | 0.01% | 0.02% |  |  |  |
|  | 0.03% | 0.10% | 0.14% | 0.00% | 0.01% | 0.02% |  |  |  |
| Class F | 0.06% | 0.08% | 0.06% | 0.04% | 0.07% | 0.08% |  |  |  |
| Enc Time[%] | 99% | | | 97% | | |  | | |
| Dec Time[%] | 99% | | | 101% | | |  | | |
|  |  |  |  |  |  |  |  |  |  |
|  | **Random Access HE** | | | **Random Access LC** | | | **Random Access HE-10** | | |
|  | Y | U | V | Y | U | V | Y | U | V |
| Class A (8bit) | 0.06% | 0.05% | 0.09% | 0.00% | 0.01% | 0.01% | 0.02% | 0.26% | 0.30% |
| Class B | 0.05% | 0.13% | 0.20% | 0.02% | 0.02% | 0.04% | 0.06% | 0.13% | 0.17% |
| Class C | 0.09% | 0.13% | 0.11% | 0.06% | 0.07% | 0.13% |  |  |  |
| Class D | 0.09% | 0.04% | 0.02% | 0.04% | 0.10% | 0.06% |  |  |  |
| Class E |  |  |  |  |  |  |  |  |  |
| **Overall** | 0.07% | 0.09% | 0.11% | 0.03% | 0.05% | 0.06% | 0.04% | 0.19% | 0.23% |
|  | 0.07% | 0.09% | 0.11% | 0.03% | 0.06% | 0.07% | 0.04% | 0.20% | 0.28% |
| Class F | -0.03% | 0.05% | 0.02% | 0.05% | 0.00% | 0.08% |  |  |  |
| Enc Time[%] | 99% | | | 99% | | | 100% | | |
| Dec Time[%] | 99% | | | 99% | | | 98% | | |
|  |  |  |  |  |  |  |  |  |  |
|  | **Low delay B HE** | | | **Low delay B LC** | | | **Low delay B HE-10** | | |
|  | Y | U | V | Y | U | V | Y | U | V |
| Class A |  |  |  |  |  |  |  |  |  |
| Class B | 0.04% | 0.10% | 0.20% | 0.01% | 0.01% | 0.04% |  |  |  |
| Class C | 0.07% | 0.07% | 0.16% | 0.05% | 0.02% | 0.07% |  |  |  |
| Class D | 0.07% | 0.03% | -0.13% | 0.01% | 0.10% | -0.03% |  |  |  |
| Class E | 0.08% | 0.04% | -0.28% | 0.04% | -0.23% | -0.30% |  |  |  |
| **Overall** | 0.06% | 0.06% | 0.02% | 0.02% | -0.01% | -0.03% |  |  |  |
|  | 0.07% | 0.06% | 0.01% | 0.02% | 0.00% | -0.04% |  |  |  |
| Class F | 0.05% | -0.01% | -0.03% | 0.23% | 0.32% | 0.00% |  |  |  |
| Enc Time[%] | 100% | | | 100% | | |  | | |
| Dec Time[%] | 100% | | | 101% | | |  | | |

*Table 4. BD-rates for low QPs (no additional contexts)*

# Conclusion

In this contribution, a scheme to allow 3-bins in parallel for the large transforms significance map while reducing the complexity is proposed. The average BD-rate is 0.1%.

# References

[1] T. Ikai, “CE11.1 Improvement of significant\_coeff\_flag context at high frequency area,” JCTVC-H0095, JCT-VC 8th Meeting, San Jose, USA, Feb. 2012.

[2] V. Seregin, J. Sole, M. Karczewicz, “Non-CE11: Support modification for parallel calculation of significant map contexts,” JCTVC-H0427, JCT-VC 8th Meeting, San Jose, USA, Feb. 2012.

# Working draft text

This text modification for working draft is based on JCTVC-G1103 version d8.

###### 9.2.3.1.1.5 Derivation process of ctxIdxInc for the syntax element significant\_coeff\_flag

Inputs to this process are the color component index cIdx, the current coefficient scan position ( xC , yC ), the transform block width log2TrafoWidth and the transform block height log2TrafoHeight.

Output of this process is ctxIdxInc.

The variable sigCtx depends on the current position ( xC, yC ), the color component index cIdx, the transform block size and previsously decoded bins of the syntax element significant\_coeff\_flag. For the derivation of sigCtx, the following applies.

* If log2TrafoWidth is equal to log2TrafoHeight and log2TrafoWidth is equal to 2, sigCtx is derived using ctxIdxMap4x4[ ] specified in Table 9‑39 as follows..

sigCtx = ctxIdxMap4x4[ ((cIdx > 0) ? 15 : 0) + (yC << 2) + xC ] (9‑55)

* Otherwise if log2TrafoWidth is equal to log2TrafoHeight and log2TrafoWidth is equal to 3, sigCtx is derived using ctxIdxMap8x8[ ] specified in Table 9‑40 as follows.

sigCtx = ((xC + yC) = = 0) ? 10 : ctxIdxMap8x8[ ((yC >> 1 ) << 2) + (xC >> 1) ] (9‑56)

sigCtx += ( cIdx > 0) ? 6: 9 (9‑56)

* Otherwise if xC + yC is equal to 0, sigCtx is derived as follows.

sigCtx = ( cIdx > 0) ? 17: 20 (9‑57)

* Otherwise ((xC>>2) + (yC>>2) is equal to or greater than 3 << (uiLog2BlkSize-4), sigCtx is derived as follows.
* cIdx is equal to 0, ctxIdxInc is derived as follows.

sigCtx = 27

* Otherwise (cIdx is greater than 0), ctxIdxInc is derived as follows.

sigCtx = 21

* Otherwise (xC + yC is greater than 0), sigCtx is derived using previously decoded bins of the syntax element significant\_coeff\_flag as follows.
* The variable sigCtx is initialized as follows.

sigCtx = 0 (9‑58)

* When xC is less than ( 1 << log2TrafoWidth ) − 1, the following applies.

sigCtx = sigCtx + significant\_coeff\_flag[ xC + 1 ][ yC ] (9‑59)

* When xC is less than ( 1 << log2TrafoWidth ) − 1 and yC is less than ( 1 << log2TrafoHeight ) − 1, and ~~the following applies.~~
  + xC % 4 is not equal to 0 or yC % 4 is not equal to 0,
  + xC % 4 is not equal to 2 or yC % 4 is not equal to 3,

the following applies.

sigCtx = sigCtx + significant\_coeff\_flag[ xC + 1 ][ yC + 1 ] (9‑60)

* When xC is less than ( 1 << log2Width ) − 2, the following applies.

sigCtx = sigCtx + significant\_coeff\_flag[ xC + 2 ][ yC ] (9‑61)

* ~~When all of the following conditions are true,~~
  + ~~yC is less than ( 1 << log2TrafoHeight ) − 1,~~
  + ~~xC % 4 is not equal to 0 or yC % 4 is not equal to 0,~~
  + ~~xC % 4 is not equal to 3 or yC % 4 is not equal to 2,~~

~~the following applies.~~

~~sigCtx = sigCtx + significant\_coeff\_flag[ xC ][ yC + 1 ]~~ (9‑62)

* When yC is less than ( 1 << log2TrafoHeight ) − 2 ~~and sigCtx is less than 4, the following applies.~~

sigCtx = sigCtx + significant\_coeff\_flag[ xC ][ yC + 2 ] (9‑63)

and if sigCtx is less than 4 and xC is less than ( 1 << log2Width ) – 2

sigCtx = sigCtx + significant\_coeff\_flag[ xC + 1 ][ yC + 2 ]

* The variable sigCtx is modified as follows.
  + If cIdx is equal to 0 and xC + yC are greater than (1 << (max(log2TrafoWidth, log2TrafoHeight) − 2)) − 1, the following applies.

sigCtx = ( (sigCtx + 1) >> 1 ) + 24 (9‑63)

* + Otherwise, the following applies.

sigCtx = ( (sigCtx + 1) >> 1 ) + ( (cIdx > 0) ? 18 : 21 ) (9‑63)

The context index increment ctxIdxInc is derived using the color component index cIdx and sigCtx as follows.

* If cIdx is equal to 0, ctxIdxInc is derived as follows.

ctxIdxInc = sigCtx (9‑64)

* Otherwise (cIdx is greater than 0), ctxIdxInc is derived as follows.

ctxIdxInc = 27 + sigCtx (9‑65)

Method 2

For method 2, the first high-lighted part above becomes:

* Otherwise ((xC>>2) + (yC>>2) is equal to or greater than 3 << (uiLog2BlkSize-4), sigCtx is derived as follows.
* cIdx is equal to 0, ctxIdxInc is derived as follows.

sigCtx = 24

* Otherwise (cIdx is greater than 0), ctxIdxInc is derived as follows.

sigCtx = 18

# Patent rights declaration(s)

**Qualcomm may have current or pending patent rights relating to the technology described in this contribution and, conditioned on reciprocity, is prepared to grant licenses under reasonable and non-discriminatory terms as necessary for implementation of the resulting ITU-T Recommendation | ISO/IEC International Standard (per box 2 of the ITU-T/ITU-R/ISO/IEC patent statement and licensing declaration form).**

**SHARP Corporation may have current or pending patent rights relating to the technology described in this contribution and, conditioned on reciprocity, is prepared to grant licenses under reasonable and non-discriminatory terms as necessary for implementation of the resulting ITU-T Recommendation | ISO/IEC International Standard (per box 2 of the ITU-T/ITU-R/ISO/IEC patent statement and licensing declaration form).**