## CABAC parsing process for slice data

### General

### Initialization process

#### General

#### Initialization process for context variables

Outputs of this process are the initialized CABAC context variables indexed by ctxTable and ctxIdx.

For each context variable, the two variables pStateIdx0 and pStateIdx1 are initialized as follows:

* Table 52 to Table 125 contain the values of the 6 bit variable initValue used in the initialization of context variables that are assigned to all syntax elements in subclauses 7.3.10.1 through 7.3.10.11, except end\_of\_slice\_one\_bit, end\_of\_tile\_one\_bit, and end\_of\_subset\_one\_bit.
* From the 6 bit table entry initValue, the two 3 bit variables slopeIdx and offsetIdx are derived as follows:

slopeIdx = initValue >> 3  
offsetIdx = initValue & 7 (1539)

* The variables m and n, used in the initialization of context variables, are derived from slopeIdx and offsetIdx as follows:

m = slopeIdx − 4  
n = ( offsetIdx \* 18 ) + 1 (1540)

* The two values assigned to pStateIdx0 and pStateIdx1 for the initialization are derived from SliceQpY, which is derived in Equation 143. Given the variables m and n, the initialization is specified as follows:

preCtxState = Clip3( 1, 127, ( ( m \* ( Clip3( 0, 51, SliceQpY ) − 16 ) ) >> 1 ) + n ) (1541)

* The two values assigned to pStateIdx0 and pStateIdx1 for the initialization are derived as follows:

pStateIdx0 = preCtxState << 3  
pStateIdx1 = preCtxState << 7 (1542)

NOTE 1 – The variables pStateIdx0 and pStateIdx1 correspond to probability state indices as further described in subclause 9.3.4.3.

In Table 51, the ctxIdx for which initialization is needed for each of the three initialization types, specified by the variable initType, are listed. Also listed is the table number that includes the values of initValue needed for the initialization for each value of ctxIdx. For P and B slice types, the derivation of initType depends on the value of the cabac\_init\_flag syntax element. The variable initType is derived as follows:

if( slice\_type = = I )  
 initType = 0  
else if( slice\_type = = P )  
 initType = cabac\_init\_flag ? 2 : 1 (1543)  
else  
 initType = cabac\_init\_flag ? 1 : 2

| Table 51 – Association of ctxIdx and syntax elements for each initializationType in the initialization process | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Syntax structure** | **Syntax element** | **ctxTable** | **initType** | | | |
| **0** | **1** | **2** | |
| coding\_tree\_unit( ) | alf\_ctb\_flag[ ][ ][ ] | Table 52 | 0..8 | 9..17 | 18..26 | |
| alf\_use\_aps\_flag | Table 53 | 0 | 1 | 2 | |
| alf\_ctb\_filter\_alt\_idx[ ][ ][ ] | Table 56 | 0..1 | 2..3 | 4..5 | |
| alf\_ctb\_cc\_cb\_idc[ ][ ] | Table 54 | 0..2 | 3..5 | 6..8 | |
| alf\_ctb\_cc\_cr\_idc[ ][ ] | Table 55 | 0..2 | 3..5 | 6..8 | |
| sao( ) | sao\_merge\_left\_flag sao\_merge\_up\_flag | Table 57 | 0 | 1 | 2 | |
| sao\_type\_idx\_luma sao\_type\_idx\_chroma | Table 58 | 0 | 1 | 2 | |
| coding\_tree( ) | split\_cu\_flag | Table 59 | 0..8 | 9..17 | 18..26 | |
| split\_qt\_flag | Table 60 | 0..5 | 6..11 | 12..17 | |
| mtt\_split\_cu\_vertical\_flag | Table 61 | 0..4 | 5..9 | 10..14 | |
| mtt\_split\_cu\_binary\_flag | Table 62 | 0..3 | 4..7 | 8..11 | |
| mode\_constraint\_flag | Table 63 | 0..1 | 2..3 | 4..5 | |
| coding\_unit( ) | cu\_skip\_flag[ ][ ] | Table 64 | 0..2 | 3..5 | 6..8 | |
| pred\_mode\_ibc\_flag | Table 65 | 0..3 | 4..7 | 8..11 | |
| pred\_mode\_flag | Table 66 |  | 0..1 | 2..3 | |
| pred\_mode\_plt\_flag | Table 67 | 0 | 1 | 2 | |
| cu\_act\_enabled\_flag | Table 68 | 0 | 1 | 2 | |
| intra\_bdpcm\_luma\_flag | Table 69 | 0 | 1 | 2 | |
| intra\_bdpcm\_luma\_dir\_flag | Table 70 | 0 | 1 | 2 | |
| intra\_mip\_flag[ ][ ] | Table 71 | 0..3 | 4..7 | 8..11 | |
| intra\_luma\_ref\_idx[ ][ ] | Table 72 | 0..1 | 2..3 | 4..5 | |
| intra\_subpartitions\_mode\_flag | Table 73 | 0 | 1 | 2 | |
| intra\_subpartitions\_split\_flag | Table 74 | 0 | 1 | 2 | |
| intra\_luma\_mpm\_flag[ ][ ] | Table 75 | 0 | 1 | 2 | |
| intra\_bdpcm\_chroma\_flag | Table 76 | 0 | 1 | 2 | |
| intra\_bdpcm\_ chroma\_dir\_flag | Table 77 | 0 | 1 | 2 | |
| cclm\_mode\_flag | Table 78 | 0 | 1 | 2 | |
| cclm\_mode\_idx | Table 79 | 0 | 1 | 2 | |
| intra\_chroma\_pred\_mode | Table 80 | 0 | 1 | 2 | |
| general\_merge\_flag[ ][ ] | Table 81 | 0 | 1 | 2 | |
| inter\_pred\_idc[ x0 ][ y0 ] | Table 82 |  | 0..5 | 6..11 | |
| inter\_affine\_flag[ ][ ] | Table 83 |  | 0..2 | 3..5 | |
| cu\_affine\_type\_flag[ ][ ] | Table 84 |  | 0 | 1 | |
| sym\_mvd\_flag[ ][ ] | Table 85 |  | 0 | 1 | |
| ref\_idx\_l0[ ][ ], ref\_idx\_l1[ ][ ] | Table 86 |  | 0..1 | 2..3 | |
| mvp\_l0\_flag[ ][ ], mvp\_l1\_flag[ ][ ] | Table 87 | 0..1 | 2..3 | 4..5 | |
| amvr\_flag[ ][ ] | Table 88 |  | 0..1 | 2..3 | |
| amvr\_precision\_idx[ ][ ] | Table 89 | 0..1 | 2..3 | 4..5 | |
| bcw\_idx[ ][ ] | Table 90 |  | 0 | 1 | |
| cu\_cbf\_flag | Table 91 | 0 | 1 | 2 | |
| cu\_sbt\_flag | Table 92 |  | 0..1 | 2..3 | |
| cu\_sbt\_quad\_flag | Table 93 |  | 0 | 1 | |
| cu\_sbt\_horizontal\_flag | Table 94 |  | 0..2 | 3..5 | |
| cu\_sbt\_pos\_flag | Table 95 |  | 0 | 1 | |
| lfnst\_idx | Table 96 | 0..2 | 3..4 | 6..8 | |
| mts\_idx | Table 97 | 0..3 | 4..7 | 8..11 | |
| palette\_coding( ) | copy\_above\_palette\_indices\_flag | Table 98 | 0..1 | 2..3 | 4..5 | |
| palette\_transpose\_flag | Table 99 | 0..1 | 2..3 | 4..5 | |
| run\_copy\_flag | Table 100 | 0..7 | 8..15 | 16..23 | |
| merge\_data( ) | regular\_merge\_flag[ ][ ] | Table 101 |  | 0..1 | 2..3 | |
| mmvd\_merge\_flag[ ][ ] | Table 102 |  | 0 | 1 | |
| mmvd\_cand\_flag[ ][ ] | Table 103 |  | 0 | 1 | |
| mmvd\_distance\_idx[ ][ ] | Table 104 |  | 0 | 1 | |
| ciip\_flag[ ][ ] | Table 105 |  | 0 | 1 | |
| merge\_subblock\_flag[ ][ ] | Table 106 |  | 0..2 | 3..5 | |
| merge\_subblock\_idx[ ][ ] | Table 107 |  | 0 | 1 | |
| merge\_idx[ ][ ]  merge\_gpm\_idx0[ ][ ] merge\_gpm\_idx1[ ][ ] | Table 108 | 0 | 1 | 2 | |
| mvd\_coding( ) | abs\_mvd\_greater0\_flag[ ] | Table 109 | 0 | 1 | 2 | |
| abs\_mvd\_greater1\_flag[ ] | Table 110 | 0 | 1 | 2 | |
| transform\_unit( ) | tu\_cbf\_luma[ ][ ] | Table 111 | 0..3 | 4..7 | 8..11 | |
| tu\_cbf\_cb[ ][ ] | Table 112 | 0..1 | 2..3 | 4..5 | |
| tu\_cbf\_cr[ ][ ] | Table 113 | 0..2 | 3..5 | 6..8 | |
| cu\_qp\_delta\_abs | Table 114 | 0..1 | 2..3 | 4..5 | |
| cu\_chroma\_qp\_offset\_flag | Table 115 | 0 | 1 | 2 | |
| cu\_chroma\_qp\_offset\_idx | Table 116 | 0 | 1 | 2 | |
| transform\_skip\_flag[ ][ ][ ] | Table 117 | 0..1 | 2..3 | 4..5 | |
| tu\_joint\_cbcr\_residual\_flag[ ][ ] | Table 118 | 0..2 | 3..5 | 6..8 | |
| residual\_coding( ) | last\_sig\_coeff\_x\_prefix | Table 119 | 0..22 | 23..45 | 46..68 | |
| last\_sig\_coeff\_y\_prefix | Table 120 | 0..22 | 23..45 | 46..68 | |
| coded\_sub\_block\_flag[ ][ ] | Table 121 | 0..6 | 7..13 | 14..20 | |
| sig\_coeff\_flag[ ][ ] | Table 122 | 0..62 | 63..125 | 126..188 | |
| par\_level\_flag[ ] | Table 123 | 0..32 | 33..65 | 66..98 | |
| abs\_level\_gtx\_flag[ ][ ] | Table 124 | 0..71 | 72..143 | 144..215 | |
| coeff\_sign\_flag[ ] | Table 125 | 0..5 | 6..11 | 12..17 | |

Table 121 – Specification of initValue and shiftIdx for ctxInc of coded\_sub\_block\_flag

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Initialization variable** | **ctxIdx of coded\_sub\_block\_flag** | | | | | | | | | | | | | | | |
| **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** |
| **initValue** | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP |
| **shiftIdx** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | **16** | **17** | **18** | **19** | **20** |  |  |  |  |  |  |  |  |  |  |  |
| **initValue** | EP | EP | EP | EP | EP |  |  |  |  |  |  |  |  |  |  |  |
| **shiftIdx** | 0 | 0 | 0 | 0 | 0 |  |  |  |  |  |  |  |  |  |  |  |

Table 122 – Specification of initValue and shiftIdx for ctxInc of sig\_coeff\_flag

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Initialization variable** | **ctxIdx of sig\_coeff\_flag** | | | | | | | | | | | | | | | |
| **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** |
| **initValue** | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP |
| **shiftIdx** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | **16** | **17** | **18** | **19** | **20** | **21** | **22** | **23** | **24** | **25** | **26** | **27** | **28** | **29** | **30** | **31** |
| **initValue** | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP |
| **shiftIdx** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | **32** | **33** | **34** | **35** | **36** | **37** | **38** | **39** | **40** | **41** | **42** | **43** | **44** | **45** | **46** | **47** |
| **initValue** | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP |
| **shiftIdx** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | **48** | **49** | **50** | **51** | **52** | **53** | **54** | **55** | **56** | **57** | **58** | **59** | **60** | **61** | **62** | **63** |
| **initValue** | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP |
| **shiftIdx** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | **64** | **65** | **66** | **67** | **68** | **69** | **70** | **71** | **72** | **73** | **74** | **75** | **76** | **77** | **78** | **79** |
| **initValue** | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP |
| **shiftIdx** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | **80** | **81** | **82** | **83** | **84** | **85** | **86** | **87** | **88** | **89** | **90** | **91** | **92** | **93** | **94** | **95** |
| **initValue** | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP |
| **shiftIdx** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | **96** | **97** | **98** | **99** | **100** | **101** | **102** | **103** | **104** | **105** | **106** | **107** | **108** | **109** | **110** | **111** |
| **initValue** | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP |
| **shiftIdx** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | **112** | **113** | **114** | **115** | **116** | **117** | **118** | **119** | **120** | **121** | **122** | **123** | **124** | **125** | **126** | **127** |
| **initValue** | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP |
| **shiftIdx** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | **128** | **129** | **130** | **131** | **132** | **133** | **134** | **135** | **136** | **137** | **138** | **139** | **140** | **141** | **142** | **143** |
| **initValue** | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP |
| **shiftIdx** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | **128** | **129** | **130** | **131** | **132** | **133** | **134** | **135** | **136** | **137** | **138** | **139** | **140** | **141** | **142** | **143** |
| **initValue** | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP |
| **shiftIdx** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | **160** | **161** | **162** | **163** | **164** | **165** | **166** | **167** | **168** | **169** | **170** | **171** | **172** | **173** | **174** | **175** |
| **initValue** | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP |
| **shiftIdx** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | **176** | **177** | **178** | **179** | **180** | **181** | **182** | **183** | **184** | **185** | **186** | **187** | **188** |  |  |  |
| **initValue** | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP |  |  |  |
| **shiftIdx** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |

Table 123 – Specification of initValue and shiftIdx for ctxInc of par\_level\_flag

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Initialization variable** | **ctxIdx of par\_level\_flag** | | | | | | | | | | | | | | | |
| **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** |
| **initValue** | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP |
| **shiftIdx** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | **16** | **17** | **18** | **19** | **20** | **21** | **22** | **23** | **24** | **25** | **26** | **27** | **28** | **29** | **30** | **31** |
| **initValue** | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP |
| **shiftIdx** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | **32** | **33** | **34** | **35** | **36** | **37** | **38** | **39** | **40** | **41** | **42** | **43** | **44** | **45** | **46** | **47** |
| **initValue** | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP |
| **shiftIdx** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | **48** | **49** | **50** | **51** | **52** | **53** | **54** | **55** | **56** | **57** | **58** | **59** | **60** | **61** | **62** | **63** |
| **initValue** | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP |
| **shiftIdx** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | **64** | **65** | **66** | **67** | **68** | **69** | **70** | **71** | **72** | **73** | **74** | **75** | **76** | **77** | **78** | **79** |
| **initValue** | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP |
| **shiftIdx** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | **80** | **81** | **82** | **83** | **84** | **85** | **86** | **87** | **88** | **89** | **90** | **91** | **92** | **93** | **94** | **95** |
| **initValue** | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP |
| **shiftIdx** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | **96** | **97** | **98** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **initValue** | EP | EP | EP |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **shiftIdx** | 0 | 0 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |

Table 124 – Specification of initValue and shiftIdx for ctxInc of abs\_level\_gtx\_flag

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Initialization variable** | **ctxIdx of abs\_level\_gtx\_flag** | | | | | | | | | | | | | | | |
| **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** |
| **initValue** | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP |
| **shiftIdx** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | **16** | **17** | **18** | **19** | **20** | **21** | **22** | **23** | **24** | **25** | **26** | **27** | **28** | **29** | **30** | **31** |
| **initValue** | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP |
| **shiftIdx** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | **32** | **33** | **34** | **35** | **36** | **37** | **38** | **39** | **40** | **41** | **42** | **43** | **44** | **45** | **46** | **47** |
| **initValue** | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP |
| **shiftIdx** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | **48** | **49** | **50** | **51** | **52** | **53** | **54** | **55** | **56** | **57** | **58** | **59** | **60** | **61** | **62** | **63** |
| **initValue** | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP |
| **shiftIdx** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | **64** | **65** | **66** | **67** | **68** | **69** | **70** | **71** | **72** | **73** | **74** | **75** | **76** | **77** | **78** | **79** |
| **initValue** | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP |
| **shiftIdx** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | **80** | **81** | **82** | **83** | **84** | **85** | **86** | **87** | **88** | **89** | **90** | **91** | **92** | **93** | **94** | **95** |
| **initValue** | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP |
| **shiftIdx** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | **96** | **97** | **98** | **99** | **100** | **101** | **102** | **103** | **104** | **105** | **106** | **107** | **108** | **109** | **110** | **111** |
| **initValue** | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP |
| **shiftIdx** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | **112** | **113** | **114** | **115** | **116** | **117** | **118** | **119** | **120** | **121** | **122** | **123** | **124** | **125** | **126** | **127** |
| **initValue** | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP |
| **shiftIdx** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | **128** | **129** | **130** | **131** | **132** | **133** | **134** | **135** | **136** | **137** | **138** | **139** | **140** | **141** | **142** | **143** |
| **initValue** | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP |
| **shiftIdx** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | **128** | **129** | **130** | **131** | **132** | **133** | **134** | **135** | **136** | **137** | **138** | **139** | **140** | **141** | **142** | **143** |
| **initValue** | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP |
| **shiftIdx** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | **160** | **161** | **162** | **163** | **164** | **165** | **166** | **167** | **168** | **169** | **170** | **171** | **172** | **173** | **174** | **175** |
| **initValue** | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP |
| **shiftIdx** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | **176** | **177** | **178** | **179** | **180** | **181** | **182** | **183** | **184** | **185** | **186** | **187** | **188** | **189** | **190** | **191** |
| **initValue** | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP |
| **shiftIdx** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | **192** | **193** | **194** | **195** | **196** | **197** | **198** | **199** | **200** | **201** | **202** | **203** | **204** | **205** | **206** | **207** |
| **initValue** | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP |
| **shiftIdx** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | **208** | **209** | **210** | **211** | **212** | **213** | **214** | **215** |  |  |  |  |  |  |  |  |
| **initValue** | EP | EP | EP | EP | EP | EP | EP | EP |  |  |  |  |  |  |  |  |
| **shiftIdx** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |  |  |  |  |  |

Table 125 – Specification of initValue and shiftIdx for ctxInc of coeff\_sign\_flag

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Initialization variable** | **ctxIdx of coeff\_sign\_flag** | | | | | | | | | | | | | | | | | |
| **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** |
| **initValue** | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP | EP |
| **shiftIdx** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

#### Storage process for context variables

#### Synchronization process for context variables

#### Initialization process for the arithmetic decoding engine

#### Storage process for palette predictor

#### Synchronization process for palette predictor

### Binarization process

#### General

#### Rice parameter derivation process for abs\_remainder[ ] and dec\_abs\_level[ ]

#### Truncated Rice binarization process

#### Truncated Binary (TB) binarization process

#### k-th order Exp-Golomb binarization process

#### Limited k-th order Exp-Golomb binarization process

#### Fixed-length binarization process

#### Binarization process for intra\_chroma\_pred\_mode

#### Binarization process for inter\_pred\_idc

#### Binarization process for cu\_qp\_delta\_abs

#### Binarization process for abs\_remainder[ ]

#### Binarization process for dec\_abs\_level[ ]

#### Binarization process for palette\_idx\_idc

#### Binarization process for abs\_mvd\_minus2

### Decoding process flow

#### General

#### Derivation process for ctxTable, ctxIdx and bypassFlag

##### 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 131:

* If the entry in Table 131 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 51
* The variable ctxInc is specified by the corresponding entry in Table 131 and when more than one value is listed in Table 131 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 in Table 51 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 131 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.a
* Otherwise, if the entry in Table 131 is equal to "terminate", the values of binIdx are decoded by invoking the DecodeTerminate process as specified in clause 9.3.4.3.5 and the following applies:
* ctxTable is set equal to 0.
* ctxIdx is set equal to 0.
* bypassFlag is set equal to 0.
* Otherwise (the entry in Table 131 is equal to "na"), the values of binIdx do not occur for the corresponding syntax element.

| Table 131 – Assignment of ctxInc to syntax elements with context coded bins | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **Syntax element** | **binIdx** | | | | | |
| **0** | **1** | **2** | **3** | **4** | **>= 5** |
| end\_of\_slice\_one\_bit | terminate | na | na | na | na | na |
| end\_of\_tile\_one\_bit | terminate | na | na | na | na | na |
| end\_of\_subset\_one\_bit | terminate | na | na | na | na | na |
| alf\_ctb\_flag[ ][ ][ ] | 0..8 (clause 9.3.4.2.2) | na | na | na | na | na |
| alf\_use\_aps\_flag | 0 | na | na | na | na | na |
| alf\_luma\_fixed\_filter\_idx | bypass | bypass | bypass | bypass | bypass | bypass |
| alf\_luma\_prev\_filter\_idx | bypass | bypass | bypass | bypass | bypass | bypass |
| alf\_ctb\_filter\_alt\_idx[ 0 ][ ][ ] | 0 | 0 | 0 | 0 | 0 | 0 |
| alf\_ctb\_filter\_alt\_idx[ 1 ][ ][ ] | 1 | 1 | 1 | 1 | 1 | 1 |
| alf\_ctb\_cc\_cb\_idc[ ][ ] | 0..2 (clause 9.3.4.2.2) | bypass | bypass | bypass | bypass | bypass |
| alf\_ctb\_cc\_cr\_idc[ ][ ] | 0..2 (clause 9.3.4.2.2) | bypass | bypass | bypass | bypass | bypass |
| sao\_merge\_left\_flag | 0 | na | na | na | na | na |
| sao\_merge\_up\_flag | 0 | na | na | na | na | na |
| sao\_type\_idx\_luma | 0 | bypass | na | na | na | na |
| sao\_type\_idx\_chroma | 0 | bypass | na | na | na | na |
| sao\_offset\_abs[ ][ ][ ][ ] | bypass | bypass | bypass | bypass | bypass | na |
| sao\_offset\_sign[ ][ ][ ][ ] | bypass | na | na | na | na | na |
| sao\_band\_position[ ][ ][ ] | bypass | bypass | bypass | bypass | bypass | bypass |
| sao\_eo\_class\_luma | bypass | bypass | na | na | na | na |
| sao\_eo\_class\_chroma | bypass | bypass | na | na | na | na |
| split\_cu\_flag | 0..8 (clause 9.3.4.2.2) | na | na | na | na | na |
| split\_qt\_flag | 0..5 (clause 9.3.4.2.2) | na | na | na | na | na |
| mtt\_split\_cu\_vertical\_flag | 0..4 (clause 9.3.4.2.3) | na | na | na | na | na |
| mtt\_split\_cu\_binary\_flag | ( 2 \* mtt\_split\_cu\_vertical\_flag ) + ( mttDepth < = 1 ? 1 : 0 ) | na | na | na | na | na |
| mode\_constraint\_flag | 0,1 (clause 9.3.4.2.2) | na | na | na | na | na |
| cu\_skip\_flag[ ][ ] | 0,1,2 (clause 9.3.4.2.2) | na | na | na | na | na |
| pred\_mode\_flag | 0,1 (clause 9.3.4.2.2) | na | na | na | na | na |
| pred\_mode\_ibc\_flag | 0,1,2 (clause 9.3.4.2.2) | na | na | na | na | na |
| pred\_mode\_plt\_flag | 0 | na | na | na | na | na |
| cu\_act\_enabled\_flag | 0 | na | na | na | na | na |
| intra\_bdpcm\_luma\_flag | 0 | na | na | na | na | na |
| intra\_bdpcm\_luma\_dir\_flag | 0 | na | na | na | na | na |
| intra\_mip\_flag[ ][ ] | (Abs( Log2(cbWidth) − Log2(cbHeight) ) > 1) ?  3 : ( 0,1,2 (clause 9.3.4.2.2) ) [Ed. (YK): Something is wrong here.] | na | na | na | na | na |
| intra\_mip\_transposed[ ][ ] | bypass | na | na | na | na | na |
| intra\_mip\_mode[ ][ ] | bypass | bypass | bypass | bypass | bypass | na |
| intra\_luma\_ref\_idx[ ][ ] | 0 | 1 | na | na | na | na |
| intra\_subpartitions\_mode\_flag | 0 | na | na | na | na | na |
| intra\_subpartitions\_split\_flag | 0 | na | na | na | na | na |
| intra\_luma\_mpm\_flag[ ][ ] | 0 | na | na | na | na | na |
| intra\_luma\_not\_planar\_flag[ ][ ] | intra\_subpartitions\_mode\_flag | na | na | na | na | na |
| intra\_luma\_mpm\_idx[ ][ ] | bypass | bypass | bypass | bypass | na | na |
| intra\_luma\_mpm\_remainder[ ][ ] | bypass | bypass | bypass | bypass | bypass | bypass |
| intra\_bdpcm\_chroma\_flag | 0 | na | na | na | na | na |
| intra\_bdpcm\_chroma\_dir\_flag | 0 | na | na | na | na | na |
| cclm\_mode\_flag | 0 | na | na | na | na | na |
| cclm\_mode\_idx | 0 | bypass | na | na | na | na |
| intra\_chroma\_pred\_mode | 0 | bypass | bypass | na | na | na |
| palette\_predictor\_run | bypass | bypass | bypass | bypass | bypass | bypass |
| num\_signalled\_palette\_entries | bypass | bypass | bypass | bypass | bypass | bypass |
| new\_palette\_entries | bypass | bypass | bypass | bypass | bypass | bypass |
| palette\_escape\_val\_present\_flag | bypass | na | na | na | na | na |
| palette\_transpose\_flag | 0 | na | na | na | na | na |
| palette\_idx\_idc | bypass | bypass | bypass | bypass | bypass | bypass |
| copy\_above\_palette\_indices\_flag | 0 | na | na | na | na | na |
| run\_copy\_flag | 0..7 (clause 9.3.4.2.11) | na | na | na | na | na |
| palette\_escape\_val | bypass | bypass | bypass | bypass | bypass | bypass |
| general\_merge\_flag[ ][ ] | 0 | na | na | na | na | na |
| regular\_merge\_flag[ ][ ] | cu\_skip\_flag[ ][ ] ? 0 : 1 | na | na | na | na | na |
| mmvd\_merge\_flag[ ][ ] | 0 | na | na | na | na | na |
| mmvd\_cand\_flag[ ][ ] | 0 | na | na | na | na | na |
| mmvd\_distance\_idx[ ][ ] | 0 | bypass | bypass | bypass | bypass | bypass |
| mmvd\_direction\_idx[ ][ ] | bypass | bypass | na | na | na | na |
| merge\_subblock\_flag[ ][ ] | 0,1,2 (clause 9.3.4.2.2) | na | na | na | na | na |
| merge\_subblock\_idx[ ][ ] | 0 | bypass | bypass | bypass | bypass | na |
| ciip\_flag[ ][ ] | 0 | na | na | na | na | na |
| merge\_idx[ ][ ] | 0 | bypass | bypass | bypass | bypass | na |
| merge\_gpm\_partition\_idx[ ][ ] | bypass | bypass | bypass | bypass | bypass | bypass |
| merge\_gpm\_idx0[ ][ ] | 0 | bypass | bypass | bypass | bypass | na |
| merge\_gpm\_idx1[ ][ ] | 0 | bypass | bypass | bypass | na | na |
| inter\_pred\_idc[ x0 ][ y0 ] | ( cbWidth + cbHeight ) > 12 ? 7 − ( ( 1 +  Log2( cbWidth ) + Log2( cbHeight ) ) >> 1 )   : 5 | 5 | na | na | na | na |
| inter\_affine\_flag[ ][ ] | 0,1,2 (clause 9.3.4.2.2) | na | na | na | na | na |
| cu\_affine\_type\_flag[ ][ ] | 0 | na | na | na | na | na |
| sym\_mvd\_flag[ ][ ] | 0 | na | na | na | na | na |
| ref\_idx\_l0[ ][ ] | 0 | 1 | bypass | bypass | bypass | bypass |
| ref\_idx\_l1[ ][ ] | 0 | 1 | bypass | bypass | bypass | bypass |
| mvp\_l0\_flag[ ][ ] | 0 | na | na | na | na | na |
| mvp\_l1\_flag[ ][ ] | 0 | na | na | na | na | na |
| amvr\_flag[ ][ ] | inter\_affine\_flag[ ][ ] ? 1 : 0 | na | na | na | na | na |
| amvr\_precision\_idx[ ][ ] | 0 | 1 | na | na | na | na |
| bcw\_idx[ ][ ] NoBackwardPredFlag = = 0 | 0 | bypass | na | na | na | na |
| bcw\_idx[ ][ ]  NoBackwardPredFlag = = 1 | 0 | bypass | bypass | bypass | na | na |
| cu\_cbf\_flag | 0 | na | na | na | na | na |
| cu\_sbt\_flag | ( cbWidth \*  cbHeight <= 256 ) ? 1 : 0 | na | na | na | na | na |
| cu\_sbt\_quad\_flag | 0 | na | na | na | na | na |
| cu\_sbt\_horizontal\_flag | ( cbWidth  = = cbHeight ) ? 0 : ( cbWidth < cbHeight ) ? 1 : 2 | na | na | na | na | na |
| cu\_sbt\_pos\_flag | 0 | na | na | na | na | na |
| lfnst\_idx | ( treeType != SINGLE\_TREE ) ? 1 : 0 | 2 | na | na | na | na |
| mts\_idx | 0 | 1 | 2 | 3 | na | na |
| abs\_mvd\_greater0\_flag[ ] | 0 | na | na | na | na | na |
| abs\_mvd\_greater1\_flag[ ] | 0 | na | na | na | na | na |
| abs\_mvd\_minus2[ ] | bypass | bypass | bypass | bypass | bypass | bypass |
| mvd\_sign\_flag[ ] | bypass | na | na | na | na | na |
| tu\_cbf\_luma[ ][ ] | 0,1,2,3 (clause 9.3.4.2.5) | na | na | na | na | na |
| tu\_cbf\_cb[ ][ ] | intra\_bdpcm\_chroma\_flag ? 1 : 0 | na | na | na | na | na |
| tu\_cbf\_cr[ ][ ] | intra\_bdpcm\_chroma\_flag ? 2 : tu\_cbf\_cb[ ][ ] | na | na | na | na | na |
| cu\_qp\_delta\_abs | 0 | 1 | 1 | 1 | 1 | bypass |
| cu\_qp\_delta\_sign\_flag | bypass | na | na | na | na | na |
| cu\_chroma\_qp\_offset\_flag | 0 | na | na | na | na | na |
| cu\_chroma\_qp\_offset\_idx | 0 | 0 | 0 | 0 | 0 | na |
| transform\_skip\_flag[ ][ ][ cIdx ] | cIdx = = 0 ? 0 : 1 | na | na | na | na | na |
| tu\_joint\_cbcr\_residual\_flag[ ][ ] | 2\*tu\_cbf\_cb[ ][ ] + tu\_cbf\_cr[ ][ ] − 1 | na | na | na | na | na |
| last\_sig\_coeff\_x\_prefix | 0..22 (clause 9.3.4.2.4) | | | | | |
| last\_sig\_coeff\_y\_prefix | 0..22 (clause 9.3.4.2.4) | | | | | |
| last\_sig\_coeff\_x\_suffix | bypass | bypass | bypass | bypass | bypass | bypass |
| last\_sig\_coeff\_y\_suffix | bypass | bypass | bypass | bypass | bypass | bypass |
| coded\_sub\_block\_flag[ ][ ] | 0..6 (clause 9.3.4.2.6) | na | na | na | na | na |
| sig\_coeff\_flag[ ][ ] | 0..62 (clause 9.3.4.2.8) | na | na | na | na | na |
| par\_level\_flag[ ] | 0..32 (clause 9.3.4.2.9) | na | na | na | na | na |
| abs\_level\_gtx\_flag[ ][ ] | 0..71 (clause 9.3.4.2.9) | na | na | na | na | na |
| abs\_remainder[ ] | bypass | bypass | bypass | bypass | bypass | bypass |
| dec\_abs\_level[ ] | bypass | bypass | bypass | bypass | bypass | bypass |
| coeff\_sign\_flag[ ]  transform\_skip\_flag[ x0 ][ y0 ][ cIdx ] = = 0 | | RemCcbs = = 0 | | slice\_ts\_residual\_coding\_disabled\_flag | bypass | na | na | na | na | na |
| coeff\_sign\_flag[ ] transform\_skip\_flag[ x0 ][ y0 ][ cIdx ] = = 1 && RemCcbs >= 0 && !slice\_ts\_residual\_coding\_disabled\_flag | 0..5 (clause 9.3.4.2.10) | na | na | na | na | na |

##### Derivation process of ctxInc using left and above syntax elements

##### Derivation process of ctxIncfor the syntax element mtt\_split\_cu\_vertical\_flag

##### Derivation process of ctxInc for the syntax elements last\_sig\_coeff\_x\_prefix and last\_sig\_coeff\_y\_prefix

##### Derivation process of ctxInc for the syntax element tu\_cbf\_luma

##### Derivation process of ctxInc for the syntax element coded\_sub\_block\_flag

##### Derivation process for the variables locNumSig, locSumAbsPass1

##### Derivation process of ctxInc for the syntax element sig\_coeff\_flag

##### Derivation process of ctxInc for the syntax elements par\_level\_flag and abs\_level\_gtx\_flag

Inputs to this process are the colour component index cIdx, the luma location ( x0, y0 ) specifying the top-left sample of the current transform block relative to the top-left sample of the current picture, the current coefficient scan location ( xC, yC ), the binary logarithm of the transform block width log2TbWidth, and the binary logarithm of the transform block height log2TbHeight.

Output of this process is the variable ctxInc.

The variable ctxInc is derived as follows:

* If transform\_skip\_flag[ x0 ][ y0 ][ cIdx ] is equal to 1 and slice\_ts\_residual\_coding\_disabled\_flag is equal to 0, the following applies:
* If the syntax element is par\_level\_flag[ n ], the following applies:

ctxInc = 32 (1585)

* Otherwise, if the syntax element is abs\_level\_gtx\_flag[ n ][ 0 ], the following applies:
* If BdpcmFlag[ x0 ][ y0 ][ cIdx ] is equal to 1, ctxInc is derived as follows:

ctxInc = 67 (1586)

* Otherwise, if xC is greater than 0 and yC is greater than 0, ctxInc is derived as follows:

ctxInc = 64 + sig\_coeff\_flag[ xC − 1 ][ yC ] + sig\_coeff\_flag[ xC ][ yC − 1 ] (1587)

* Otherwise, if xC is greater than 0, ctxInc is derived as follows:

ctxInc = 64 + sig\_coeff\_flag[ xC − 1 ][ yC ] (1588)

* Otherwise, if yC is greater than 0, ctxInc is derived as follows:

ctxInc = 64 + sig\_coeff\_flag[ xC ][ yC − 1 ] (1589)

* Otherwise, ctxInc is derived as follows:

ctxInc = 64 (1590)

* Otherwise, if the syntax element is abs\_level\_gtx\_flag[ n ][ j ] with j > 0, the following applies:

ctxInc = 67 + j (1591)

* Otherwise (transform\_skip\_flag[ x0 ][ y0 ][ cIdx ] is equal to 0 or slice\_ts\_residual\_coding\_disabled\_flag is equal to 1), the following applies:
* The variable locNumSig and locSumAbsPass1 is derived by invoking the derivation process for the variables locNumSig and locSumAbsPass1 specifies in clause 9.3.4.2.7 with colour component index cIdx, the luma location ( x0, y0), the current coefficient scan location (xC, yC ), the binary logarithm of the transform block width log2TbWidth, and the binary logarithm of the transform block height log2TbHeight as input.
* The variable ctxOffset is set equal to Min( locSumAbsPass1 − locNumSig, 4 ).
* The variable d is set equal to xC + yC.
* If xC is equal to LastSignificantCoeffX and yC is equal to LastSignificantCoeffY, ctxInc is derived as follows:

ctxInc = ( cIdx  = =  0 ? 0 : 21 ) (1592)

* Otherwise, if cIdx is equal to 0, ctxInc is derived as follows:

ctxInc = 1 + ctxOffset + ( d  = =  0 ? 15 : ( d < 3 ? 10 : ( d < 10 ? 5 : 0 ) ) ) (1593)

* Otherwise (cIdx is greater than 0), ctxInc is derived as follows:

ctxInc = 22 + ctxOffset + ( d  = =  0 ? 5 : 0 ) (1594)

##### Derivation process of ctxInc for the syntax element coeff\_sign\_flag for transform skip mode

##### Derivation process of ctxInc for the syntax element run\_copy\_flag