

The background of the slide is a vibrant red, overlaid with a complex network of white lines and dots, resembling a data or communication network. On the left side, there are faint, semi-transparent digital interface elements, including a code editor window with a '</>' symbol, a waveform graph, and various binary digits (0s and 1s) scattered throughout.

# AHG17: Signaling of Rectangular Slices

**JVET-P0240**

**T. Hellman, W. Wan, M. Zhou, B. Heng,  
P. Chen**

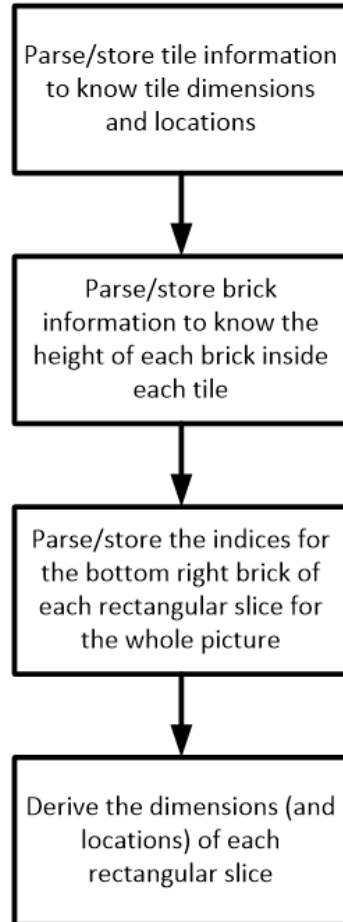
**Geneva, October 2019 meeting**



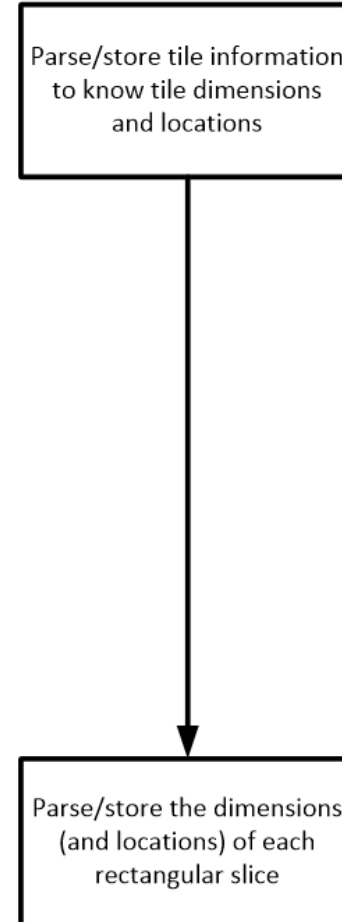
# Introduction

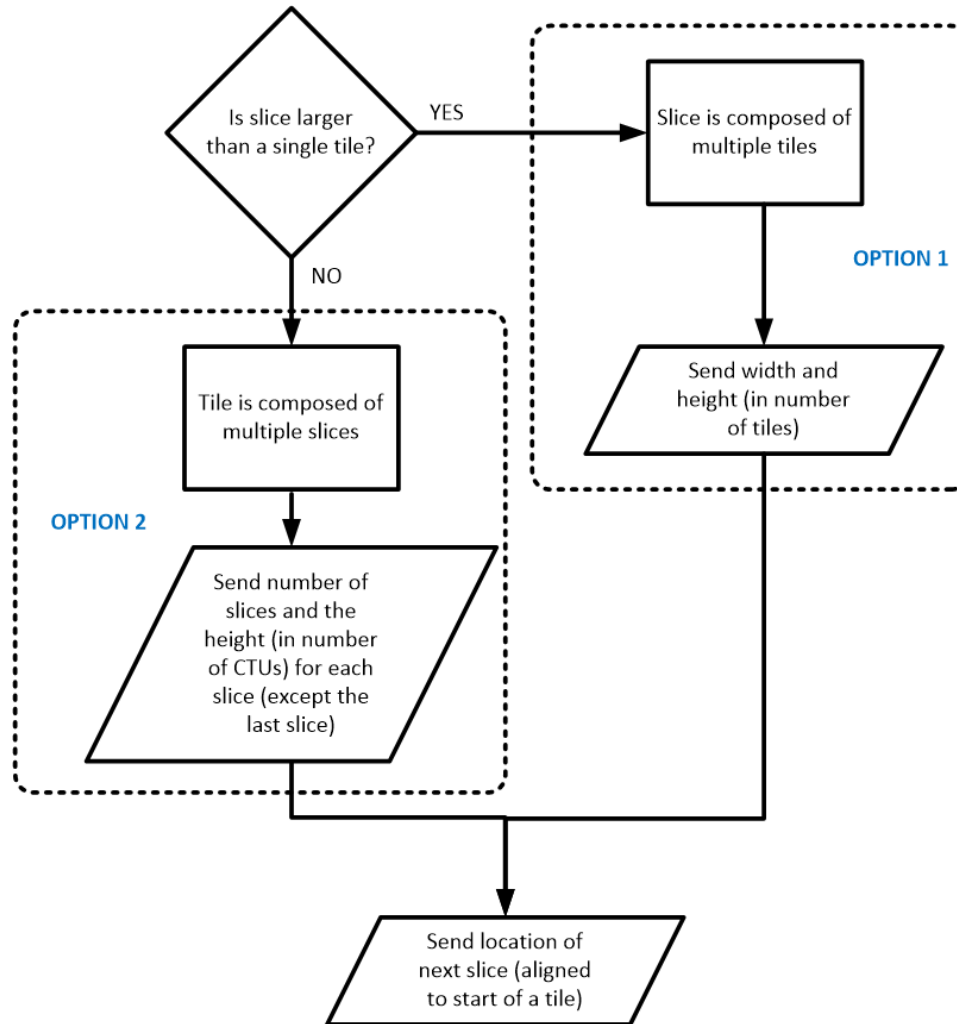
- Propose dimensions and locations of rectangular slices be signalled directly by using tile information rather than bricks
  - Each CTU row can potentially be a brick boundary
  - PPS for 8K picture with 64x64 CTUs potentially has 8192 bricks!
- Removes the need to predefine and store brick information at the PPS level
- No change to the slice and brick partitioning options
- Proposed changes only affect the signalling

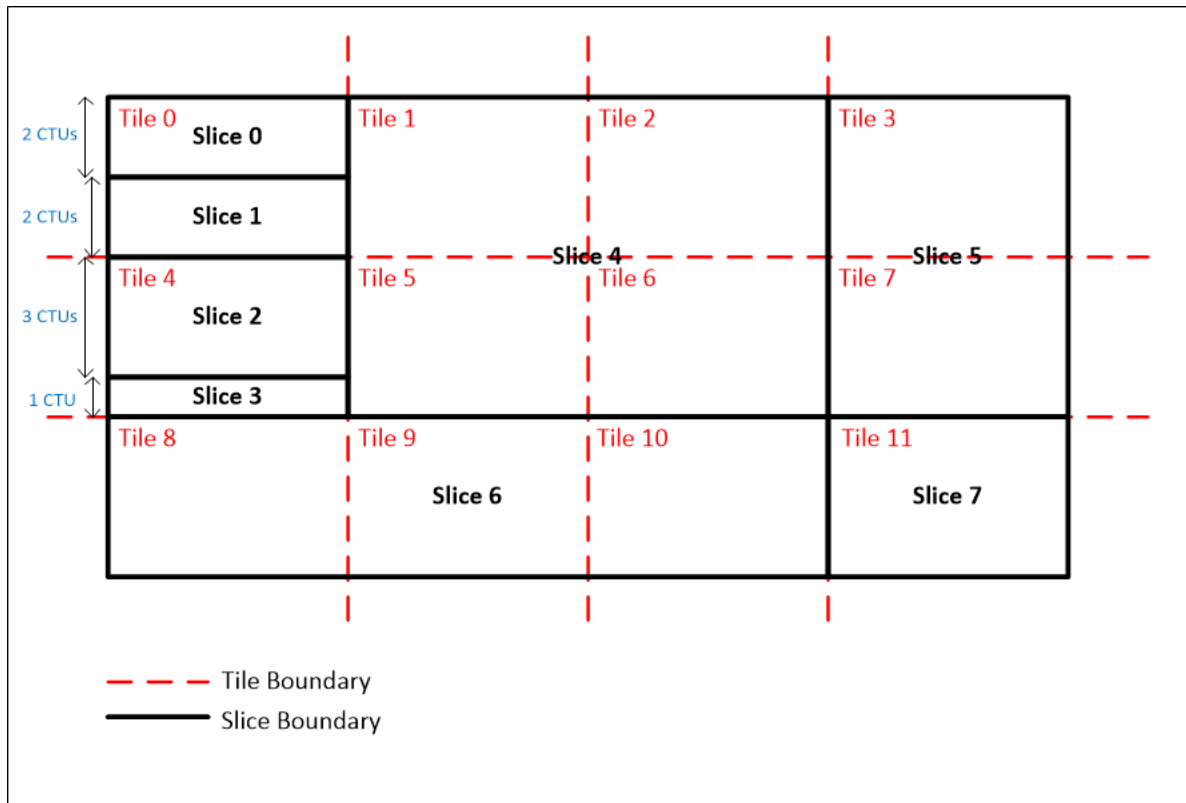
### Current Draft Text



### Proposed Method







Tile 0: Option 2

CTU\_rows = 2

Infer CTU\_rows = 2

Tile 4: Option 2

CTU\_rows = 3

Infer CTU\_rows = 1

Tile 1: Option 1

2x2 Tiles

Tile 3: Option 1

2x1 Tiles

Tile 8: Option 1

1x3 Tiles

Tile 11: Inferred

# Overview of new syntax elements

- **num\_slices\_in\_pic\_minus1** plus 1 specifies the number of rectangular slices in each picture referring to the PPS.
- **slice\_width\_in\_tiles\_minus1[ i ]** plus 1 specifies the width of the i-th rectangular slice in units of tile columns.
- **slice\_height\_in\_tiles\_minus1[ i ]** plus 1 specifies the height of the i-th rectangular slice in units of tile rows.
- **num\_slices\_in\_tile\_minus1[ i ]** plus 1 specifies the number of slices in the current tile for the case where the i-th slice contains a subset of bricks from a single tile.
- **slice\_height\_in\_ctu\_minus1[ i ]** plus 1 specifies the height of the i-th rectangular slice in units of CTU rows for the case where the i-th slice contains a subset of bricks from a single tile.
- **tile\_idx\_delta[ i ]** specifies the difference in tile index between the i-th rectangular slice and the (i+1)-th rectangular slice. The value of **tile\_idx\_delta[ i ]** shall not be equal to 0.

# Syntax Changes to slice\_data

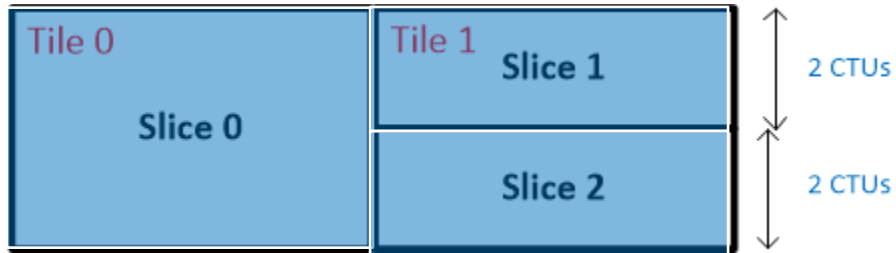
- Introduction of **end\_of\_brick\_flag** allows each CTU tile row within a slice to start a new brick.
- All existing brick splitting options are still available, without the need to pre-define and store all this information in the PPS.

# Summary of Text Simplifications

- Reduce brick/slice related syntax elements in the PPS from 13 to 7
- More significantly, removes the following lookup tables:
  - BrickColBd[ ], BrickRowBd[ ], BrickWidth[ ], BrickHeight[ ]
  - CtbAddrRsToBs[ ], CtbAddrBsToRs[ ], BrickId[ ]
  - NumCtusInBrick[ ], FirstCtbAddrBs[ ], BricksToSliceMap[ ], BottomRightBrickIdx[ ]
  - TopLeftBrickIdx[ ], NumBrickInSlice[ ]



# Example



- $\text{num\_slices\_in\_pic\_minus1} = 2$
- $i=0 \rightarrow \text{slice\_width\_in\_tiles\_minus1} = 0$ ,  
 $\text{slice\_height\_in\_tiles\_minus1} = 0$ 
  - $\text{num\_slices\_in\_tile\_minus1} = 0$
  - $j=0 \rightarrow$  do not send slice height (rest of tile = 4 CTUs)
  - $\text{tile\_idx\_delta}[i] = 1$
- $i=1 \rightarrow \text{slice\_width\_in\_tiles\_minus1} = 0$ ,  
 $\text{slice\_height\_in\_tiles\_minus1} = 0$ 
  - $j=0 \rightarrow$  do not send slice height (rest of tile = 4 CTUs)
  - $j=0 \rightarrow \text{slice\_height\_in\_ctu\_minus1} = 1$  (note  $i++$ )
- $i=2, j=1 \rightarrow$  do not send slice height (rest of tile = 2 CTUs)

<b>num_slices_in_pic_minus1</b>
for( i = 0; i < num_slices_in_pic_minus1; i++ ) {
<b>slice_width_in_tiles_minus1</b> [ i ]
<b>slice_height_in_tiles_minus1</b> [ i ]
if( slice_width_in_tiles_minus1[ i ] == 0 && slice_height_in_tiles_minus1[ i ] == 0 ) {
<b>num_slices_in_tile_minus1</b> [ i ]
numSlicesInTileMinus1 = num_slices_in_tile_minus1[ i ]
for( j = 0; j < numSlicesInTileMinus1; j++ )
<b>slice_height_in_ctu_minus1</b> [ i++ ]
}
<b>tile_idx_delta</b> [ i ]
}

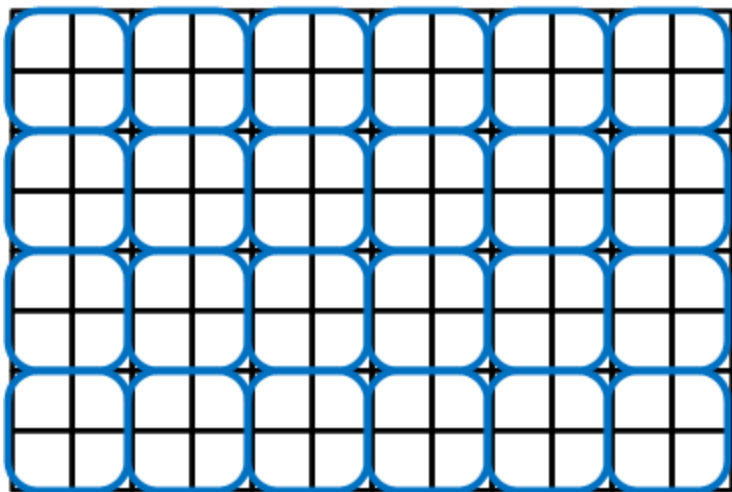
# Overhead examples

Picture size is 4608x3072

12x8 tile structure, so 96 tiles that are 384x384 pixels

Each slice is 2x2 tiles so 24 slices that are 768x768 pixels

Note if there is a flag to assume  
slices are in raster scan order,  
`tile_idx_delta` is not needed



— Tile  
— Slice

With proposed changes

`num_slices_in_pic_minus1` = 23 (9 bits)

`slice_width_in_tiles_minus1` = 1 (3 bits) x 23 = 69 bits

`slice_height_in_tiles_minus1` = 1 (3 bits) x 23 = 69 bits

For tiles not at the right edge (20)

`tile_idx_delta` = 2 (5 bits) x 20 = ~~100 bits~~

For tiles at the right edge except last one (3)

`tile_idx_delta` = 14 (9 bits) x 3 = ~~27 bits~~

Total bits = 274 bits → 148 bits

Current spec

`brick_splitting_flag` = 0 (1 bit)

`num_slices_in_pic_minus1` = 23 (9 bits)

`bottom_right_brick_idx_length_minus1` = 4 (5 bit)

`bottom_right_brick_idx_delta` = x (4 bits) x 23 = 92 bits

`brick_idx_delta_sign_flag` = x (1 bit) x 23 = 23 bits

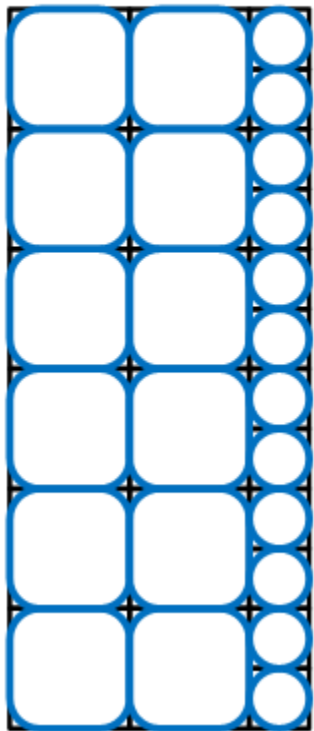
Total bits = 130 bits

# Overhead examples

## OMAF example

Assumed 4K height to calculate brick heights

24 bricks each in its own slice



— Tile  
— Slice

With proposed changes:

$\text{num\_slices\_in\_pic\_minus1} = 23$  (9 bits)

Tiles which are slices (12)

$\text{slice\_width\_in\_tiles\_minus1} = 0$  (1 bit)  $\times 12 = 12$  bits

$\text{slice\_height\_in\_tiles\_minus1} = 0$  (1 bit)  $\times 12 = 12$  bits

$\text{num\_slices\_in\_tile\_minus1} = 0$  (1 bits)  $\times 12 = 12$  bits

Tiles which are split into bricks (6)

$\text{slice\_width\_in\_tiles\_minus1} = 0$  (1 bit)  $\times 6 = 6$  bits

$\text{slice\_height\_in\_tiles\_minus1} = 0$  (1 bit)  $\times 6 = 6$  bits

$\text{num\_slices\_in\_tile\_minus1} = 1$  (3 bits)  $\times 6 = 18$  bits

$\text{slice\_height\_in\_ctu\_minus1} = x$  (3 bits)  $\times 6 = 18$  bits

$\text{tile\_idx\_delta} = 1$  (3 bits)  $\times 17 = 51$  bits

Total bits = 144 bits  $\rightarrow 94$  bits

Note if there is a flag to assume  
slices are in raster scan order,  
 $\text{tile\_idx\_delta}$  is not needed

Current spec:

$\text{brick\_splitting\_flag} = 1$  (1 bit)

Tiles which are slices (12)

$\text{brick\_split\_flag}[i] = 0$  (1 bit)  $\times 12 = 12$  bits

Tiles which are split into bricks (6)

$\text{brick\_split\_flag}[i] = 1$  (1 bit)  $\times 6 = 6$  bits

$\text{uniform\_brick\_spacing\_flag}[i] = 1$  (1 bit)  $\times 6 = 6$  bits

$\text{brick\_row\_height\_minus1}[i][j] = x$  (3 bits)  $\times 6 = 18$  bits

$\text{num\_slices\_in\_pic\_minus1} = 23$  (9 bits)

$\text{bottom\_right\_brick\_idx\_length\_minus1} = 1$  (1 bit)

$\text{bottom\_right\_brick\_idx\_delta} = x$  (1 bits)  $\times 23 = 23$  bits

$\text{brick\_idx\_delta\_sign\_flag} = x$  (1 bit)  $\times 23 = 23$  bits

Total bits = 99 bits



**BROADCOM<sup>®</sup>**

connecting everything<sup>®</sup>