

# JCTVC-H0463: Comments on specifying tile support in HEVC Profiles and Levels

**Wade Wan and Tim Hellman**

Broadcom Corporation



- Parallel techniques (e.g. tiles) targeted towards assisting encoders
- Real-world decoders would not be able to rely on a stream to be encoded in a specific parallelizable fashion:
  - Unless operating in a closed-system
  - Profile/Level constrain streams to that specific parallelizable structure
- Unclear how tiles maps into appropriate Profile/Level definitions
- Determine whether and how tiles would fit within a particular Profile/Level definition where:
  - Every decoder conformant to that definition is required to support decoding of a stream encoded using tiles
  - Note some applications will not use tiles at all

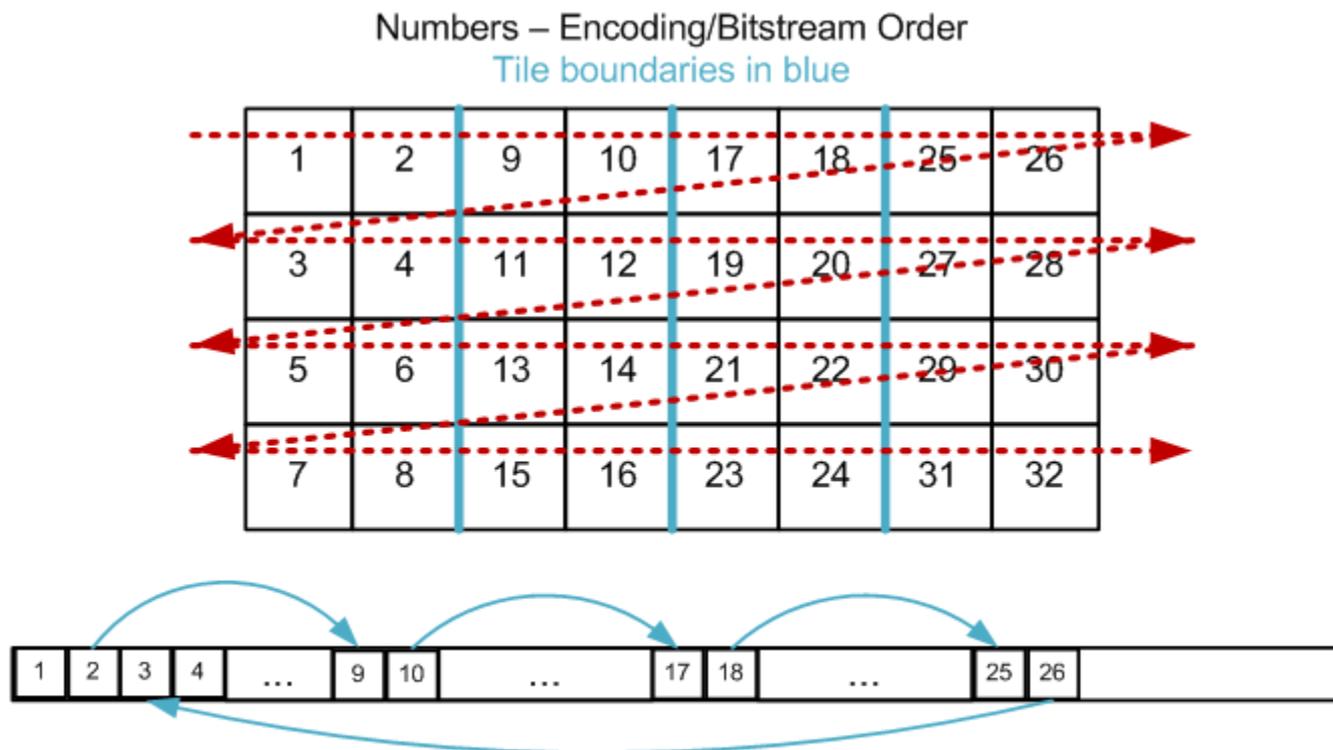
# In-loop filtering across vertical tile boundaries

- Current definition of independent tiles:
  - Entropy and decoding are independent across tile boundaries
  - In-loop filtering does span across tile boundaries so the tiles are not “completely independent”
- Flag can disable/enable in-loop filtering across tile boundary
  - This does not change worst-case requirement for decoders
- Visual artifacts seen when all in-loop filtering (DBLK, SAO, ALF) removed
- Unclear if just deblocking across boundaries is sufficient
- Study whether deblocking is sufficient to eliminate artifacts without additional SAO and ALF processing along tile boundaries

- Unclear whether dependent tiles are really needed as independent tiles (which are simpler to support) may be able to provide sufficient functionality.
  - Private discussion with proponents of original tile proposal indicate they do not anticipate use of dependent tiles in their products
- Prohibit dependent tiles

# Constraining the maximum number of tile columns

- Issues with vertical tile boundaries already highlighted in G110
- Storage, logic and handling that must be performed at each vertical boundary to keep the decoder in the same state/context as the encoder



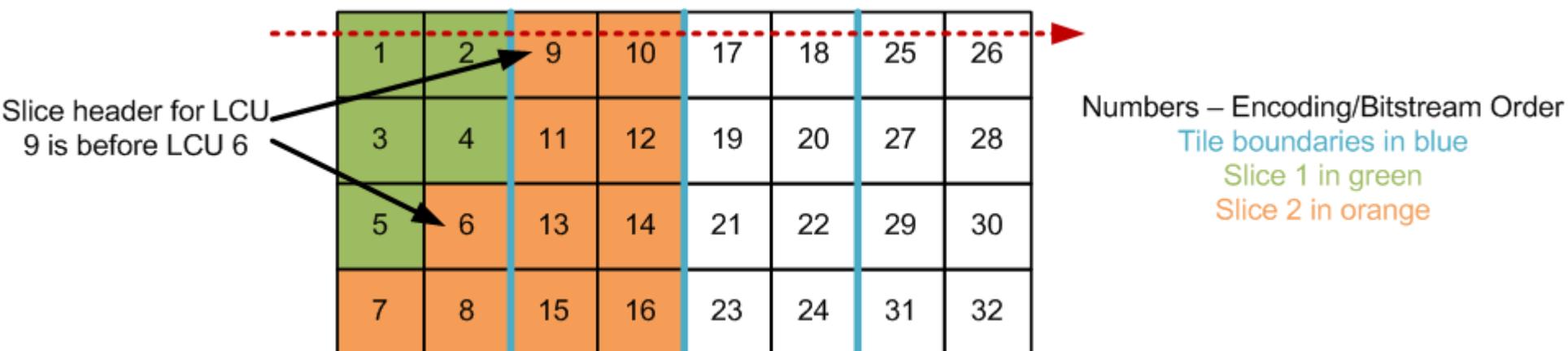
Blue arrows represent bitstream jumps to handle tile transitions to process one LCU row when processing a stream with tiles in LCU raster scan order

# Constraining the maximum number of tile columns

- Save/reload at every vertical boundary (but does not exist for independent tiles)
  - As an aside, similar (but not as prominent) issues for wavefronts
    - Bitstream jumping does not exist if substream interleaving is not used
    - Wavefront schemes do not involve more than one save/reload per LCU row
- **Constrain the maximum number of tile columns to two**

- Tiles and slices are currently defined to be completely orthogonal to one another
- Decoder processing is typically designed to operate in a hierarchical fashion with resources allocated to blocks and then subdivided accordingly.
- The orthogonality of tiles breaks this paradigm and results in inconsistent and sporadic processing affecting efficiency.

- For slices that span tile boundaries:
  - How does one distinguish which slice should be used when crossing across tile boundaries?
  - Bitstream offset pointers? Emulation prevention considerations?



- Restrict slices such that they do not cross tile boundaries. More specifically, every tile boundary shall align with a slice boundary.

- G199 reported improvements of 0.3% compared to tiles for two threads per tile and 0.6% better than tiles for four threads per tile
  - Significant amount of complexity with supporting each of these concepts by themselves
  - Combination of these two techniques presents a very difficult problem for implementers to support all the different possible permutations.
  - Unclear whether significant additional functionality beyond what is supportable with each of the tools themselves is provided by combining them
- Prohibit the combination of tiles and wavefronts

- In-loop filtering across vertical tile boundaries
  - Study whether deblocking is sufficient to eliminate artifacts without additional SAO and ALF processing along tile boundaries
- Dependent tiles
  - Prohibit dependent tiles
- Constraining the maximum number of tile columns
  - Constrain the maximum number of tile columns to two
- Tiles and slices
  - Restrict slices such that they do not cross tile boundaries. More specifically, every tile boundary shall align with a slice boundary.
- Combining tiles and wavefronts
  - Prohibit the combination of tiles and wavefronts

# Thank You

