

# H0348 – Simplification on tiles and slices

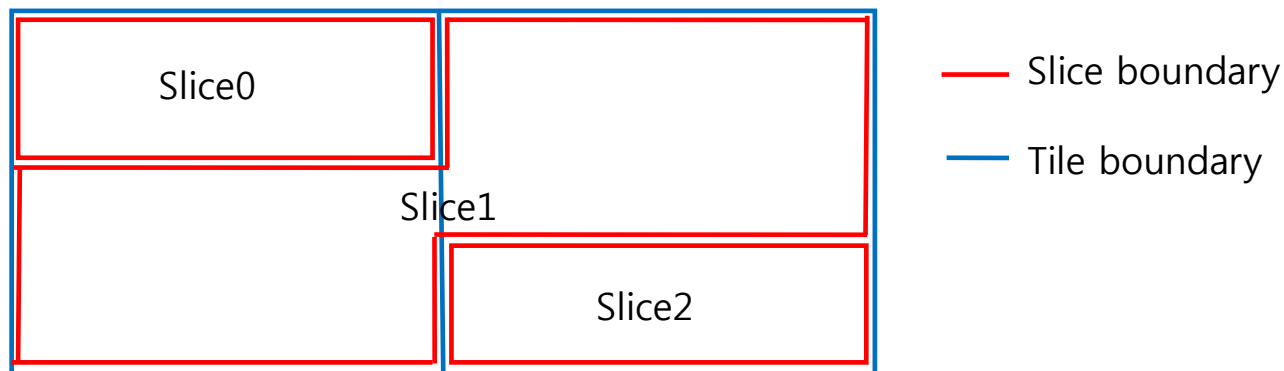
---

Tammy Lee, Byeongdoo Choi, Jeonghoon Park  
Samsung Electronics Co., Ltd

Feb. XX, 2012

# Introduction

- ❖ Picture partitioning schemes in the current HEVC
  - Picture partitioning schemes:
    - Slices, Entropy slices
    - Independent tiles, Dependent tiles
    - Wavefront parallel processing (WPP)
  - Entropy slice is regarded as a kind of slice or a subset of a slice
    - Entropy slice is signalled by short slice header
  - WPP is defined as processing in a tile
  - No restriction between slices and tiles



→ For in-picture prediction and context modeling, every PU, CU or TU needs to check availability of neighbors over both of the slice boundaries and tile boundaries

# Introduction (Cont'd)

## ❖ Properties of slices and tiles in the current HEVC

Feature	Slices	Entropy Slices	Independent Tiles	Dependent Tiles
In-picture prediction	X	O	X	O
Context prediction	X	<b>X</b>	X	<b>O</b>
LCU order changing	X	X	O	O
Packetization	O	O	X	X

- In in-picture prediction
  - Entropy slices = dependent tiles
- Entropy context prediction

	Entropy Slices	Dependent Tiles	
At the beginning	Init probabilities	Inherit probabilities (At the first tile, init probabilities)	Performance difference
At the end	Flushing and Terminating	Flushing (At the last tile, flushing and terminating)	
Over boundaries	Disallowed	Allowed	

Even though dependent tile has benefit in performance,  
this requires extra buffer to store CABAC probabilities on the specified position

... **TEST 1**

# TEST 1

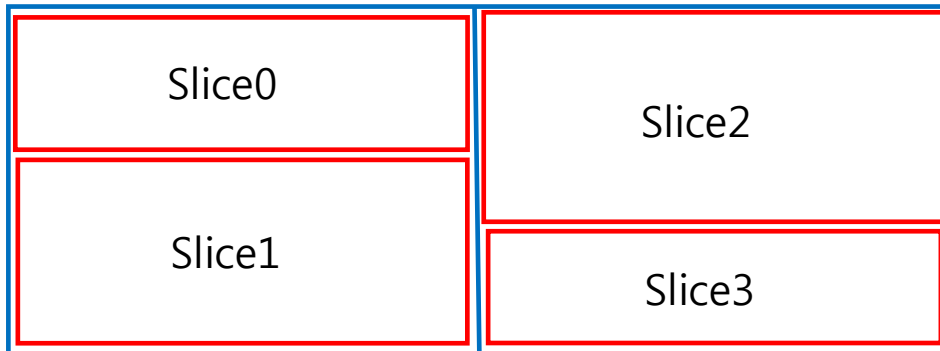
- ❖ Code change
  - To test of efficiency probabilities inheritance at the beginning of dependent tiles,
  - Probabilities at the beginning of dependent tiles are initialized by default
- ❖ Experimental result

	2 Columns	4 Columns
AI-HE	0.0	0.0
RA-HE	0.0	0.1
LB-HE	0.1	0.1
AI-LC	0.0	0.0
RA-LC	0.0	0.1
LB-LC	0.1	0.2
RA-HE10	0.0	0.1
Average	<b>0.0</b>	<b>0.1</b>

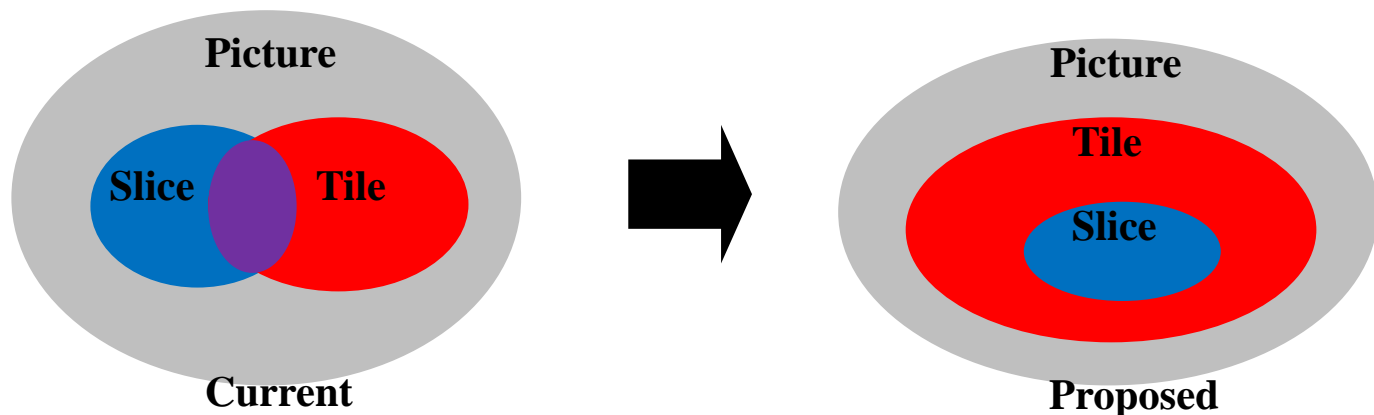
**Performance loss is negligible  
with no requirement of storing  
CABAC probabilities**

# Proposed design

- ❖ Inclusion relation between Slices and Tiles
  - By focusing on the following in terms of properties,
    - Independent tile = (Normal/non-entropy) slice
    - Dependent tile  $\approx$  Entropy slice
  - A Slice is proposed to be inside a Tile



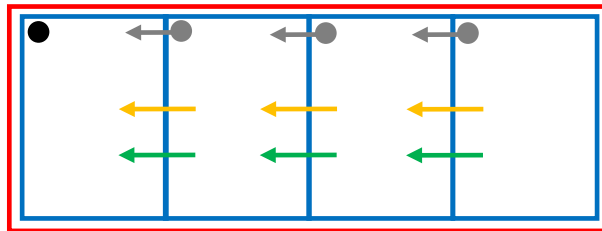
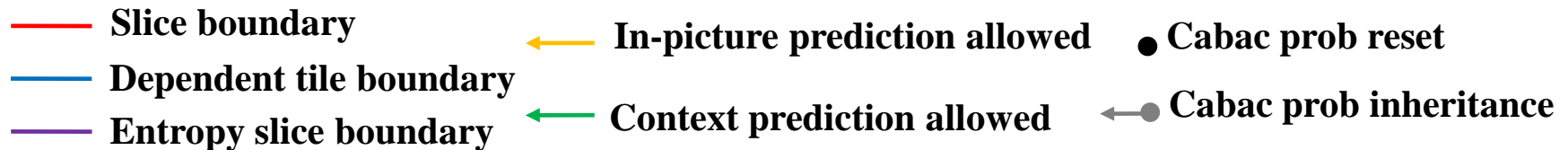
**In the proposed design,  
Slices never cross tile boundaries**



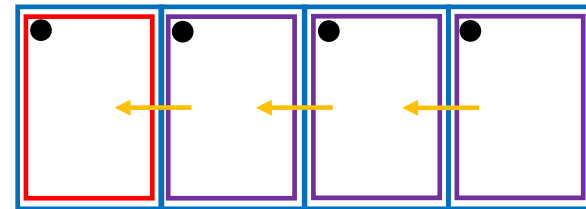
# TEST 2

## ❖ Test configuration

- Since the properties of (non-entropy) slices and independent tiles are same
- And dependent tiles by rows are meaningless
- It is tested how much performance drop can be happened in the following case



Reference



Tested

- Performance loss of tested configuration may be resulted from:
  - Cabac probabilities reset at the first LCU in each dependent tiles
  - Cabac context prediction is disallowed over entropy slice boundaries
  - **Entropy slice header signalling instead of signalling of locations for markers**

# TEST 2 (Cont'd)

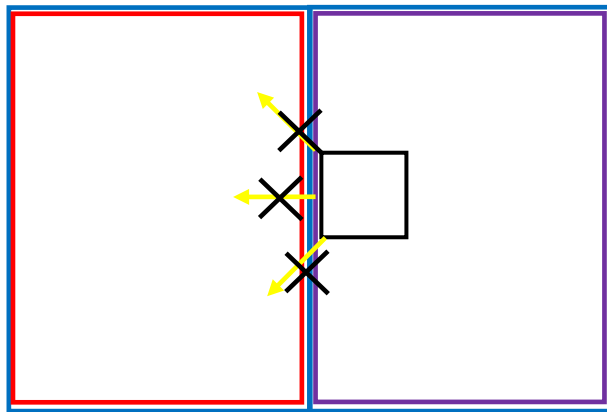
- ❖ Code change
  - Every tile is terminated by entropy slice termination
  - The max number of merge candidates syntax is excluded in the short slice header
- ❖ Configuration difference between reference and tested
  - TileLocationInSliceHeaderFlag and TileMarkerFlag to 0 to exclude location markers without code change in tested while those are set to 1 in reference
- ❖ Experimental result

	2 Columns	4 Columns
AI-HE	0.0	0.0
RA-HE	0.2	0.7
LB-HE	0.3	0.9
AI-LC	0.0	0.1
RA-LC	0.3	0.8
LB-LC	0.3	1.0
RA-HE10	0.2	0.7
Avg. All	<b>0.2</b>	<b>0.6</b>
Avg. ABC	<b>0.1</b>	<b>0.4</b>

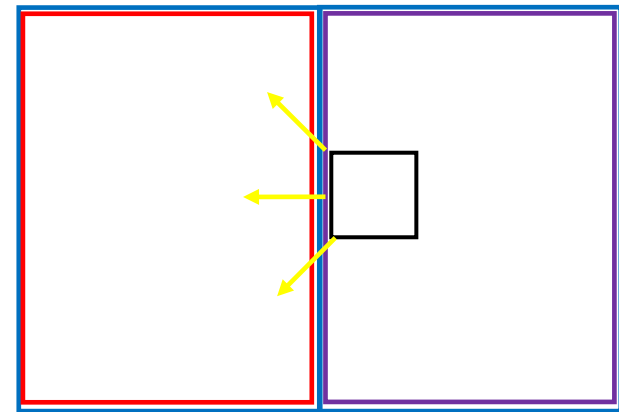
**The larger loss  
in the smaller resolution and  
when the number of tiles is increased**

# TEST 2 (Cont'd)

- ❖ Main reason of loss in the current SW (Not applied in Test 2)
  - Bug in the current reference SW
    - LCU address should be changed to be ordered within a tile when tiles are used with multiple columns, but in the software, LCU address is still raster scan address not within a tile but within a picture
    - In-picture prediction on the right LCUs to the dependent tile boundaries doesn't work when getting neighboring information to the left which is belong to the different tile



Problem in the reference SW



To be fixed

**If this bug is fixed, loss of the proposed design will be reduced while the performance of the reference SW remains the same with the tested configuration**



# TEST 2 (Cont'd)

- ❖ Possible future change to reduce loss (Not applied in Test 2)
  - Removing tile\_boundary\_independence\_flag
    - Separation of dependent and independent tiles  
→ encoder side configuration and realized by entropy and non-entropy slices
  - Remove loop\_filter\_across\_tile\_flag
    - Tile boundary is the subset slice boundary →  
loop\_filter\_across\_tile\_flag can be replaced by loop\_filter\_across\_slice\_flag
  - Location markers for low latency processing can be replaced by slice header (Tested by configuration change in TEST 2)

# Summary

- ❖ By changing Cabac inheritance to Cabac probabilities reset at the first LCU in each dependent tiles,
  - Loss is negligible and cabac probabilities is not needed to be saved
- ❖ By enforcing each tile to be terminated by a slice,
  - Concept of slice and tile can be cleaned up in relationship.
  - Any boundary related issue is simplified by checking only slice boundary in such cases of availability checking for in-picture prediction and context prediction
- ❖ It is recommended to remove separation between independent and dependent tiles and make slices included within a tile

Thank you !