

H0349 – Simplification on WPP

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

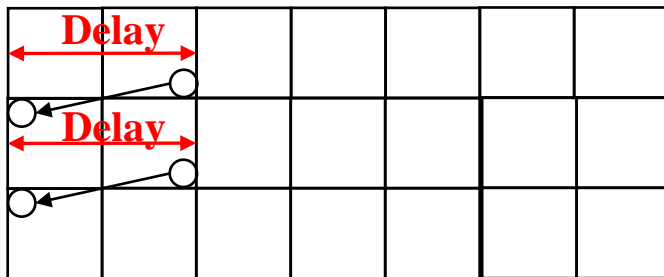
Feb. XX, 2012

CABAC Synchronization (1)

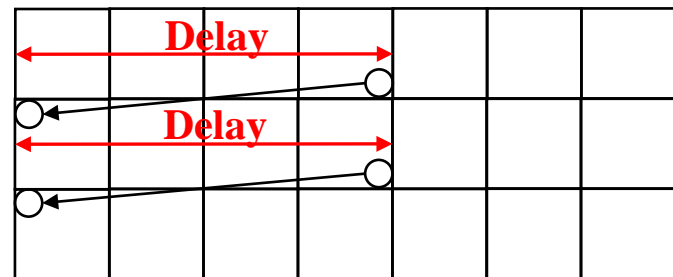
❖ CABAC synchronization in WPP

- WPP consists of synchronizing the CABAC probabilities of the first LCU in each line from the specified LCU of the line above
- Synchronization offset to get position of the specified LCU is signalled named as `entropy_coding_synchro`
- `entropy_coding_synchro` gives flexibility to find and use the better probability
- Latency for starting the first LCU increases as `entropy_coding_synchro` increases

Cabac synchro by 1



Cabac synchro by 3



CABAC Synchronization (2)

❖ Experimental result

- The following table shows the performance when WaveFrontSynchro is changed from 1 to 2 for 2 substreams in test configuration setting

| | WaveFrontSynchro 1 vs 2 | | |
|---------|-------------------------|-----|------|
| | Y | U | V |
| AI-HE | 0.0 | 0.0 | 0.0 |
| RA-HE | 0.0 | 0.0 | 0.0 |
| LB-HE | 0.0 | 0.0 | 0.0 |
| AI-LC | 0.0 | 0.0 | 0.0 |
| RA-LC | 0.0 | 0.0 | 0.1 |
| LB-LC | -0.1 | 0.1 | -0.1 |
| RA-HE10 | -0.1 | 0.1 | 0.1 |
| Average | 0.0 | 0.0 | 0.0 |

There is almost no performance difference with change of cabac synchronization positions

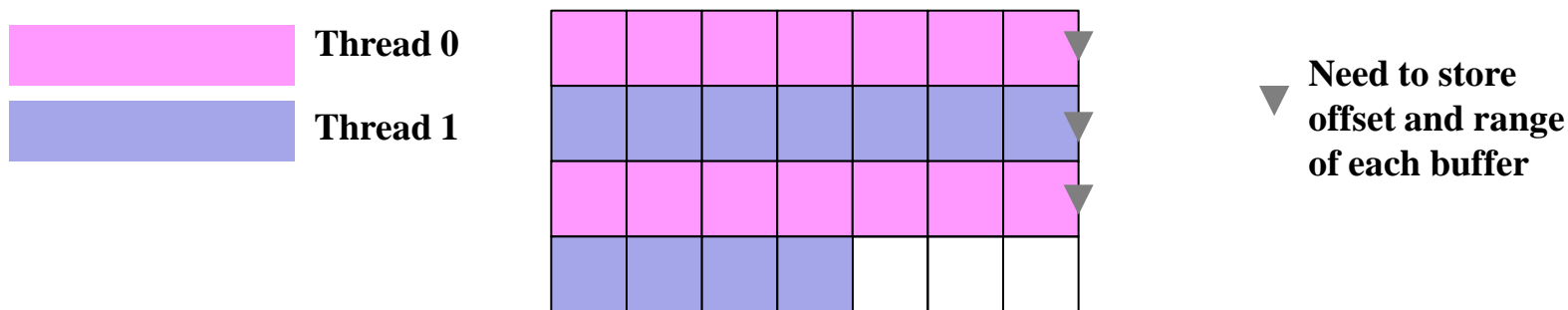
Flushing/Non-flushing (1)

❖ Flushing mode in WPP

- A flush and re-initialization of the internal state variables of CABAC at the end of each line of LCUs
- Each line of LCUs is compressed into a "chunk" of bits that is independent from the desired level of parallelism

❖ Non-flushing mode in WPP

- Non-flush mode have better performance compared to flush mode due to no flush at the end of LCU line
- Non-flush mode has disadvantages in single core encoding and decoding
 - It needs to store the offset and the range values of each buffer in each thread at the end of each LCU line to switch to another thread
 - Since each line's information cannot be delimited, the amount of bitstream to be loaded is not neat
→ overloaded with the redundant information of another line's information within a thread



Flushing/Non-flushing (2)

❖ Experimental result

- The following table shows the performance when NoFlush mode is changed to Flush mode for 2 substreams in test configuration setting

| | NoFlush vs Flush | | |
|----------------|------------------|------------|------------|
| | Y | U | V |
| AI-HE | 0.0 | 0.0 | 0.0 |
| RA-HE | 0.2 | 0.3 | 0.2 |
| LB-HE | 0.4 | 0.5 | 0.2 |
| AI-LC | 0.0 | 0.0 | 0.0 |
| RA-LC | 0.2 | 0.1 | 0.2 |
| LB-LC | 0.4 | 0.5 | 0.4 |
| RA-HE10 | 0.2 | 0.3 | 0.1 |
| Average | 0.2 | 0.3 | 0.2 |

Summary

- ❖ By changing Cabac synchronization position by 1, there is no performance loss
 - Adaptive cabac synchronization position has benefit only in performance in design
 - but it turns out it shows almost no change with additional delay requirement
- It is recommended to remove Cabac synchronization adaptation by using only 1

- ❖ By changing non-flush mode to flush mode, there is only 0.2 performance loss in average
 - Non-flush mode has benefit only in performance in design
 - but it shows negligible performance loss with requirement to store offset and range and to overload bitstream in single core design
- It is recommended to remove non-flush mode to simply WPP implementation

Thank you !