Implication of parallelized bitstreams on single core decoder architectures (JCTVC-G110)

Wade K. Wan & Tim Hellman

Broadcom Corporation

Overview

- Several methods proposed and adopted in previous meetings to enable parallelism for multi-core architectures
- Redundant ? Possibly unified?
- Single core decoder architectures may not benefit and in fact, may be penalized by having to support some of these parallelism techniques.
- This contribution discusses the impact caused by supporting these methods on single core hardware decoder designs
- Provided to aid effort to better understand the trade-offs between different techniques to enable parallelism.

Regular Slices

- **No addition data buffering or DRAM storage is required.**
- **Slices can have an impact on hardware decoder performance:**
  - Large amount of data needs to be reloaded on every slice
  - Issue when slice rate (number of slices per picture) grows too high
- **Can generally be addressed with appropriate level limits in the standard, as was done for AVC with its slice rate limit.**

Entropy Slices

- CABAC context to be reinitialized at each entropy slice
- Minor impact unless there are a very larger number of entropy slices (similar to regular slices)
- Can be addressed with entropy slice rate limits.

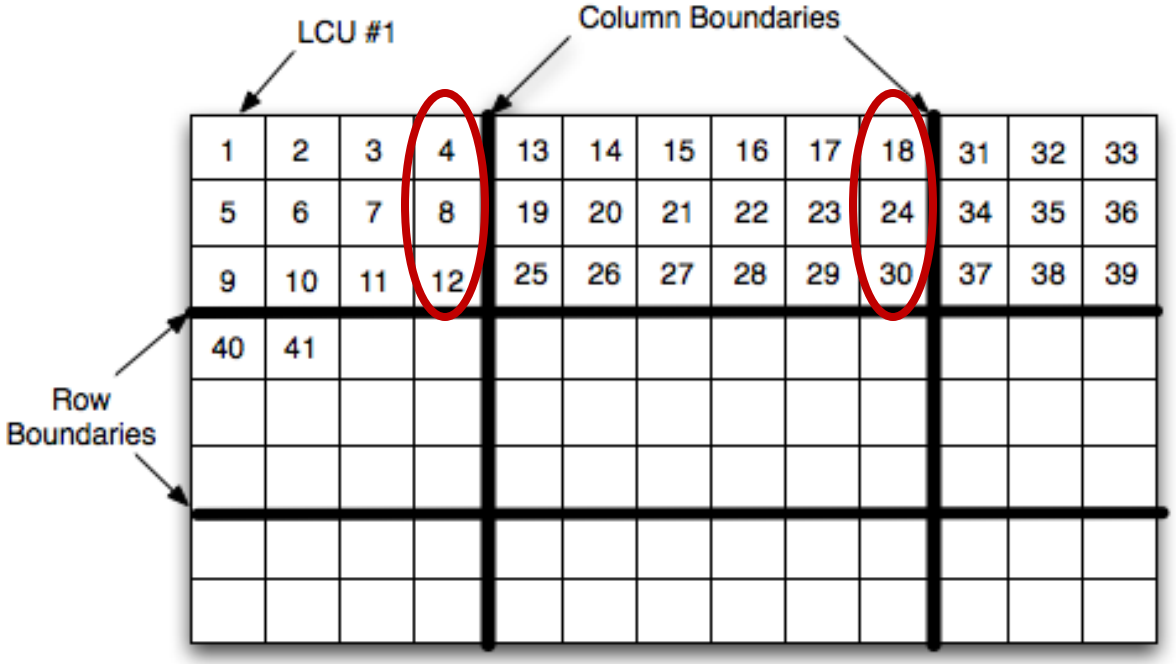
Wavefront Parallel Processing

- **CABAC state is saved from one substream and restored at the start of the next row to initialize next substream**
- **Impact is minor :**
  - Additional RAM (~512 bytes) needed to store snapshot
  - Processing overhead of restoring snapshot is similar to reinitialization for entropy slices
- **Requires multiple stream pointers to handle stream data ordering**
  - Processing affected by maximum number of substreams and interleaving
- **Can be addressed with limits on maximum substreams and frequency of interleaving**

Tiles

- **Two variants that have different impact to single core HW designs**
  - The *tile_boundary_independence_idc* flag controls whether tile-crossing coding dependencies are permitted or not
  - Appears that loop filtering is unaffected by this flag
- **Allowing any dependencies across vertical tile boundaries presents an extraordinary difficulty for a single core hardware decoder**
  - Single core decoders are designed to process in raster-LCU order
  - Top and left neighbor information is needed for various processing stages and stored (often on-chip)
  - Vertical tile boundaries with dependencies destroys this neighbor storage paradigm

Tiles



- **Long-term storage must be found for all the horizontally adjacent neighbors on the boundary**
  - Special on-chip storage or DRAM storage costly
- **Tile coding maps well to a single core hardware decoder design if all tile crossing dependencies are forbidden.**
  - In this case, each tile can be decoded as if it were a separate picture

Conclusions

- **Need to further study and understand the implications of the different techniques considered to enable parallelism**
  - Contribution discusses impact on single core hardware decoder designs
- **Regular/Entropy slices (minor impact)**
  - Does not require significant additional architecture
  - Slice rate limit would help constrain worst case
- **Wavefront parallel processing (minor impact)**
  - Limits on maximum substreams and frequency of interleaving to bound processing required for stream data ordering
- **Tiles (impact can be minor or significant!)**
  - Inter-tile coding dependencies are key aspect to consider
  - If not permitted, a single core HW design can support without significant additional implementation burden by processing tiles as “subpictures”
  - If permitted, difficultly mapping general tile methodology efficiently to a single core HW design