Entropy Coders:
How many do we need in HEVC?

Ken McCann

(Samsung/ZetaCast)

Architectural View of Entropy Coding

- Best architecture is a single entropy coder
  - One tool per function unless there is a very good reason to do otherwise
  - Clean standard with no entropy-related interoperability problems
- Second best is a scalable entropy coder
  - Relatively clean standard
  - Manageable interoperability issues (e.g. onion ring profile structure)
- Third best is to switch between two entropy coders
  - Many decoder ICs will have to implement both entropy coding schemes
    - more complex design than either one alone
    - increased cost of verification testing
  - Less elegant standard
    - Increased risk that application-focused standardisation bodies (e.g. DVB) define their own application-specific profiles
  - More effort required during standardisation work
    - Many more lines of code and more pages of documentation to maintain

So why do we have third best in HM?

- Historical Reasons
  - JCT-VC participants are familiar with AVC architecture
    - Made sense at the time it was developed
  - Step change in entropy coding performance became possible (but difficult) to implement when AVC was developed 10 years ago
    - We are now 5 cycles of Moore’s Law later - a factor of 32
- HM entropy coders grew out of TMuC compromises
  - Performance/complexity trade-off points offered by the two entropy coders have subsequently converged
- Now is the right time to perform a sanity-check on the HM entropy coding architecture

Suggested way Forward

- Agree principle of “only one entropy coder unless there is a very good reason to do otherwise”
- Agree to determine whether or not there is a very good reason
  - Decision should be based on verifiable facts
- Set up a CE to gather information and perform preliminary analysis to enable decision at next meeting
  - Or could be AHG if the tasks cannot be sufficiently well defined this week
- After deciding on architecture, then decide on solution