COMPLEXITY ANALYSIS OF NEXT-GENERATION VVC ENCODING AND DECODING

Session: SS-11 -- Complexity Reduction and Real Time Implementations of the Versatile Video Coding Standard

Farhad Pakdaman[1], Mohammad Ali Adelimanesh[1], Moncef Gabbouj[2], and Mahmoud Reza Hashemi[1]

[1]School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran
[2]Department of Computing Sciences, Tampere University, Tampere, Finland

University of Tehran

M.P.L
MULTIMEDIA PROCESSING LABORATORY

Tampere University

AbuDhabi ICIP 2020

Versatile Video Coding (VVC)

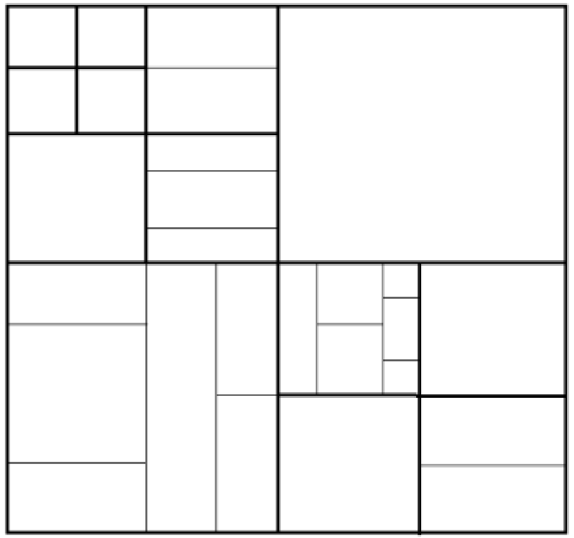
- VVC (H.266), developed by JVET
  - Growth in spatial, temporal, bit depth, and view dimensions
  - Various devices should support HD video
- Supports different content and applications
  - SDR and HDR
  - Camera and computer generated content
  - 360°, light-field, depth, and volumetric video
- Finalized on July 2020
  - ~40% bitrate saving compared to HEVC

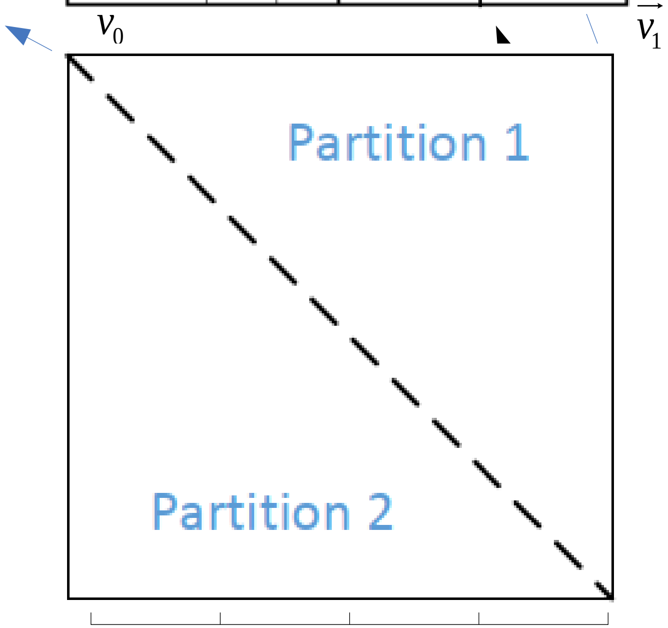
Complexity analysis of VVC

- High computational complexity for encoding and decoding
  - Real time implementation
  - Low power consumer devices
  - Complexity analysis to measure the design requirements
  - Measure the memory bandwidth requirements

Overview of VVC coding tools

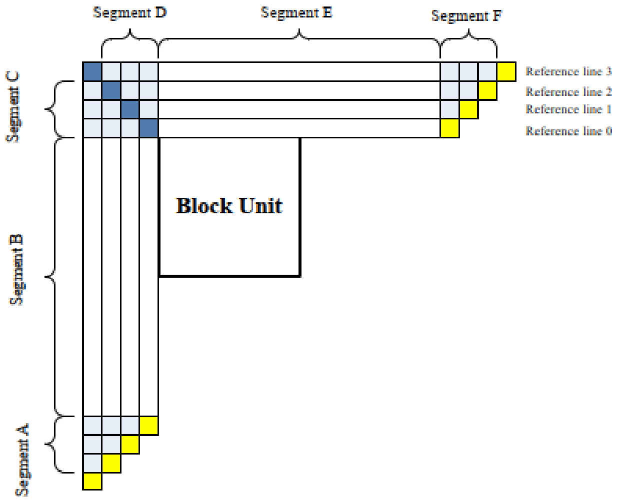
- Block partitioning
  - CTUs as large as 128×128
  - Quad-tree with nested multi-type tree partitioning
- Motion estimation
  - Motion signaled explicitly, skip, or merge
  - Merge mode with motion vector difference (MMVD)
  - Symmetric MVD
  - Affine motion compensation for merge and AMVP
  - Adaptive MVD resolution (1/16 to 4 luma-samples)
  - Triangle partition for inter prediction

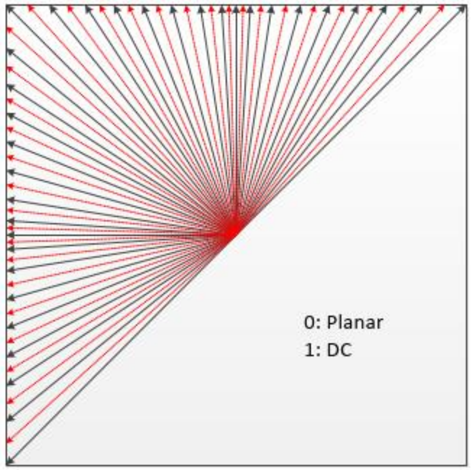




Overview of VVC coding tools (cont.)

- Intra prediction
  - 67 intra modes
  - Non-square blocks and wide-angle modes
  - Multiple reference lines
- Transform
  - Up to 64×64 for luma and 32×32 for chroma
  - High-frequency zeroing
  - Multiple transform selection
- Loop filters
  - Deblocking Filter (DBF)
  - Sample Adaptive Offset (SAO)
  - Adaptive Loop Filter (ALF)



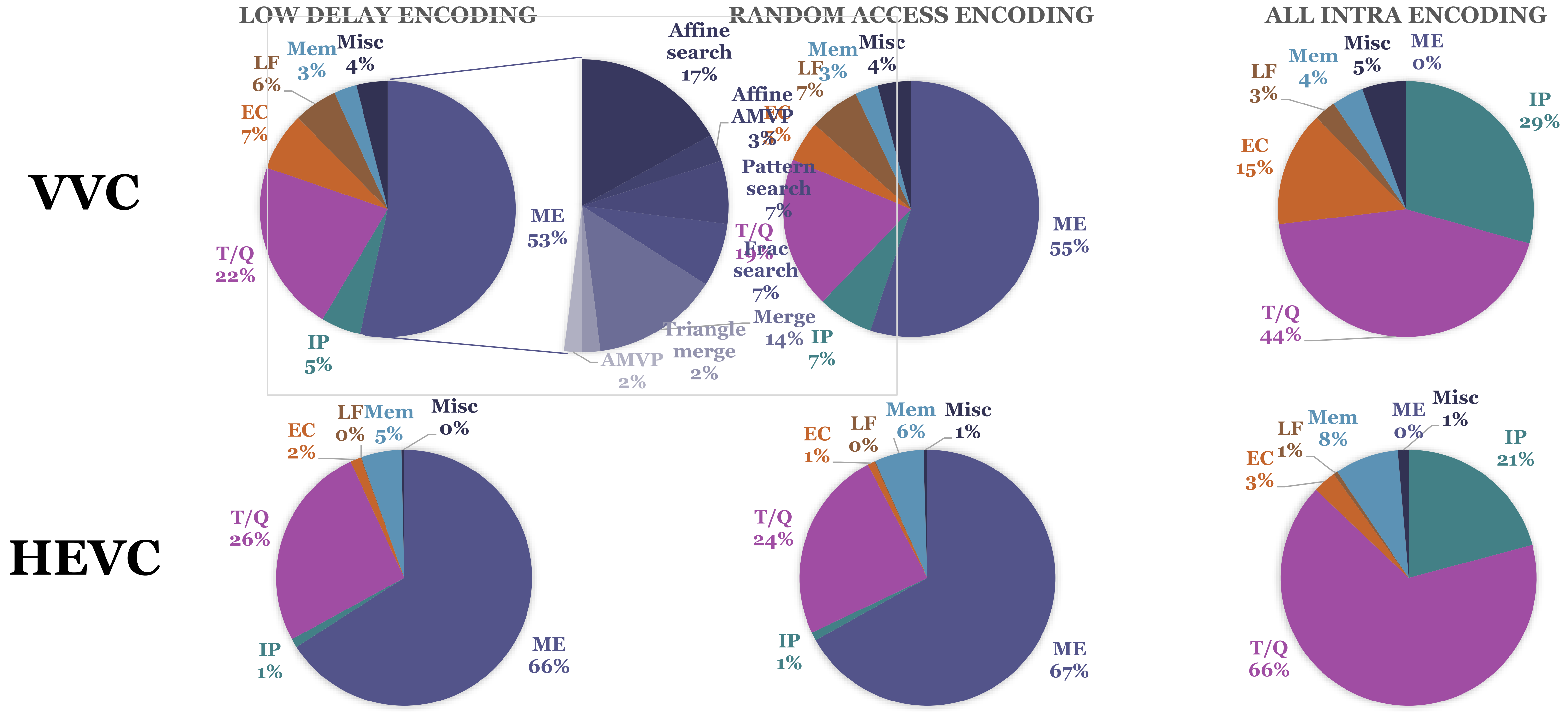


Test environment

- VTM 6 and HM 16 as VVC and HEVC codecs
  - Low Delay (LD), Random Access (RA), and All-Intra (AI) configs
  - QPs 22, 27, 32, and 37
- Machine configurations
  - Single thread, Intel core i7 4790K, 4GHz
  - 8 GBs memory
  - Windows 10
- Test video sequences
  - Six sequences from 720p to 2160p
- Complexity analysis
  - Intel VTune™ Amplifier 2019 for
  - Hotspot analysis to measure complexity of each coding tool
  - Memory bandwidth analysis

Sequence	Resolution	Frame rate	Bit depth	Tested configs
BQTerrace	1920×1080	60	8	AI, LD, RA
BasketballDrive	1920×1080	50	8	AI, LD, RA
Cactus	1920×1080	50	8	AI, LD, RA
Johnny	1280×720	60	8	AI, LD, RA
KristenAndSara	1280×720	60	8	AI, LD, RA
DaylightRoad2	3840×2160	60	10	AI, LD

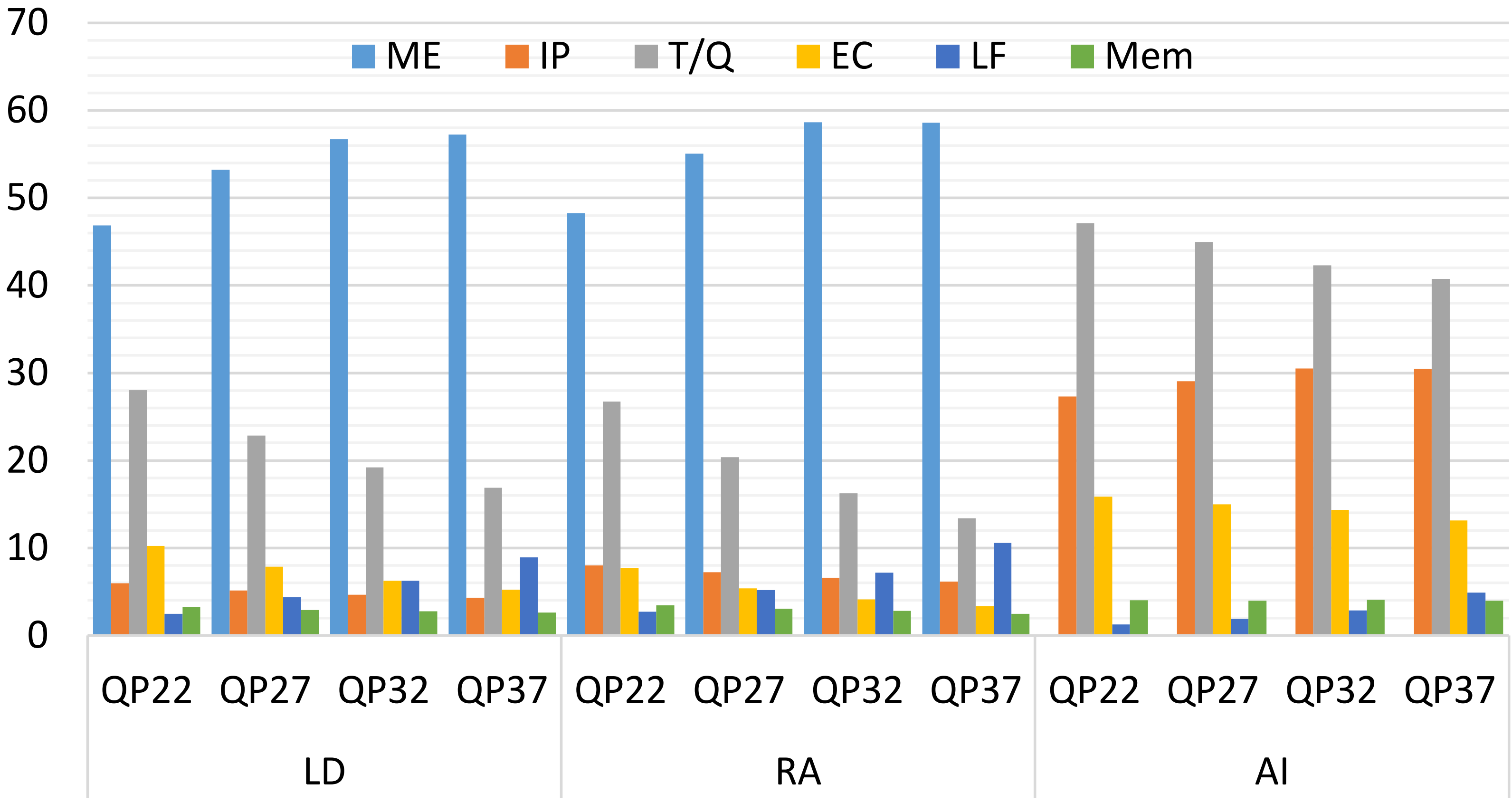
Complexity break-down of encoding



Complexity break-down of encoding

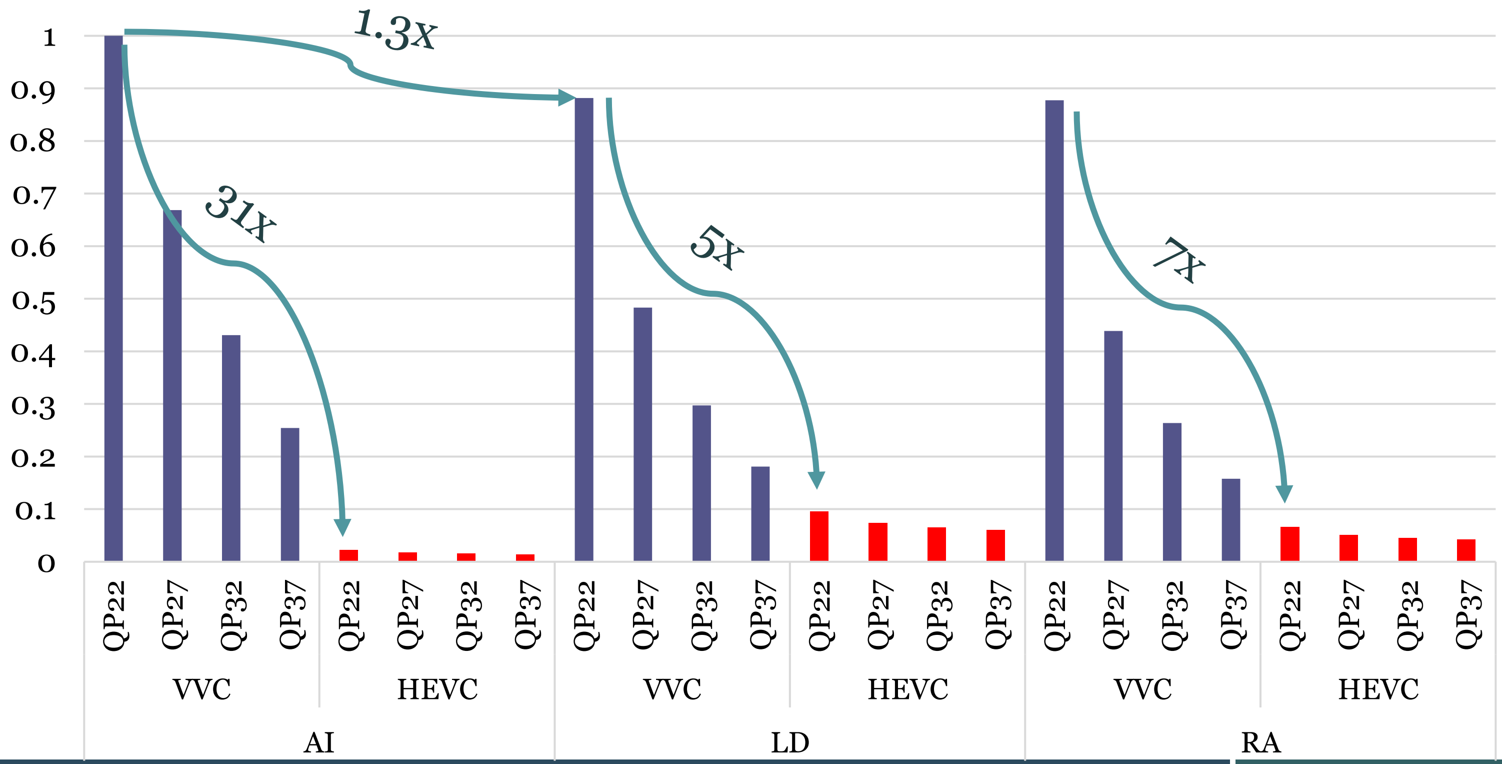
- Increasing the QP will:
  - Increases ME, and LF
  - Decreases T/Q and EC
- Recursive calls over CTU structure
  - Reduce max coding depth (QT+MTT depth)

Max Depth	1	2	3	4
Complexity reduction	82%	76%	58%	26%



VVC vs. HEVC encoding

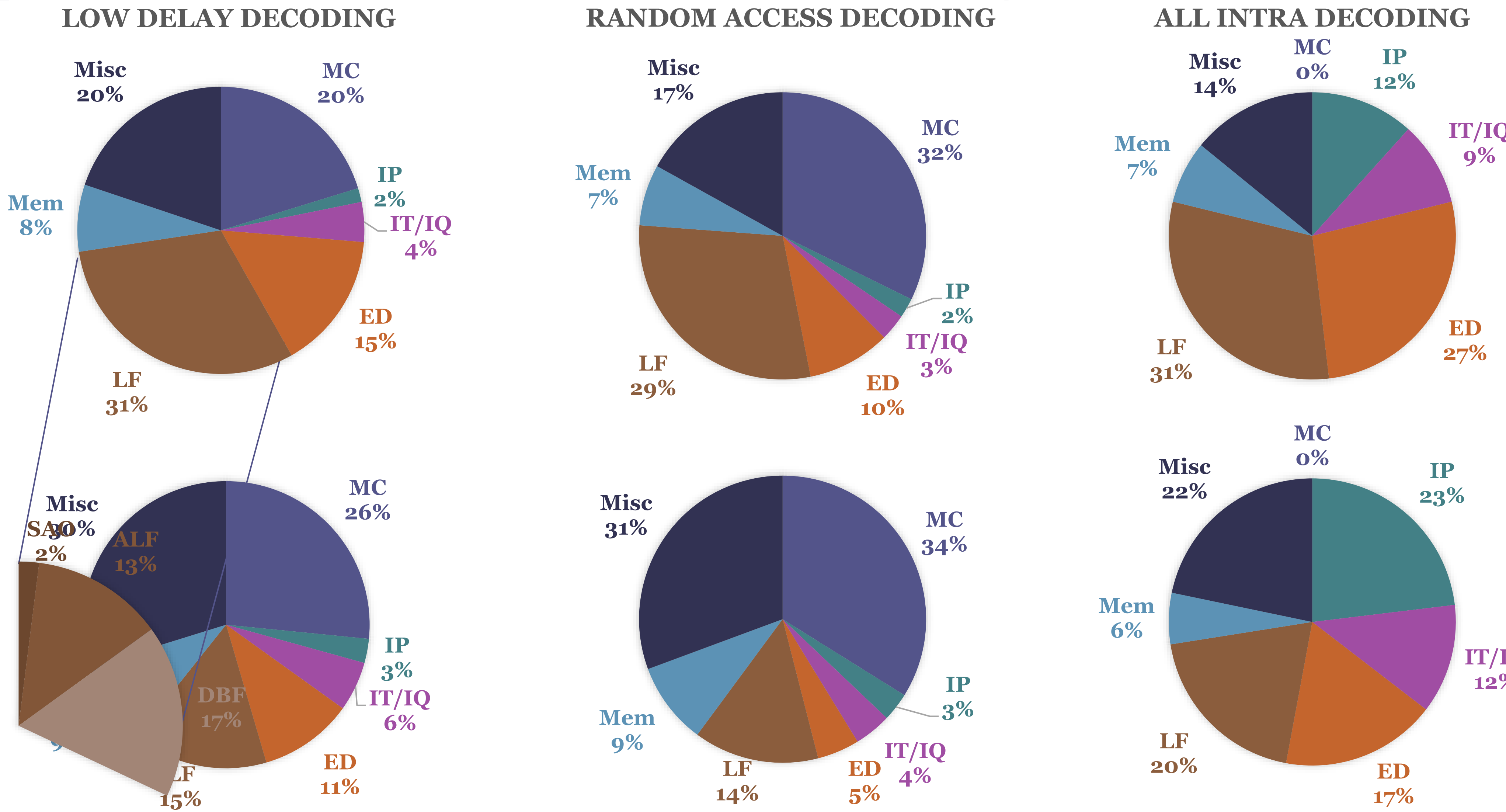
- Normalized complexity
- LD, RA, and AI are 5x, 7x, and 31x of HEVC encoding
- AI takes 1.3x and 1.4x of LD and RA
- VVC encoding is more QP-dependent (higher quality requires more power as well as higher bitrate)
  - QP=22 is 4.8x of QP=37
  - 1.6x for HEVC



Complexity break-down of decoding

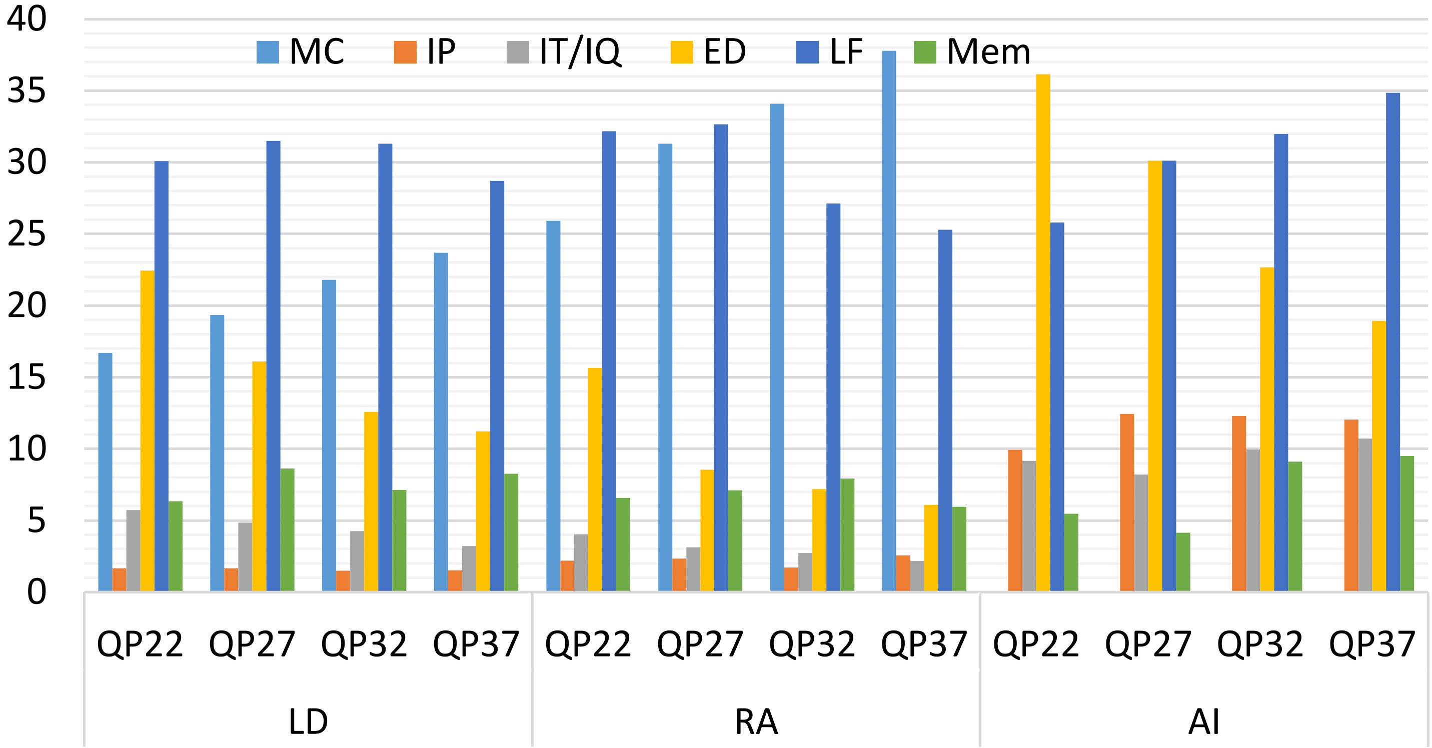
VVC

HEVC



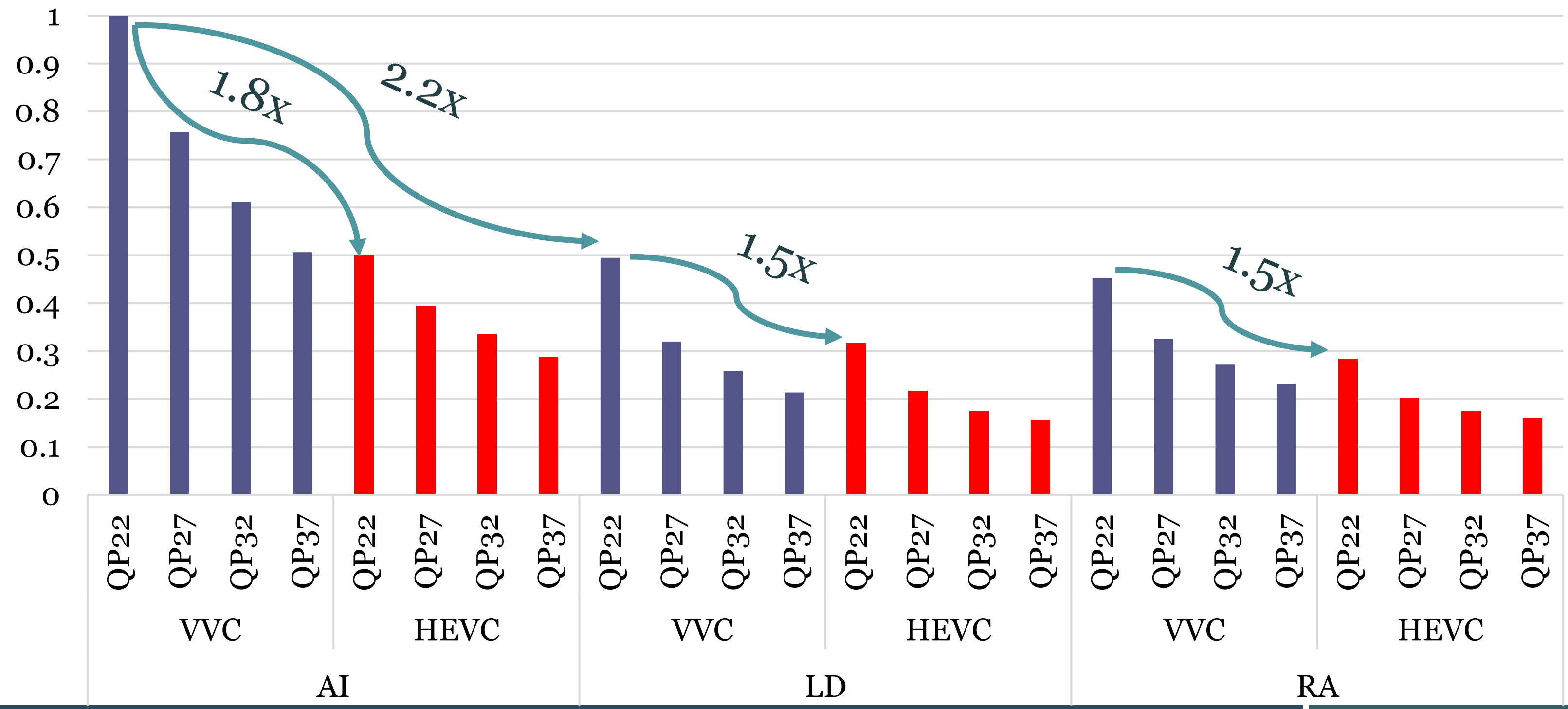
Complexity break-down of decoding

- Increasing the QP will
  - Increases MC
  - Decreases ED and IT/IQ



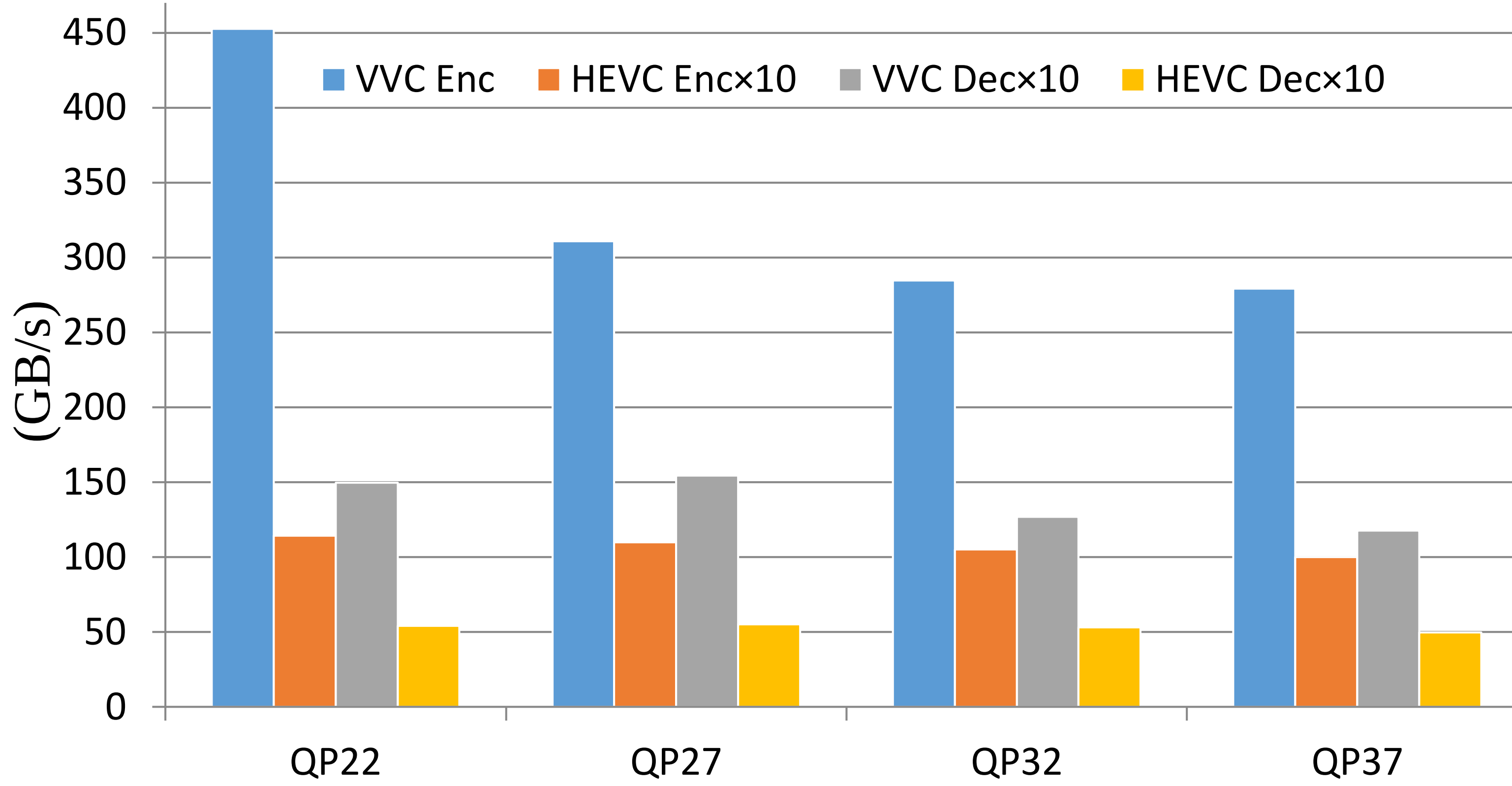
VVC decoding complexity vs. HEVC

- Normalized complexity
- LD, RA, and AI are 1.5x, 1.5x, and 1.8x of HEVC decoding
- AI in VVC takes 2.2 of LD and RA
- VVC decoding is also more QP-dependent
  - QP=22 is 2.3x of QP=37
  - 2x for HEVC



Memory bandwidth requirements

- For 1080p and LD
- Scaled to real-time enc/dec
- 30x and 3x HEVC enc/dec
- Larger maximum CU size, and several prediction candidates



- [22] reports 14.83 GB/s for encoding
  - Reduces to 2.97 via efficient memory sub-system

Conclusion

- New coding tools require new complexity reduction techniques
  - Affine ME, triangle mode, multiple transform cores, ALF, etc.
- Some existing solutions for HEVC can be adapted/extended for VVC
  - Intra prediction [25][26], CTU partitioning [24], fast mode decision [27][28]
  - Parallel processing and pipelining for efficient decoding [31][32]
  - Low-power implementations for ALF, interpolation filter, etc. [29][30]
- Memory access reduction
  - Memory-efficient algorithms [21], data-reuse [23], reference frame compression [20], efficient memory sub-systems [22]

THANKS FOR WATCHING

Questions?

Please email us:
farhad.Pakdaman@ut.ac.ir

Please visit our Lab page for other relevant researches in this area:

https://www.researchgate.net/lab/Multimedia-Processing-Laboratory-MPL-Mahmoud-Reza-Hashemi