Video on Mobile CPU: UHD Video Parallel Decoding for Asymmetric Multicores

Yeongil Ryu
Department of Computer Engineering
Gachon University, Korea
yiryu@gc.gachon.ac.kr

Eun-Seok Ryu
Department of Computer Engineering
Gachon University, Korea
esryu@gachon.ac.kr

ABSTRACT

This paper proposes a novel High Efficiency Video Coding (HEVC) Tile partitioning method for parallel processing by analyzing the computing ability of asymmetric multicores. The proposed method (i) analyzes the computing ability of asymmetric multicores and (ii) makes a regression model of computational complexity per video resolutions. Finally, the model (iii) determines the optimal HEVC Tile resolution for each core and partitions/allocates the Tiles to suitable cores.

The proposed method minimizes the decoding time gap between faster CPU cores and power-efficient cores (big/LITTLE cores). Experimental results with 4K ultra-high definition (UHD) test sequences show an average improvement of 25% in decoding speed for most recent Android smart phones.

CCS CONCEPTS

• **Computing methodologies** → **Parallel algorithms**; **Image compression**; *Image processing*;

KEYWORDS

HEVC; Parallel video processing; Asymmetric multicores; Tile

ACM Reference format:
Yeongil Ryu and Eun-Seok Ryu. 2017. Video on Mobile CPU: UHD Video Parallel Decoding for Asymmetric Multicores. In *Proceedings of MMSys'17, Taipei, Taiwan, June 20-23, 2017,* 3 pages.
https://doi.org/http://dx.doi.org/10.1145/3083187.3083229

1 INTRODUCTION

In recent years, parallel ultra-high definition (UHD) video processing has emerged as a preferred technology and the usage of the computing systems that have asymmetric multicore processor such as ARM big.LITTLE is actively increasing [1].

A new international video standard High Efficiency Video Coding (HEVC) provides two new parallel processing tools employing different picture partitioning strategies such as Tiles and Wavefront Parallel Processing (WPP) [3, 9]. Tiles partition a picture with horizontal and vertical boundaries so that it provides better coding gains compared to multiple slices. However, it does not take into account computational abilities of asymmetric CPU cores such as ARM's big/LITTLE cores, and divides a picture into a grid of equal-sized rectangular regions. This results in performance degradation of multicore parallel processing. Thus, this paper proposes a new HEVC Tile partitioning method for parallel processing by analyzing the computing ability of asymmetric multicores as well as the computational complexity of each Tile. In addition, this paper demonstrates the results of the study by implementing the proposed method on Samsung Galaxy S7 Edge, a smartphone introduced in the market in the recent past.

2 VIDEO PARALLEL PROCESSING USING THE PROPOSED NON-UNIFORM TILE PARTITIONING METHOD

On asymmetric multicore systems, the conventional uniform Tile partitioning method causes performance bottlenecks, because the faster decoding threads(on big cores) are forced to wait for slower decoding threads(on little cores) to finish decoding of each picture. This paper proposes a method to minimize the relative workload gap between the cores to minimize the performance bottlenecks.

Diverse researches have been conducted in the parallel video processing field to equalize the relative workload of each core. One of the researches proposes the HEVC Tile partitioning algorithm by estimating decoding complexities. The method counts the encoded bits of each coding tree unit (CTU) and segments multiple Tiles by distributing workloads to multicores as uniformly as possible. It is quite practical, but it does not consider the asymmetric multicore environments. Hence, this paper focuses on relative workload equalization for asymmetric multicore systems. The proposed method works on the concept of dividing video pictures into multiple non-uniform Tiles and allocating them to big and little cores that have asymmetric performances. Figure 1 depicts the concept of the mapping HEVC Tiles onto multiple cores.

Among many factors affecting video decoding complexity, this paper focuses on the resolution of each Tile to estimate the decoding complexity [2]. The proposed Tile partitioning method involves the following processing steps: (i) Analyze the computational ability of asymmetric multicores and (ii) Apply the pre-defined regression model [5–8] for a computational complexity per video resolutions. (iii) Determine

MMSys'17, June 20-23, 2017, Taipei, Taiwan

ACM ISBN 978-1-4503-5002-0/17/06.
https://doi.org/http://dx.doi.org/10.1145/3083187.3083229

the optimal HEVC Tile resolution for each core. (iv) Partition/Allocate Tiles to the best cores as shown in Figure 2.

The proposed method does not work for pre-encoded videos and broadcast systems, which does not take into account each decoder side. However, the proposed method works for real-time video communication systems such as video conference applications and first person view (FPV) video streaming systems on unmanned aerial vehicles (UAV), which is a key differentiator over competing methods because it allows encoders of the video systems to employ particular non-uniform Tile partitioning options by taking into account environment of the decoder side real-time.

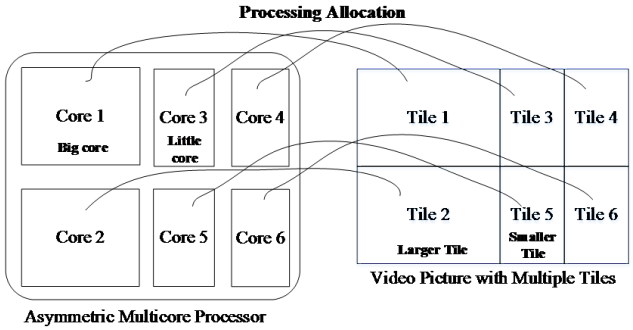

Figure 1: The concept of the mapping relationship between video Tiles and asymmetric multicores (big and little.)

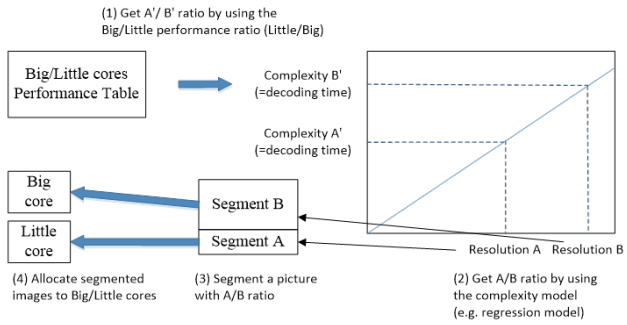

Figure 2: The procedure of the proposed Tile partitioning method.

3 IMPLEMENTATION

HM15.0 encoder, HEVC reference software, and JCT-VC official two UHD test sequences which are *PeopleOnStreet* (3840 × 2160) and *Traffic* (3840 × 2048) are used for encoding. The encoding options are as shown in Table 1. The internal option *TileUniformSpacing* is set to value '0' for non-uniform Tile partitioning. *TileColumnWidthArray* and *TileRowHeightArray* options are used to adjust resolutions of each Tile. Figure 3 and 4 show the partitioned Tiles using conventional and proposed methods.

Table 1: Coding Options for Demonstration

Coding option	Parameter
Coding structure	Random Access (RA) All Intra (AI) Low-Delay B (LDB)
QP	22, 27, 32, 37
Number of Tiles	6 (3 × 2)

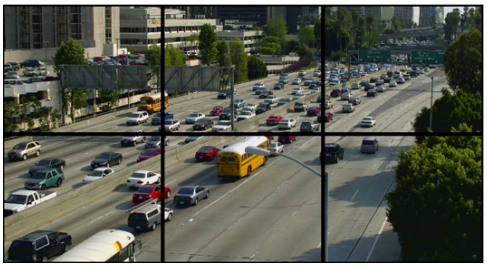

Figure 3: Conventional uniform Tile partitioning method considering Samsung Galaxy S7 Edge environments.

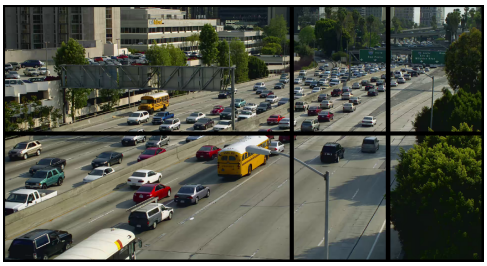

Figure 4: Proposed non-uniform Tile partitioning method considering Samsung Galaxy S7 Edge environments.

The proposed method is implemented as additional functional modules with typical HEVC decoder, and Figure 5 shows the block diagram of the proposed HEVC decoder structure. For real-time demonstration, open source OpenHEVC decoder is used [4]. This paper describes the modification of function *hls_decode_entry_tiles* in OpenHEVC decoder to implement the proposed method. A function *sched_setaffinity* is used to allocate video decoding threads to big and little cores.

4 EXPERIMENTAL RESULTS AND DEMONSTRATION

This paper conducts a demonstration on two Android smart phones(Samsung Galaxy S7 Edge) which have asymmetric multicores, as shown in Figure 6. These two phones decode test sequences segmented by conventional uniform and the proposed non-uniform Tile partitioning methods using the modified OpenHEVC decoder, and this paper calculates decoding speed differences between the two phones. The Samsung Galaxy S7 Edge has four big and four little cores, but two big cores are always on online state, on the other hand,

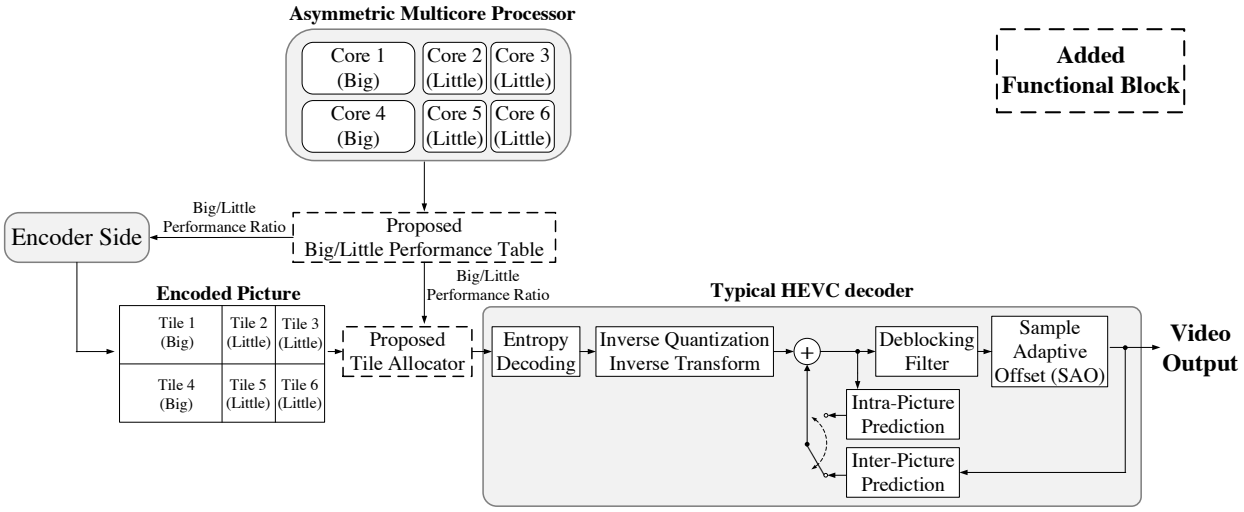

Figure 5: Block diagram of the modified HEVC decoder (in case of 2 big and 4 little asymmetric multicores.)

the other two big cores are normally on offline state for power saving. Thus, this paper considers only two big and four little cores.

Figure 7, 8 and Table 2 show the measured performance gains in decoding time through *PeopleOnStreet* and *Traffic* test sequences. The results show that the proposed method achieved an average 25% decoding time gains. The decoding time gains are achieved by increasing decoding complexity for big cores and reducing decoding complexity for little cores. In addition, Figure 9 and 10 show utilization rates of each cores during conventional and the proposed Tile partitioning–based decoding. In the Figure 9, utilization rates of 2 big cores show large fluctuations. A cause of the large fluctuations is that big cores wait for little cores to complete decoding of a picture, although the big cores complete to decode Tiles which is allocated to them. On the other hand, the figure 10 shows relatively stable utilization rates of big cores compared to figure 9, because wait time of big cores is minimized by the proposed Tile partitioning method. The minimized wait time enhances overall decoding performance.

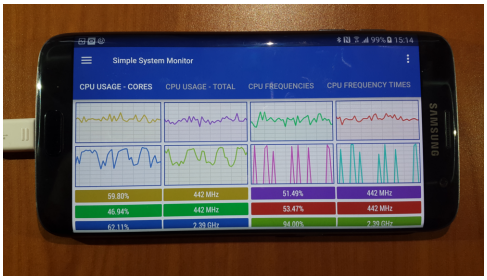

Figure 6: Screen capture of experiments with Samsung Galaxy S7 Edge.

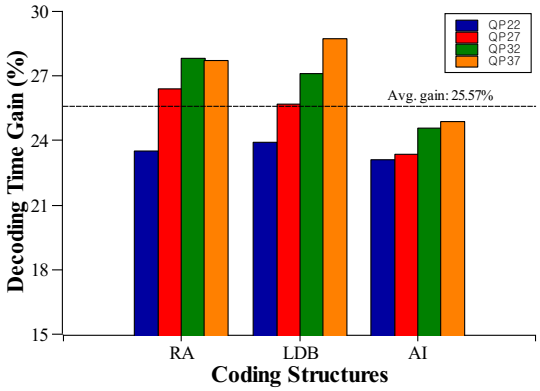

Figure 7: Decoding time gain from *PeopleOnStreet* at Exynos 8890 Octa.

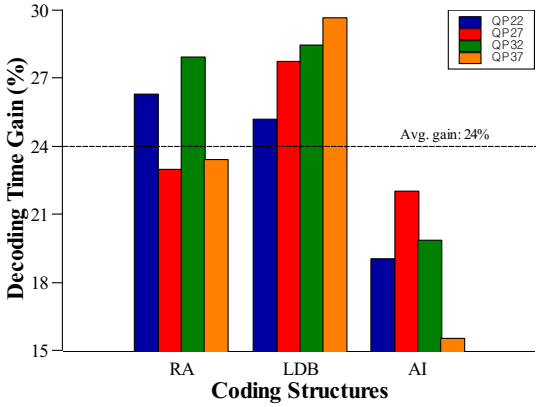

Figure 8: Decoding time gain from *Traffic* at Exynos 8890 Octa.

Table 2: Decoding time gains(%) for Samsung Galaxy S7 Edge

Test sequences	QP	Decoding time gain (%)		
		RA	*LDB*	*AI*
PeopleOnStreet	22	23.51	23.91	23.12
	27	26.42	25.69	23.37
	32	27.84	27.12	24.59
	37	27.73	28.72	24.86
Traffic	22	26.31	25.17	19.02
	27	22.96	27.73	22.01
	32	27.91	28.44	19.87
	37	23.42	29.65	15.54

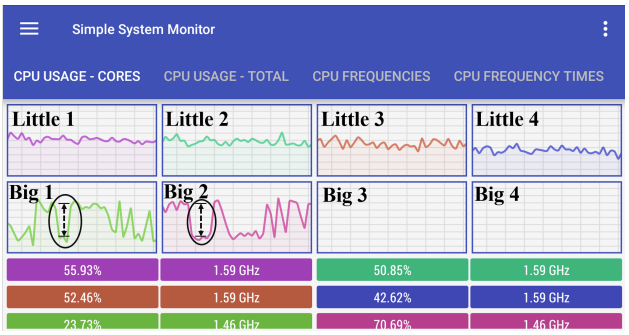

Figure 9: CPU core utilization rates with conventional Tile partitioning.

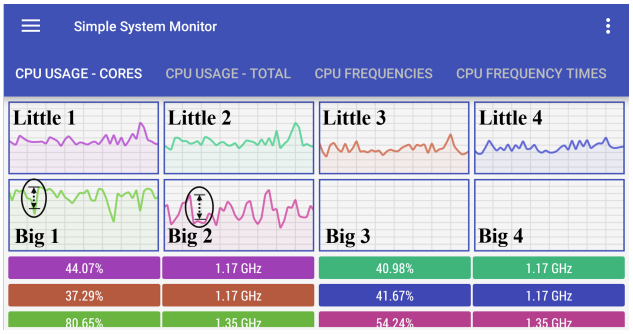

Figure 10: CPU core utilization rates with proposed Tile partitioning.

5 CONCLUSION

This paper proposes the novel HEVC Tile partitioning method using asymmetric multicores for UHD parallel video processing. The method minimizes the decoding time gap between big (faster) and little (power efficient) cores by allocating non-uniform HEVC Tiles to the cores. Experimental results with standard 4K UHD test sequences show an average 25% performance improvement on the Android smart phone introduced recently.

6 ACKNOWLEDGEMENTS

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT & Future Planning(NRF–2015R1C1A1A02037743)

Demo Link: https://youtu.be/a95sek6Oy_c

REFERENCES

[1] ARM. 2017. *big.LITTLE Technology - ARM.* https://www.arm.com/products/processors/technologies/biglittleprocessing.php.
[2] Frank Bossen, Benjamin Bross, Karsten Suhring, and David Flynn. 2012. HEVC complexity and implementation analysis. *IEEE Trans. Circuits Syst. Video Technol.* 22, 12 (2012), 1685–1696.
[3] Chi Ching Chi, Mauricio Alvarez-Mesa, Ben Juurlink, Gordon Clare, Félix Henry, Stéphane Pateux, and Thomas Schierl. 2012. Parallel scalability and efficiency of HEVC parallelization approaches. *IEEE Trans. Circuits Syst. Video Technol.* 22, 12 (2012), 1827–1838.
[4] OpenHEVC. 2017. *OpenHEVC GitHub page.* https://github.com/OpenHEVC/openHEVC.
[5] Hyun-Joon Roh, Yeongil Ryu, and Eun-Seok Ryu. 2015. Video Decoding Complexity Analysis Based on HEVC Resolution. In *Korea Information Processing Society (KIPS) Fall Conference.* KIPS, 1603–1606.
[6] Eun-Seok Ryu, Yeongil Ryu, and Hyun-Joon Roh. 2016. Tile Information Signaling for Video Parallel Processing. *Telecommunications Technology Association (TTA) Standard* TTAK.KO-10.0959 (2016).
[7] Yeongil Ryu, Hyun-Joon Roh, Shin Jin Kang, Soo-Kyun Kim, and Eun-Seok Ryu. 2016. Non-Uniform HEVC Tile Partitioning Method for Asymmetric Multicores. In *The 11th Asia Pacific International Conference on Information Science and Technology.* KSII, 229–231.
[8] Yeongil Ryu, Hyun-Joon Roh, and Eun-Seok Ryu. 2016. Tile Partitioning-based HEVC Parallel Decoding Optimization for Asymmetric Multicore Processor. *Journal of The Korean Institute Of Information Scientists and Engineers* 43, 9 (2016), 1060–1065.
[9] Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. 2012. Overview of the high efficiency video coding (HEVC) standard. *IEEE Trans. Circuits Syst. Video Technol.* 22, 12 (2012), 1649–1668.