MULTICORE BASED HIGHLY PARALLEL AND FLEXIBLE FRAMEWORK FOR HEVC MOTION ESTIMATION

Yufeng Bai, Yibo Fan, Yanheng Lu, and Xiaoyang Zeng*
State Key Lab of ASIC & System, Fudan University, Shanghai 201203, CHINA
*Corresponding Author's Email: fanyibo@fudan.edu.cn

ABSTRACT

In this work, a highly parallel and flexible framework based on a multicore processor which is especially optimized for computation-intensive execution is proposed to accelerate motion estimation for HEVC. Using multilevel on-chip communication mechanism greatly enhances efficiency and flexibility of data exchange in motion estimation. Experimental results not only validate the feasibility of the framework, but also show that ME achieves 8.5 times speedup comparing to typical frameworks while 16 cores are utilized.

INTRODUCTION

High Efficiency Video Coding (HEVC) is the state-of-art video coding standard developed by Joint Collaborative Team on Video Coding (JCT-VC). A highly flexible hierarchy of unit representation is the most significant feature that has been introduced in HEVC [1]. In this representation, a frame is divided into LCUs (largest coding unit). Then an LCU is further divided into CUs (coding unit), in a quad-tree structure. A CU is further divided into small CUs recursively, until it is a SCU (smallest CU).

Motion estimation (ME), a block-matching method to exploit temporal correlation, is inherited in HEVC for inter-prediction. A CU is split into one, two or four PUs (prediction unit) for prediction, if it would not be further divided. For inter-prediction in HEVC, the main partition modes for HEVC inter-prediction include 2Nx2N, 2NxN and Nx2N, indicating that the CU is not split, split into two equal-size horizontally, and split into two equal-size vertically. Types of PU would be more various if asymmetric motion partitions (AMP) are considered. In order to obtain the best PU candidate, candidates should be evaluated as much as possible. Thus there is a trade-off between coding efficiency and computation complexity.

In software solution for ME, many algorithms have been proposed, targeting at decreasing the searching points of candidates. But ME still occupies huge computation consumption in software solution with its sequential processing. Analyses from [2] show that the complexity of ME occupies over 60% in an encoder.

VLSI hardware design would be a good solution to accelerate ME, which has been widely used in H.264/AVC encoders [3]. However, evaluations in [4] indicate that, hardware solution for ME in HEVC faces huge challenge with larger CU sizes and more partition modes. Moreover, hardware design usually use full search to maximize parallelism, compromising the flexibility to adopt different fast search algorithms.

Recently, multicore-processor is gaining prevalence due to its high parallelism and flexibility characteristic. Therefore it could be used to accelerate the computation-intensive ME process, combining advantage of software and hardware solution.

In this paper, firstly, an efficient framework is proposed for parallelizing HEVC ME. Secondly, we map this framework to a multicore processor, using the multilevel on-chip communication in order to improve efficiency of data exchange and memory access. In all, speedup has been achieved comparing to the typical framework.

COMPARISON WITH THE TYPICAL

When processing motion estimation, CU depth is set at first, specifying the CU size of the current ME. Then ME on all PU sizes is processed individually and successively. Finally, a mode decision is made at the end of last PU size, picking up the best PU for the current CU, based on rate-distortion (RD) optimization The general equation of Lagrangian cost function is given by

$$RD-\cos t = D+\lambda \times R \qquad (1)$$

where λ is the Lagrangian multiplier, D is the distortion between original and reference pixels, and R is the coding rate associated with motion vectors.

The above procedure processes ME in a sequential way which is adopted by the HEVC reference software (HM) [5], shown in Figure 1(a). This strategy of ME is called typical framework in this paper. In the typical framework, sequential procedure and enormous data dependency contribute to large duration time of ME.

In this paper, we propose a ME framework to process ME in a parallel way based on multicore processors. As shown in Figure 1(b), ME engines are responsible for all depth CU. Within the engine, ME on all PU sizes (2Nx2N, 2NxN, Nx2N and other partitions) is processed simultaneously, and RD cost of every candidate is saved. A mode decision is determined using the saved RD costs with ME processing. Once all PU candidates are handled, the best partition mode of the current CU level is gained immediately.

The proposed framework processes ME in a highly parallel way, diminishing the expenditure in terms of time. At the same time, it reduces the data dependency among

CUs and PUs level by adopting communication network to transfer data between different ME engines. Moreover, fast-speed and high-efficiency of mode decision with same CU size are achieved, also including considerable improvement in computation resource utilization rates of ME.

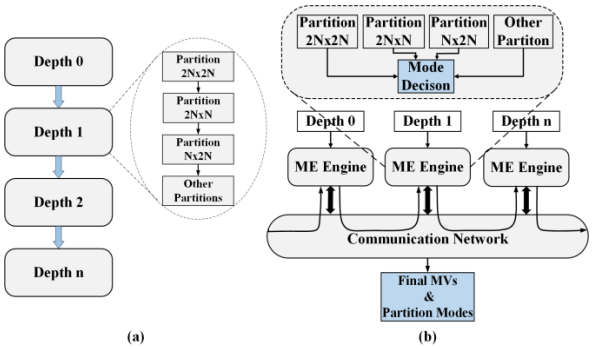

Figure 1: (a) Typical framework (b) Proposed framework

FRAMEWORK IMPLEMENTATION

Multicore-processor is drawing attention because of their high flexibility and low implementation cost. [6] describes a high-performance energy-efficient 24-core processor with a packet controlled circuit-switched double-layer network-on-chip (NoC) and a cluster-shared NoC. The features of this multicore processor make it become a good candidate for supporting the proposed framework implementation capable of exploiting the parallelization and eliminating data dependency of motion estimation for HEVC.

The proposed ME framework for HEVC is mapped to 16 cores of the processor mentioned above. The size of LCU used in this paper is 32x32. Max partition depth equals to 4, which means the size of smallest coding unit is 8x8. In order to obtain the maximized ME parallelism, the following features are adopted.

Multicore Mapping Rules

A cluster consists of a set of tightly connected cores that work together to obtain better shared-memory utilization and higher-efficient message transmission. A cluster is a suitable hardware implementation of ME engine for the reason that three cores of the cluster would be used for ME and one for mode decision. Therefore we allocate each cluster for each CU size, while three cores processes each PU size and one makes decision, shown in Figure 2(a). Owing to pattern search algorithms, time consumption on ME become longer with the reduction of CU size. Thus, the CUs with size 8x8 are mapped to 2 clusters. To balance load of cores in other clusters, searching windows for the smaller CUs are enlarged, which also helps to gain better performance.

Multilevel On-chip Communication

Highly data dependency and frequent memory access are two of the challenging problems in HEVC ME [4]. RD costs for mode decision need to be exchanged among different cores, as well as clusters. Meanwhile, original pixels and reference pixels transmission are considerable and frequent. To deal with these problems, multilevel on-chip communication mechanism plays a great role in the proposed framework. (Figure 2(b))

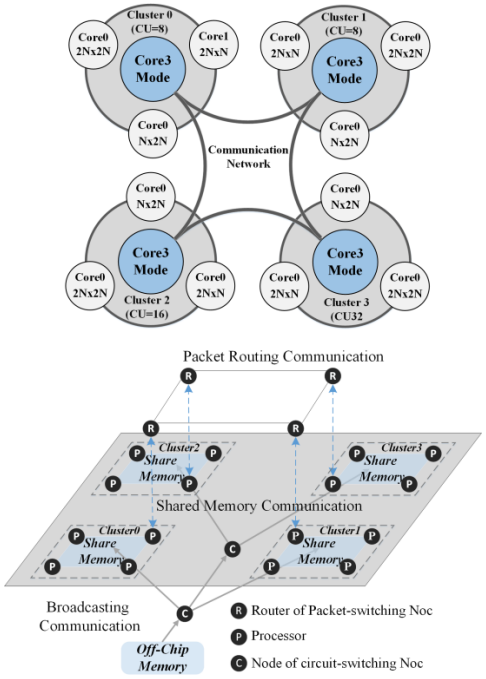

Figure 2: (a)Mapping rules (b)Multi-level communication

Shared Memory within Cluster: In this level, shared memory serves for cores within the same cluster. On one hand, original pixels and reference pixels are stored in shared memory, so that they would be fetched together by 3 cores inside the cluster. On the other hand, cores could set flags and store RD-costs in shared memory when processing motion estimation, thus other cores could detect flags and make mode decision by accessing these data.

Packet Routing Communication: For cores in different cluster, an advanced communication method is adopted in this level. Packet-switching NoC is a high speed channel for small amounts of data transfer [6], which is appropriate for transferring mode decision data in different clusters. Mode decision data is submitted to routers of packet-switching NoC by cores, and routers search for suitable route and transmit the data according to their destination and the current status of NoC.

Broadcasting Data Transmission: In this level, shared memory serves for cores within the same cluster. On one hand, original pixels and reference pixels are stored in shared memory, so that they would be fetched together by 3 cores inside the cluster. On the other hand, cores could set flags and store RD-costs in shared memory when processing motion estimation, thus other cores could

detect flags and make mode decision by accessing these data.

EXPERIMENS AND DISCUSSION

The propose framework is aimed at fast search algorithm, so three different fast ME search algorithms, 4-Step-Search (4SS), Diamond-Search (DS) and Hexagon-Search (HS), are conducted in our experiments.

In Figure. 3(a), it can be seen that loads of every core are balanced. It costs approximately 180k cycles to process a LCU in 32x32 for average using HS, while 200k cycles for 4SS. Time spent on loading original and reference pixels is about 2600 cycles at the beginning, while actually only about 900 cycles are needed considering of overlapped searching window. It also should be noticed that partition mode decision costs less time because it is distributed to many clusters and processed simultaneously with motion estimation.

The speedup (S) is a common measurement for parallelism, and it is calculated as follows:

$$S = \frac{T_{serial}}{T_{parallel}} \qquad (2)$$

where T_{serial} and $T_{parallel}$ are the ME processing time of serial execution and proposed parallel execution.

In Figure. 3(b), it is shown that speedup increases as number of core add. Compared with serial execution (the typical framework), our proposed framework achieves 8.5 times acceleration when using 16 cores. It should be noticed that from Figure. 3, the bottleneck of our experiment locates in CU=16x16, which would be broken up by using another cluster to deal with ME on this CU size. Moreover, in order to eliminate the data dependency of ME in this multicore based framework, communication consumption is inevitable, which will lead to the fact that it cannot achieve the same speedup as the number of cores.

CONCLUSION

A highly parallel and flexible ME framework for HEVC is proposed in this paper. The motivation is to accelerate ME and achieve maximal resource utilization, while enhancing flexibility for fast search algorithm comparing to hardware design. The proposed framework makes fully use of the resources in the multicore processor, including multilevel on-chip communication mechanism. Rational mapping rules on multicore contributes to balancing load of cores. Experimental results show that the proposed framework applies to different fast search algorithm and speed up the process of ME effectively compared with the typical serial framework.

ACKNOWLEDGEMENTS

This paper was supported by National Natural Science Foundation of China (61306023),Specialized Research Fund for the Doctoral Program of Higher Education (SRFDP, 20120071120021), STCSM (13511503400), National High Technology Research and Development Program (863, 2012AA012001)

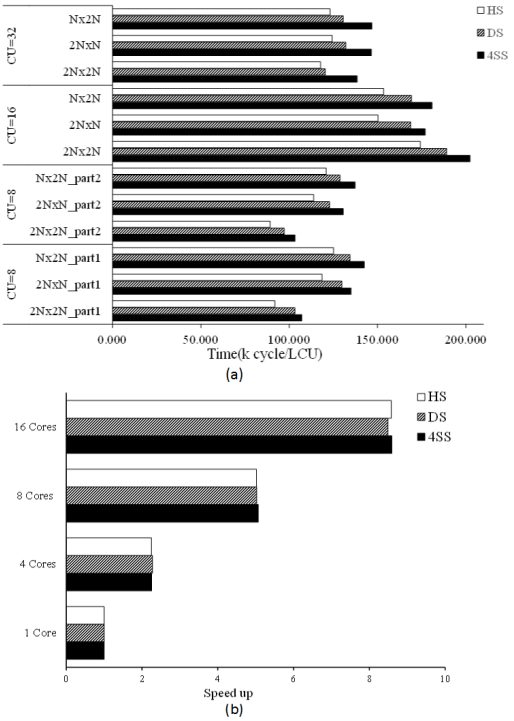

Figure 3: Experimental results

REFERENCES

[1] G. Sullivan, *et.al*, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[2] F. Bossen, B. Bross, K. Suhring, and D. Flynn, "HEVC complexity and implementation analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1685–1696, 2012.

[3] Y. Fan, X. Zeng, and S. Goto, "Optimized 2-d SAD tree architecture of integer motion estimation for H.264/AVC,"*IEICE TRANSACTIONS on Electronics*, vol. E94-C, no. 4, pp. 411–418, 2011.

[4] M. E. Sinangil, *et.al*, "Memory cost vs. coding efficiency trade-offs for HEVC motion estimation engine," in 2012 19th *IEEE International Conference on Image Processing (ICIP)*, 2012, pp. 1533–1536.

[5] I.-K. Kim, *et.al*, "High efficiency video coding (HEVC) test model 10 (HM 10) encoder description," *JCT-VC, Doc*. JCTVC-L1002, 2013.

[6] P. Ou, *et.al*, "A 65nm 39GOPS/W 24-core processor with 11Tb/s/W packet-controlled circuit-switched double-layer network-on-chip and heterogeneous execution array," *Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, 2013 IEEE International, pp. 56–57, 2013.