BANDWIDTH-EFFICIENT CACHE-BASED MOTION COMPENSATION ARCHITECTURE WITH DRAM-FRIENDLY DATA ACCESS CONTROL

Tzu-Der Chuang, Lo-Mei Chang, Tsai-Wei Chiu,Yi-Hau Chen, and Liang-Gee Chen

DSP/IC Design Lab, Graduated Institute of Electronics Engineering
National Taiwan University, Taipei, Taiwan

ABSTRACT

For H.264/AVC decoder system, the motion compensation bandwidth comes from two parts, the reference data loading bandwidth and the equivalent bandwidth from DRAM access overhead latency. In this paper, a bandwidth-efficient cache-based MC architecture is proposed. It exploits both intra-MB and inter-MB data reuse and reduce up to 46% MC bandwidth compared to conventional scheme. To reduce the equivalent bandwidth from DRAM access overhead latency, the DRAM-friendly data mapping and access control scheme are proposed. They can reduce averagely 89.8% of equivalent DRAM access overhead bandwidth. The average MC burst length can be improved to 9.59 words/burst. The total bandwidth reduction can be up to 32~71% compared to previous works.

***Index Terms*—**Cache, Motion Compensation, Cache-based Motion Compensation, H.264/AVC

1. INTRODUCTION

H.264/AVC is the advance video coding standard developed by the ITU-T – ISO/IEC Joint Video Team (JVT). H.264/AVC provide several new coding tools including sub-pixel inter prediction, variable block size (VBS) motion compensation, and it can save about 25-45% bit-rate compared with MPEG-4 standard. Unfortunately, in order to support these new coding tools, it requires huge memory bandwidth to fetch reference pixels for motion compensation (MC). For H.264/AVC decoder system, based on our simulation, the MC will take up to 75~83% memory bandwidth without applying any MC bandwidth reduction technique.

Several techniques have been proposed to reduce MC bandwidth requirement in H.264 decoder system. They can be classified into two directions. One is to reduce MC data loading bandwidth, and the other is to reduce the DRAM access latency. Tsai proposes an interpolation window reuse (IWR) scheme that load reference data according to macro-block (MB) type [1]. This data reuse scheme can reuse overlapped reference data in the same partition. But the performance is limited when the motion vectors (MV) of neighboring blocks are different. Li utilizes a cache scheme to further reduce the memory bandwidth [2]. However, its architecture can only support P-frame coding and cannot reuse overlapped data across MB boundary. The other direction is to reduce the equivalent bandwidth from DRAM access overhead latency. Zhu proposes a data arrangement and memory mapping method for MC in [3] and a DRAM command out-of-order techniques that can efficiently reduce the DRAM access latency in [4].

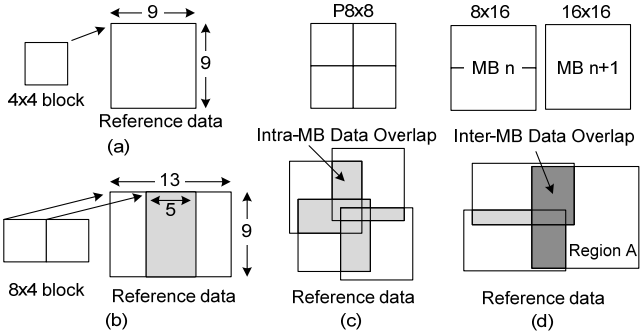

Fig. 1. Example of required reference data region for MC, intra-MB data overlap and intra-MB data overlap

In this paper, we propose a bandwidth-efficient cache-based MC architecture and DRAM-friendly data access control for H.264/AVC decoding system. This cache-based MC architecture can reuse overlapped data across different block partition and MB boundary. The DRAM-friendly data access can efficiently reduce the row precharge/active frequency and access latency of DRAM.

The rest of this paper is organized as follows. In Sec. 2, the design challenges of H.264 motion compensation are illustrated. Section 3 presents the proposed cache-based MC architecture and DRAM-friendly data access scheme. Simulation result and comparison are shown in Sec. 4. Finally, Sec. 5 concludes this paper.

2. DESIGN CHALLENGES OF H.264/AVC MOTION COMPENSATION

For H.264 motion compensation, the bandwidth comes from two parts. They are the reference data loading bandwidth and its equivalent bandwidth from DRAM access overhead latency. In H.264, it adopts the 6-tap sub-pixel interpolation filter to support luma sub-pixel MC. Therefore, interpolation of an X×Y luma sub-pixel block requires (X+5)×(Y+5) integer pixels as shown in Fig.1(a). Under this constraint, the bandwidth of loading reference frame data becomes a main design challenge in H.264 decoding system. Fortunately, there are overlapped regions between neighboring blocks. IWR data reuse scheme was [1] proposed to reduce data access for the overlapped data in the same block partition as shown in Fig.1(b). Li [2]

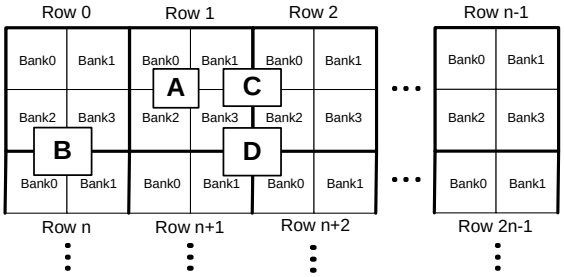

Fig. 2. Data arrangement and memory mapping for MC.

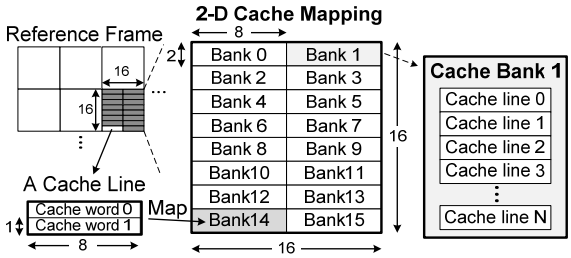

Fig. 3. Data mapping in proposed cache organization

proposes a cache architecture to reuse intra-MB overlapped data in different block partition as shown in Fig.1(c), but it cannot exploit the data reuse for inter-MB data overlapped as shown in Fig.1.(d). Hence a bandwidth-efficient MC architecture that can achieve both intra-MB data reuse and inter-MB data reuse is necessary for H.264 decoder system.

The other design challenge is the equivalent bandwidth of DRAM access overhead latency. In modern DRAM architecture, a DRAM is composed of several rows and each row is capable of storing 1~8Kbyte data. However, a 1920× 1080 frame needs 2MB data storage, and it must be distributed into several hundreds of rows. In DRAM access behavior, if the target access data are in an un-active row, the DRAM needs time to precharge and active the target data row. Hence it may result in a lot of extra overhead cycles if the DRAM needs to precharge/active different rows frequently. The overhead cycles from access latency can be represented as equivalent latency bandwidth. The memory mapping affects the row precharge/active frequency deeply. Zhu [3] proposes a memory mapping method for MC as shown in Fig.2, which will be discussed in Sec.3. However, it still has some cases that need to access data across row boundary in this memory mapping. How to minimize the row precharge/active frequency and hide the access latency is another important issue for MC architecture.

3. PROPOSED MOTION COMPENSATION ARCHITECTURE AND DATA ACCESS SCHEME

To alleviate the MC bandwidth requirement, a cache-based MC architecture with DRAM-friendly access control is proposed. The cache-based MC architecture can reduce MC data loading bandwidth significantly and the DRAM-friendly access control can reduce the precharge/active frequency and hide the access latency.

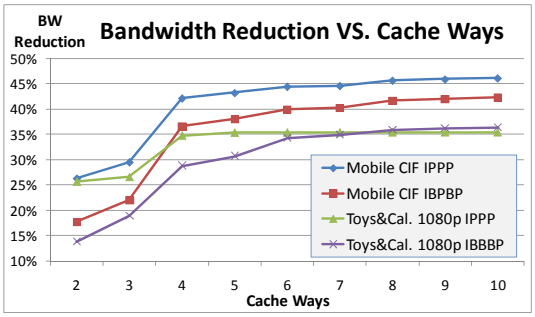

Fig. 4. BW reduction performance of different cache size.

3.1. Cache-based Motion Compensation Architecture

In order to save memory bandwidth for MC, a cache-based data reuse strategy is proposed. The data fetch pattern is based on MB partition like IWR data reuse scheme, but it will check whether the target data are in the cache ahead of accessing the DRAM. In our MC architecture, the loaded data are stored in the cache banks, unlike traditional MC architectures which store data in the temporary buffer. The cache system allows the neigh-boring blocks which are in different block partition or in different MB, to achieve intra-MB and inter-MB overlapped data reuse. In our MC architecture, the N-way associative cache architecture is adopted.

3.1.1. Data mapping in proposed cache organization

In our proposed cache architecture, it adopts 2-D data mapping method as shown in Fig.3. The reference frame data are partitioned into 16×16 blocks and data in a 16×16 block can be partitioned into 16 cache line. A cache line consists of two cache words and each cache line can be mapped into an unique cache bank according to its position within a 16×16 block. The size of a cache word is equivalent to the external memory bus width, which is 64-bits. In the N-way associative cache architecture, a cache bank contains N cache lines.

3.1.2. Cache system architecture for motion compensation

The cache size affects bandwidth reduction performance a lot. Base on the proposed cache organization and data mapping, the cache size is proportional to the number of cache lines in a cache bank. A cache system with larger cache size has better bandwidth reduction because it contains more data for data reuse. To find the best trade-off point between cache size and performance, we evaluate the bandwidth reduction performance versus N as shown in Fig.4. Notice that the bandwidth reduction result is compared with IWR data reuse scheme. According to this figure, the 6-way cache system is adopted in our MC architecture because N=6 is a good trade-off point.

The proposed 6-way cache system architecture is shown in Fig.5. The proposed cache system has 16 cache banks and each cache bank has 6 cache lines. One cache line represents 16 pixels data which are stored in a SRAM set of four on-chip SRAM. Each cache line has four tag registers to store the information of these 16 pixels data, which are lock control (Lock), reference frame index (RefIdx), x-position (X-tag), and y-position (Y-tag). Lock is a one-bit signal which represents the

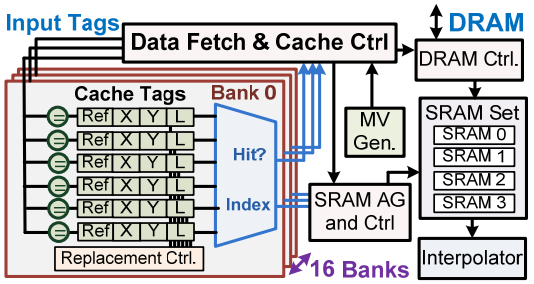

Fig. 5. Proposed 6-way cache MC architecture.

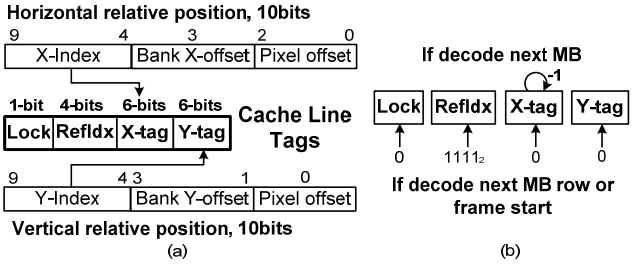

Fig. 6. Tag contents and update behavior.

availability of data for replacement. If the cache line data has not been used for interpolation, the Lock signal is set as 1 and indicates that the cache line should not be replaced for storing new cache line. As long as the cache line data has been used for interpolation, the Lock signal is set as 0. RefIdx indicates the reference frame's index. This 4-bits signal is capable of representing LIST_0 and LIST_1 two lists for B-frame, and each list can support up to five reference frames. In proposed cache system, it uses relative position to record the cache line data's location instead of absolute position. The two dimensional relative coordinates of each cache line can be split into tag part and offset part as shown in Fig.6(a). These tags can support ±512 search range which is sufficient for high definition video. Because it uses relative coordinates, the tag data must be updated while decoding next MB or next frame as shown in Fig.6(b). For area efficiency consideration, we group two cache words in one cache line as described in Sec.3.1.1. Two cache words which are in the same cache line can share the tag registers and 50% tag memory can be saved. The FIFO replacement policy is used in the proposed cache system. If the cache bank is full and the new data want to store in this bank, the oldest cache line will be replaced as long as its Lock is not 1.

3.2. DRAM-friendly Data Mapping and Access Control

To alleviate the equivalent overhead bandwidth caused by DRAM row precharge/active, two design methodologies can be applied. One is to reduce the precharge/active frequency and the other is to hide the access latency. A DRAM data mapping method that suitable for MC has been proposed by Zhu [3] as shown in Fig.2. However, it still has some cases that need to precharge/active operation when the target data are distributed in different rows, like the case B, C, and D in Fig.2. In case B, the MC access data are crossing two vertical neighboring rows, and they're crossing two horizontal neighboring rows in case C. In conventional access pattern which is in raster scan, only one

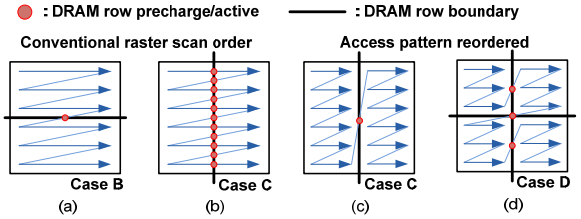

Fig. 7. Precharge/active commands in different access order

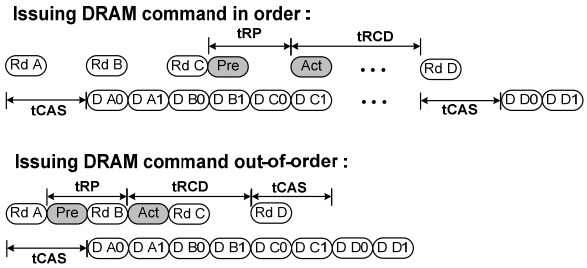

Fig. 8. Illustration of DRAM command out-of-order

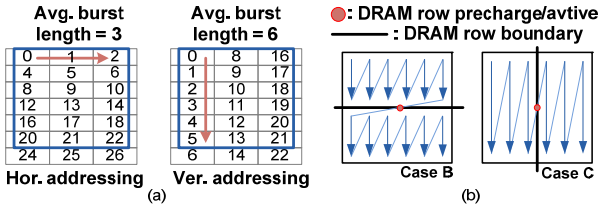

Fig. 9. (a)Average burst length in different addressing mapping. (b) Access pattern reordered in vertical addressing mapping

precharge /active command is needed in case B as shown in Fig.7(a). However, in case C, it needs to precharge and active rows several times as shown in Fig.7(b). This situation will occur in case D, too. To reduce the precharge/active frequency in case C and D, we proposed an access pattern reordered scheme to alleviate the DRAM access overhead. As shown in Fig.7(c), the MC controller adjusts the access order when the access pattern encounters row boundaries. The controller will change the access order to load reference data in the same row first and then to load the data in different row later. The access pattern for case D is shown in Fig.7(d). The proposed access pattern reordered scheme can reduce 79~85% precharge/active frequency for MC compensation.

To further reduce the DRAM access overhead, the DRAM command out-of-order technique [4] is adopted as shown in Fig.8. The precharge and active commands will be taken forwa rd in advance to hide the precharge/active latency cycles. By applying these two techniques, the average equivalent latency bandwidth can be reduced 89.8%.

The average burst length is another important index in MC design. According to our observation, the shapes of data access pattern in proposed cache-based MC are usually tall rectangle, like the region-A in Fig.1. Besides, the shape for a basic memory word in DRAM is flat rectangle and it limits the average burst length for 2-D block MC data loading in conventional horizontal addressing data mapping. Therefore, we proposed a vertical addressing data mapping method that maps the reference frame data in vertical direction as shown in

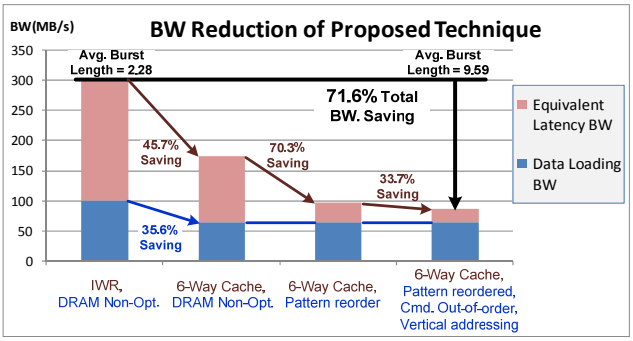

Fig. 10. Bandwidth reduction of proposed technique in Pedestrian_area 1080p HD, IPPP @15Mbps 30fps

Fig.9(a). The data access pattern of case B and case C in Fig.2 are modified as shown in Fig.9 (b). This data mapping method can improve the average burst length from 2.28 to 9.59 words/burst.

4. RESULTS AND COMPARISON

To evaluate the performance of proposed bandwidth reduction methods, we modify the H.264/AVC scalable extension reference software JSVM 9.14 for simulating the MC memory access. In our simulation environment, the external DRAMs are Micron MT48LC4M32B2 [5], and our target operating frequency is 166 MHz. The latency for a precharge/active command is 60ns and is equivalent to 10 clock cycles in our simulation environment. Three different size sequences are simulated, which are CIF, 720pHD, and 1080pHD sequences. The target bitrates are 768kbps, 6Mbps and 16Mbps respectively. Fig.10 shows the simulation result of our proposed techniques in 1080p HD Pedestrian_area sequence. From this figure, the proposed case-base data reuse scheme can exploit intra-MB and inter-MB data reuse and thus reduces 35.6% of MC data loading bandwidth. Besides, it saves 45.7% DRAM access latency bandwidth because of reduced data loading access. To further reduce DRAM access overhead equivalent band-width, the proposed access pattern reordered scheme can reduce 70.3% equivalent bandwidth. With DRAM command out-of-order technique, the equivalent bandwidth can be further reduced 33.7%. By applying vertical addressing mapping, the burst length can be improved from 2.28 to 9.59 words/burst. Table 1 shows the performance result of more different sequence. Table 2 shows the comparison of the proposed scheme and previous works. We implement the previous works [1][2][6] in our evaluation environment according to their published papers. Our proposed scheme has 29~71% total MC bandwidth reduction.

The implementation of cache-based MC hardware architecture is designed with Verilog HDL and synthesized with UMC 90-nm technology. The total gate count is 72.3k and 1.5kB single port SRAM.

5. CONCLUSION

In this paper, we propose a bandwidth-efficient cache-based MC architecture for H.264/AVC decoding system. The cache-based MC architecture exploits both intra-MB and inter-MB data reuse and has up to 46% MC bandwidth reduction compared to conventional IWR data reuse scheme. Besides, the DRAM-friendly data mapping and access control scheme are proposed to reduce the equivalent bandwidth from DRAM access overhead. They can reduce averagely 89.8% of DRAM access overhead equivalent bandwidth. The MC average burst length can be improved to 9.59 words/burst. The total bandwidth reduction can be up to 32~71% compared to previous works.

Table 1. MC bandwidth comparison in different sequence

Sequence	Coding Structure	Data Loading BW		Equivalent Latency BW		Total BW Reduction
		IWR	Cache	Non-Opt.	Proposed	
Foreman (CIF)	IPPP	6.56	3.69	13.86	1.54	74.4%
	IBBBP	8.99	5.31	20.75	2.35	72.5%
Mobile (CIF)	IPPP	7.86	3.68	17.59	2.06	77.4%
	IBBBP	10.26	6.16	24.76	2.25	76.0%
Harbour (720p)	IPPP	66.08	35.63	141.09	14.12	76.0%
	IBBBP	98.10	63.60	219.84	19.26	73.9%
Crew (720p)	IPPP	40.66	27.04	79.28	8.37	70.5%
	IBBBP	69.77	48.35	154.31	12.99	72.6%
Rush hour (1080p)	IPPP	99.19	69.38	180.88	21.06	67.7%
	IBBBP	153.65	112.16	312.90	32.25	71.6%
Toys&Cal. (1080p)	IPPP	115.01	74.33	217.77	22.08	71.0%
	IBBBP	182.56	119.89	362.51	35.74	71.4%

Bandwidth unit : Mbytes/s

Table 2. BW comparison with previous work

	[1]	[2]	[6]	Proposed
Data Reuse Scheme	IWR	Single MB Cache	IWR	6-Way Cache
DRAM Access Strategy	None	None	Cmd. out-of-order	Access pattern reordered Cmd. out-of-order Vertical addring mapping
Data Loading BW	100.26	94.98	100.26	64.56
Equi. Latency BW	203.33	172.15	27.13	21.73
Total BW	303.59	267.13	127.39	86.29
Proposed method BW Reduction	71.58%	67.70%	32.26%	-

Sequence : Pedestrian_area 1080p, IPPP @15Mbps 30fps, with 64-bits bus
Bandwidth unit : Mbytes/s

6. REFERENCES

[1] C.-Y. Tsai, T.-C. Chen, T-W. Chen, and L.-G. Chen, "Band-width optimized motion compensation hardware design for H.264/AVC HDTV decoder," *ISCAS 2005*, pp. 273-276, August 2005.

[2] Y. Li, Y. Qu, and Y. He, "Memory Cache Based Motion Compensation Architecture for HDTV H.264/AVC Decoder," *ISCAS 2007*, pp. 2906 - 2909, May 2007.

[3] J. Zhu, L. Hou, W. Wu, R. Wang, C. Huang, J.-T. Li, "High Performance Synchronous DRAMs Controller in H.264 HDTV Decoder," *ICSCIT 2004*, vol. 3, pp. 1621 - 1624, Oct. 2004

[4] J. Zhu, P. Liu, D. Zhou, "An SDRAM controller optimized for high definition video coding application", *ISCAS 2008*, pp. 3518 – 3521, May 2008

[5] Micron SDRAM MT48LC4M32B2 datasheet, www.micron.com

[6] C.-D. Chien, et al., "A 252kgate/71mW Multi-Standard Multi-Channel Video Decoder for High Definition Video Applications," *ISSCC 2007*, pp. 282 - 603, Feb., 2007.