ELectronics EXpress

LETTER

IEICE Electronics Express, Vol.11, No.15, 1–6

Memory bandwidth reduction using frame pipeline in video codec chips

Yun Gu Lee[1a)], **Ki-Hoon Lee**[2], **and Woosaeng Kim**[1]

[1] *Department of Computer Science and Engineering, Kwangwoon University, Seoul, 139–70, South Korea*

[2] *Department of Computer Engineering, Kwangwoon University, Seoul, 139–70, South Korea*

a) *harmony96@gmail.com*

Abstract: An image processing chip includes many image processing engines requesting heavy external memory access. So bus traffic is a critical issue in real-time systems. An video encoder is one of blocks requesting the most heavy external memory access among them. This letter presents a method to reduce the external memory requests for accessing reference frames in video codec chips. The method pipelines a video in a frame level, and the intermediate reconstructed data is used in the next frame without storing it to external memory. Thereby the proposed algorithm reduces the external memory access with reasonable internal memory increase.

Keywords: video coding, codec, memory bandwidth reduction

Classification: Electron devices, circuits, and systems

References

[1] T. Wiegand, G. J. Sullivan, G. Bjoentegaard and A. Luthra: IEEE Trans. Circuits Syst. Video Technol. **13** (2003) 560. DOI:10.1109/TCSVT.2003.815165

[2] J. Kim and C.-M. Kyung: IEEE Trans. Circuits Syst. Video Technol. **20** (2010) 848. DOI:10.1109/TCSVT.2010.2045923

[3] L. Guo, D. Zhou and S. Goto: European Signal Processing Conference (EUSIPCO) (2013).

[4] H. Jeong, J. Kim, K. Lee, K. Yoo and J. Kim: 2012 IEEE 16th International Symposium on Consumer Electronics (2012) 1. DOI:10.1109/ISCE.2012.6241723

[5] M. Budagavi and Z. Minhua: International Conference on Acoustics, Speech, and Signal Processing (2008) 1165. DOI:10.1109/ICASSP.2008.4517822

[6] Y. G. Lee, B. C. Song, N. H. Kim, T. H. Kim and W. H. Joo: IEEE International Conference on Image Processing (2009) 2329. DOI:10.1109/ICIP.2009.5414420

[7] A. D. Gupte, B. Amrutur, M. M. Mehendale, A. V. Rao and M. Budagavi: IEEE Trans. Circuits Syst. Video Technol. **21** (2011) 225. DOI:10.1109/TCSVT.2011.2105599

DOI: 10.1587/elex.11.20140592
Received June 19, 2014
Accepted July 3, 2014
Publicized July 25, 2014
Copyedited August 10, 2014

1 Introduction

Nowadays, an advanced image sensor technology makes it easy to obtain high quality videos with a large resolution through digital cameras and camcorders. In general, an image processing-purpose SoC embedded in the digital cameras requires significant external memory access to handle large size images. Since the SoC includes many kinds of image processing blocks such as image enhancement, restoration, pre/post processing, and noise reduction, reduction of the external memory access is one of critical and important issues in designing the SoC. When several processing blocks simultaneously request the external memory access like DDR read or write, only one memory request is available while the other requests are waiting. As a result, this bus traffic becomes serious and the external memory bandwidth should be reduced efficiently.

An video encoder is one of blocks requesting the most heavy external memory access among image processing blocks in the above SoC. International video coding standards such as H.264 [1] divide the current (or target) frame into non-overlapped blocks, and sequentially encode each current block. The video encoder first reads the current block within the current frame from external memory. In order to reduce temporal redundancy, the encoder finds the best matching block of the current block within a reference frame that is a reconstructed frame of the previous frame. Here, the reference frame is read from the external memory. Then, the difference block between the best matching block and the current block is coded, and the encoder writes the new reference (or reconstructed) block to the external memory. In summary, the video encoder reads the current and reference frames from the external memory, and writes the new reference frame to the external memory.

If a frame buffer for storing the reference frame is embedded in the SoC, the external memory access to read and write the reference frames can be eliminated. However, it requires high implementation cost in terms of gate count and may not be a feasible solution. Therefore, this letter introduces a compromise method to significantly reduce the external memory access for accessing reference frame in video encoding for the SoC with reasonable internal memory increase.

2 Related works

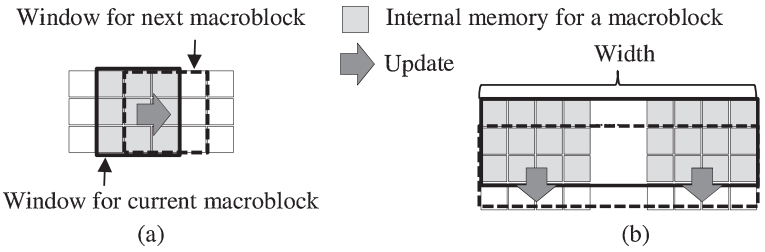

Fig. 1. (a) Sliding window and (b) buffering of macroblock row

DOI: 10.1587/elex.11.20140592
Received June 19, 2014
Accepted July 3, 2014
Publicized July 25, 2014
Copyedited August 10, 2014

Reference frames can be compressed using simple image compression techniques, which reduces the size of the reference frame, before storing them to external memory. The method is categorized into two approaches, namely lossless and lossy compression techniques. In [2], a lossless embedded coding algorithm and its hardware architecture were proposed. The method achieves a 60% reduction of memory bandwidth on average. Guo proposed lossless embedded compression using multi-mode DPCM and averaging prediction [3]. A lossless embedded compression with context-based error compensation was proposed to reduce memory bandwidth requirement [4]. Lossy compression achieves higher compression ratio with small quality degradation compared to lossless compression [5, 6]. However, since reference frames used in an encoder becomes slightly different from those in a decoder due to coding error of reference frame compression, the lossy compression approach can cause drift. In [7], lossy compression uses in motion estimation, and lossless compression is used in motion compensation to avoid the drift. This is achieved by separately storing the quantization error in the lossy compression.

In a large size SoC, many advanced image processing algorithms require high external memory access. Shared internal memory such as SRAM is used to reduce redundancy in accessing the external memory. The shared internal memory can be also used to temporally store macroblocks in the reference frame, as in Fig. 1(a). This sliding window approach updates only three macroblocks for each macroblock encoding [7]. When the shared internal memory is big enough to store several macroblock rows, external memory access can be reduced further as in Fig. 1(b). This approach needs to update only one macroblock for each macroblock encoding. Although a sliding window approach significantly reduces the external memory access, the amount of the external memory access is still very huge even for the SoC. Hence, the approach in Fig. 1(b) is sometimes considered in industry in spite of expensive implementation cost. Typically, when the number of reference frames is 1, the amount of external memory access for reading reference frames in a sliding window is about 3 Gbps for encoding full HD video with a frame rate of 60 Hz (A search range is ± 16). The amount of memory access is about 1 Gbps in the method in Fig. 1(b) with extra implementation cost.

When system bus for real-time SoC is designed, bandwidth required in the worst case scenarios should be considered. Although the lossless compression can significantly reduce memory traffic on average, since its compression ratio is not constant, the amount of external memory access in the worst case is not guaranteed. Hence, although the lossless approach can provide a significant bandwidth margin in the system bus, it is difficult to decrease a clock frequency of the system bus. Meanwhile, the amount of memory access in the worst case in the methods in Fig. 1(a) and (b) is deterministic, and they can be considered in determining the clock frequency of the system bus.

DOI: 10.1587/elex.11.20140592
Received June 19, 2014
Accepted July 3, 2014
Publicized July 25, 2014
Copyedited August 10, 2014

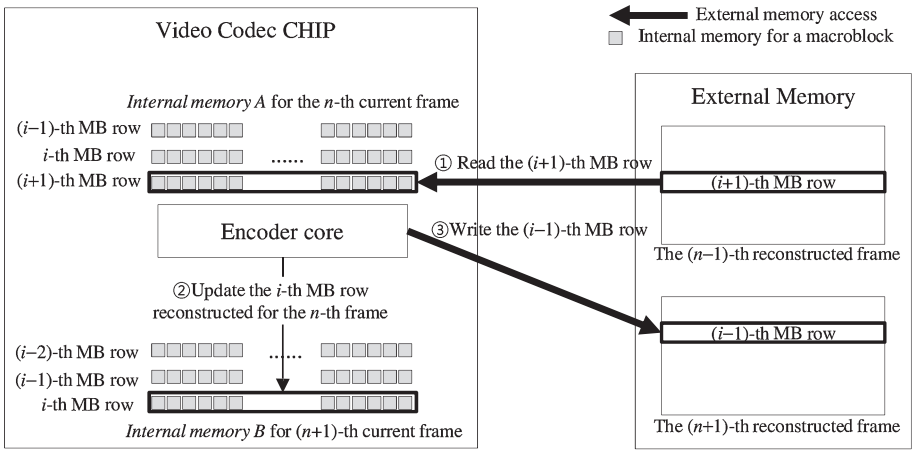

Fig. 2. Structure of the proposed algorithm. The n-th reconstructed frame is not stored in the external memory. It is directly fed to *Internal memory B*.

3 Proposed method

Fig. 2 illustrates the basic structure of the proposed method where memory bandwidth is further reduced using additional internal memory. Here, a vertical search range is assumed to be ± 16. Since the internal memory includes all pixels in the horizontal direction, there is no limitation in a horizontal search range. An encoder first loads the $(i+1)$-th macroblock row in a reference frame from external memory to *Internal Memory A*, and processes the i-th macroblock row in the n-th frame. Its reconstructed macroblock row is directly stored to *Internal Memory B* instead of storing it to external memory. Then, the encoder processes the $(i-1)$-th macroblock row in the $(n+1)$-th frame. Since image data required for motion estimation and compensation is stored in *Internal Memory B*, external memory access to read the reference frame is not necessary. Finally, the $(i-1)$-th macroblock row reconstructed for the $(n+1)$-th frame is stored to the external memory. The proposed algorithm can reduce the read and write of reference frames by using additional internal memory.

In the example, two successive frames are alternately encoded macroblock row by macroblock row. Hence, an encoder should alternately handle two successive frames based on a macroblock row. Since a video encoder hardware usually processes macroblock by macroblock, the encoder hardware can be modified to handle two successive frames by saving some necessary internal coding parameter such as motion vectors and mode information. Note here that since the encoder hardware adopts efficient macroblock pipeline structure, it is not usually efficient to alternately encode two frames in a macroblock level. Accordingly, the proposed algorithm adopts switching in a macroblock row level.

Fig. 2 illustrates an example of two successive frame encoding, and the depth of frame pipeline is 2. By investing more internal memory, its depth can be increased to further reduce the external memory access as shown in Fig. 3.

DOI: 10.1587/elex.11.20140592
Received June 19, 2014
Accepted July 3, 2014
Publicized July 25, 2014
Copyedited August 10, 2014

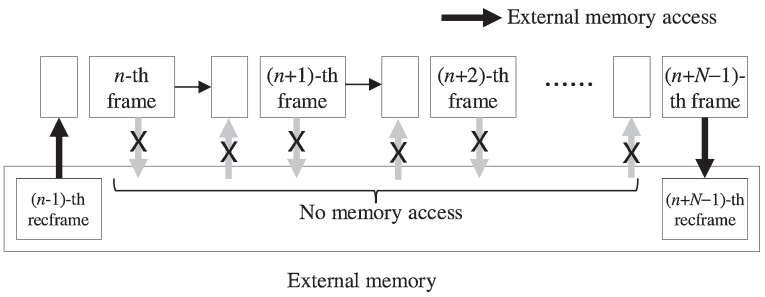

Fig. 3. Reduction of external memory access in the depth of frame pipeline with N

An image processing SoC always lacks available memory bandwidth. Due to the limited memory bandwidth, its performance is limited or the processing time becomes slow. Hence the SoC for expensive high-end products considers even expensive solution to increase memory bandwidth. In industry, an advanced image processing SoC that requires high memory traffic frequently considers the approach in Fig. 1(b). Therefore, although the proposed algorithm requires expensive internal memory to reduce the external memory access, this work should be meaningful in the SoC for the high-end products. Also since advanced image processing algorithms in SoC for digital cameras usually refer many pixel rows, the SoC for the high-end products in digital cameras usually includes a large size of shared internal memory (or cache) for reducing external memory access. Hence, the existing internal memory can be shared to significantly alleviate the increase of internal memory (or implementation cost) for the proposed algorithm.

4 Analysis

Fig. 4 depicts the relationship between external memory access and implementation cost for reading and writing reference frames. Note here that the memory bandwidth for reading the current frames is not considered in the figure. For analysis, it is assumed that an encoder performs motion estimation in the luminance domain. A resolution and a frame rate are set to 1920×1080 and 60 Hz, respectively. A vertical search range is set to be ± 16. X and Y axes represent implementation cost and external memory access per second. Additional implementation cost for saving coding parameters are ignored in the figure. The proposed algorithm efficiently reduces the external memory access by investing internal memory as shown in the figure.

5 Conclusion

This letter presents a method to reduce external memory access in video encoding for the SoC with reasonable internal memory increase. Since the amount of its external memory access is always constant, it can be easily considered in designing system bus. The proposed algorithm is especially suitable for an image processing SoC for expensive high-end products that always lack available memory bandwidth. However, the proposed algorithm

DOI: 10.1587/elex.11.20140592
Received June 19, 2014
Accepted July 3, 2014
Publicized July 25, 2014
Copyedited August 10, 2014

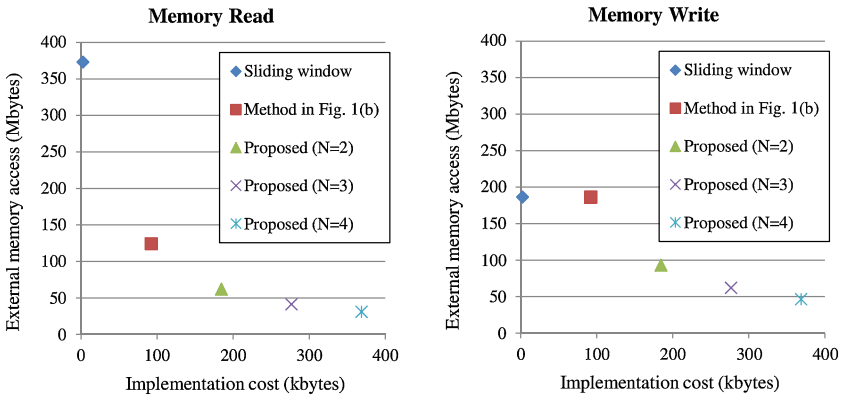

Fig. 4. Relationship between external memory access and implementation cost for reading and writing reference frames. Here, N represents the depth of frame pipeline

considers only external memory access to read and write reference frames. The reduction of the external memory access for reading the current frame is required to further save the total memory bandwidth.

Acknowledgments

This present research has been conducted by the Research Grant of Kwangwoon University in 2013. This research was supported by a fund from the Forensic Research Program of the National Forensic Service, Korea.

DOI: 10.1587/elex.11.20140592
Received June 19, 2014
Accepted July 3, 2014
Publicized July 25, 2014
Copyedited August 10, 2014