

# A Flexible Two-Layer External Memory Management for H.264/AVC Decoder

Chang-Hsuan Chang, Ming-Hung Chang, and Wei Hwang, *Fellow, IEEE*

Department of Electronics Engineering & Institute of Electronics,  
National Chiao-Tung University, HsinChu 300, Taiwan  
[hsuan.ee89@nctu.edu.tw](mailto:hsuan.ee89@nctu.edu.tw), [hwang@mail.nctu.edu.tw](mailto:hwang@mail.nctu.edu.tw)

## ABSTRACT

In this paper, a flexible two-layer external memory management for H.264/AVC decoder is proposed. Power consumption and data access latency caused by being fetched to/from the off-chip memory greatly affect multimedia system performance. The proposed memory controller consists of two layers. The first layer is the address translation which provides an efficient pixel data arrangement to reduce the row-miss occurrence. The second layer is the external memory interface (EMI) which can further reduce access latency up to 70% by using the specific command FIFO and a unified FSM with generic scheduling. Particularly, the memory utilization can be increased about 3 times as compared with traditional method after combining the address translation layer with external memory interface. Similarly, the proposed memory controller unit is feasible and beneficial for future memory-bandwidth-constraint System-on-Chip applications.

## I. INTRODUCTION

Memory sub-system dominates the performance of a video processing system in terms of access latency, bandwidth utilization, and power consumption. Data in a video decoder is usually stored in off-chip memories in view of cost efficiency. In order to achieve the higher coding efficiency, several advanced coding methods in H.264/AVC require a large amount of data transferred to/from the off-chip memory for pixel prediction, de-blocking, and motion prediction. As the resolution of video processing applications becomes higher, data exchange between processors and external memory should be completed within a limited time period to achieve real-time requirement. However, the bandwidth of the off-chip memory is limited due to the scarce

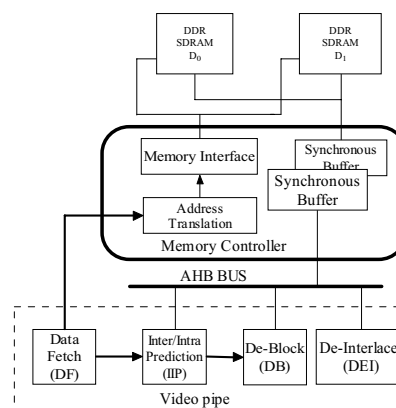


Figure 1: The Memory Subsystem of H.264 Decoder

resource of I/O pins. An efficient memory management is desperately needed to loose bottle neck.

Several literatures are dedicated to improve the performance of the memory controller within a video decoder. Firstly, memory interface architecture [1] recognized an efficient data arrangement in SDRAM to increase the row-hit rate. However, its address translation method is not well suitable for H.264/AVC decoder. Secondly, history-based memory mode controller [2] reduces row-miss rate. Energy consumption and memory latency are reduced to 23.3 % and 18.8 % respectively. Nevertheless, the history-based prediction results in additional overhead cycles when the prediction is not correct. Thirdly, SDRAM controller in H.264/AVC HDTV decoder [3] focuses on efficient memory mapping and data arrangement in SDRAM. However, it adopts auto pre-charge rather than manual pre-charge leads to some loss of bus bandwidth and increase the access latency.

To improve memory bandwidth and power consumption in video applications, we propose a

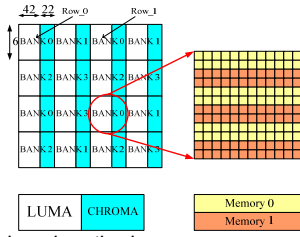


Figure 2: Interlaced method

memory sub-system to efficiently manage the off-chip memory for H.264/AVC decoder as shown in Fig. 1. There are three modules accessing the external memory via AHB-bus and memory controller. 32-bit mobile double-data-rate (DDR) synchronous DRAMs are operating at 162 MHz to deliver sufficiently high levels of total memory bandwidth. The proposed H.264/AVC decoder targets on high profile at level 4. On-chip ping-pong structure SRAM stores row data to optimize the accesses to the external memory. In memory controller, address translation (AT) is designed for H.264/AVC to generate the physical addresses of external memory according to the request of each internal module. External memory interface (EMI) generates associated commands to external memory based on the output of AT. In addition, EMI can be configurable for different types of external memory. The performance of the memory controller can be improved by AT with EMI.

The rest of the paper is organized as follows: Section II describes the detail of address translation (AT). Section III presents the detail of external memory interface (EMI). Section IV shows the simulation results and Section V concludes this paper.

## II. ADDRESS TRANSLATION

Data arrangement in the external memory has a great effect upon the memory bandwidth. Once row-miss occurs during the access of the external memory, the original row should be pre-charged and then re-activate the new row. Therefore, the overhead cycles are introduced so as to decrease the memory bandwidth. In addition, row-activation and pre-charge operations induce much more power consumption than read/write access. The dynamic energy consumption of the SDRAM is computed by the following formula:

$$E_{dynamic}^i = N_{active} \cdot E_{active} + N_{precharge} \cdot E_{precharge} + N_{rw} \cdot E_{rw} \quad (1)$$

According to the data sheet provided by Micron [5], each active or pre-charge access expend

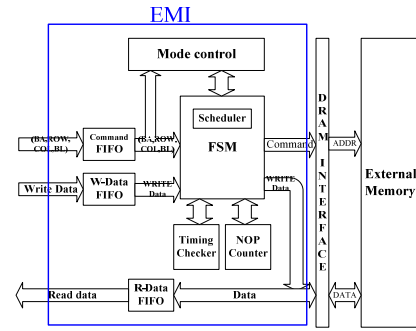


Figure 3: Architecture of EMI

14000 PJ while each read or write access only consumes 2000 PJ. In other words, active and pre-charge commands require seven times power consumption than the read/write commands. Thus, reducing the number of active and pre-charge is crucial in the view of low power application. It means a good data arrangement is essential to minimize the probability of row-miss.

A novel data arrangement suitable for H.264/AVC decoder is proposed to increase the memory bandwidth and reduce the power consumption. In order to minimize the number of active and pre-charge, chessboard-based arrangement memory mapping is presented as shown in Fig. 2. It is further compounded with the fact that Luma and Chroma are placed interleaved.

The interlaced memory mapping method put the luminance block and chrominance block in the same row of the bank. Because the decoder accesses a chrominance block after each luminance block, it doesn't need to re-activate the row when accessing the chrominance block. Thus, it leads to the latency and power consumption reduced.

In addition, the targeted memory is changed every two lines of pixels. The data, then, is equally distributed in two external memories no matter which block are transmitted. A motion-compensated block could be a frame block or a field block because H.264/AVC supports adaptive frame/field coding. Similarly, the de-blocking module may write a frame block or a field block into the external memory. This kind of data arrangement guarantees the data is equally distributed in both frame and field coding. Thus, the transmission cycles could be minimized.

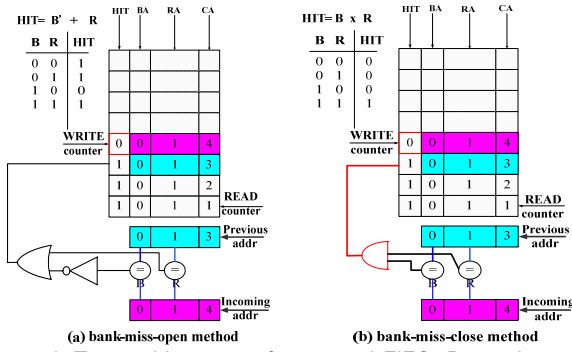


Figure 4: Two architectures of command FIFO. B equals to one means bank hit. R equals to one means row hit.

### III. EXTERNAL MEMORY INTERFACE

To decrease the latency of row-miss and bank-miss status, an efficient external memory interface shown in Fig. 3 is proposed. Specifically, the physical addresses produced by AT are stored in specific command FIFO. Finite state machine (FSM) combined with mode control block, timing checker, and counter generates the appropriate commands to external memory. Data is transferred via W-data FIFO and R-data FIFO. In addition, the proposed EMI is configurable for different type of external memory by setting several particular parameters.

#### A. Command FIFO

The specific command FIFO adaptively makes use of auto-pre-charge capability. It can reduce the latency of row-miss while prior art has penalty in row-hit case. As shown in Fig. 4, the addresses are stored in both of the command FIFO and previous-address register (PAR). The incoming address is compared with PAR. If bank address and column address are the same as PAR, we set hit bit of the previous command to one. It leads to auto-pre-charge capability turned off. Otherwise, the hit bit remains zero such that auto-pre-charge capability turns on to reduce the latency of row-miss.

In addition, two dynamic methods are also proposed. They are bank-miss-open and bank-miss-close for the decision of bank-miss, respectively. When bank-miss occurs, the original bank could be pre-charged or remain row-open. Pre-charging the original bank has benefit of less latency when the next access to this bank is to/from another row. However, it results in the overhead cycles when the next access to this bank is to/from the same row. Thus, the benefit of auto-

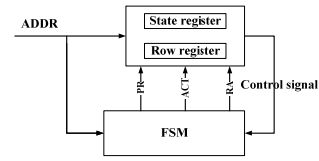


Figure 5: Mode control block

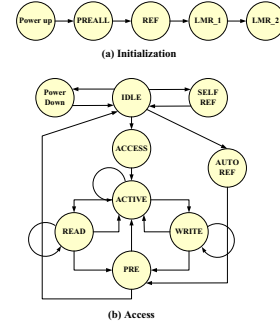


Figure 6: Finite State Machine

pre-charge for bank-miss depends on the probability of accessing the same row when returning to the original bank. The bank-miss-open method as shown in Fig. 4(a) is suitable for video application which has high probability of accessing the same row in a bank. The EMI keeps the row of the current bank remained open if the next command leads to bank-miss. On the contrary, the bank-miss-close method as shown in Fig. 4(b) is suitable for random process which has high probability of accessing the different rows in a bank. The EMI automatically pre-charges the current bank if the next command leads to bank-miss. Therefore, our specific command FIFO has advantage of reducing the access latency.

#### B. Mode Control Block

The mode control block that records the state and the active row of each bank is used to control the finite state machine block as shown in Fig. 5. Each bank could be in idle or active state. Idle state means there is no row open in this bank. Active state means that the bank has a particular open row whose address is recorded in a row register. Several signals such as pre-charge (PR), active (ACT), and row address (RA) from finite state machine is used to update the state register and the row register in mode control block. When the current command is updated, the mode control block will compare the current command with the state register and row register to generate the control signal that can decide next state of finite state machine.

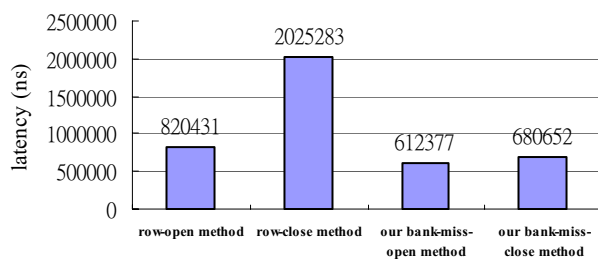


Figure 7. Latency of different method for video process

### C. Unified and Scheduled Finite State Machine

The functionality of the finite state machine (FSM) is to produce the appropriate commands to the external memory. Particularly, the operations of all the banks in the DRAM are controlled by one unified FSM instead of multiple FSMs. A common scheduling method similar to [2] is used to minimize the overhead cycles. In addition, the FSM obeys timing constraints that are specified in each type of DRAM. The simplified state diagram of FSM is shown in Fig. 6. There are two major parts in the FSM. Shown in Fig. 6a, Initialization process deals with the process from power up to stable state for access. Access process as shown in Fig. 6b deals with operations including read, write, self refresh, auto refresh, pre-charge, and power-down such that the data could be correctly transferred under the consideration of power saving.

In summary, the purpose of EMI is to reduce the latency of bank-miss and row-miss status by using dynamic pre-charge method and a unified scheduler. In addition, the proposed EMI is independent of applications.

## IV. SIMULATION RESULTS

Fig. 7 shows that our dynamic pre-charge method can reduce the access latency 27%-70% as compared with traditional row-open and row-close method. We can adaptively control the auto-pre-charge capability because FSM can look ahead before accessing the next command by using the specific command FIFO. Therefore, the overhead cycles introduced by false prediction could be reduced. Particularly, the bank-miss-open method is more suitable for the application of H.264/AVC. The reason is the probability of accessing the same row is higher than the different row when returning to the original bank.

The proposed AT can reduce the probability of the row-miss occurrence in De-Blocking module and Intra-Inter-Prediction module to about 2.5% and 0.56% respectively. Considering the interaction of these two modules, the probability of row-miss occurrence is less than 4.5% and the probability of the row-hit status is about 85%.

## V. CONCLUSIONS

In this paper, a memory sub-system that contains a flexible memory controller for the H.264/AVC decoder has been presented. Our flexible controller contains two layers which are address translation (AT) and external memory interface (EMI). The AT layer combined with specific data arrangement in DRAM is designed for H.264/AVC decoder to decrease the number of row-miss status and bank-miss status. Therefore, the active and pre-charge operation can be minimized such that latency and power consumption are reduced. The EMI layer that contains specific command FIFO, mode control block, and unified and scheduled FSM can increase the bandwidth utilization by automatically pre-charge and scheduling. The proposed EMI layer can reduce the access latency up to 30% as compared to conventional approach in H.264/AVC application. Consequently, the memory utilization can be increased from 15% to 51% after combining address translation with external memory interface. Any future memory-bandwidth-constraint System-on-Chip applications can be benefited from the proposed memory management controller.

## REFERENCES

1. H. Kim, and I. C. Park, "High-performance and low-power memory-interface architecture for video processing applications," in *IEEE Trans. Circuits and Systems for Video Technology*, vol. 11, no 11, pp.1160–1170, 2001.
2. S. I. Park, Y. Yongseok, and I. C. Park, "High performance memory mode control for HDTV decoders," in *IEEE Trans. Consumer Electronics*, vol.49, no.4, pp. 1348–1353, 2003.
3. J. Zhu, L. Hou, R. Wang, C. Huang, and J. Li, "High performance synchronous DRAMs controller in H.264 HDTV decoder" in *Proc. IEEE Int. Conf. Solid-state and Integrated Circuits Technology*, vol. 3, pp.1621–1624, Oct. 2004.
4. Chih-Hung Li, Wen-Hsiao Peng, and Tihao Chiang, "Design of Memory Sub-System with Constant-Rate Bumping Process for H.264/AVC Decoder," in *IEEE Trans. on Consumer Elec.*, vol. 53, Issue 1, pp. 209-217, Feb. 2007.
5. Micron, <http://www.micron.com>.