Displays 40 (2015) 62–67

Contents lists available at ScienceDirect

Displays

journal homepage: www.elsevier.com/locate/displa

# An efficient memory control method for video and image processing in digital TV

Chanho Lee

*School of Electronic Engineering, Soongsil University, Seoul, Republic of Korea*

ARTICLE INFO

*Article history:*
Received 5 October 2014
Received in revised form 6 February 2015
Accepted 3 May 2015
Available online 22 May 2015

*Keywords:*
Memory controller
Image signal processing
Address generation
SDRAM

ABSTRACT

High definition (HD) and ultra-high definition (UHD) digital TV require high-resolution images and lots of data transfers between processors and memory devices often become the bottleneck of the system. Video and image signal processing usually require blocks of square or rectangular shaped pixel data for signal processing. It requires frequent precharging and activating new rows, and results in extra latencies for reading and writing pixel data in memory devices. This paper proposes an efficient memory controller for video and image processing to reduce the latencies for reading and writing blocks of pixel data. The controller stores a frame of pixel data by distributing contiguous lines of pixel data to multiple banks in sequence. Its efficiency is enhanced more with an interface protocol such as AMBA AXI in which outstanding transactions are allowed. Memory controllers according to the proposed scheme are designed and the performance and the efficiency are compared with the previous works.



## 1. Introduction

A digital image consists of two-dimensional array of pixel data and a video consists of multiple digital image frames. The number of pixels in an image frame depends on the resolution of image. The resolution of digital TVs has increased from standard definition (SD, $720 \times 480$) to ultra-high definition (UHD, $3840 \times 2160$ and $7680 \times 4320$) according to the demands of high quality videos. Video and image processing operations usually require blocks of two-dimensional pixel data in a frame [1]. The size of the block varies from $2 \times 2$ pixels to $128 \times 128$ pixels depending on the image processing operations [1,2]. The shape of the block is a square or a rectangle.

The size of pixel data in one frame of 4 k UHD image is about 33 MB for 32bit RGBA format. In general, the pixel data are stored in the memory according to the same shape as the frame of an image in order to reduce the amount of calculation to obtain addresses for reading and writing. As a result, it requires the larger memory space for the pixel data of a frame than 33 MB because some memory spaces are unused unless the numbers of vertical or horizontal pixels are coincident with the powers of 2. In addition, several frames of pixel data are required for lots of image and video processing, and the memory space for frame data is large in most cases. The memory devices for frame data are usually SDRAM installed on the same PCB as off-chip devices. There are waiting cycles when the next data are stored in a different row in DRAM devices. The pixel data of a frame and therefore a block are usually stored in several contiguous rows of a bank in two-dimensional shape. New rows are opened and closed several times while reading a block of pixel data, and the waiting cycles are accumulated. In order to solve the problem, required data can be read to a buffer memory in advance so that they can be fed to processing units immediately. As the size of a block becomes larger, the size of the buffer memory need to be larger. In addition, prefetching pixel data using a buffer memory is not applicable when the addresses of the data for the next processing are not known in advance.

In order to eliminate changing rows when reading or writing a block data, the two-dimensional array of the block can be converted to a one-dimensional array so that the block data can be stored contiguously in a row. However, the waiting cycles still arise when the size of blocks are variable, for example, in compressing and decompressing movies because the data of larger blocks other than basic blocks may not be contiguously stored. The block array conversion method sometimes degrades the performance of memory access. For example, when neighboring pixel data are required for motion compensation of a macroblock in H.264 decoder, the neighboring blocks of the block need to be read although all pixels of the neighboring blocks are not necessary. The converted one-dimensional array needs to be converted to a 2-dimensional array again for a display unit. Moreover, complex calculation for addresses and the corresponding datapath circuits are required.

*E-mail address:* chlee@ssu.ac.kr

http://dx.doi.org/10.1016/j.displa.2015.05.002
0141-9382/

The array conversion method seems to work for a specific processing of blocks with a fixed size. However, the waiting cycles are unavoidable in most of real applications.

The problem of the waiting cycles due to reading and writing a block of pixel data in a frame memory can be solved using multiple banks of DRAMs. Each row of a block is stored in a different bank so that the bank is also changed when the row is changed. Changing a bank does not generate waiting cycles if the bank is already open.

Efforts to increase the performance of memory access in image signal processing system using the multiple banks have been reported several times [7–12]. One of them is reducing waiting cycles by analyzing motion image decoding processes and estimating the next addresses. Another effort is employing cache memories to reduce the number of access to a DRAM. The two methods can be applied to specific systems and dedicated memory controllers are required. Another approach reduces waiting cycles by sending PRECHAGRE and ACTIVE commands early to memory controllers of systems supporting outstanding address communications when addresses are provided in advance. It does not reduce waiting cycles for the data in different rows of the same bank although it can reduce waiting cycles due to PRECHARGE and ACTIVE operations between memory access requests to the data in different banks.

In this paper, I propose a memory controller architecture for image processing systems which requires two-dimensional block data of square or rectangular shapes. The proposed memory controller changes the bank when changing row is required so that block data are written or read without waiting cycles in system with an on-chip network supporting outstanding address transactions. A simple conversion of the address of the data is enough for the proposed operation of changing bank without modifying the interface logic of masters and it does not increase the complexity of hardware. The proposed memory controller can increase the performance of the video codec and the image processing system in digital TVs with high resolution and it also works for general data other than block data of rectangular shape without the degradation of performance.

## 2. Operations of conventional memory controller

### 2.1. Architecture of SDRAM

SDRAM is widely used as a main memory of IT devices since its manufacturing cost is the lowest. The 1 transistor 1 capacitor (1T1C) structure which enables the economical manufacturing has the disadvantage that the stored information can be lost due to leakage current. The problem is solved by REFRESH operation which refreshes the stored periodically. DRAM also adopts address multiplexing which split a destination address into two pieces and supplies them one by one in order to reduce the manufacturing costs by reducing the number of pins of a package. As a result, the performance is degraded and the architecture of the controller becomes complicated. The performance of SDRAM has been increased by introducing the interface of double data rate (DDR). DDRx($x$ = 1, 2, 3, 4) SDRAMs transfer data at both clock edges and the operating frequencies are greatly increased [3–6].

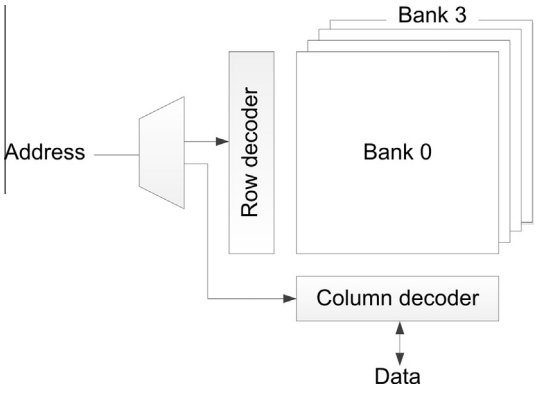


**Fig. 1.** Block diagram of a conventional DRAM with 4 banks.

Memory architecture of DRAMs consists of several banks which can be controlled separately as shown in Fig. 1. The memory controllers generate bank information from the address of the data, and the bank information is delivered to a DRAM so that a bank is selected for a read or a write operation. The bank system enables more efficient control of DRAMs when reading and writing data. The proposed algorithm utilizes the bank system in conjunction with the advanced interface protocol.

### 2.2. Waiting cycles of SDRAM

The maximum data bandwidth of SDRAMs, *BWmax* is represented as Eq. (1).

$$BW_{max} = dfW_{DQ} \ [bps] \tag{1}$$

where $f$ is the operating frequency, $W_{DQ}$ is the number data pins, and $d$ is 2 for DDRx SDRAMs and is 1 otherwise. However, the maximum bandwidth can be achieved instantaneously, and the average bandwidth is usually much smaller than that of Eq. (1). It is caused by waiting cycles in which memory cells are not reachable for reading or writing due to REFRESH, ACTIVE, and PRECHARGE. They are investigated further below.

#### 2.2.1. REFRESH

REFRESH operation prevents the DRAM from losing data due to the leakage currents. It requires the number of cycles proportional to that of rows in a bank since it is reading every cell row by row. The period of REFRESH is usually tens of milliseconds though it depends on operation voltage, temperature, and process variations. Every cell in a row should be read at least once in the period, and the memory cells are not accessible during the REFRESH operation.

#### 2.2.2. PRECHARGE

PRECHARGE operation precharges bit lines of a bank before a word line is activated. It closes a row of a bank when the row is no longer necessary for reading or writing. It also prepares to open a new row, and must precede the ACTIVE operation. It usually requires 2–3 cycles in single data rate (SDR) SDRAMs depending on the operation frequencies due to the large capacitance of the bit lines.

#### 2.2.3. ACTIVE

ACTIVE operation activates a word line of a bank corresponding to a destination address. It opens a row to read or write data. It follows the PRECHARGE operation. Once a row is open, it is accessible without another ACTIVE command until the PRECHARGE command is issued. It usually requires 2–3 cycles in single data rate (SDR) SDRAMs depending on the operation frequencies due to the large RC delays of the word lines.

Most of the SDRAM contains more than 4 banks, which can be controlled independently. For example, bank 1 can be activated or bank 2 can be precharged while bank 0 is active. Using the properties, ACTIVE or PRECHARGE command for another bank can be issued if the next request is available before the current request is finished while waiting for data in reading operation for a bank. Otherwise, NOP commands are issued. It will increase the effective bandwidth by issuing PRCHARGE or ACTIVE command for another bank in advance.