Design of eMMC Controller with Multiple Channels

Chulhoon Kim
Department of Electronic Engineering
Soongsil University
Seoul, Korea
ch.kim@ssu.ac.kr

Chanho Lee
Department of Electronic Engineering
Soongsil University
Seoul, Korea
chlee@ssu.ac.kr

Abstract— **Embedded multimedia card (eMMC) is expected to replace secure digital (SD) card which is widely used for external memory and to be used widely in the embedded systems due to the improved performance and package. In this paper, we propose architecture of eMMC controller with multiple channels. It is connected to a host system using an AXI master interface for data transfer and an APB slave interface for writing command and reading responses and status. The interface for eMMC devices has multiple channels for multiple devices and each channel can be enabled so that multiple processors can request memory access. An eMMC controller is designed based on the proposed architecture using Verilog-HDL and is implemented using an FPGA.**

Keywords – eMMC; Multi-Channel; AXI; APB; Verilog-HDL; DMA Controller; SoC

I. Introduction

Embedded multimedia card (eMMC) [1] is a high performance non-volatile memory which includes NAND flash memory devices and a controller, and is expected to replace secure digital (SD) cards. Multiple channels can increase the data bandwidth since the data width of an eMMC device is limited to 8 bits. The simultaneous operation of multiple channels is not effective for multiple requests of multiple processors although it increases the data bandwidth.

In this paper, we propose an eMMC controller architecture with multiple channels for multiple requests of multiple processors. It consists of an application adaptor and eMMC adaptors. The application adaptor includes an AMBA APB slave interface [2] and an AMBA AXI master interface [3]. The APB interface is used for configuring the controller, writing eMMC commands and reading eMMC responses. The AXI interface is used for the data transfer. The application adaptor enables the eMMC adaptors. The eMMC adaptor generates commands, controls data transfer, and analyzing responses. The proposed eMMC controller architecture is designed using Verilog-HDL, and is implemented using an FPGA.

II. Architecute of proposed controller

A. Application adaptor

The APB slave interface of the application adaptor is used by a host system to configure the controller, writing eMMC commands and reading eMMC responses. It also includes command and argument queue so as to issue the next command without latency when the controller receives a response. The AXI master interface includes a direct memory access controller (DMAC) which reads and writes data in a host system. It sends enable signals to eMMC adaptors, and divides and combines data when more than one channels are enabled. It can distinguish the commands and the responses of the multiple channels. Other functions are similar to that of the controller with a single channel [4] except for the added functions.

B. eMMC adaptor

The eMMC adaptor is the core component of the controller. A channel controller is connected to one eMMC device to transfer data. The channel controller can be enabled or disabled by a host system through the APB interface. It issues commands at the right timing and receives responses to analyzed and to deliver them to the APB interface. It has a buffer to hold a block of data which is the minimum transfer unit between an eMMC adaptor and a device. Since the DMAC also has a buffer, we can increase the performance using double buffering [5]. The basic structure is similar those of the controller with a single channel [4]. Fig. 1 shows the block diagram of the proposed eMMC controller with multiple channels.

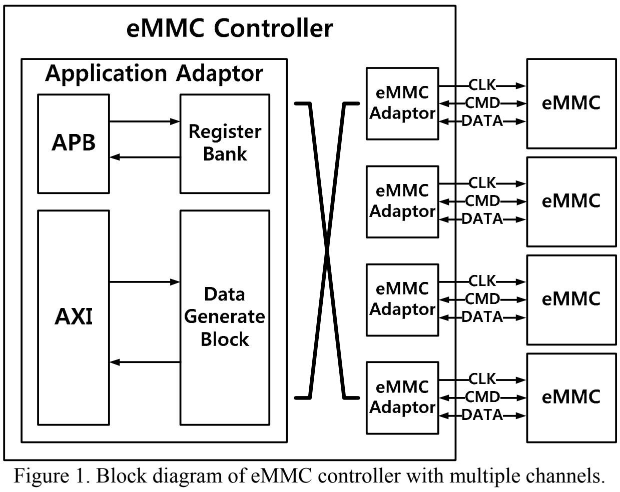

Figure 1. Block diagram of eMMC controller with multiple channels.

III. Operation of controller

A processor should configure the eMMC controller with multiple channels to determine which channel is enabled before

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MOE) (NRF-2016R1D1A1B01008846) and the Industrial Core Technology Development Program (10049192, Development of a smart automotive ADAS SW-SoC for a self-driving car) funded by the Ministry of Trade, industry & Energy. The EDA tools were supported by IDEC.

it requests a memory access. If all the channels are enabled and a memory access is requested, the application adaptor divides the data by 8 bits and distributes them to all the channels for writing request, or collects 8 bit data form each channel and combines them. The data bandwidth increases by the number of enabled channels. Multiple eMMC devices act as a single device with the increased transfer rate.

Processors can write data to specific eMMC devices since the controller transfers data to eMMC devices which are connected to enabled channels only. We can assign a channel or channels to a processor so that the processor can have a dedicated eMMC device. A processor enables its channel before it transfers data normally, and enables another channel when data sharing is necessary. We can give variable data bandwidth to each processor by assigning different number of channels. Therefore, the controller can process multiple memory access requests from multiple processors by enabling each channel independently.

IV. SIMULATION RESULT AND IMPLEMENTATION

A eMMC controller with 4 channels is designed using Verilog-HDL based on the proposed architecture and is verified by simulation using an eMMC simulation model and ModelSim. Fig. 2 shows the simulation environment. The AHB master behaves processor operations and the test slave stores data in the host system.

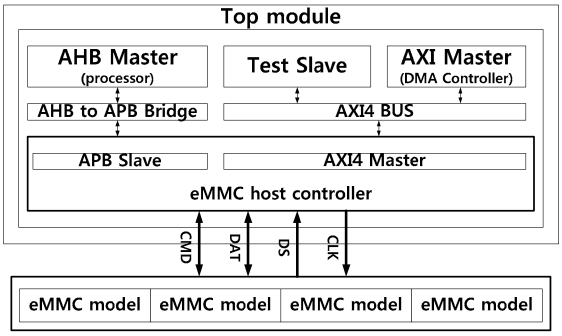

Figure 2. Simulation environment.

Fig. 3 shows the simulation results of the write data operation when all the channels are enabled. The eMMc devices operate in HS400 mode using 8bit data bus [6]. The AHB master writes 2KB data which are divided into four 512B blocks. Each data block is written to the corresponding eMMC device simultaneously. The data bandwidth increases by 4 times compared with that of a single channel operation.

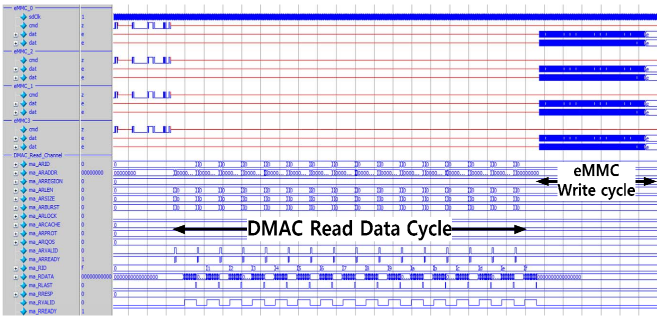

Figure 3. Simulation results of the write data operation with 4 channels enabled.

Fig. 4 shows the simulation results of reading 512B data by 4 processors with one channel enabled at a time. The simulation environment is the same as that of Fig. 3. It shows that data are read through only one channel. Each processor read data from its own eMMC device.

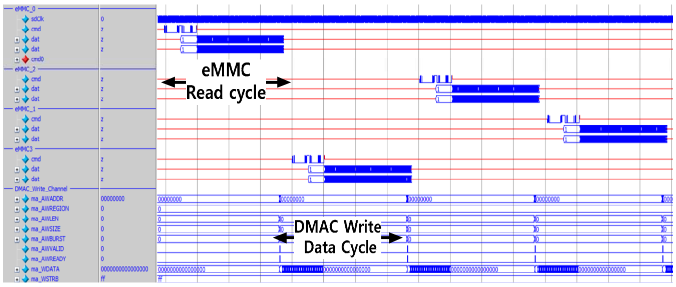

Figure 4. Simulation result of the read data operation with one channel enabled at a time.

Table 1 shows the synthesis result of the eMMC controller using Synopsys Design Compiler and a 0.18um CMOS standard library.

TABLE I. SYNTHESIS RESULT WITH EMMC DEVICES

Maximum Frequency	285MHz
Logic Gate Count	42,085
SRAM size in buffers	12.2KB

V. CONCLUSTIONS

We propose an eMMC memory control architecture and design a controller with multiple channels. Each channel can be enabled before a request of memory access and variable configurations are possible. The controller can operate with increased data bandwidth or process multiple requests of multiple processors for the dedicated devices. The operation is verified by simulation for various requests. The maximum operation frequency is 285MHz and the area for logic is 42,085 gates with a 0.18um CMOS standard library.

REFERENCES

[1] JEDEC, Embedded Multi-Media Card (eMMC) Electrical Standard(5.1), JEDEC JESD84-B51, February 2015.

[2] ARM, AMBA APB Protocol Specification., ARM IHI 0024C, April 2010.

[3] ARM, AMBA AXI and ACE Protocol Specification., ARM IHI 0022E, Febuary 2013, pp.22-122.

[4] Chulhoon Kim and Chanho Lee, “Design of programmable high performance eMMC controller,” IEIE SoC Conference, Seoul, Rep. of Korae, May 2016.

[5] Manil Dev Gomony, Benny Akesson, and Kees Goossens, “Architecture and optimal configuration of a real-time multi-channel memory controller,” Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, France, pp. 1307–1312, March 2013.

[6] Ning Fu, Yuanheng Li, Baowen Liu, Hongwei Xu and Yigang Zhang, “Realization of controlling eMMC 5.0 device based on FPGA for automation test system,” IEEE AUTOTESTCON, National Harbor, MD, pp. 251–255, November 2015.