Journal of KIIT. Vol. 20, No. 9 pp. 63-71, Sep. 30, 2022. pISSN 1598-8619, eISSN 2093-7571
http://dx.doi.org/10.14801/jkiit.2022.20.9.63

Memory Controller for 1Gb LPDDR2 Memory Initialization Sequence

Jin-Su Kang*[1], Chang-Yong Lee*[2], and Yong-Hwan Lee*[3]

This research was supported by Kumoh National Institute of Technology(202000980001)

Abstract

Each memory has its own initialization sequence before proceeding with its normal operation. This sequence must be performed as a step for the memory to operate normally without errors. In this paper, we propose a memory controller sequence that automatically performs the initialization sequence of 1Gb LPDDR2(Low Power Double Data Rate). The memory initialization sequence is divided into three steps: delay time section, memory setting, and timing calibration. The memory controller calculates the delay, adjusts the initial set value of the memory according to the user's setting, and writes and reads data of a specific pattern for timing calibration. If these steps are performed externally, unnecessary time consumption and accurate operation are difficult. By using the memory controller proposed in this paper, it is possible to proceed with an accurate memory initialization sequence without unnecessary time consumption.

요 약

각 메모리들은 정상 동작을 진행하기 전에 각자의 초기화 시퀀스가 존재한다. 이러한 시퀀스는 메모리가 오류 없는 정상 동작을 하기 위한 단계로서 필수적으로 진행되어야 한다. 본 논문에서는 1Gb LPDDR2(Low Power Double Data Rate)의 초기화 시퀀스를 자동으로 진행해 주는 메모리 컨트롤러의 시퀀스를 제안한다. 메모리의 초기화 시퀀스는 딜레이 타임 구간, 메모리 설정, 타이밍 조정 총 3단계로 나누어져 있다. 메모리 컨트롤러는 딜레이를 계산하고, 사용자의 설정에 맞추어 메모리의 초기 설정값을 조정하며 타이밍 조정을 위한 특정 패턴의 데이터를 쓰고 읽는다. 이러한 단계를 외부에서 진행하면 불필요한 시간 소모 및 정확한 운용이 어렵다. 본 논문에서 제안하는 메모리 컨트롤러를 사용하면 불필요한 시간 소모 없이 정확한 메모리 초기화 시퀀스의 진행이 가능하다.

Keywords

LPDDR2 memory, initialization sequence, memory controller, calibration, FPGA

* Dept. of Electronic Engineering, Kumoh National Institute of Technology (*[3] Corresponding Author)
- ORCID[1]: http://orcid.org/0000-0001-9943-448X
- ORCID[2]: http://orcid.org/0000-0002-8868-3373
- ORCID[3]: http://orcid.org/0000-0003-1222-8283

· Received: Jul. 29, 2022, Revised: Aug. 11, 2022, Accepted: Aug. 14, 2022
· Corresponding Author: Yong-Hwan Lee
Dept. of Electronic Engineering, Kumoh National Institute of Technology
Tel.: +82-54-478-7432, Email: yhlee@kumoh.ac.kr

Ⅰ. Introduction

Recently, technology-related research using large-capacity data or high-resolution image data has been actively conducted. In addition, technologies for recognizing and processing a rapidly moving object are rapidly developing[1]. The technology for recognizing a moving object requires a low-power, high-performance memory to recognize a moving object because fast processing of the memory is essential. In order for the low-power, high-performance memory to operate normally, an accurate initialization sequence must be performed in advance. The initialization sequence of the 1Gb LPDDR2(Low Power Double Data Rate 2) memory targeted in this paper consists of three sections: time delay, memory register setting, and read timing adjustment[2].

In order to proceed with the time delay step, it is necessary to calculate a time delay that must elapse before some signals are transmitted and to execute a specific operation. In the memory register setting step, a specific instruction must be transmitted to the memory according to the user's setting. In the read timing adjustment step, it is necessary to write and read data of a specific pattern. If a specific operation and delay calculation suitable for these steps are externally performed, unnecessary time is consumed and it is difficult to proceed with an accurate initialization sequence. In this paper, the initialization sequence is performed using the memory controller to minimize unnecessary time consumption and to proceed with the correct initialization sequence.

In addition, the PHY modeling that replaces the role of the PHY that connects the memory and the memory controller is designed and the test using the simulation is performed instead of the role of the PHY. In the case of PHY modeling, the clock generation, signal transmission delay, and read timing calibration algorithms that the existing PHYs are designed to be performed instead are designed and tested.

The memory controller can only pass signals to the memory through the PHY[3]. Many signals are connected between the memory controller and the PHY, including the signals transmitted to the memory and the signals for the operation of the PHY.

Table 1 shows the signals connected to the PHY based on the memory controller. Of the signals in Table 1, only Rise_CA, Fall_CA, Clock_Enable, Chip_Select, and Clk_base are transferred to the memory, and data and data signals transferred from the PHY to the memory are additionally generated and transferred using other signals.

Table 1. Signals connected to the memory controller and the PHY

Signal port	Direction	Width
Clk_Base	Input	1
Clk_Div	Input	1
PLL_Locked	Input	1
Reset_n	Input	1
DQ_Rise_Data	Input	32
DQ_Fall_Data	Input	32
Logic_Rise_Data	Output	32
Logic_Fall_Data	Output	32
Rstrobe	Output	1
Rise_CA	Output	10
Fall_CA	Output	10
Clock_Enable	Output	1
Chip_Select	Output	1
ZQ_Init	Output	1
ZQ_Short	Output	1
Read_cal_Enable	Output	1
Read_Enable	Output	1
Write_Enable	Output	1

Ⅱ. LPDDR2 Memory

The target memory in this paper is FIDELIX's 1Gb LPDDR2 Memory. The memory can transmit data in 32-bit units and has 8 banks, 512 rows, and 8192 columns. It has signals including CA(Command Address) for allocating commands to memory as pins.

Table 2 shows the input/output pins of FIDELIX's 1Gb LPDDR2 memory. CK and CK# are a clock for operating a memory and an inverted clock of the clock.

Table 2. FIDELIX's 1Gb LPDDR2 memory pin

Signal port	Direction
CK, CK#	Input
CKE	Input
CS#	Input
DQ0-DQ31	Input / Output
DQS0-DQS3 DQS0#-DQS3#	Input / Output
CA0-CA9	Input

The CKE signal is Clock Enable and must remain HIGH when receiving a clock. CS# is a Chip Select signal, and it must be kept LOW when transmitting a command to the corresponding memory. DQ0 – DQ31 are pins in charge of input/output of data, and DQS is a signal for input/output of 8 DQ pins. DQS0 is responsible for DQ0-DQ7, DQS1 is for DQ8-DQ15, DQS2 is for DQ16-DQ23, and DQS3 is for DQ24-DQ31.

2.1 Memory time delay initialization sequence

In LPDDR2, there is an initialization sequence using a timing delay before memory operation[4][5]. This step is necessary because other signals can be connected after a specific time has elapsed from the moment when the power is first turned on, and there are signals that can be connected after a specific signal is input from the connected signal and time has elapsed. In this step, the memory controller measures the time delay by counting the clock using CLK_Div. In addition, a necessary command is transmitted to the memory together with other signals using the CA signal.

Fig. 1 shows the time delay initialization process of LPDDR2. As shown in Fig. 1, because the time that must be satisfied between certain operations is all different, the memory controller must use clock counting to calculate the corresponding time delay[6]. In addition, a specific operation must be performed between time delays, and the output of the signal must be adjusted. The Memory RESET command resets the memory using the Memory Register Write (MRW) command using CA in the memory.

Table 3 shows the MRW Command of LPDDR2 Memory. LPDDR2 uses the MRW instruction to reset the memory. For Memory Reset, 8'h3F must be given to MA[7:0], and the memory is reset regardless of what value is given to OP[7:0]. After the LPDDR2 memory executes the Memory Reset command, the time delay initialization sequence is completed when the subsequent time delay is satisfied.

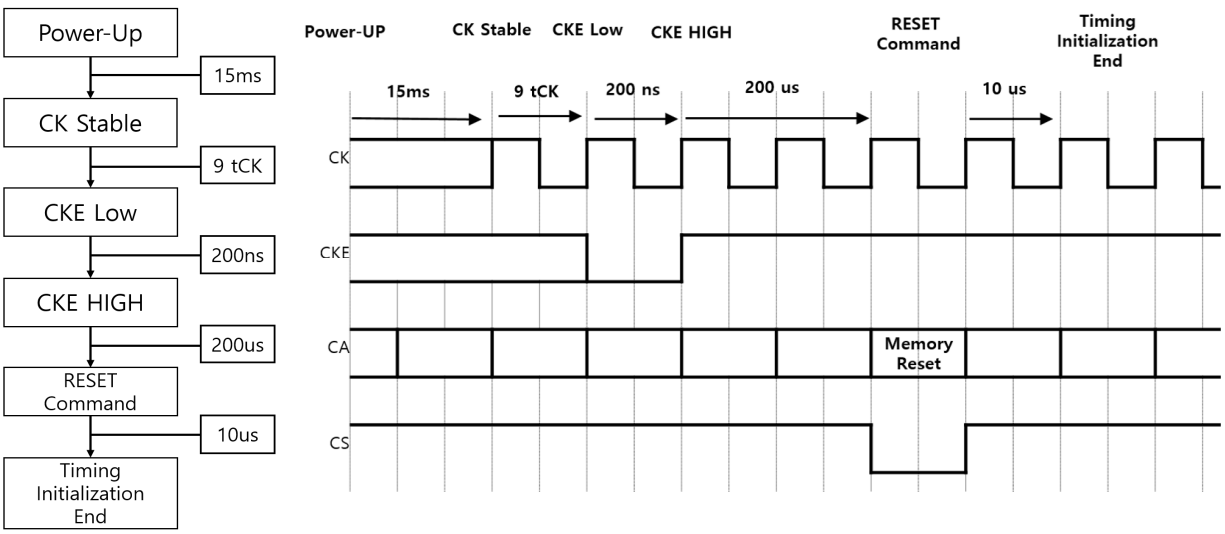

Fig. 1. Memory time delay initialization

Table 3. Memory command address(MRW)

Command	MRW(Mode Register Write)	
Edge	Rise edge	Fall edge
CA0	Low	MA6
CA1	Low	MA7
CA2	Low	OP0
CA3	Low	OP1
CA4	MA0	OP2
CA5	MA1	OP3
CA6	MA2	OP4
CA7	MA3	OP5
CA8	MA4	OP6
CA9	MA5	OP7

2.2 Memory register setting

When the memory's time delay initialization sequence is complete, you need to set a register in the LPDDR2 memory using the MRW instruction. The register initially sets only the values related to the operation of the memory, and the values set by the user and the PHY are set in the memory and Long phase are performed. In the case of the PHY used in this paper, when executing the ZQ calibration initialization step of the memory, the ZQ calibration INIT step must be performed. ZQ calibration INIT in PHY requires 1152 clocks based on Clk_base when ZQINIT signal is received. The memory controller must provide the ZQINIT signal for PHY calibration while delivering the ZQ calibration initialization command to the memory. In addition, both the time required for ZQ calibration initalization in memory and the time delay for ZQ calibration INIT in PHY should be calculated. After completing ZQ calibration initialization, proceed with ZQ calibration Long and Zero using the same method.

Fig. 2 is a diagram showing the register settings in progress in memory. LPDDR2 memory requires a ZQ calibration step[7]. ZQ calibration can be set using the LPDDR2 memory register with a minimum value of 8'h0A. There are 4 types of ZQ calibration initialization long, short, and reset. In the memory initialization phase, only ZQ calibration initialization.

The ZQ calibration zero step is a step to input the value of 8'h00, which is the default value of the corresponding register. The 127 clock counting calculated after the ZQ calibration zero step is the clock counting used in the initialization step, and the maximum value of the time the command is applied to the LPDDR2 memory is used.

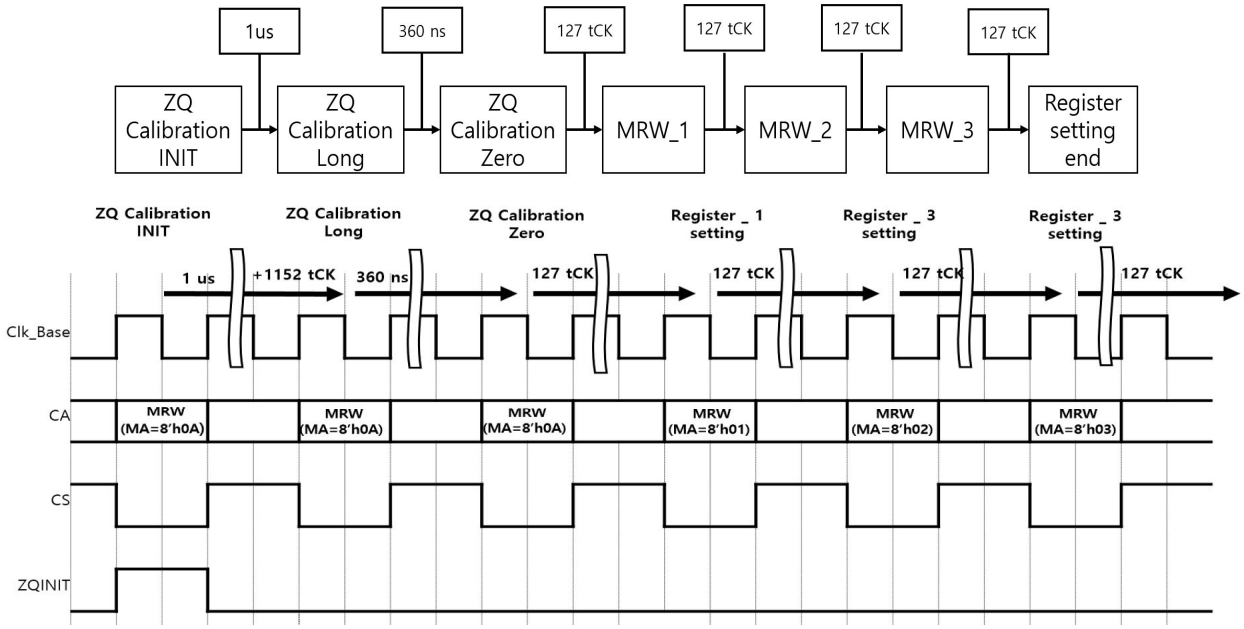

Fig. 2. Memory register setting

After ZQ Calibration is all completed, use the MRW instruction to input values to registers 1, 2, and 3 for initial memory setting. Registers 1, 2, 3 of LPDDR2 set values for basic operation of memory. Many initial values, including latency in memory and clock delay for recovery, are set based on the memory operation clock and user settings.

Talbe 4 is register 1 of LPDDR2 memory. Register 1 sets the values required in the memory write phase. In this paper, the memory operating frequency is set to 332MHz as the default, and the default value of register 1 has 8 Burst Length, Sequential, Wrap, and 6 Write Recovery.

Table 5 is register 2 of LPDDR2 memory. Register 2 sets the required clock delay after memory write/read commands. When the operating frequency is 332MHz, the values of 5 RL and 2 WL are initial values.

Table 4. Memory device feature_1(MR)

MR_1	Description
OP0	BL(Burst Length) 3'b010 : 4 BL 3'b011 : 8 BL 3'b100 : 16 BL All others : reserved
OP1	
OP2	
OP3	1'b0 : Sequential 1'b1 : Interleaved
OP4	1'b0 : Wrap 1'b1 : No Wrap
OP5	WR(Write Recovery) 3'b001 : 3 WR 3'b010 : 4 WR 3'b011 : 5 WR 3'b100 : 6 WR 3'b101 : 7 WR 3'b110 : 8 WR All other : reserved
OP6	
OP7	

Table 5. Memory device feature_2(MR)

MR_2	Description
OP0	RL(Read Latency)/ WL(Write Latency) 4'b0001 : 3 RL / 1 WL 4'b0010 : 4 RL / 2 WL 4'b0011 : 4 RL / 2 WL 4'b0100 : 6 RL / 3 WL 4'b0101 : 7 RL / 4 WL 4'b0110 : 8 RL / 4 WL
OP1	
OP2	
OP3	

Register 3 of LPDDR2 memory is used to set the resistance required for I/O configuration, and 48 ohm is used based on the PHY used in this paper. When ZQ calibration and register setting are completed using the MRW instruction, the memory setting step is completed during memory initialization.

2.3 Memory read timing calibration

Memory read timing calibration is a necessary step in order to transmit the data to the user without error from the memory controller when data is read from the memory. If this step is not performed, even if normal data is read from memory, normal data cannot be delivered to the user. In the case of read timing calibration presented in this paper, it is primarily performed in the PHY. In the case of read timing calibration performed in the PHY, the timing of the DQ used when data is output from the memory and the DQS connected to the corresponding DQ is adjusted[8]. Thereafter, data is transmitted to the memory controller in synchronization with the clock transmitted to the memory controller[9].

In order to perform read timing calibration in PHY, a specific pattern must be continuously output from memory. The memory controller writes a specific pattern to the memory for PHY read timing calibration and then operates the memory continuously by timing successive read commands. At this time, in order to write a specific data pattern to the memory, the storage space of the memory is activated based on the delay between operations of the memory, and the data is written. The memory controller delivers CA data for the ACTIVATE and WRITE commands. Based on the write latency, after the clock delay, the memory controller delivers the memory data for writing according to the characteristics of the PHY through Logic_Rise_Data and Logic_Fall_Data. At the same time, the controller transmits two signals, Write_Enable and Rstrobe, for the data write status of the PHY.

When writing of a specific data pattern is completed, data is read using the same sequence. When reading data, the controller calculates the read latency and sends successive READ commands so that the data is read continuously. As in the write phase, two signals, Read_cal_Enable and Read_Enable, are transmitted to read data from the PHY and proceed with the calibration phase. In the PHY, a maximum of 2048 clocks are required for read timing calibration, so the memory controller must calculate this and continuously transmit the signal.

After the sequence for PHY read timing calibration is completed, read timing calibration inside the controller is performed. The read timing calibration of the memory controller is a step to match the read timing calculated based on the READ command and the minimum clock delay and the actual read timing. This step writes and reads specific data similar to the step for PHY read timing calibration. However, since the read timing calibration of the memory controller is performed based on the READ command, the data timing is checked after one READ command. If the two timings do not match, add a clock delay and thenexecute the READ command again to correct the finally calculated read timing and the actual data read timing.

Fig. 3 is a diagram of the read timing calibration performed in the controller. As shown in the figure, the calculated read timing, which is the sum of the read latency and the latency generated by the internal logic, and the timing of the data output from the real memory are mostly different. At this stage, Fig. As in 3, add latency to correct the calculated read timing to match the actual data read timing. Since the read timing calibration of the memory controller is performed for each DQS group, in the case of the 1Gb LPDDR2 memory presented in this paper, a total of 4 read timing calibrations are performed. When the read timing calibration is completed 4 times, the activated memory is deactivated using the PRECHARGE command to return to the initial state for normal operation. If the read timing does not match even if latency is continuously added in the calibration stage of the memory controller, the memory controller presented in this paper can use an additional register to input the additional latency of the corresponding stage to complete the memory initialization. As the initialization of the memory is completed in this way, the preparation for the normal operation of the memory is completed.

III. Additional features

In the memory controller presented in this paper, in addition to the sequence for memory initialization, it is possible to test the normal operation after the completion of memory initialization.

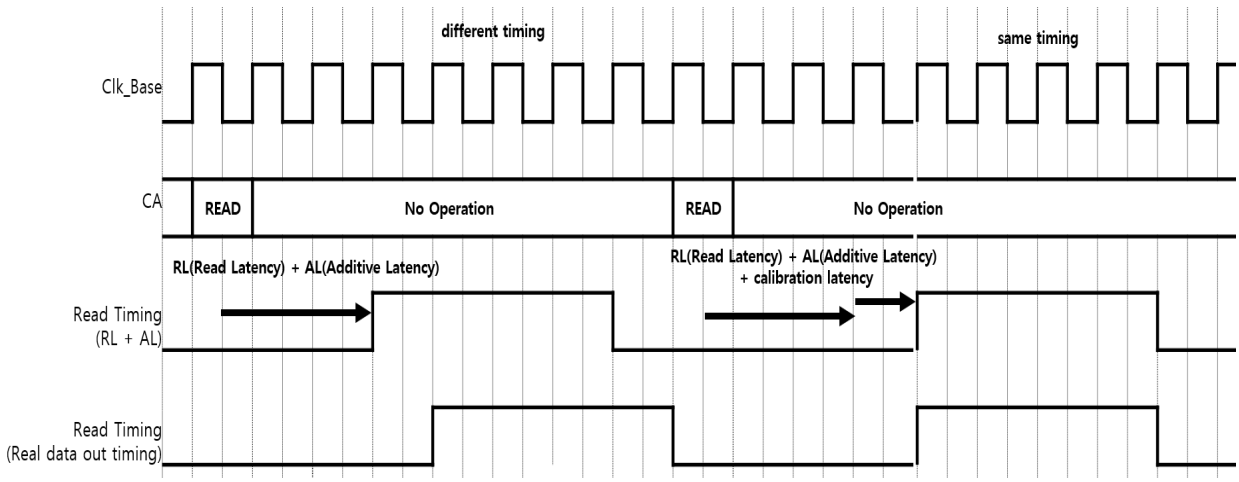

Fig. 3. Controller read timing calibration

After the memory initialization is completed, it is possible to test whether the information in the normal memory is read by using the MRR(Mode Register Read) command to the register containing the basic information of the memory. In addition, by using the BIST(Built In Self Test) mode[10], it is possible to check whether normal data operation is being performed while writing and reading a pattern of specific data for each bank.

Ⅳ. Board test

The memory controller designed in this paper is tested using FPGA. For FPGA, a Zynq UltraScale+ FPGA(XCZU2EG-SFVC784) equipped with 1Gb LPDDR2 was manufactured and used. When all of the memory initialization steps presented in this paper are completed, an initialization completion signal is output, and the completion of memory initialization is confirmed by checking the corresponding signal[11].

Fig. 4 is the figure to check the operation by loading the designed memory controller into the FPGA. It can be seen that the memory initialization through the memory controller has been completed using the board.

Fig. 4. Board test

Ⅴ. Conclusion

The memory controller presented in this paper automatically performs the initialization step of the corresponding memory for normal operation of the memory. The memory initialization sequence proceeds by executing a specific instruction after a specific time has elapsed. The memory controller presented in this paper proceeds with this sequence by itself, and checks the timing progress by itself to minimize unnecessary time lapse. If such a sequence is performed directly from the outside, a lot of time is consumed even after a very short time elapses. It reduces unnecessary time wastage and minimizes errors that may occur in the initialization stage by performing the initialization of the memory through the internal sequence of the controller rather than an external additional input. In addition, in order to confirm the simulation stage, the PHY modeling[12][13] to replace the PHY was designed and tested. Since the operation of the PHY and the operation of the PHY modeling was completed in advance, the error of the digital logic in the actual design stage can be minimized. Additionally, by adding a sequence for testing whether the normal operation of the memory is possible, it is possible to check and correct errors in advance due to the abnormal operation of the memory. Based on the corresponding memory controller, it brings many advantages from the research and development of memory controllers for initialization of other memories in the future[14].

References

[1] Soojung Lim and Hyokyung Bahn "Policy for Selective Flushing of Smartphone Buffer Cache using Persistent Memory", The Journal of The Institute of Internet, Broadcasting and Communication, Vol. 22, No. 1, pp. 71-76, Feb.

2022. https://doi.org/10.7236/JIIBC.2022.22.1.71.

[2] Daeyeop Yang, Manhyun Chung, Jaeik Cho, Taeshik Shon, and Jongsub Moon, "Research on User Data Leakage Prevention through Memory Initialization", Journal of The Institute of Electronic Engineers of Korea, Vol. 49, No. 7, pp. 71-79, Jul. 2012.

[3] Young-Deuk Jeon, Minhyung Cho, InSan Jeon, and Seong-Min Kim, "Low-power and small-area 8Gbps data receiver design for LPDDR5 Memory Interface", Korean Society of Electronic Engineering Conference, Vol. 2020, No. 8, pp. 217-218, Aug. 2020.

[4] Jin-Su Kang, "Design of LPDDR2 Memory Controller for Image Data Trasnfer", M.S. dissertation, Department of Electronic, Engineering, Kumoh National Institute of Technology, Gumi, Korea, 2021.

[5] Jin-Su Kang and Yong-Hwan Lee, "Low Power DDR2 Controller for Video Data Transmission", Proc. of KIIT Conference, pp. 17-18, Jun. 2021.

[6] Gil-Hwan Kim, "Design of LPDDR2(DDR3) memory controller for multiple request", M.S. dissertation, Department of Electronic, Engineering, Graduate School of Soongsil University, Seoul, Korea, 2015.

[7] Min-Hyung Cho, Young-Deuk Jeon, and Jin-Ho Han, "Design of ZQ Calibration circuit for high-speed LPDDR memory interface to improve signal integrity", Korean Society of Electronic Engineering Conference, Vol. 2021, No. 6, pp. 196-197, Jun. 2021.

[8] Chanho Lee and Kyochul Koo, "Memory Controller Architecture with Adaptive Interconnection Delay Estimation for High Speed Memory", Journal of IKEEE Korea Council, Vol. 17, No. 2, pp. 168-175, Jun. 2013. https://doi.org/10.7471/ikeee.2013.17.2.168.

[9] Chang-Yong Lee, Song-Hee Park, Tae-Min Kim, Yong-Hwan Lee, and Young-Hyung Kim, "Study of Data Calibration Algorithm of TDR Sensor", Proc. of KIIT Conference, pp. 153-154, Jun. 2022.

[10] Sunwoong Yang and Hoon Chang, "Automatic BIST Circuit Generator for Embedded Memories", Journal of the Institute of Electronic Engineers of Korea, Vol. 38, No. 10, pp. 746-753, Oct. 2001.

[11] Do-Sun Son and Sang-Hoon Lee, "Implementation of a Data Acquisition System with Large Number of Channels", Proc. of KIIT Conference, pp. 624-627, Jun. 2009.

[12] Min-Seok Seo, Seung-Yong Lee, Hyun Kim, and Hyuk-Jae Lee, "Implementation of Memory Controller Verification Platform Using FPGA", Korean Society of Electronic Engineering Conference, Vol. 2021, No. 6, pp. 429-430, Jun. 2021.

[13] Tae Ho Kim, Byoung Joo Yoo, Woo-Yeol Shin, and Deong-Kyoon Jeong, "Verilog Modeling for Verifying Systematic Operation of Memory Controller Physical Layer", Korean Society of Electronic Engineering Conference, Vol. 2010, No. 6, pp. 1246-1249, Jun. 2010.

[14] Jongbok Lee, "A Study in the Effects of DRAM on The Microprocessor Performance", The Journal of The Institute of Internet, Broadcasting and Communication, Vol. 17, No. 1, pp. 219-224, Feb. 2017. https://doi.org/10.7236/JIIBC.2017.17.1.219.

Authors

Jin-Su Kang

2020 : B.S degrees in Dep. of Electronic Engineering, Kumoh National Institute of Technology

2022 : MS degrees in Dep. of Electronic Engineering, Kumoh National Institute of Technology

Research interests : Digital SoC, Memory controller, Verilog HDL

Chang-Yong Lee

2017 : B.S degrees in Dep. of Electronic Engineering, Kumoh National Institute of Technology
2019 : M.S degrees in Dep. of Electronic Engineering, Kumoh National Institute of Technology
Research interests : Digital SoC, Image Processing, Verilog HDL

Yong-Hwan Lee

1993 : B.S. degrees in Department of Electronic Engineering, Yonsei University, Korea.
1999 : Ph.D degrees in Department of Electronic Engineering, Yonsei University, Korea.
2004 ~ present : Proc. in School of Electronic Eng., Kumoh National Institute of Technology
Research interests : Digital SoC, Image, Verilog HDL