Ph.D. Dissertation

Design of a Controller PHY for High-Capacity DRAM with Pulse-Based Feed-Forward Equalizer

펄스 기반 피드 포워드 이퀄라이저를 갖춘
고용량 DRAM을 위한 컨트롤러 PHY 설계

by

Hyeongjun Ko

August 2020

Department of Electrical and Computer Engineering
College of Engineering
Seoul National University

Design of a Controller PHY for High-Capacity DRAM with Pulse-Based Feed-Forward Equalizer

펄스 기반 피드 포워드 이퀄라이저를 갖춘 고용량 DRAM을 위한 컨트롤러 PHY 설계

지도교수 김 수 환

이 논문을 공학박사 학위논문으로 제출함

2020 년 7 월

서울대학교 대학원

전기정보 공학부

고 형 준

고형준의 공학박사 학위논문을 인준함

2020 년 7 월

위 원 장 : 정 덕 균 (印)
부위원장 : 김 수 환 (印)
위 원 : 홍 용 택 (印)
위 원 : 최 우 석 (印)
위 원 : 김 진 태 (印)

ABSTRACT

Design of a Controller PHY for High-Capacity DRAM with Pulse-Based Feed-Forward Equalizer

Hyeongjun Ko
Department of Electrical and
Computer Engineering
College of Engineering
Seoul National University

A controller PHY for managed DRAM solution, which is a new memory structure to maximize capacity while minimizing refresh power, is presented. Inter-symbol interference is critical in such a high-capacity DRAM interface in which many DRAM chips share a command/address (C/A) channel. A pulse-based feed-forward equalizer (PB-FFE) is introduced to reduce ISI on a C/A channel. The controller PHY supports all the training sequences specified in the DDR4 standard. A glitch-free DCDL is also adopted to perform link training efficiently and to reduce training time.

The DQ transmitter adopts quarter-rate architecture to reduce output latency. For the quarter-rate transmitters in DQ, we propose a quadrature error corrector (QEC), in which

clock signal phase errors are corrected using two replicas of the 4:1 serializer of the output stage. Pulse shrinking is used to compare and equalize the outputs of these two replica serializers.

A controller PHY was fabricated in 55nm CMOS. The PB-FFE increases the timing margin from 0.23UI to 0.29UI at 1067Mbps. At 2133Mbps, the read timing and voltage margins are 0.53UI and 211mV after read training, and the write margins are 0.72UI and 230mV after write training.

To validate the QEC effectiveness, a prototype quarter-rate transmitter, including the QEC, was fabricated to another chip in 65nm CMOS. Adopting our QEC, the experimental results show that the output phase errors of the transmitter are reduced to a residual error of 0.8ps, and the output eye width and height are improved by 84% and 61%, respectively, at a data-rate of 12.8Gbps.

Keywords: DRAM interface, feed-forward equalizer, glitch-free digitally-controlled delay line, memory controller, pulse-shrinking delay line, quadrature error corrector, replica serializer

Student Number: 2014-21669

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 MOTIVATION

The recent information and communication technology (ICT) topics can be represented by big data, internet of things (IoT), and cloud services, which are data-centric technologies. After the Exabyte Era, we have entered the Zettabyte Era, and technology for effectively storing and processing such massive data becomes very important [1.1.1], [1.1.2]. The International Data Corporation (IDC) predicts that the global datasphere will grow from 33 Zettabytes (ZB) in 2018 to 175 ZB by 2025, as shown in Figure 1.1.1. According to TrendForce's investigations, the number of hyperscale datacenters constructed globally will continue to rise up to a projected 1070 in 2025, registering a compound annual growth rate (CAGR) of 13.7% over the 2016-2025 period, as shown in Figure 1.1.2. As such, the demand for high capacity, low power, and low cost memory continues to increase.

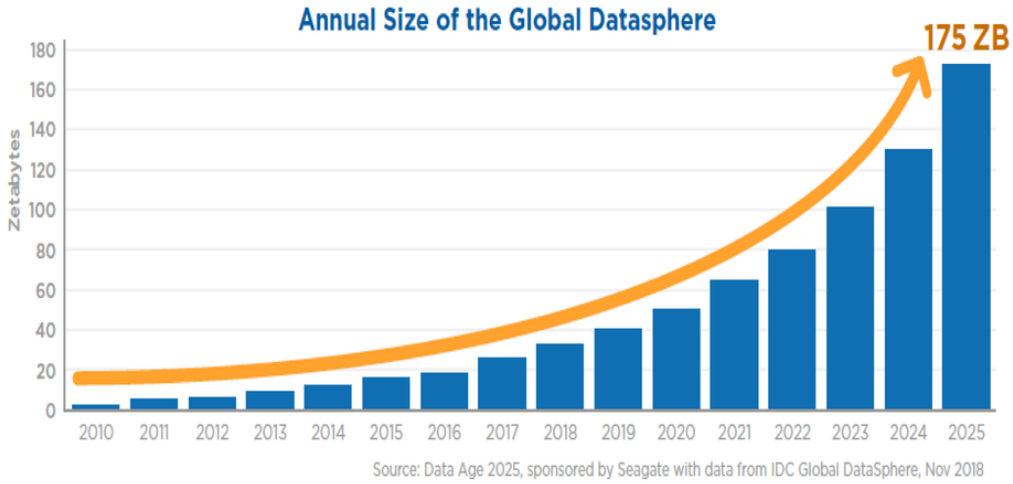


Figure 1.1.1 Annual size of the global datasphere, which is the amount of data created, captured, and replicated in any given year across the world

Figure 1.1.2 Estimated global number of hyperscale datacenters constructed

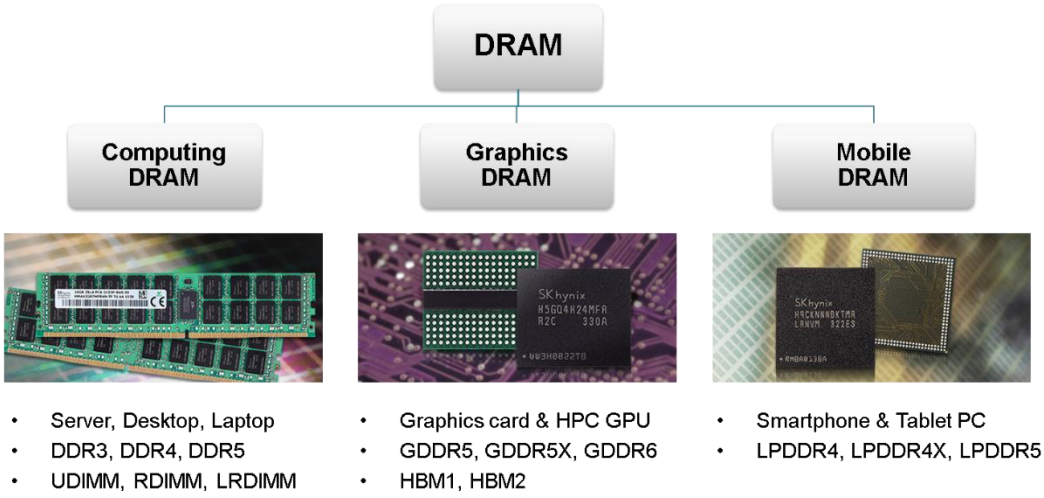


Figure 1.1.3 Classification of DRAM according to the application.

Dynamic random-access memory (DRAM) is a kind of random access memory (RAM), which is a storage device for storing each information bit in capacitors. Most computing systems adopt DRAM as the main memory. This is because DRAM can achieve a higher capacity than static random access memory (SRAM) and have shorter latency and higher bandwidth than nonvolatile memory such as phase-change memory (PCM), flash, or magnetic disk. The DRAM can be categorized into three types depending on an application: computing DRAM, graphics DRAM, and mobile DRAM, as shown in Figure 1.1.3.

The computing DRAM is typically configured in the form of the dual in-line memory modules (DIMMs) to increase storage capacity and bandwidth. Multiple DRAMs are connected in parallel to configure 64-bit as a rank, which is a group that can be accessed simultaneously, and an additional 8-bits for error-correcting code (ECC) are also

configured. Figure 1.1.4 shows the structure of each standard DIMM type. In the un-buffered DIMM (UDIMM), CLK, command, address, and DQ signals communicate directly with the memory controller without buffers [1.1.3]. CLK, command, and address signals are connected in a fly-by topology to distribute the capacitive loading of the connected memory chip, and the data lines communicate with the central processing unit (CPU) in parallel. When configured with multiple ranks or multiple DIMMs to realize high capacity, there is a problem that the loading of the command and address (C/A) lines is excessively increased.

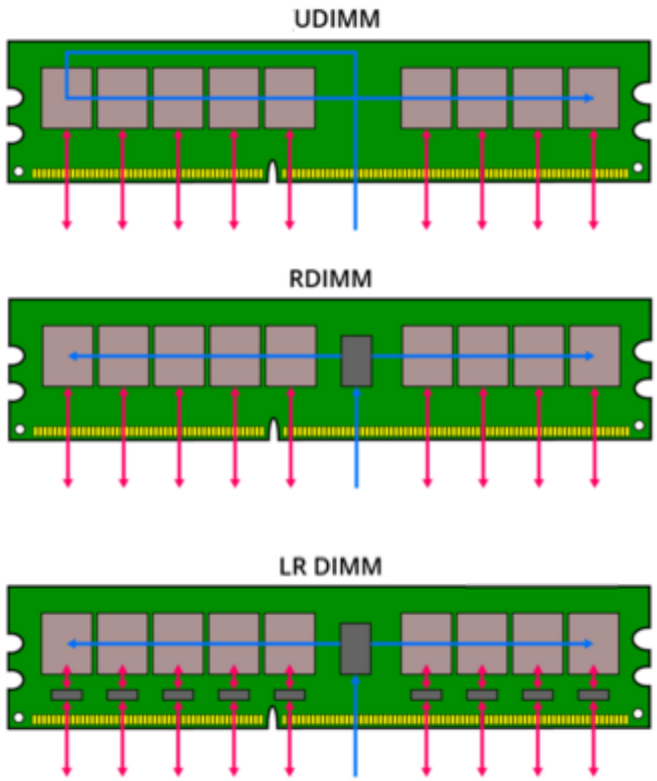


Figure 1.1.4 Structure of the standard DIMMs

When using multiple DIMMs for additional capacity, registered DIMMs (RDIMMs) are used to reduce the loading on the C/A lines [1.1.4]. Because an RDIMM has a registering clock driver (RCD) to buffering CLK and C/A signals, more DRAMs or more DIMMs can be used with reduced C/A line loading than UDIMM. However, since the data line is not buffered in the same way as the UDIMM, DQ loading becomes a bottleneck when it is composed of multi-rank and multi-DIMM. A load reduced DIMM (LRDIMM) that has data buffers (DBs) of the DQ signals are used to have the further capacity by reducing the loading of the data bus [1.1.5]. Because the DQ line load of multi-rank LRDIMM is reduced, more DIMMs than RDIMM can be used in a channel.

1.1.1 HEAVY LOAD C/A CHANNEL

A high-capacity solution with four times the storage of an LRDIMM was recently proposed [1.1.6]. In such a high-density DIMM, a plurality of DRAM chips is stacked in each package and C/A lines are connected to all DRAM chips. Therefore, the C/A lines have a very large capacitive loading. In particular, in the case of MDS DIMM in [1.1.6], each DRAM package has eight DRAMs stacked, so the C/A transmitter of the controller has to drive 80 DRAM chips. In addition, there is no input termination at the C/A receivers in DRAM, and the resulting impedance mismatch causes reflections on the C/A lines.

Approaches such as impedance-matched bidirectional multi-drop (IMBM) and parallel branching with write-direction impedance matching (PBIM) topologies have been

used to improve impedance matching in multi-drop memory channels [1.1.7], [1.1.8]. In IMBM, the input impedance is adjusted to Z_0 by using resistors with resistances of $n \times Z_0$ and Z_0/n in each branch of the multi-drop, where n=1, 2, ..., k-1 and k is the number of the branch. Because a large number of resistors are needed and there is not enough space to mount these resistors, it is difficult to implement in a DIMM. The PBIM topology also needs an additional resistor to match impedance and needs channels with a characteristic impedance of $Z_0/2$ that requires twice the line width. Therefore, it is also difficult to implement in a DIMM.

A decision feedback equalizer (DFE) can be used on the receiver side to improve inter-symbol interference (ISI) in multi-drop memory channels with impedance discontinuities [1.1.9]-[1.1.11]. However, DFE cannot remove pre-cursor ISI that needs to be addressed for dispersive channels such as the high-capacity C/A channel [1.1.12]. Continuous-time linear equalizer (CTLE) or feed-forward equalizer (FFE) can compensate for the pre-cursor ISI. However, it is disadvantageous in terms of cost and power consumption to implement equalizers such as CTLE and FFE in each DRAM receivers. Therefore, a transmitter equalizer is appropriate to compensate for the pre-cursor ISI. The conventional transmitter feed-forward equalizing scheme is not suitable for DRAM interfaces because it shifts the output data, sums the tap currents, which wastes power when there are no transitions in the signal, and changes the output impedance of the drivers [1.1.13]. Therefore, a transmitter equalizer suitable for the DRAM interface that can solve the above problems is needed.

1.1.2 QUARTER-RATE ARCHITECTURE IN DQ TRANSMITTER

The low latency memory controllers play an important role in memory-intensive applications that could reduce the overhead cycles for providing high memory access efficiency [1.1.14]. In our controller PHY, the DQ transmitter adopts quarter-rate architecture to reduce the output latency. The DQ transmitter receives 4-phase data from the digital part and serializes it 4 to 1 using a quarter-rate clock to prevent the inevitable increase in latency when serializing it sequentially 4 to 2, and 2 to 1. However, during four-phase clock generation and distribution, factors such as process, voltage, and temperature (PVT) variations, mismatches between the clock trees of each phase, and supply and ground noise can cause quadrature phase errors. As these errors increase, the signal integrity of the transmitter (TX) output deteriorates, reducing its effective bandwidth. A data-dependent jitter (DDJ) in the output stage due to the PVT variation is another important factor for the signal integrity of a high-speed interconnection [1.1.15]. The output impedance is calibrated using external ZQ resistor [1.1.16], and considering large PVT variation in DRAM process, the distortion in the serializer (SER) should also be corrected to assure the integrity of the output signal.

A quadrature error corrector (QEC) can be implemented using analog circuits [1.1.17], but a QEC of this type is vulnerable to PVT variations, current mismatches, voltage offsets, and noise. In addition, because a DRAM is often idle, and low standby power is a key part of its specification, a digital QEC that consumes far less standby power than an analog one is more suitable for a DRAM TX. A digital QEC with reference delay line and phase

detector requires iteration or locking procedure of the reference delay line, like delay-locked loop (DLL). A single shared loop filter can be used to minimize the effect of the mismatches between phase detectors [1.1.18], but it is difficult to operate at a higher frequency due to the tight internal timing margin. A time-to-digital converter (TDC) can be used to detect phase error, but counter-based TDC requires an additional clock source [1.1.19]. Alternatively, a vernier TDC architecture is widely adopted thanks to the simplicity of its design concept [1.1.20], but it consumes large area and power. On the other hand, the above QECs detect and correct phase errors in the clock trees, thus the distortion in the SER still remains.

1.1.3 SUMMARY

We present a controller PHY with pulse-based feed-forward equalizing (PB-FFE) Tx on heavy load DRAM interface. The proposed 3-tap FFE only injects current before and after a signal transition to compensate for pre-cursor and post-cursor ISI; and makes no current flows through the Tx FFE when there is no transition, thus consumes no DC power and keeps Ron unchanged. The third tap can cancel the reflection by introducing an adjustable delay. Our controller PHY supports all the training sequences specified in the DDR4 standard. Digitally controlled delay-lines (DCDLs) are used in all transmitter and receiver blocks for link training, and when changing the delay of the DCDL, a glitch problem may occur. A glitch-free DCDL is also adopted to perform link training efficiently

and to reduce training time.

We also propose a QEC including two replicas of the SER of the output stage, to correct quadrature errors including distortions in the SER. Each replica SER converts associated clock phase difference to pulse width. Two pulse-shrinking delay lines (PSDLs) are used as a digital pulse-width comparator instead of area and power-hungry TDCs. A phase adjuster equalizes the two pulses based on the result of the comparison. As a result, the PVT variations in the SER of the output stage can be compensated during phase error correction by this QEC. A fine delay unit is added to the first stage of each PSDL to improve the accuracy of phase error detection.

1.2 THESIS ORGANIZATION

This thesis is organized as follows. In Chapter 2, the overall architecture of the managed DRAM solution (MDS) with its controller are introduced, and the controller PHY architecture is also presented. In Chapter 3, the thesis describes the heavy load command/address channel and the proposed pulse-based feed-forward equalizer. Detail sub-blocks and functional verification results are presented in Chapter 4. In Chapter 5, a prototype quadrature error corrector (QEC), which is implemented in another chip, is presented. Chapter 6 shows the experimental results of both the controller PHY and the prototype QEC. Finally, in Chapter 7, the thesis is summarized with the discussion of contribution.

CHAPTER 2

ARCHITECTURE

2.1 MDS DIMM STRUCTURE

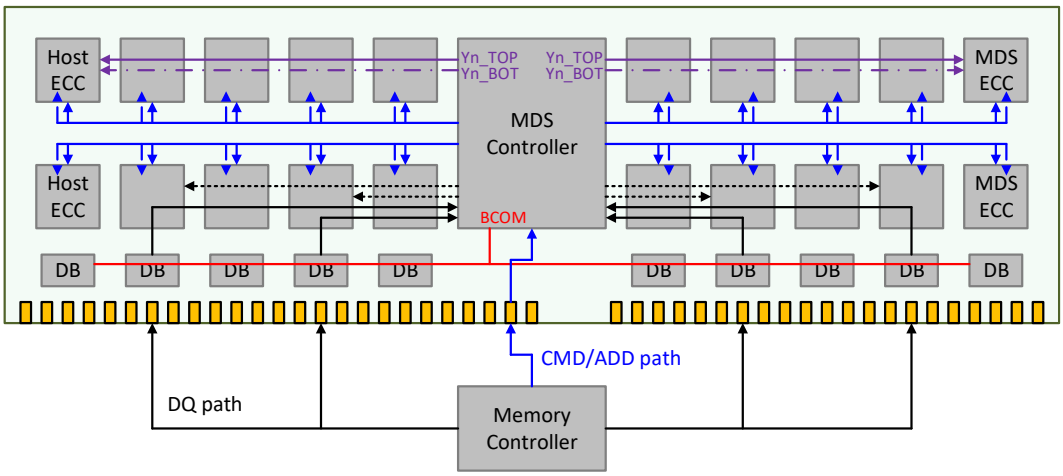


Figure 2.1.1 Structure of the MDS DIMM

Figure 2.1.1 shows the structure of a MDS DIMM. Similar to Joint Electron Device Engineering Council (JEDEC) standard high-capacity DIMM, LRDIMM (Load-Reduced DIMM) [1.1.5], but differs in DRAM configuration, the role of the MDS controller, and DQ path. In addition to CLK and C/A lines, the data line is also buffered by the MDS controller. To maximize capacity, each DRAM package has a capacity of 128Gb (16GB)

by stacking eight 16Gb DRAM chips as an octal-die package (ODP). There are 20 DRAM packages each on the front and back sides, of which 8 are for host ECC and 8 are for MDS ECC. The MDS DIMM of this structure has a capacity of 512 GB (32×16GB) whereas the maximum capacity of the LRDIMM is 128 GB based on standard DDR4 SDRAM. When the capacity is increased, the refresh power also increases. By applying the Error-Correcting Code (ECC), refresh power is reduced by improving the data retention time issue. A very small part of the data retention time of the entire DRAM cell determines the refresh time. Since some of the weak cells limit the refresh time, the refresh rate can be reduced by remedying the failure of these cells through ECC, and accordingly, the refresh power is also reduced [2.1.1].

Figure 2.1.2 compares conventional DDR4 and managed DRAM. The MDS has the advantage that a higher capacity can be realized at a lower cost than standard DDR4. By applying additional ECC, refresh power is reduced. However, due to the increase in loading and latency, a slight loss in performance must be considered.

Item	Conventional DRAM	Managed DRAM	MDS methodology
Capacity	up to 128GB	up to 512GB	16Gb x ODP x 32
Cost	Good	Better	Relax spec.
Performance	Great	Good	Relax spec.
Refresh Power	Bad	Good	ECC for tail bit errors
Operating Power	Good	Good	DRAM cell
Endurance	Great	Great	DRAM cell
Process Maturity	Great	Great	DRAM process
System Compatibility	Great	Good	DDR4 spec.

Figure 2.1.2 Comparison between conventional DDR4 and MDS

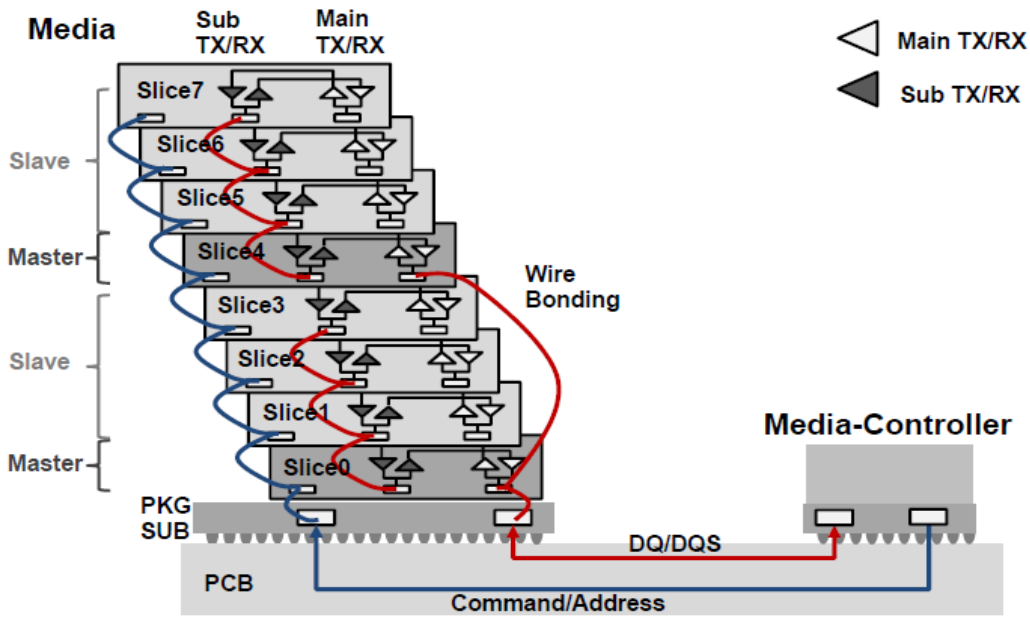


Figure 2.1.3 Architecture of the MDS ODP including wire-bonding diagram

The MDS ODP uses wire-bonding to reduce the cost than a conventional 3D stacked DDR4 SDRAM [2.1.2], which connects each slice using through-silicon via (TSV). Command/Address (C/A) signals are shared between 8 slices. However, in the case of the DQ/DQS signal, it is not possible to have this structure because signal integrity is reduced due to heavy load on the pad. As shown in Figure 2.1.3, only the DQ/DQSs of the two master slices (slice 0 and 4) are directly connected to the media-controller, using the main TX/RX. Slave slices 1, 2, 3, and slices 5, 6, 7 are connected to the master slice 0 and slice 4, respectively, via sub-TX/RX and repeater in the master slices. Since only the pads of the two slices are directly connected to the media controller, the load on the pads is significantly reduced compared to when all eight pads are connected. ODPs on the front and back share

the DQ line, so one media controller transceiver and four DQ transceivers of the media are connected to one DQ line.

2.2 MDS CONTROLLER

MDS controller supports all functions of DDR4 registering clock driver (RCD) [2.2.1] and additionally performs ECC operation. For this, DQ of all DRAM, including ECC DRAM, must be transmitted through the MDS controller. That is, in the case of DDR4 LRDIMM, the memory controller and the DRAM send and receive data through the data buffer (DB). However, in the MDS DIMM case, the data lines are buffered by the MDS controller. For a write operation, the MDS controller computes the ECC and writes the data to all the DRAMs include MDS ECC DRAMs. When reading data from DRAMs, the MDS controller calculates the ECC and passes the corrected data to the memory controller through the DB if any error exists.

The MDS DIMM consists of 20 nibbles, including Host ECC and MDS ECC, and is composed of 2-rank divided into front and backside. In this configuration, if C/A signals are transmitted separately to the left and right, like the existing LRDIMM, 160 DRAMs are connected to each driver. Thus each of the left and right sides is further divided into up and down to reduce loading. Each driver is configured to drive 80 DRAMs. Here, Yn (DRAM CK) drives the 40 DRAMs by further dividing the front and back sides.

Figure 2.2.1 shows the architecture of the MDS controller and the interface diagram. The controller consists of a flow control block, media control block, and PHY blocks that communicate with the host controller and the DRAMs, respectively. The F-PHY communicates to the host controller with a standard DDR4 interface and the B-PHY interfaces to the DRAMs with a modified DDR4-like interface. The flow control block is

needed because the data-rate of the host interface and the media interface is different. Media controller supports training such as read pre-amble training, write leveling, read training, and write training, and additionally supports CA training through parity check.

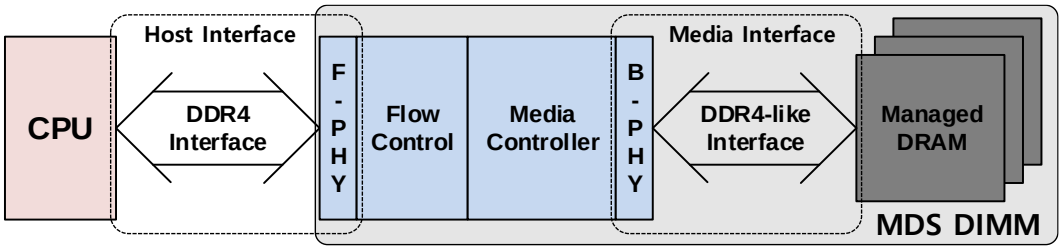


Figure 2.2.1 MDS controller architecture and interface diagram

2.3 MDS CONTROLLER PHY

Figure 2.3.1 shows the architecture of the MDS controller PHY, and how it communicates with MDS media [1.1.6]. The controller PHY consists of an all-digital phase-locked loop (ADPLL), an all-digital delay-locked loop (ADDLL), a clock distribution circuit, a link training finite-state machine (LTFSM), 8 pairs of clock signal (CK) transmitters, 4 groups of 33 transmitters for the C/A lines, 80 data signal (DQ) transceivers for 20 nibbles of data, each connecting to a X4 DRAM, and 20 transceiver pairs for the corresponding strobe signals (DQS).

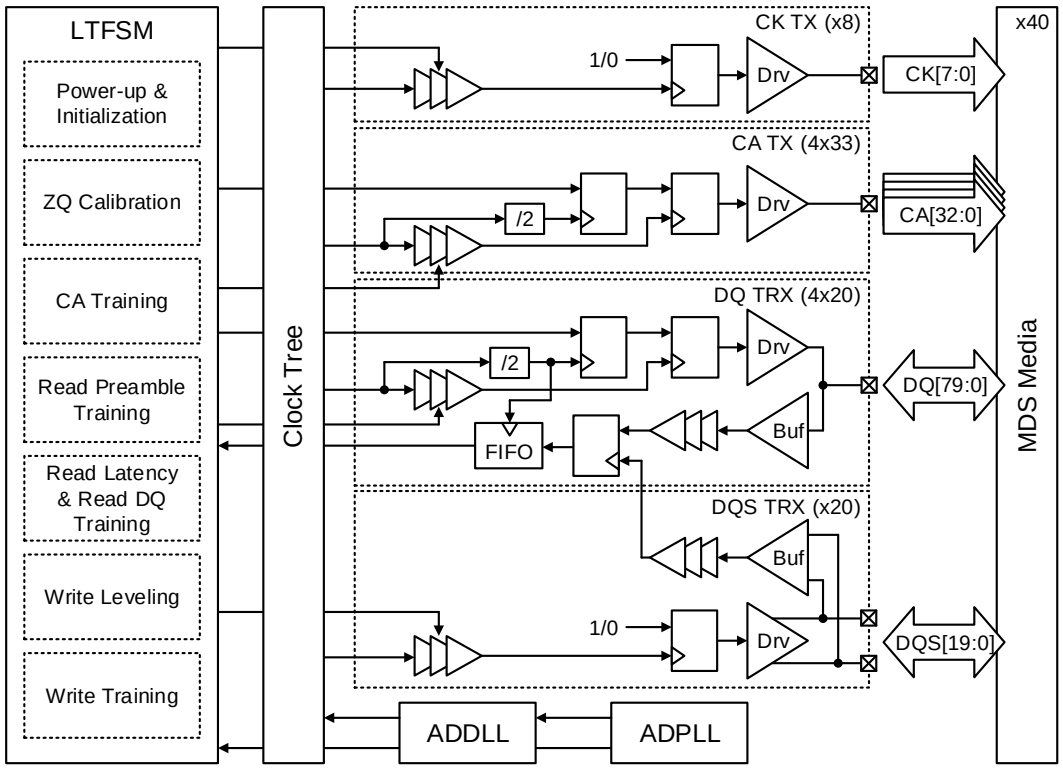


Figure 2.3.1 PHY architecture of the MDS Controller

The ADPLL generates both the global clock signal (PHYCLK) used by the transceivers and the system clock (SYSCLK) used by the LTFSM. The frequency of PHYCLK is 1066MHz, and that of SYSCLK is 533MHz. The ADDLL has the same delay line as that in each transceiver, and provides delay control-code corresponding to 1-cycle of PHYCLK to each transceiver. Each delay line divides the received control-code by 128 allowing the clock phase to be adjusted in 1/128 cycle intervals. The clock tree minimizes the skew between each C/A group and between each nibble of DQ. To prevent skews between clock phases, PHYCLK is distributed as a single phase, and complementary clock signals are generated in each transceiver. The LTFSM contains a number of digital modules, including circuits for power-up and initialization, ZQ calibration, CA training, read preamble training, read latency and DQ training, write leveling, and write training. The power-up and initialization module performs power-up and initial reset procedures for both the controller and the DRAM, and also programs the mode register sets (MRS) for each DRAM chip, using gear-down mode in which CA signals are widened to improve the initial sampling margin before CA training. The ZQ calibration module transmits the ZQ calibration command to the DRAM, and also calibrates the drive strengths of the CA drivers and DQ drivers in the controller PHY. Other training modules adjust the DCDLs in each transceiver during the training sequences specified in the DDR4 standard. To avoid a clock domain crossing problem between the LTFSM and each transmitter, regardless of the delay being produced by the DCDL, each transmitter divides the PHYCLK before it passes through the DCDL and samples the input DQ and C/A signals using this divided clock signal. Each transmitter outputs DQ and C/A signals to DRAM using the optimized timing

determined by the training procedure. The DQ receiver accepts data read from the DRAM and passes it to the LTFSM through asynchronous FIFO. The structures of the transmitter and receiver are described in more detail later.

2.3.1 INITIALIZATION SEQUENCE

For the PHY to operate correctly, it must be initialized in the order specified. The initialization sequence required for the PHY is shown in Figure 2.3.2.

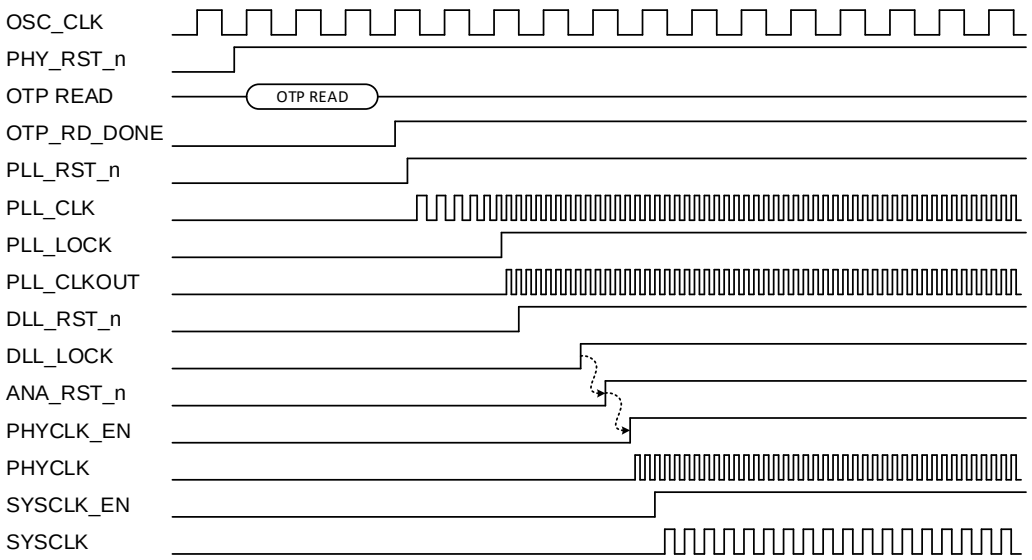


Figure 2.3.2 Initialization sequence of the controller PHY

After power on, OSC_CLK is applied from the external oscillator. When the PHY reset signal is released, the one-time programmable (OTP) memory is read. The OTP memory consists of e-fuses and stores default values such as operating mode and latency of the PHY and the DRAM. When the OTP read is completed, the LTFSM releases the PLL reset signal to perform the PLL locking operation. Before the PLL is locked, the PLL does not output the clock signal, and after it is locked, the PLL_CLKOUT signal is output to the

DLL. When the PLL_LOCK signal occurs, LTFSM sends the DLL_RST_n signal to DLL. The DLL finds delay information corresponding to the 1-cycle of PLL_CLKOUT signal and generates the DLL_LOCK signal.

The clock signal must be applied to the transceivers after reset is released so that each transmitter and receiver can operate in the correct clock phase. The LTFSM releases the reset of all the transmitters and receivers by de-asserting the ANA_RST_n signal to high. Then, it sends the PHYCLK_EN signal to the PLL. The PLL receiving this signal outputs the PHYCLK signal to all the transmitters and receivers. After all the transceivers are ready for operation, the PLL receives the SYSCLK_EN signal from the LTFSM and outputs the SYSCLK signal. Since SYSCLK is not output from the PLL before the SYSCLK_EN signal is generated, the LTFSM operates by OSC_CLK, and after SYSCLK_EN signal generation, it operates by SYSCLK.

When the PHY initialization sequence is completed, each DRAM initialization sequence is performed, as shown in Figure 2.3.3.

After power is on, the RESET_n signal is maintained below 0.2×VDD for 200us. CKE is pulled ‘low’ before RESET_n being de-asserted. When RESET_n is de-asserted, each DRAM starts internal initialization during 500us before the PHY pulls CKE ‘high’. The PHY starts transmitting clock signals before CKE goes active and sends deselect command (DES) to be registered at clock edge Td. Subsequently, the mode register set (MRS) command for operating mode setting is sequentially transmitted, and when the ZQ calibration is completed, the DRAM initialization sequence is completed. As described in chapter 2.1, the MDS ODP has two master chips that communicate with the controller.

However, a ZQ resistor is shared for the two master chips, and the ZQ calibration cannot be performed simultaneously. Therefore, the ZQ calibration command is transmitted separately as ZQCL1 and ZQCL2 for each master slice.

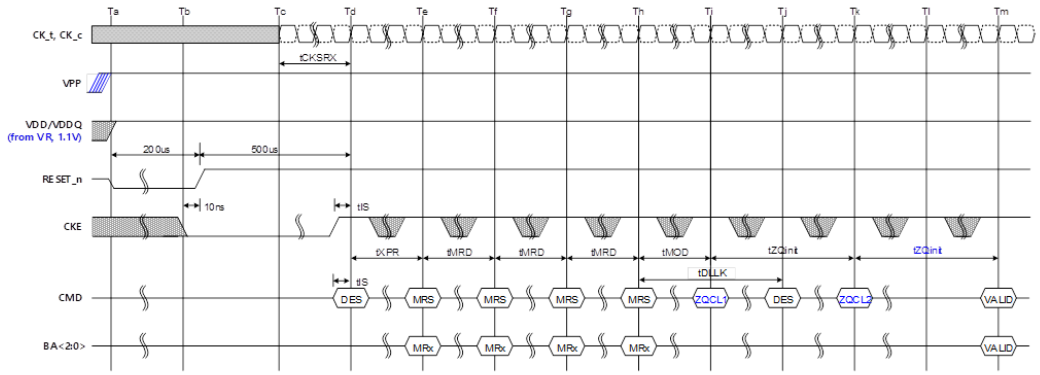


Figure 2.3.3 Reset and initialization sequence of MDS media at power-on ramping

2.3.2 LINK TRAINING FINITE-STATE MACHINE

Figure 2.3.4 shows the overall block diagram of the link training finite-state machine (LTFSM). As mentioned earlier, the LTFSM is consists of various training modules such as in the figure, and detailed input and output signals are included in the figure. The DFI_CH is a digital control logic to perform functions such as training requests, scheduling, and ECC.

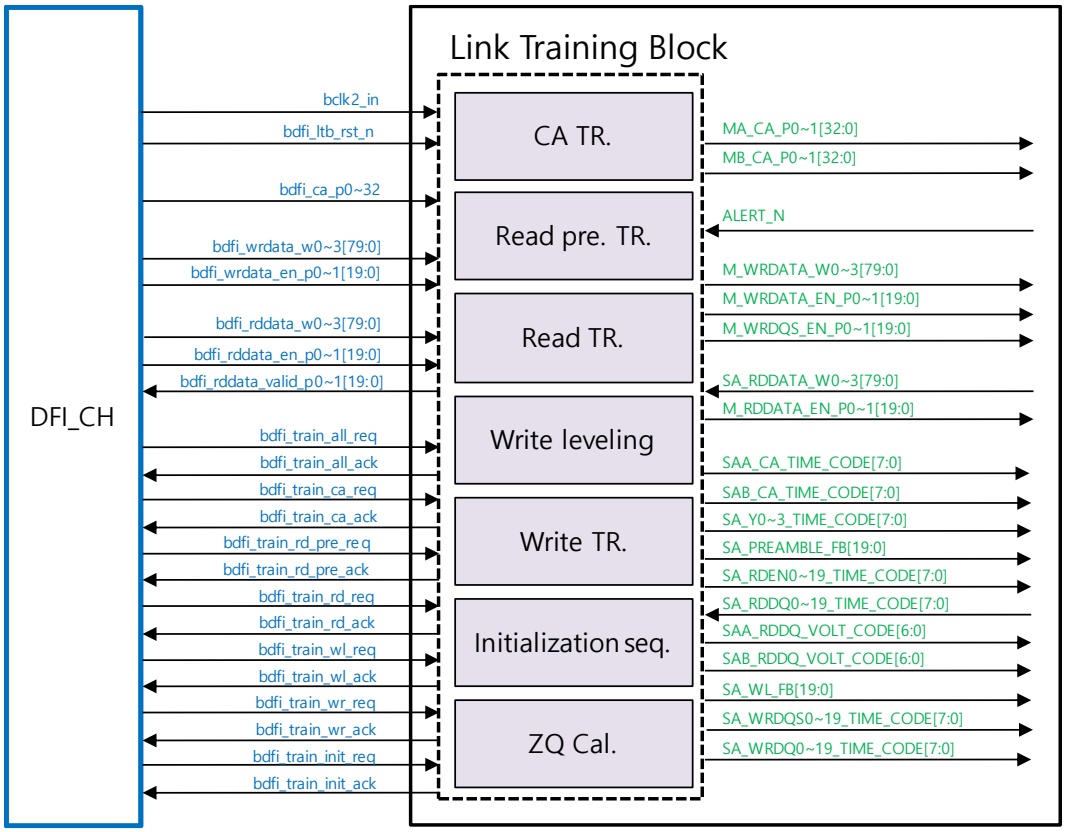


Figure 2.3.4 Block diagram of the link training finite-state machine

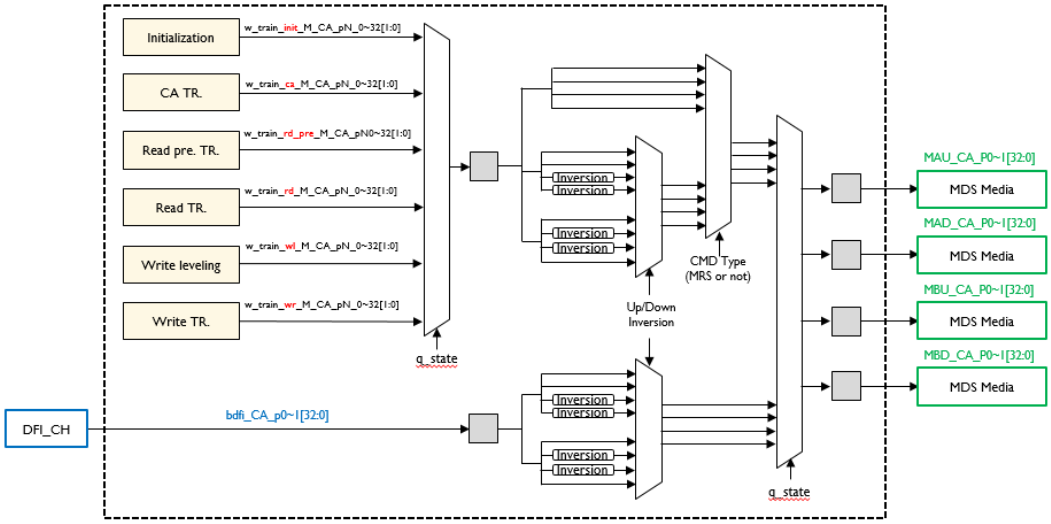


Figure 2.3.5 Block diagram of the C/A training module

Figure 2.3.5 shows the C/A operation of the LTFSM. There are two types of commands: commands used in training and commands used in regular operation. The command received from the DFI_CH is transferred to the C/A transmitters in the regular operation. In the training operation, the commands are generated by each training block and transmitted to the transmitter.

Commands are divided into four groups (MAU, MAD, MBU, MBD) in the LTFSM and output. The LTFSM supports the C/A Inversion function to reduce power noise by inverting the C/A signals in the B-side and making DC power balancing. The up and down inversion scheme is also implemented to reduce the power noise further. Because the inversion mode is not implemented in DRAM, the C/A signals for the MRS command is not inverted.

Figure 2.3.6 shows the operation of the write channel of the LTFSM. During write leveling and write training, the data, write enable, and dqs_enable signals of each nibble are generated by the LTFSM and transmitted to the DQ and DQS transmitter. However, in the regular operation, the LTFSM delays the data, write enable, and dqs_enable signals received from DFI_CH by the latency found during training and transmits them to the transmitter.

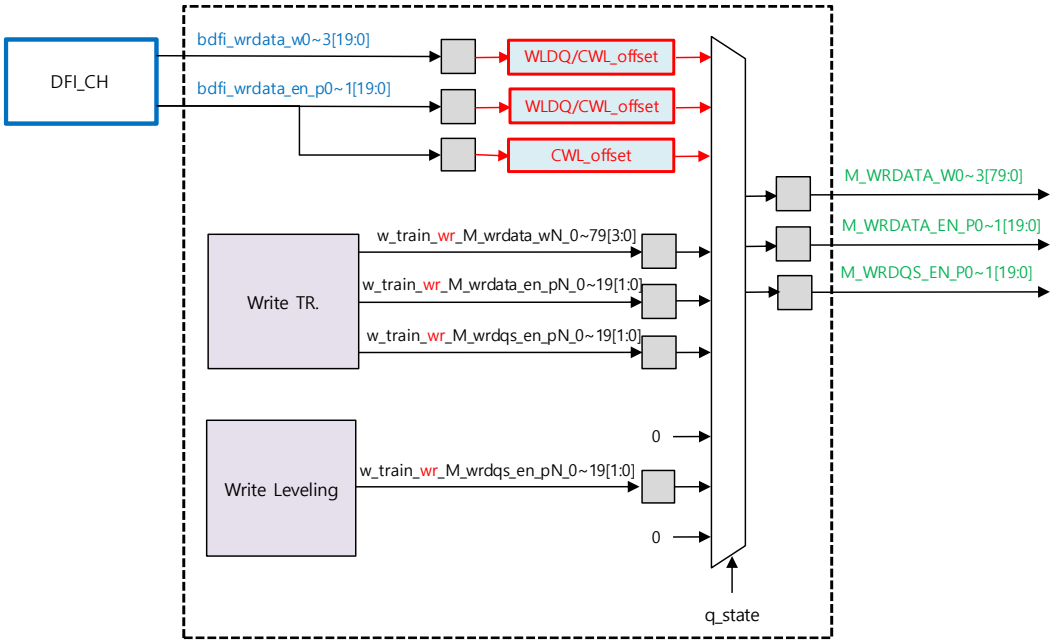


Figure 2.3.6 Block diagram of the write training module

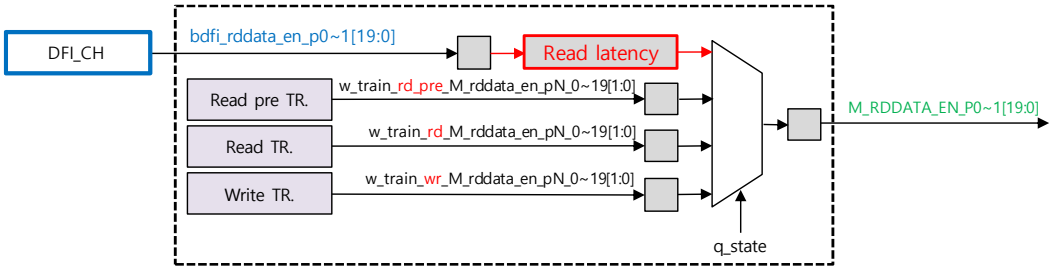


Figure 2.3.7 Block diagram of the read enable training module

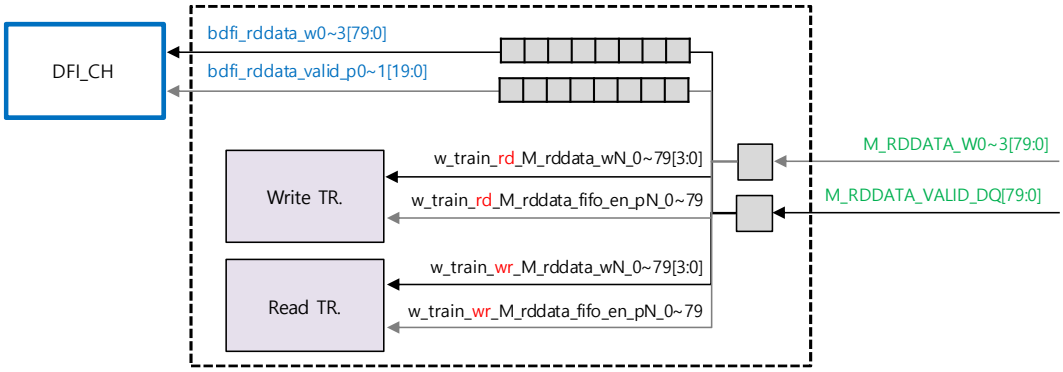


Figure 2.3.8 Block diagram of the read training module

Figure 2.3.7 and Figure 2.3.8 show the read enable control block and read data control block of the LTFSM, respectively. Similar to the write operation, the read enable signals of each nibble generated by LTFSM are transmitted to the DQ receivers in the training process. The nibbles of the training block receive data and valid signals from the media through the DQ receivers.

In the regular operation, the LTFSM delays the enable signal of nibble received from the DFI_CH by the read latency trained in the training operation and delivers it to the receivers. When receiving data and valid signals from the receivers in regular operation, the LTFSM should wait until all the data and valid signals from all nibbles arrive, and then

synchronize with FIFO to deliver to DFI_CH at once.

2.3.3 POWER DOWN MODE

The PHY supports the following three power down modes (PDM): 1) power down, 2) maximum power saving mode, 3) self-refresh. When entering power down, C/A transmitter except for CKE, and all DQ and DQS transceivers are disabled. When entering self-refresh, the CK transmitters stop operating additionally, and in the maximum power saving mode, the CKE transmitter and internal clock tree are additionally disabled. Even in all PDM operations, ADPLL is always running to provide SYSCLK to the LTFSM. When entering each PDM, the block stops immediately by asynchronous operation, and when the PDM exit, the PDM control signal is sampled synchronously to re-enable PHYCLK to prevent clock phase problems after exit.

CHAPTER 3

PULSE-BASED FEED-FORWARD EQUALIZER

3.1 COMMAND/ADDRESS CHANNEL

Figure 3.1.1 shows the structure of the MDS C/A channel. Each channel is connected to 10 DRAM packages, each of which has an ODP structure in which 8 DRAMs are stacked and connected by inter-chip bond wire. Thus each C/A transmitter drives 80 DRAMs, except for the chip select (CS) signals, which drive each of the five DRAM packages on the front and backside of the DIMM separately. Although the channel is terminated with a 20Ω resistor at the end, the DRAM input is not terminated and therefore reflections occur.

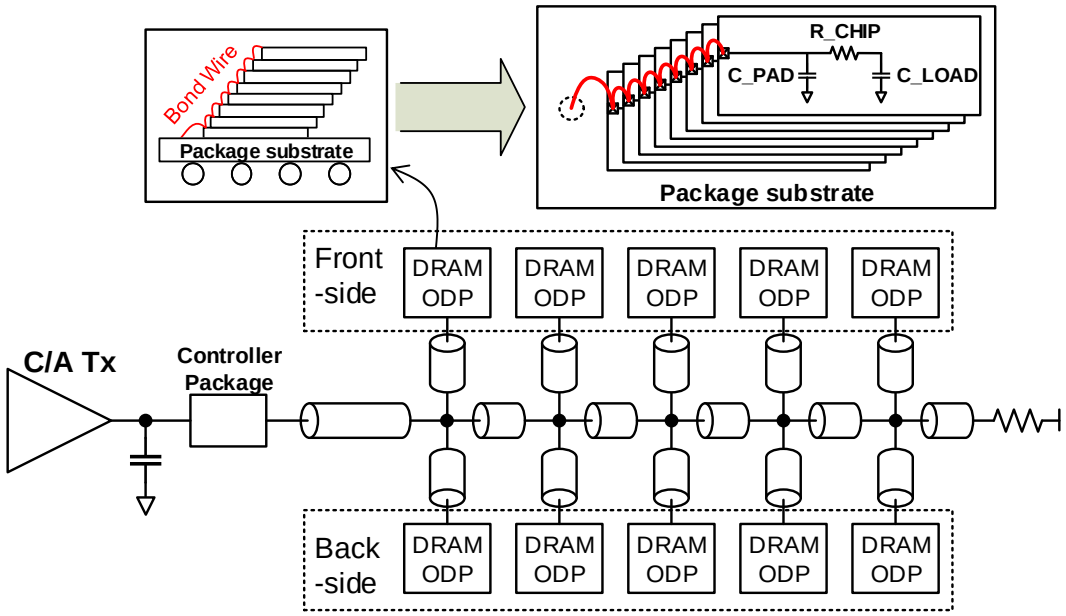


Figure 3.1.1 MDS C/A Channel

The input parasitic of each DRAM chip can be modeled as a π-network [3.2.1], as shown in the upper right corner of Figure 3.1.1. The input parasitic capacitance of each DRAM input, C_PAD, includes the capacitance of the PAD, electrostatic discharge (ESD) protection diodes, and the metal interconnects. The resistance R_CHIP between PAD and C/A receiver circuit includes the resistance of the metal lines and ESD protection resistor. The capacitive load C_LOAD includes the gate capacitance of the receiver circuit and controllable capacitor. Reflections can be reduced by increasing R_CHIP, but that also increases the insertion loss due to C_LOAD, and this tradeoff is important to signal integrity.

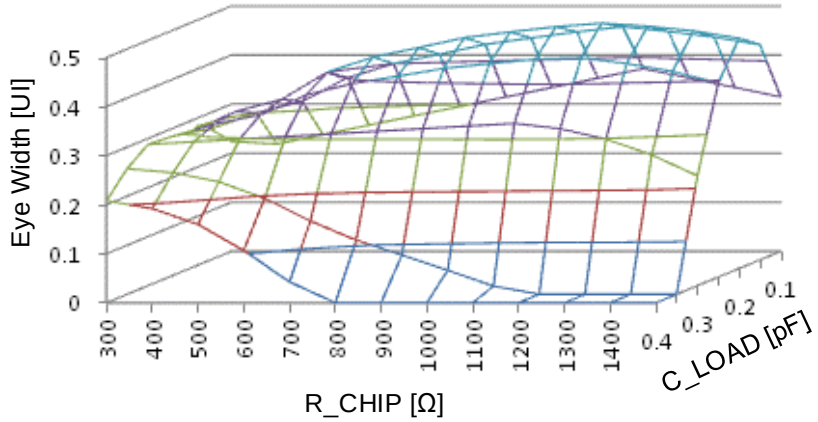


Figure 3.1.2 Variation of the simulated eye width with the input parasitic resistance and capacitance of each DRAM.

Figure 3.1.2 shows the C/A input timing margin for different values of R_CHIP and C_LOAD, obtained from a simulation in which the bond wires were modeled by their s-parameters. When C_LOAD is less than 0.3pF, the timing margin tends to increase due to the reduced reflection with increasing R_CHIP; but when the load exceeds 0.3pF, the timing margin decreases rapidly due to increased insertion loss with increasing R_CHIP. The maximum timing margin is achieved when C_LOAD is 0.2pF and R_CHIP is 1.2kΩ. When R_CHIP becomes larger than 1.2kΩ, the timing margin is reduced regardless of C_LOAD. As a result of this simulation, C_LOAD was set to 0.2pF and R_CHIP was set to 1kΩ having some margin not to exceed 1.2kΩ, in each DRAM.

As the DRAM input parasitic values found above, Figure 3.1.3 and Figure 3.1.4 show the insertion loss and return loss of the C/A channel, respectively. The insertion loss for the farthest DRAM is 11.7dB and increases rapidly around the operating frequency; thus, the ISI due to these insertion losses should be compensated. The return loss is 2.8dB, which is

relatively small, according to the large R_CHIP.

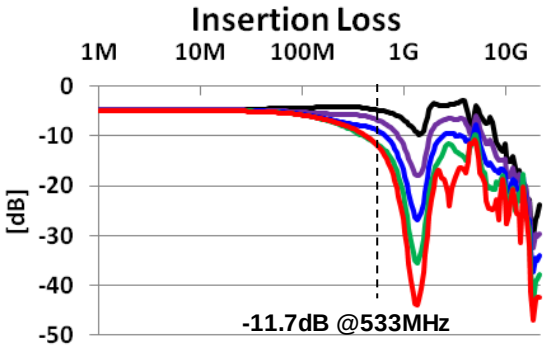


Figure 3.1.3 Insertion loss of the C/A channel. The red line is for the farthest DRAM and the black line is for the nearest DRAM. The insertion loss for the farthest DRAM is -11.7dB at 533MHz. Due to the termination at the end of the channel, the insertion loss of -4.6dB is shown even at the low frequency.

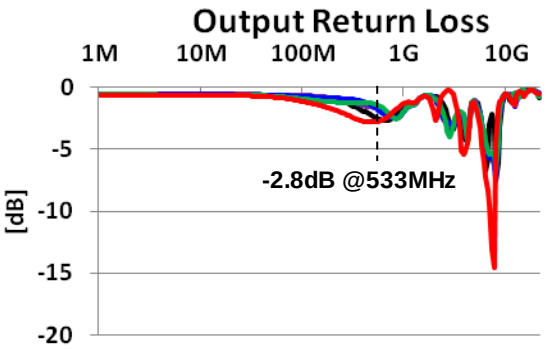


Figure 3.1.4 Return loss of the C/A channel. The red line is for the farthest DRAM and the black line is for the nearest DRAM. The return loss for the farthest DRAM is -2.8dB at 533MHz

3.2 COMMAND/ADDRESS TRANSMITTER

The controller PHY transmits 4 groups of 33 C/A signals including CS[0: 1], C[0: 2], ACT_n, A[0:17], BG[0: 1], BA[0: 1], CKE[0:1], ODT[0:1], and PAR as standard DDR4 for each group. Because each group of 33 C/A signals are sampled by a common clock signal at the DRAM input, the skew among these signal group should be minimized. Per-pin training for the C/A signals requires too many delay lines and consumes much power. To reduce the skew between these 33 signals, we divide these into 11 subgroups to train separately. Because the internal timing margin of the CS[0:1] signals in the DRAM chip is different from that of the other C/A signals due to the pre-CMD scheme described in [4], the CS[0:1] signals are composed of one subgroup. Four of the remaining 31 C/A signals are composed of another subgroup, and the other 27 signals are composed of nine subgroups of three each.

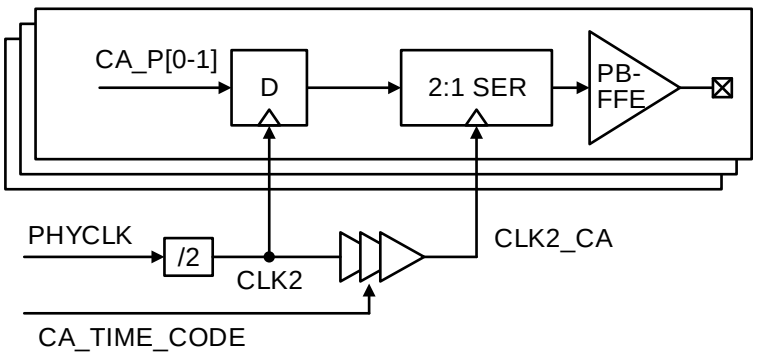


Figure 3.2.1 The transmitter for one subgroup of 3 C/A signals.

Figure 3.2.1 shows the block diagram of one of the subgroup C/A transmitter of 3 signals. The 2-phase data sequences for each C/A are applied from the LTFSM. To avoid domain crossing problems between the PHYCLK and the signals from the LTFSM depending on the training results, the signals CA_P[0-1] are sampled first with a divided PHYCLK (CLK2), and then serialized with a delayed clock (CLK2_CA). The serialized C/A data is transmitted in SDR mode.

CA training is performed using the command address parity mode described in the DDR4 standard. CS signals are trained first. During CS training, the controller transmits C/A signals with a bit pattern designed to produce a failure during parity checking. When CS is sampled successfully by the DRAM, it asserts the ALERT_N signal, which is the error flag signal of the DDR4, to low. By increasing the delays applied to the CS signals, the LTFSM trains output timing of the CS signals and sets the delay at 1/4 of the pass window so as to provide more setup than hold time. After the CS signal have been trained, each subgroup of C/A signals is trained sequentially in the same manner. The LTFSM sets the delay at the center of the pass window for C/A signals except for the CS signals.

3.3 PULSE-BASED FEED-FORWARD EQUALIZER

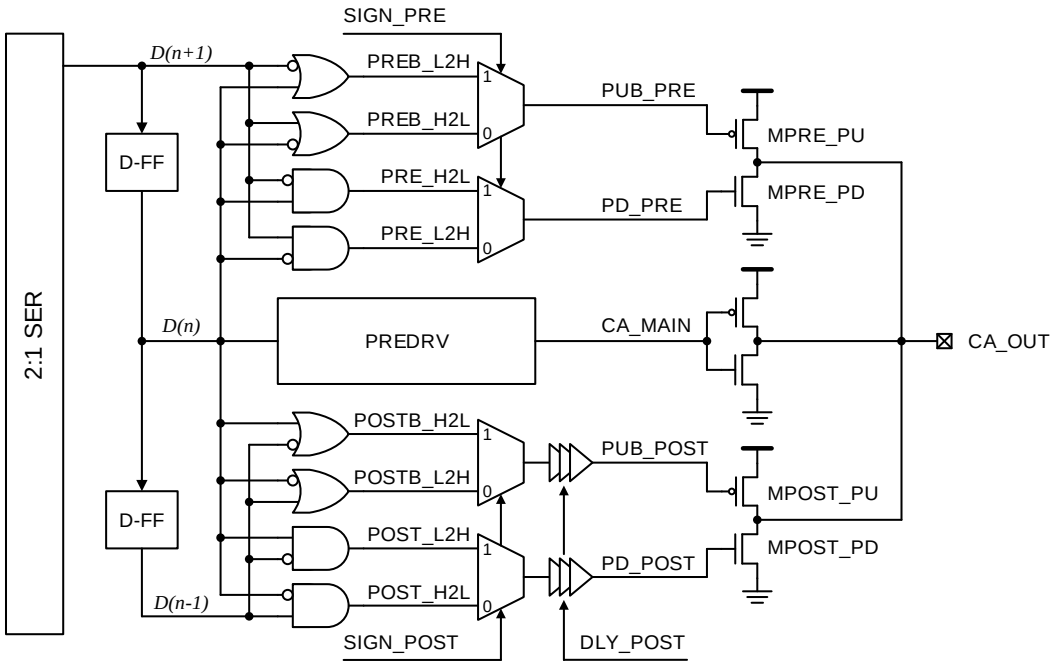


Figure 3.3.1 Block diagram of the PB-FFE.

Figure 3.3.1 shows a block diagram of the proposed PB-FFE using edge-detection logic. The incoming serialized data is shifted using D flip-flops, and the combinational logic gates generate pulse signals of 1 unit-interval (UI) in length before and after each transition as follows; just before the rising edge (PRE_L2H), just before the falling edge (PRE_H2L), just after the rising edge (POST_L2H), and just after the falling edge (POST_H2L). The inverted signals PREB_L2H, PREB_H2L, POSTB_L2H, and POSTB_H2L are generated separately, so that the pull-up and pull-down drivers can be controlled independently. Signals with the prefix ‘PRE’ act as pre-taps to remove the pre-

cursor, and the direction of compensation is selected by the SIGN_PRE signal. Signals with the prefix ‘POST’ are pre-emphasis pulses, with the compensation direction selected by the SIGN_POST signal. When the PB-FFE is providing simple pre-emphasis, inverted pre-emphasis pulses are not required; but it is available for use as a post-tap to remove the post-cursor by adding a delay to the pre-emphasis pulse. By equalizing channel loss using pulses, the proposed FFE does not consume DC power, and the output driver impedance problem during the no-transition region can be addressed. Because MDS C/A channel is an ISI-dominant dispersive channel rather than reflection, we set SIGN_PRE to 0 to compensate for the pre-cursor and POST_SIGN to 0 to boost the transition edge using a pre-emphasis pulse. DLY_POST is set to 0.

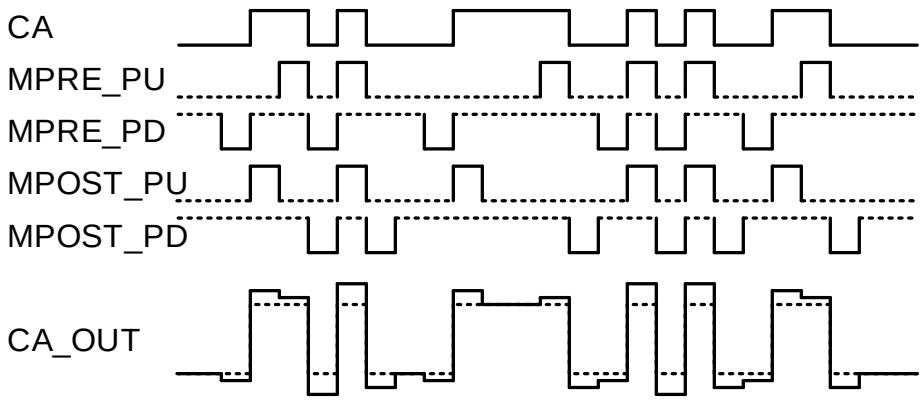
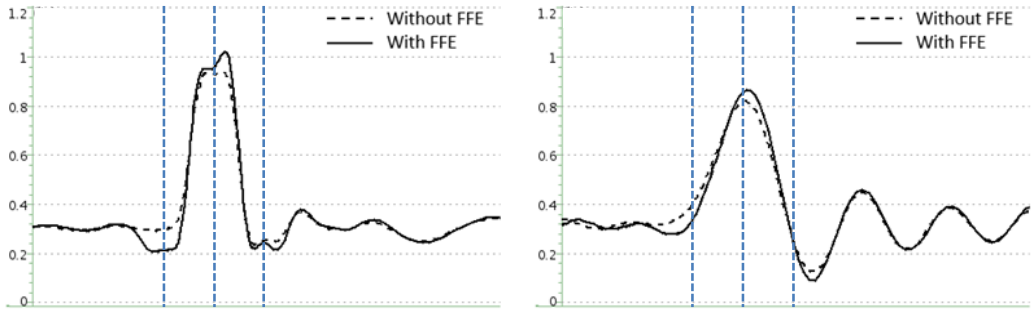


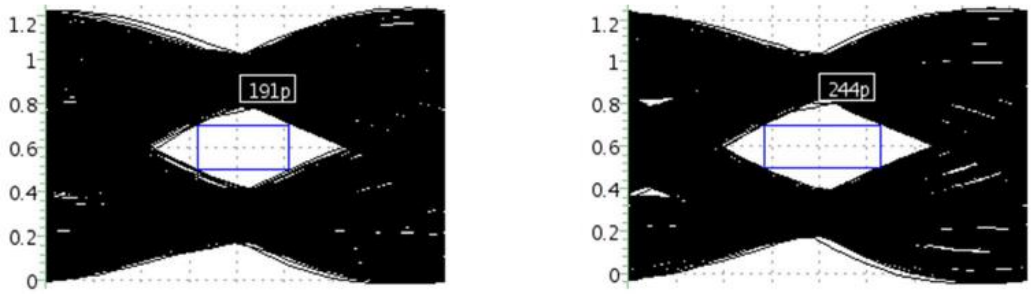
Figure 3.3.2 Example timing of the PB-FFE.

Figure 3.3.2 is an example timing diagram. The signals of MPRE_PU, MPRE_PD, MPOST_PU, and MPOST_PD are the equalizing pulses of each pre-tap and post-tap, as shown in Figure 3.3.1. Each pre-tap and post-tap driver operates before and after the transition, and the PB-FFE performs the same function as a conventional FFE at the output of the C/A transmitter while keeps Ron unchanged. Figure 3.3.3 shows the simulated single-bit response of the transmitter with and without applying the PB-FFE. The simulation result shows that the pre-cursor is reduced by 40mV, and the transition is boosted by 20mV with pre-emphasis. Figure 3.3.4 shows the simulated eye diagram of the farthest DRAM input with and without applying the PB-FFE. The simulation result shows that the timing margin is increased by 53ps for the same input mask.



(a) (b)

Figure 3.3.3 Simulated single-bit response of (a) the nearest and (b) the farthest DRAM from the controller PHY.



(a) (b)

Figure 3.3.4 Simulated eye diagram of the farthest DRAM input (a) without and (b) with applying the PB-FFE at 1.2Gbps. The vertical eye mask is 200mV.

CHAPTER 4

CIRCUIT IMPLEMENTATION

4.1 BUILDING BLOCKS

4.1.1 ALL-DIGITAL PHASE-LOCKED LOOP (ADPLL)

The ADPLL used for the MDS controller is an Integer-N structure, and the supported frequency range is 533MHz ~ 1067MHz, and the reference frequency is 133MHz. Figure 4.1.1 shows the block diagram of the ADPLL. The ADPLL consists of the phase-frequency detectable time-to-digital converter (PFDTDC), digital loop filter (DLF) with 1st order delta-sigma modulator (DSM), digitally controlled oscillator (DCO) and frequency divider (/N).

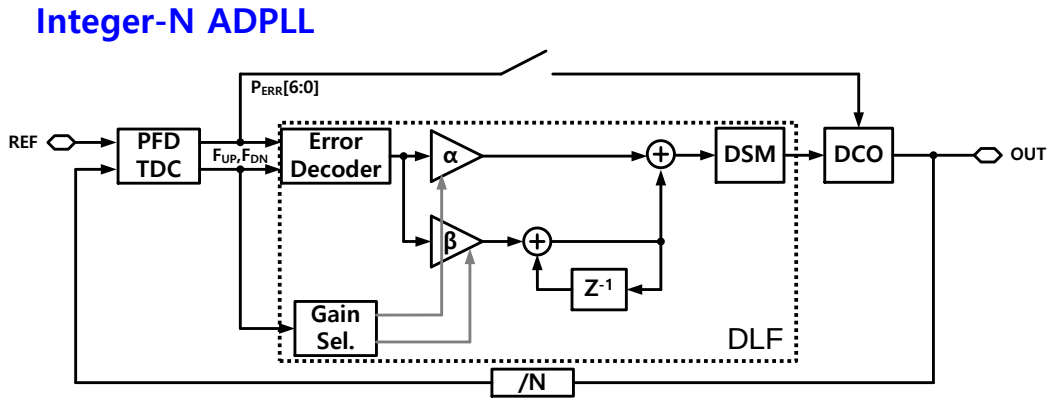


Figure 4.1.1 Block diagram of the ADPLL

A PFDTDC combines vernier TDC and the PFD to generate both high resolution and wide range TDC information with up and down information. It has dynamic range of 360ps with the resolution 12ps. The DLF provides control code of the DCO by proportional and integral path. The DSM reduces the in-band noise by shaping quantization noise and increases the effective resolution of the DCO.

The quality of the clock synthesized by the ADPLL is determined after the phase is locked. In the previous process, in the section tracking the frequency, finding the locking point as quickly as possible is efficient. In this structure, the frequency locking bandwidth and the phase-locking bandwidth are designed differently using gain selection block, so that the frequency lock is performed quickly and the PLL operates with the optimal bandwidth after the phase is locked.

If a bandwidth with optimal clock quality is used, the frequency locking time will be determined by the initial code and bandwidth. Even if the initial code matches the target code, the long locking time is inevitable because the optimum phase-locking bandwidth is usually very low as a range of several MHz. Therefore, in this structure, locking is performed using a high gain for high bandwidth in a frequency locking section. This gain control can be easily implemented using the structure of PFDTDC, which is one of the building blocks of ADPLL, and PFDTDC delivers FUP, FDN, and PERR [6: 0] signals to DLF. At this time, if an FUP or an FDN occurs, the DLF calculates it using the previously set high gain and transmits it to the DCO control code. Through this process, the frequency can be quickly tracked in the frequency locking section. The optimum proportional and

integral gain value in the phase-locking section can be found by measurement. Figure 4.1.2 shows the simulated locking procedure that shows the fast frequency locking with high gain and the optimum phase locking with moderate gain.

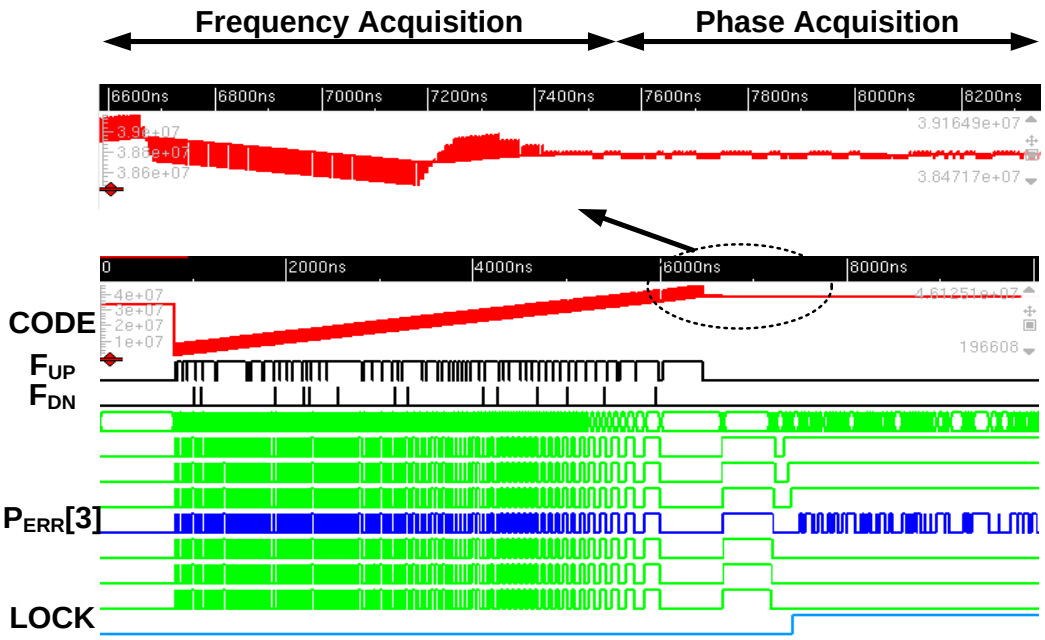


Figure 4.1.2 Simulation of the locking procedure

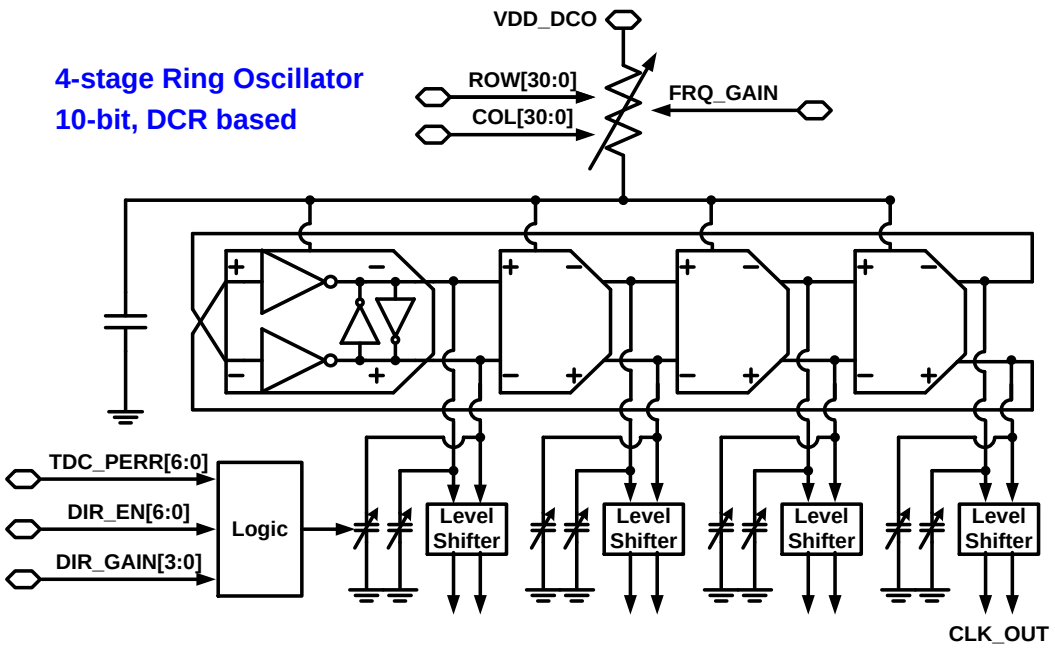


Figure 4.1.3 Block diagram of digitally controlled oscillator

The conventional structure is a method to change the control code of DCO by calculating phase error in DLF. However, since the dividing factor of our PLL is only 8, the timing guaranteed from the time of the phase error to the control code of the DCO is only 4-cycle calculated as the 'Reference clock period/DLF clock period'. The resulting loop latency not only degrades clock quality but also adversely affects stability. A proportional control structure that directly feeds the output from the PFDTDC into the input of the DCO was selected to solve this problem, as shown in Figure 4.1.3.

In the above structure, there is no delay due to DLF calculation because the output of the PFDTDC directly controls the loading capacitance of the delay cell. Therefore, a clock with better performance can be synthesized. Since the high gain in the frequency locking

section, mentioned above, is still available with this structure, fast locking can be implemented using this structure.

4.1.2 ALL-DIGITAL DELAY-LOCKED LOOP (ADDLL)

The PHY must perform training at 1/128 cycle intervals, and for this, each transceiver must operate at the corresponding phase resolution. To this end, we have one DLL in our PHY, and the DLL does not directly provide the delayed clock signal but only finds the delay code corresponding to the 1-cycle of PHYCLK and provides it to each transceiver. The local DCDL of each transceiver generates a clock signal with phase resolution of 1/128 clock cycle by dividing the 1-cycle lock code supplied by ADDLL into an 8-bit delay-control code.

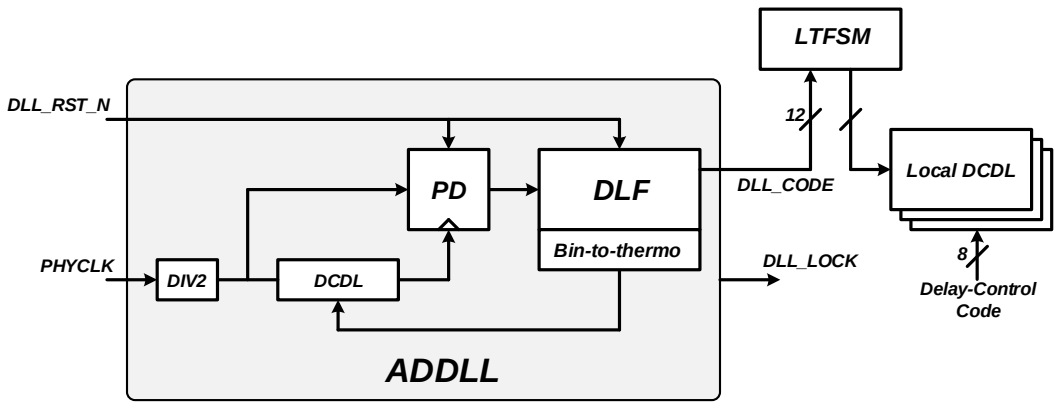
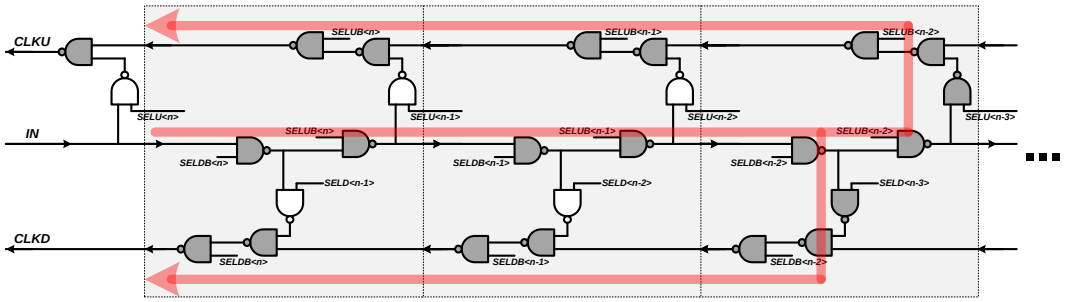
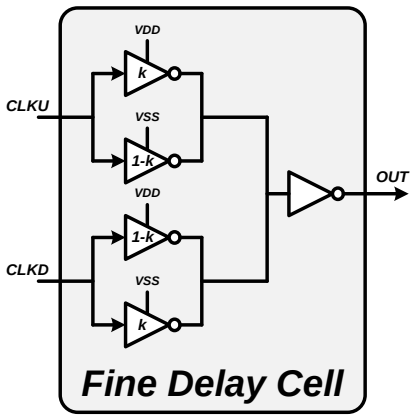


Figure 4.1.4 Block diagram of the ADDLL



(a)



(b)

Figure 4.1.5 (a) NAND-based coarse delay cell and (b) phase interpolating fine delay cell

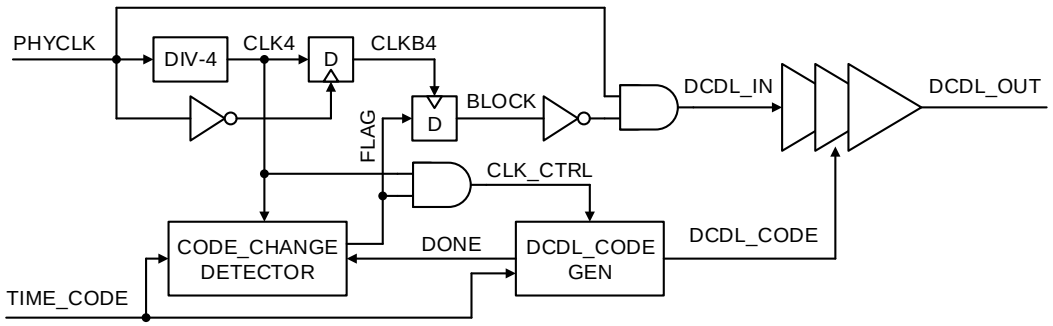
ADDLL consists of input divider, DCDL, phase detector, and digital loop filter, as shown in Figure 4.1.4. To prevent harmonic lock, the input clock is divided, and the DCDL is locked at a delay corresponding to the 180° phase of the divided clock. The DCDL consists of the coarse delay line and the fine delay line in Figure 4.1.5. The coarse delay line is a NAND-based lattice structure with seamless boundary switching scheme [4.1.1], and the fine delay is generated by phase interpolator using starved inverters. Starting with the minimum delay of the DCDL, the coarse code of the DCDL increases until the coarse

lock occurs. After the coarse lock, the DLL code decreases, and the DLF counts the output of the phase detector. When the counting value is 4, the DLF outputs DCDL_CODE of 12-bit with DLL_LOCK signal to the LTFSM. The LTFSM passes the DLL_CODE to the local DCDLs in each transceiver.

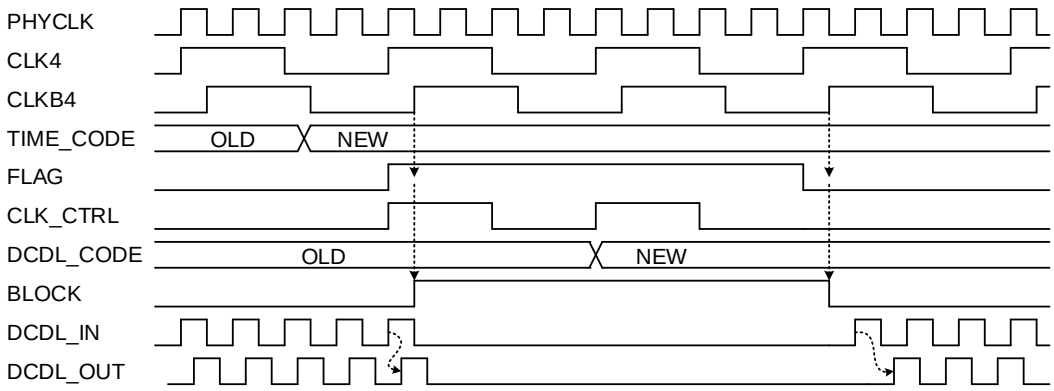
4.1.3 GLITCH-FREE DCDL CONTROL

When changing the delay of the DCDL, a glitch problem may occur. A glitch in the output of a DCDL causes a phase error in the divided clock signal reaching the serializer (SER) and deserializer (DES), and write or read failure occurs accordingly. To avoid this problem, the delay of the DCDL may change step by step from the final value to the target value after finishing the training. The length of a step is T_{CK}/128, and the DCDLs have a range of 0-127 steps in our PHY to cover the delay of 1.5×T_{CK}. Thus it may take 768×T_{CK} to adjust the DCDL for the worst-case when changing delay in every 4 cycles of PHYCLK.

An alternative way to suppress glitches is to switch each delay stage sequentially, but an implementation [4.1.2] of this scheme requires 1.5 times the area and power. Since our PHY has 5 DCDLs in each nibble and 11 in each CA group, making a total of 152 DCDLs, it is difficult to employ a DCDL structure with a large area and power consumption as described in [4.1.2].



(a)



(b)

Figure 4.1.6 (a) Block diagram and (b) timing diagram of the proposed glitch-free DCDL.

In our controller, there is a period of bus idle time when changing the delay after training, and the DCDL is implemented so that a glitch is not transmitted to the SER or DES by blocking DCDL output using this period. Figure 4.1.6 (a) is a block diagram of the glitch-free DCDL in our PHY, and Figure 4.1.6 (b) is its timing diagram. When TIME_CODE, which determines the delay, is changed by the LTFSM, the code change detector asserts the FLAG signal, and the clock signal DCDL_IN to the DCDL is blocked

accordingly. Then the internal delay-control code DCDL_CODE is changed. Finally, the FLAG signal is de-asserted and the clock signal is again supplied to the DCDL. The code change detector is supplied with CLK4, which is a ¼-rate version of PHYCLK. The clock signal to the DCDL is blocked for two cycles of CLK4, which avoids EVEN/ODD phase inversion problems. Adopting this glitch-free DCDL, the required time to update the training result is reduced to only $8\times T_{CK}$ and the overall training time is reduced compared to the step-by-step methodology described above. During training procedure, the blocking period of the DCDL does not affect the overall training time because there is a waiting time, longer than the blocking period, to receive feedback from the DRAM after transmitting a training pattern.

4.1.4 DUTY-CYCLE CORRECTOR (DCC)

DQ signal operating in DDR mode uses both rising and falling edges of the clock signal. When the clock signal's duty-cycle error occurs, the data valid time (tDV) decreases due to the difference between the even and the odd phase widths of the output DQ signal. Therefore, the DQ transmitter needs duty-cycle correction (DCC) circuits of the clock signal.

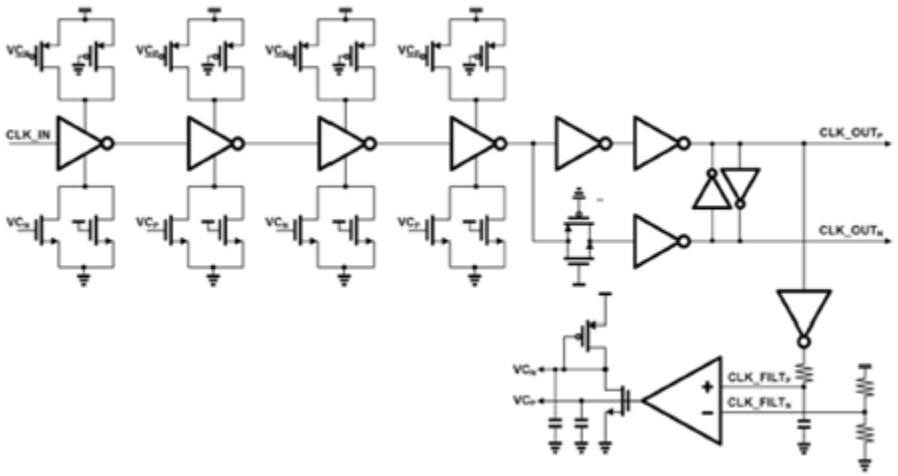


Figure 4.1.7 Schematic of the duty-cycle corrector

The DCC has a structure that receives a single-ended clock signal and outputs a differential clock. The analog method was used to compensate for the duty error, and the DCC circuit is composed of RC filter, OP-AMP, and multiple stages of starved inverter type delay cells, as shown in Figure 4.1.7. The output duty error is converted into a voltage through the RC filter. The OP-AMP converts the difference between the converted voltage

and the target voltage VDD/2 into a duty control voltage. This control voltage affects the delay cells so that the duty-cycle is adjusted to 50%. The DCC operates up to 1200MHz and corrects input duty error of 40-60% to less than 5%.

In the case of analog type DCC, the duty-cycle error of the first output clock may be larger than that of the input clock signal according to the initial bias condition before the clock is applied. Therefore, it is necessary to appropriately limit this to the extent that malfunction does not occur, which is a trade-off relationship with the correcting range. Our DCC limits the output duty error in the initial state to 20% and achieves an output duty-cycle error of less than 5% within eight cycles.

4.1.5 DQ/DQS TRANSMITTER

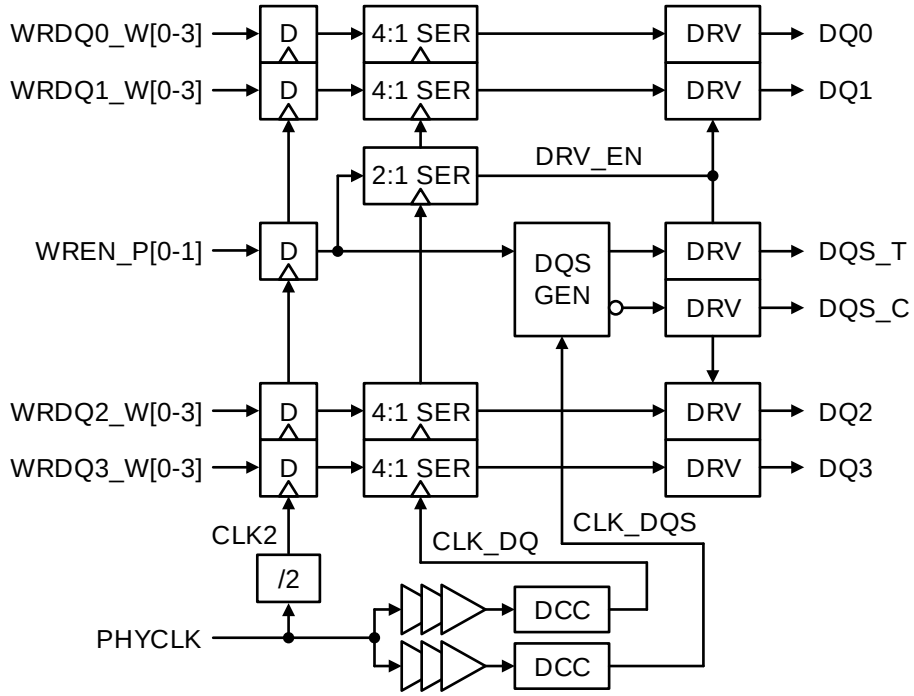


Figure 4.1.8 Block diagram of a nibble of DQ/DQS Tx.

Figure 4.1.8 shows a section of the DQ/DQS transmitter, which serializes a nibble of data from the LTFSM and outputs it in DDR mode. The transmitter receives the four 4-phase data signals WRDQ0-WRDQ3 from the LTFSM, together with the 2-phase write-enable signal WREN_P. The arriving data is sampled using the CLK2 signal to avoid domain crossing problems caused by variable delay. The strobe generator DQS_GEN generates the DQS preamble pattern specified by the mode register, and subsequently the clock pattern; and these are output as the strobe signal DQS_T, together with its differential pair DQS_C, by two of the drivers DRV. These drivers, together with the other four

drivers DRV which output the data signals, are turned on by the driver-enable signal DRV_EN, which is generated by serializing WREN_P.

The controller PHY supports per-nibble write leveling and write training. The DCDL for CLK_DQS is adjusted so that CK and DQS are aligned at the DRAM input after write leveling. The DCDL for CLK_DQ produces 0.5 UI less delay, so that the phase of CLK_DQ leads that of CLK_DQS by 90°. This delay, which is fine-tuned during write training, allows DRAM to sample DQ with DQS. Because both the rising and falling edges of the clock signals are used to output DQ in DDR mode, the duty-cycle error of each DCDL is corrected by the DCC described in chapter 4.1.4.

4.1.6 DQ/DQS RECEIVER

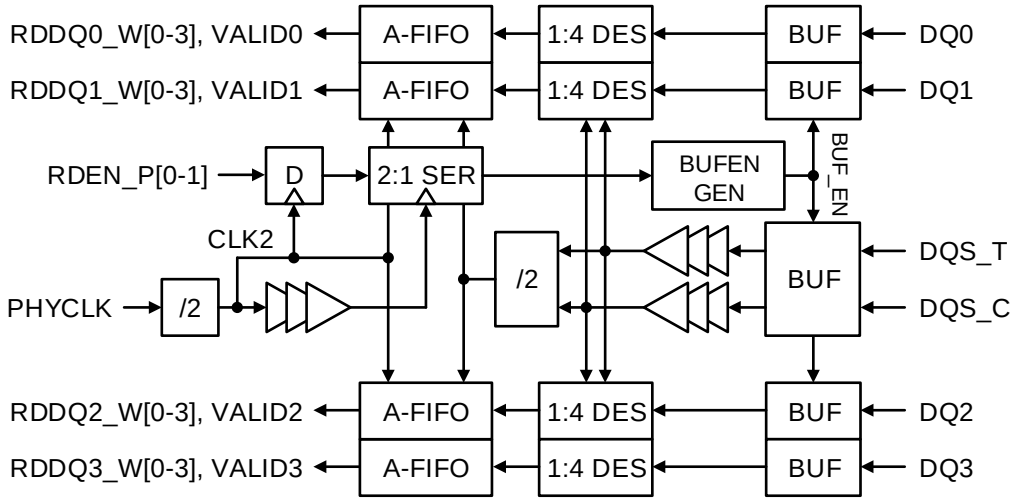


Figure 4.1.9 Block diagram of a nibble of DQ/DQS Rx.

Figure 4.1.9 shows a 1-nibble DQ / DQS receiver, which receives the four data signals DQ0-DQ3 from the DRAM, together with the strobes DQS_T and DQS_C. The time that read data arrives at the PHY after transmitting the read command is '$T_{CMD} + RL \times T_{CK} + T_{DQ}$', where T_{CMD} is the flight time of the C/A signals from the transmitter to each DRAM; T_{CK} is the period of PHYCLK; and T_{DQ} is the flight time of the DQ and DQS signals from the transmitter of each DRAM to the PHY. *RL*, which is defined in mode register, denotes read latency that is the number of clock cycles until a DRAM outputs the read data after receiving a read command.

The time at which the data arrives is not synchronized to PHYCLK and each nibble arrives at different times. Thus the buffer enable time must be different for each nibble, and

this is determined by read preamble training, which is performed as follows. The LTFSM sends the 2-phase read enable signal RDEN_P to the receiver after a delay of RL, and the receiver serializes it to generate the BUF_EN signal. The correct buffer enable time is determined by issuing a read command and then waiting for the rising edge of the DQS. This edge is found by sampling the incoming DQS using the BUF_EN signal which is delayed by a step of T_{CK}/128 during read preamble training. When both the rising edge of the incoming DQS and the BUF_EN signal are aligned, the sampled DQS changes from 0 to 1, and the LTFSM terminates read preamble training and adjusts the DCDL shorter by 0.5 UI than stopped value to have buffer enable timing margin. If 'T_{CMD} + T_{DQ}' is greater than T_{CK}, the rising edge of the incoming DQS could not be found because the DQS does not arrive at the expected read latency, and in this case, LTFSM increases internal latency of RDEN_P. Since DQ and DQS for each nibble arrive at the same time, the DQ signals are sampled by a DQS with a phase lag of 90° and this delay is fine-tuned during read training. The received DQ signals are deserialized to 4-phases, and transmitted to the LTFSM through asynchronous FIFO, together with the VALID signal. When all the VALID signals from 20 nibbles are received, the LTFSM accepts the deserialized data.

4.1.7 ZQ CALIBRATION

For the signal integrity, accurate driver strength and termination is required. At READ operation, impedance matching is needed to minimize reflection caused by an impedance mismatch between the channel and the receiver, and pull-up drivers of the DQ and DQS provide termination. The impedances are 240Ω, 120Ω, 80Ω, 60Ω, 48Ω, 40Ω, and 34Ω, which is integer divided values of 240Ω. To ensure accurate driver strength and to provide precise termination, ZQ calibration is performed using external ZQ resistor connected between ZQ pin and VSS.

Figure 4.1.10 shows the block diagram of the ZQ calibration. Calibration block consists of a VREF generator, eight comparators, four 6-bit counters, drivers for calibration, and calibration control circuitries. The CA channel is a center-tap termination (CTT) structure with termination voltage VDDQ/2, and the DQ channel is a high-tap termination (HTT) structure with termination voltage VDDQ. Therefore, the calibration voltage of the CA driver and DQ driver should be different. To use one ZQ resistor, CA driver calibration is performed first, and DQ calibration is then continued.

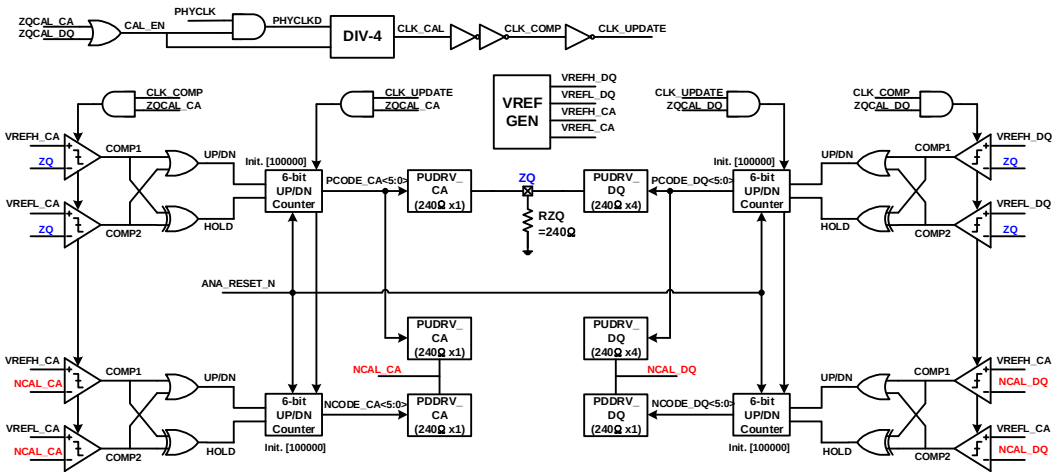


Figure 4.1.10 Block diagram of the ZQ calibration

The CA driver is calibrated at the output voltage V (ZQ) = 0.5×VDDQ. By connecting one pull-up driver with a 240-ohm ZQ resistor, and make the divided voltage to 0.5×VDDQ, the pull-up driver is calibrated to 240Ω. The calibration circuit is operated by the CLK_CAL signal, which is 1/4-rate of the PHYCLK. In every cycle of the CLK_CAL, the comparators compare the ZQ voltage to the reference voltage, and the counter value is increased or decreased according to the comparison result so that the resistance of the pull-up driver equals the ZQ resistance. To be equal to, that is, the voltage of the ZQ node is 0.5×VDDQ. At this time, to prevent the toggling of the comparison result, two reference voltages are used to form a so-called dead-zone. If V(ZQ) enters the dead-zone, that is, if two compare results are different, the calibration circuit ignores the compare result and keeps the counter value. The pull-down driver is also calibrated to 240Ω by connecting one 240Ω replica pull-up driver and one pull-down driver and calibrating the divided voltage

to be 0.5×VDDQ.

DQ driver should be calibrated at V (ZQ) = 0.8×VDDQ. To do this, four 240Ω drivers are connected with external ZQ resistors and calibrate so that the divided voltage becomes 0.8×VDDQ. The pull-down driver is calibrated to 240Ω by connecting one pull-down driver to four pull-up replica drivers and making the divided voltage equal to 0.8×VDDQ.

In the above ZQ calibration block, it is advantageous in terms of area to selectively apply VREF and share the CA calibration circuit and the DQ calibration circuit. However, because the settling time becomes a problem when changing the reference voltage between the CA calibration and the DQ calibration, we separated each calibration circuit.

4.2 MODELING AND VERIFICATION OF LINK TRAINING

Simulation results of the link training are shown in this section. The modeling verification was performed by modeling the memory and channel corresponding to the MDS DIMM as a System Verilog, and LTFSM using Verilog code for synthesis. To avoid mistakes such as signal connection errors when schematic of the PHY is modeled, functional models were created for the basic units in Verilog, and then all the circuits are extracted to the Verilog model from the top level of the PHY, similar to the process of netlist extracting, as shown in Figure 4.2.1. Modeled delay of the CLK, C/A, DQ, and DQS lines are shown in Figure 4.2.2. In this environment, the memory controller and the memory are operated to communicate through the transient simulation.

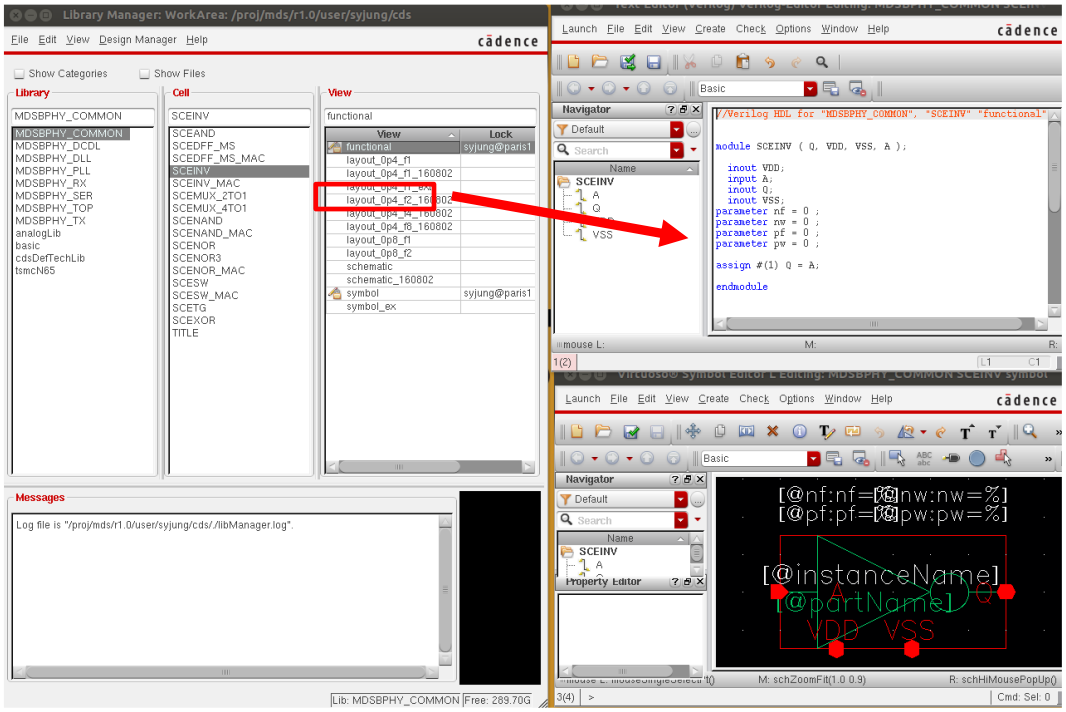


Figure 4.2.1 Verilog modeling of a functional unit

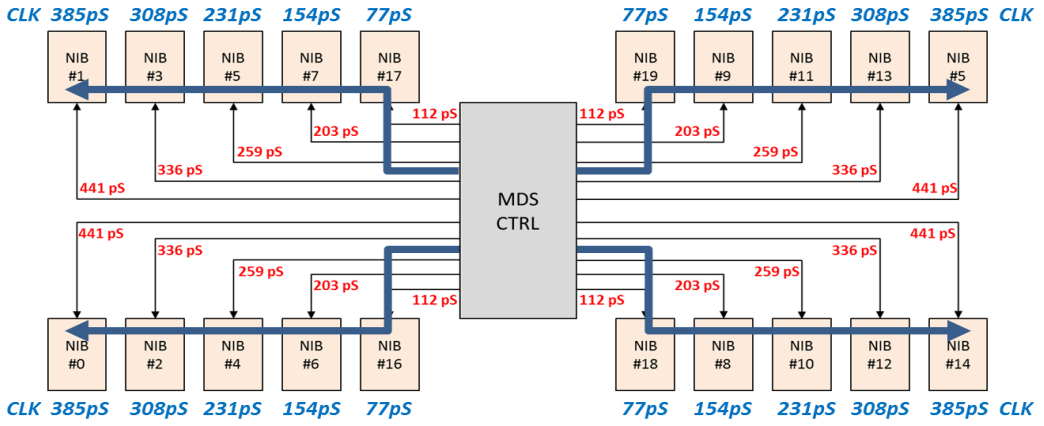


Figure 4.2.2 Modeled delay of the CLK and DQ lines. The delay of the C/A lines are equal to that of the CLK line, and the delay of the DQS lines are equal to that of the DQ lines.

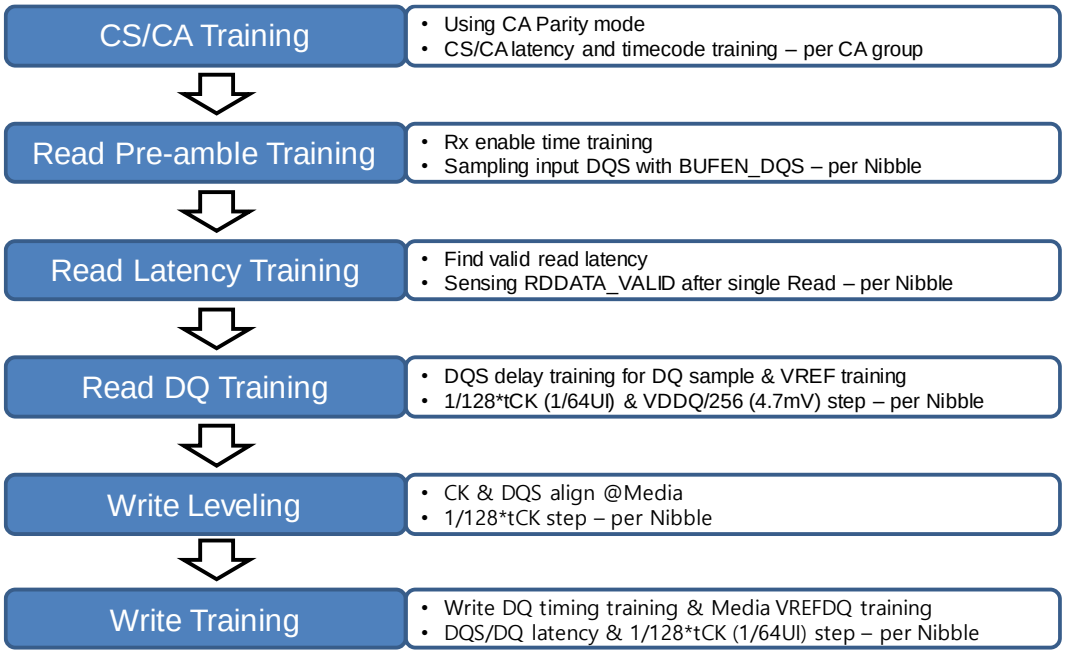


Figure 4.2.3 MDS controller PHY training sequence

Figure 4.2.3 shows the link training sequence. CA training compensates the delay difference between CLK line and the C/A lines. In the simulation environment based on delay model in Figure 4.2.2, the delay of the CLK and C/A lines are same, and all the training results of the C/A group is same.

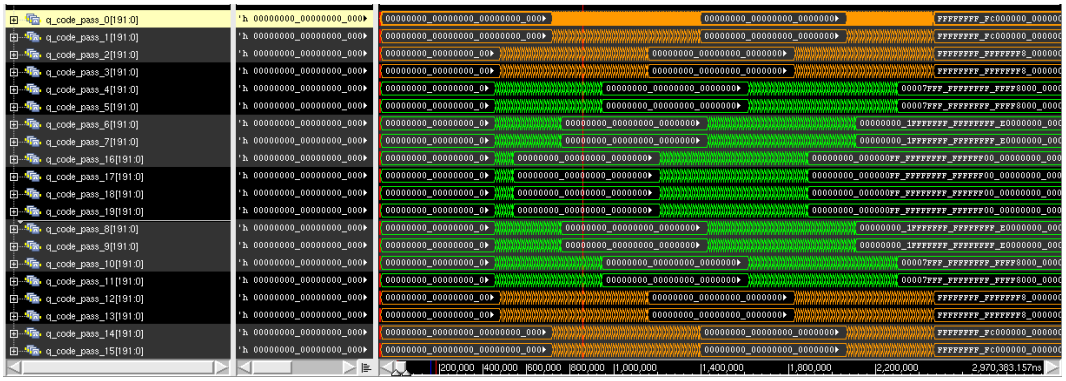


Figure 4.2.4 Simulation result of the read preamble training

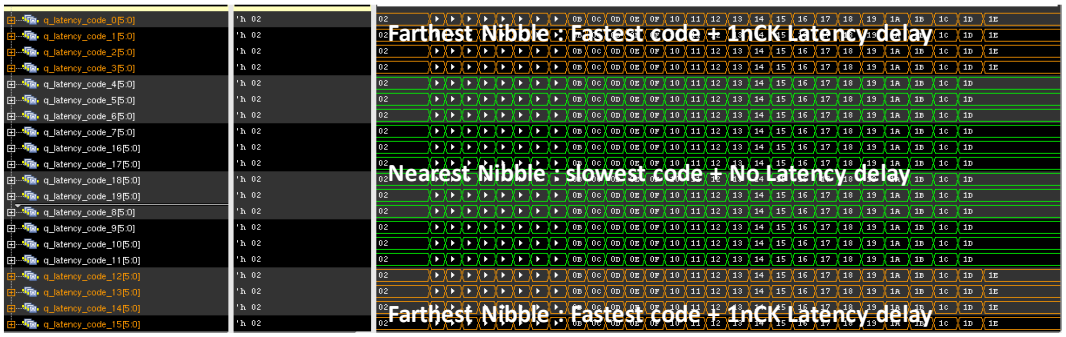


Figure 4.2.5 Simulation result of the read latency training

Read preamble training compensates the difference of the turnaround delay, which is the sum of the CLK delay and the returned DQS delay, between nibbles. Figure 4.2.4 shows the result of the read preamble training, and it is shown that the trained value is different and classified according to the delay between controller and each nibble. The trained read latency for each nibble is shown in Figure 4.2.5. If the above turnaround delay is larger than one cycle of the PHYCLK, the internal latency of the PHY is increased to compensate the large delay. As shown in figure, for the farthest nibbles, the trained latency is bigger by

1 than the other nibbles. In other words, the incoming read data and DQS from the farthest nibble is sampled at the early phase of the next cycle because they do not arrive with the present clock cycle.

The DQ output from the DRAM is aligned to the DQS signals. To capture DQ data using DQS, the incoming DQS is delayed during the read data training procedure to allow sampling DQ using the delayed DQS. If the buffer of the controller receiver receives incoming DQ and DQS, the internal delay difference between the DQ and DQS is same for each nibble, especially in the modeling environment. Therefore, the read training results are all the same for all nibbles, as shown in Figure 4.2.6. The incoming DQS is delayed from 0 to 1.5 cycle of the PHYCLK. After the training, the delay is adjusted to the center of the larger pass-zone, indicated by the blue line in the figure.

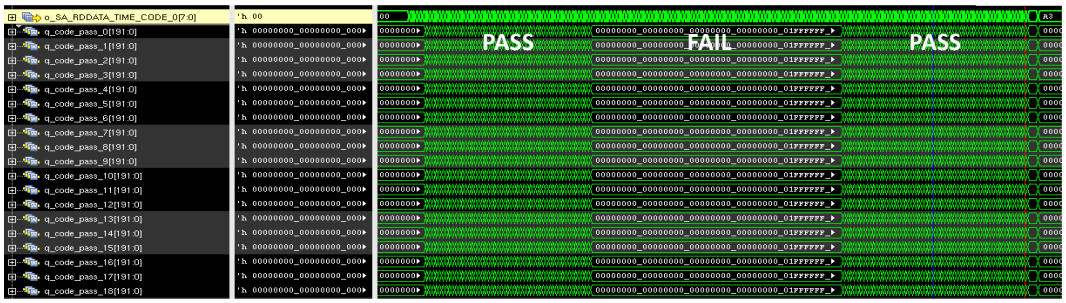


Figure 4.2.6 Simulation result of the read training

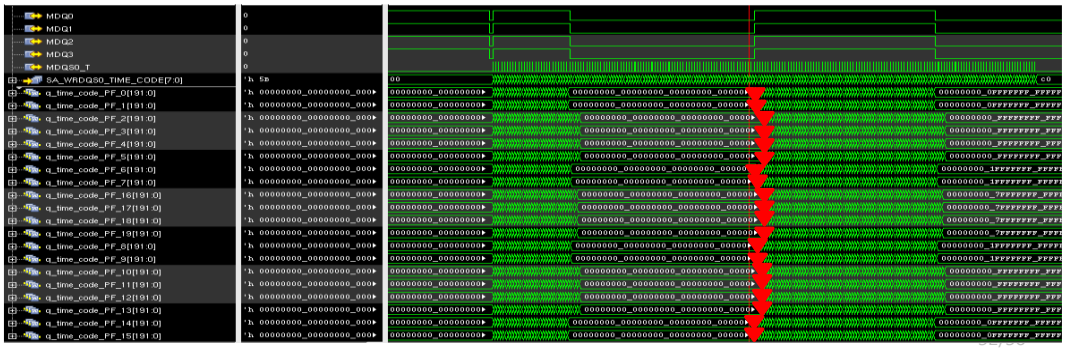


Figure 4.2.7 Simulation result of the write leveling

Because the delay between CLK and DQS from the controller to each DRAM are different, clock domain crossing problems may occur inside the DRAM. Therefore, the write leveling, which compensates the delay difference between CLK signal and the DQS signals, should be performed before the write training. In Figure 4.2.2, the delay difference between the CLK and DQ lines at the nibbles such of NIB #2-#5 and NIB #10-#13 is smaller than the others. In accordance with this delay difference, the simulation result of the write leveling, in Figure 4.2.7, shows that the output delay of the DQS of these nibbles are larger than the others.

Since the DRAM does not have the delay in the DQS, the controller should transmit DQS later than DQ to allow the DRAM to capture DQ data using DQS. During write training, the controller adjusts the delay of the DQ because the DQS timing is predetermined by the write leveling. The simulation result of the write training, in Figure 4.2.8, shows that the delay difference of the DQ between nibbles is same as the delay difference of the DQS's from the write leveling. Because the internal delay of all the DRAMs are identical in the simulation, this makes sense.

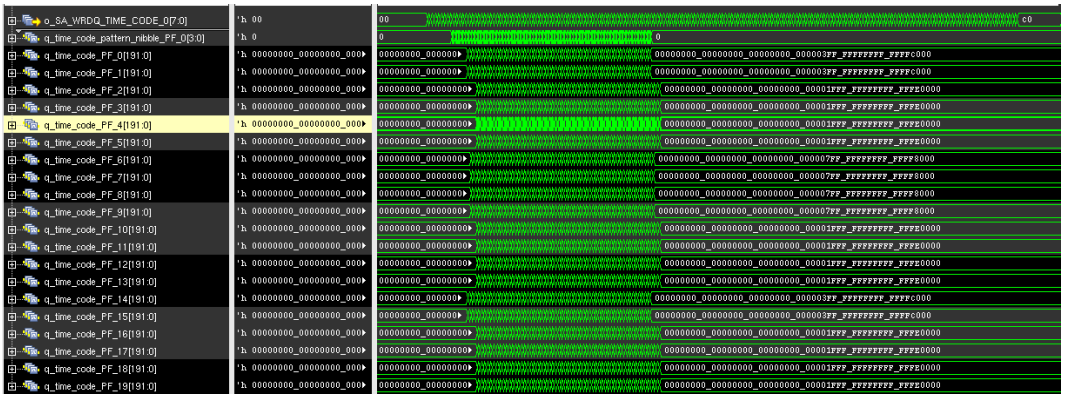


Figure 4.2.8 Simulation result of the write training

4.3 BUILT-IN SELF-TEST CIRCUITS

A simple built-in self-test (BIST) circuit is provided so that several clock signals can be monitored, and the PHY can test its own operation.

Figure 4.3.1 is a structure for monitoring the output of main circuits related to on-chip clocking such as PLL, DLL, DCC, and CLK tree. The MUX in the PLL selects one of the clock signals generated by the PLL, and SYSCLK returned through LTFSM. The MUX in front of the DLL selects one of the clock signals that have passed or not passed through the clock tree and inputs it to the DLL. Subsequently, the clock phase change according to the lock code and delay-control code of the DLL can be monitored, and the clock duty characteristic according to the DCC operation can also be observed according to the DCC bypass test-mode signal. MUX of the last output stage selects one of the PLL output signal and DLL and DCC output signal and outputs it to PAD.

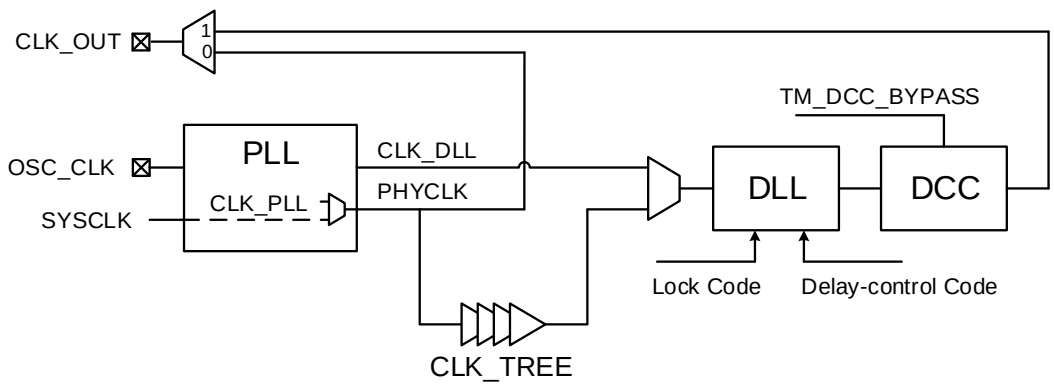


Figure 4.3.1 Block diagram of clock monitoring

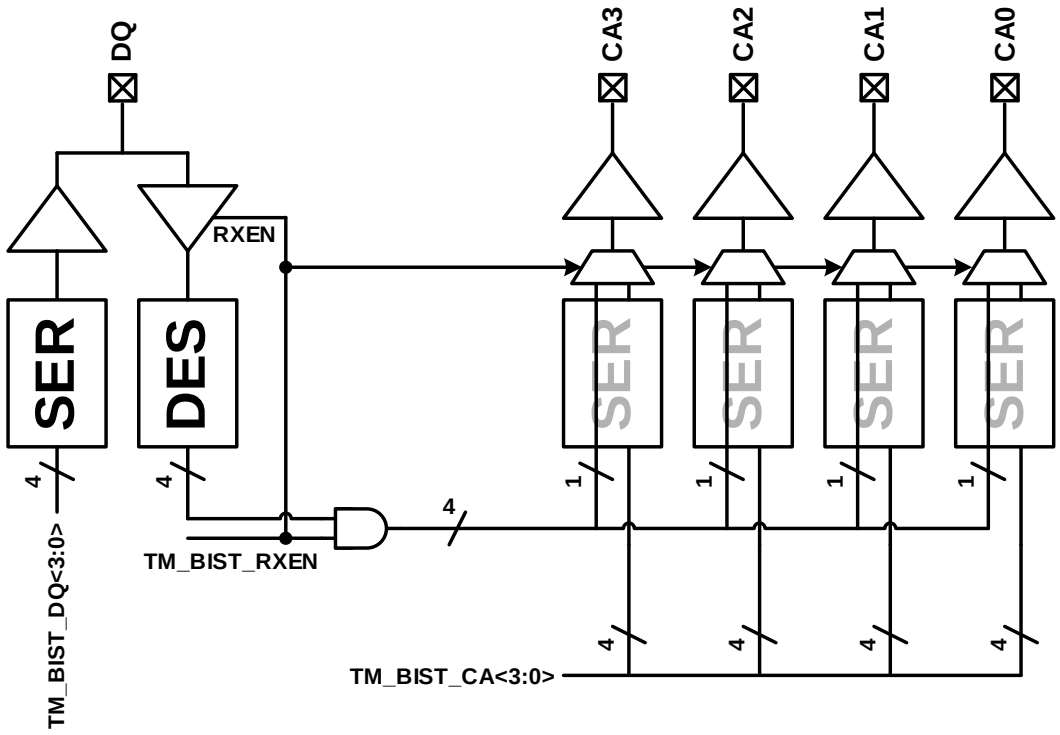


Figure 4.3.2 BIST for serializer and deserializer

Figure 4.3.2 is a BIST circuit to check the operation of the serializer and the deserializer. DQs serialize the 4-bit test-mode signal to 4:1 and output the result in the BIST mode for the DQ transmitter. For simple operation checks, the only 4-bit test signal is allocated, and output DQ is a form in which 4-bit is repeated. The DQ transmitter operates in the same way in the DQ Rx BIST mode. However, the receiver samples and deserializes the output DQ. And the result is assigned to 4 C / A pins and output in DC signal form. The absolute value may vary depending on the starting position of sampling and deserializing, but the order of the output of CA3-CA0 must match TM_BIST_DQ <3: 0>. In BIST mode for C / A serializer test, all C / A pins serialize TM_BIST_CA <3: 0> and output the result.

Besides, some circuits for the connectivity test are added. TX01 mode, which outputs 0 or 1 depending on the test-mode, is implemented in all the transmitters, including CK Tx. And on the receiver side, NAND tree test mode [4.3.1] is implemented.

CHAPTER 5

QUADRATURE ERROR CORRECTOR USING REPLICA SERIALIZERS AND PULSE-SHRINKING DELAY LINES

5.1 PHASE CORRECTION USING REPLICA SERIALIZERS AND PULSE-SHRINKING UNITS

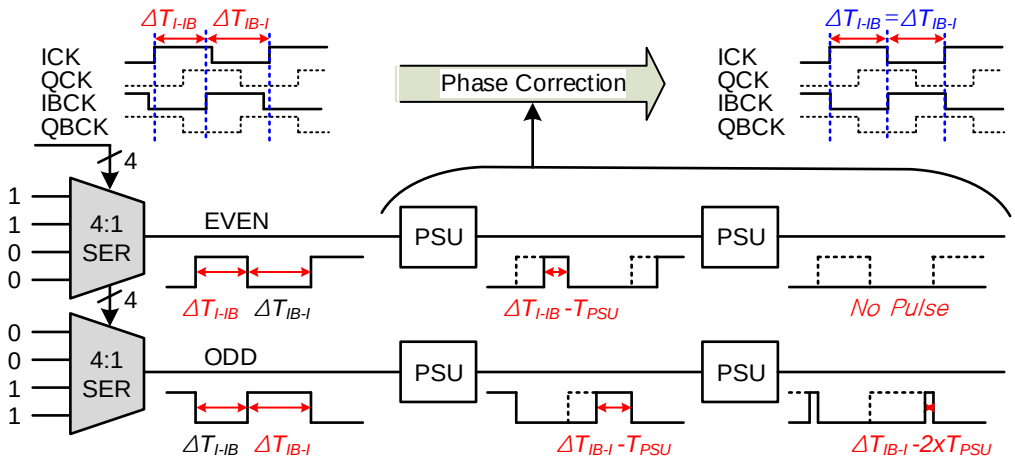


Figure 5.1.1 Simplified block diagram of phase correction using replicated serializers and pulse-shrinking units.

Figure 5.1.1 is a simplified block diagram of phase correction using two replicated SERs and pulse-shrinking units (PSUs). T_{PSU} is the delay of a rising edge through a PSU, the $\Delta T_{I\text{-}IB}$ is the time difference between the rising edges of ICK and IBCK, and $\Delta T_{IB\text{-}I}$ is the interval between the rising edges of IBCK and ICK. Each SER converts the clock phase differences $\Delta T_{I\text{-}IB}$ and $\Delta T_{IB\text{-}I}$ to pulses, and their widths are compared using multi-stage PSUs. As each signal passes through a PSU, its rising edge is delayed by T_{PSU}, reducing the pulse width, so that either the EVEN or the ODD pulse disappears. Using the result of this comparison, the clock phases are corrected and $\Delta T_{I\text{-}IB}$ and $\Delta T_{IB\text{-}I}$ become equal. Because the phase error is detected based on the output of the 4:1 SER, compensation includes errors caused by the PVT variations in the SER. Two SERs are used to prevent mismatches that can occur when the output of one SER is inverted for the comparison described above. For simplicity, Figure 5.1.1 only shows the correction of the phase between ICK and IBCK; the other phases are corrected in the same way.

5.2 OVERALL QEC ARCHITECTURE AND ITS OPERATION

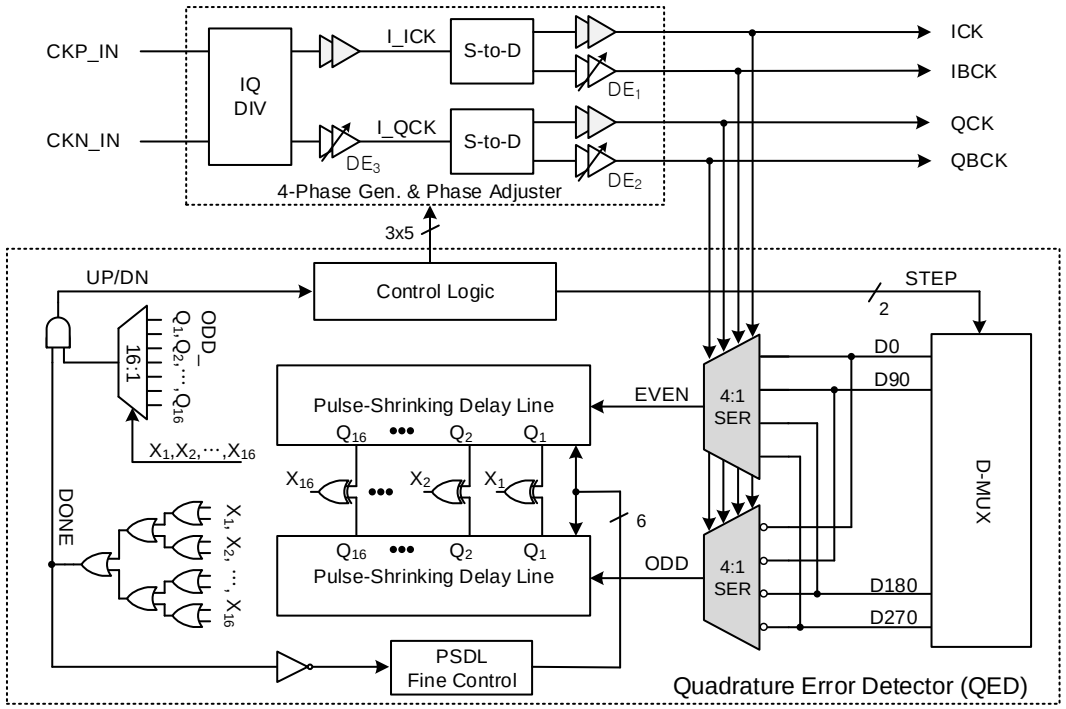


Figure 5.2.1 Block diagram of the prototype QEC

Figure 5.2.1 is a block diagram of the prototype QEC, which consists of a four-phase generator, a phase adjuster, and a quadrature error detector (QED). Four quarter-rate clock signals with 90° phase separation are generated by dividing two half-rate clocks, CKP_IN and CKN_IN, and applying the divided clocks to two single-to-differential converters.

The QEC corrects the quadrature errors between the four-phase clock signals in three steps without additional iteration. The 180° phase between ICK and IBCK is corrected first

using a delay element DE_1, and then the 180° phase between QCK and QBCK is corrected using a delay element DE_2. Finally, the phases between ICK and QCK and between IBCK and QBCK are corrected simultaneously using delay element DE_3 by correcting the 90° phase between I_ICK and I_QCK, which are the clock signals before single-to-differential conversion. Each delay element in the phase adjuster is based on binary-weighted MOSCAP delays that provide a resolution of 1ps.

The QEC corrects the phase error by comparing and equalizing the pulse widths of an EVEN and an ODD signal, which are complementary clock patterns generated by two replicated 4:1 SERs that receive the appropriate data from the D-MUX for each step of the correction process. During the first step to correcting phases between ICK and IBCK, the inputs to the 4:1 SERs, D0, D90, D180, and D270, are 1, 1, 0, and 0 respectively, so that the data pattern of EVEN is 1100 and the data pattern of ODD is 0011. By equalizing the pulse widths of these two signals, the phase difference between ICK and IBCK becomes 180°. During the second step for correcting phases between QCK and QBCK, the four inputs to the SERs are 0, 1, 1, and 0, so that EVEN is 0110 and ODD is 1001; and during the last step, the inputs are 1, 0, 1, and 0, making EVEN 1010 and ODD 0101.

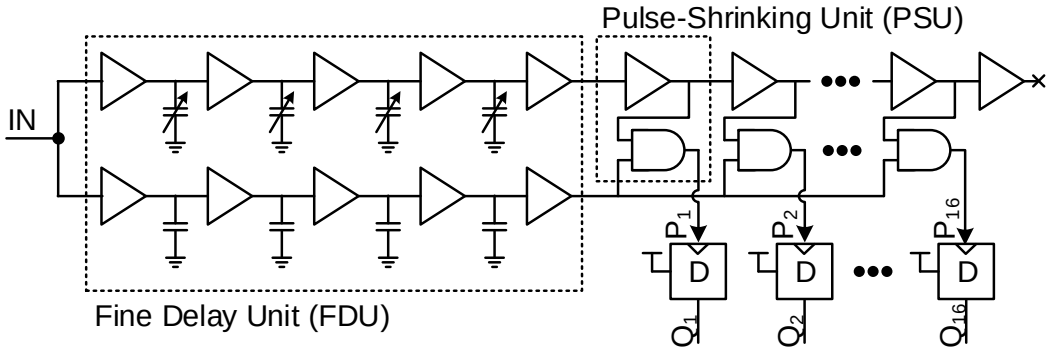


Figure 5.2.2 Block diagram of a pulse-shrinking delay line.

Two PSDLs are used to compare the pulse widths of the EVEN and ODD signals. A PSDL consists of a fine delay unit (FDU), together with 16 stages of PSUs and D flip-flops (DFFs). Each PSU includes a buffer and an AND gate, as shown in Figure 5.2.2. The pulse output by each PSU is gradually shrunk by one buffer delay, T_{PSU}, and is applied to the clock input of the corresponding DFF, whose data input is tied to VDD. The true single-phase clocked (TSPC) DFFs are used to eliminate the side-effects of generating a complementary pulse from a PSU output. Because the pulse widths of the EVEN and ODD signals in the first and second step are twice the unit-interval (UI), the number of PSU stages needs to be larger than $2\times UI/T_{PSU}$.

Starting with all DFFs reset, the output of a DFF changes to 1 if the corresponding PSU output carries the pulse, whereas the output of the DFF remains 0 if the corresponding PSU output is not pulsing. The output of each PSDL, Q_1-Q_{16}, consists of a sequence of 1s followed by a sequence of 0s, i.e. of the form '111…000', and the number of consecutive 1s corresponds to the width of the pulse at the input to the PSDL. However, the exact

number of 1s in the PSDL output sequence is not a concern in our QEC; the difference between the two outputs is important. To minimize mismatch, the two PSDLs are laid out as symmetrically as possible, including dummy patterns [5.2.1]. The detection offset can be monitored by the '1100' pattern output of the prototype Tx. The duty cycle error corresponds to the offset between the EVEN_PSDL and the ODD_PSDL, and this offset can be improved by adjusting the initial delays of the two PSDLs.

The 16 XOR gates shown in Figure 5.2.1 compare the outputs of the two PSDLs. When they differ, one of the XOR outputs, X_K from X_1 to X_{16}, becomes 1 and the corresponding output ODD_Q_K from the ODD PSDL indicates which of the pulse width of EVEN or ODD is larger. If ODD_Q_K is 1, the QEC increases the delay of the clock signal, and if ODD_Q_K is 0, the QEC decreases the delay.

The above comparison and update procedure is performed every 32 cycles, with a loop bandwidth of 200MHz at 12.8Gbps. To minimize the effects of input jitter and power noise, 14 pulses are cumulatively sampled and compared in the first and second steps, and 28 pulses in the third step.

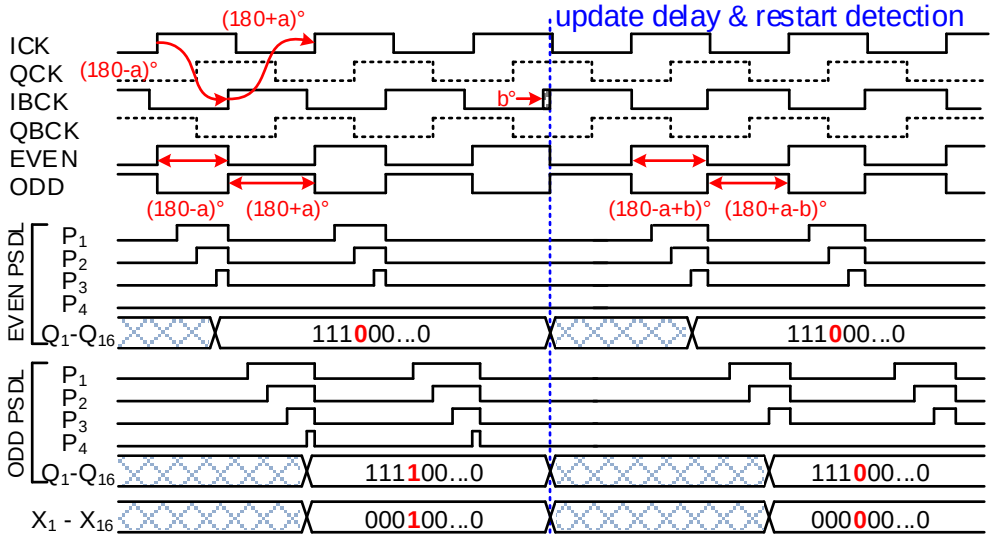


Figure 5.2.3 Timing diagram of the QEC correcting phase between ICK and IBCK.

Figure 5.2.3 shows an example timing diagram for the first step to correct the phase error between ICK and IBCK. Since IBCK has been pulled by a°, the pulse widths of EVEN and ODD are (180 – a)° and (180 + a)° respectively. The output P_4 from the fourth PSU of the EVEN PSDL is low, but the ODD PSDL is still asserted. Thus the output of the corresponding DFF, EVEN_Q_4, is 0, whereas ODD_Q_4 is 1. Then X_4 is the only 1 among the XOR gate outputs X_1 to X_{16}. The 16:1 multiplexer passes ODD_Q_4, and the UP/DN signal is also 1. In response to this signal, the control logic increases the IBCK delay by an amount corresponding to b°, which is the delay control step of the phase adjuster. Now the phase error is reduced to |a – b|° and when this is smaller than one buffer delay, the pulse traveling through each PSDL is shrunk to nothing in the same position in each PSDL, and the outputs of both PSDLs can be the same.

5.3 FINE DELAY UNIT IN THE PSDL

To improve the accuracy with which phase errors are detected, a fine delay unit (FDU) is located in front of the PSU of the first stage. The FDU applies a variable delay to a shrinking edge and a fixed delay to a non-shrinking edge, as shown in Figure 5.2.2. Four stages of 3-bit binary-weighted MOSCAP delay elements produce a total variable delay that is larger than the delay associated with one stage of a PSU. In our design, one buffer delay of a PSU is 26ps from the post-layout simulation of the typical corner. Hence the FDU is implemented in 32 steps with 1ps resolution to cover the T_{PSU}, and for all PVT corners, the delay of FDU is larger than T_{PSU}. The capacitive load is distributed across the delay elements to improve the bandwidth.

Initially, both variable and fixed delays of an FDU have the same minimum value. If the outputs of the two PSDLs are the same, then the XOR outputs X_1 to X_{16}, shown in Figure 5.2.1, are all 0, and the DONE signal is not asserted. In this case, the FDUs of both PSDLs increase the variable delay and push the shrinking edges so that the shorter pulse disappears one stage earlier and the outputs of the two PSDLs differ. The QEC repeatedly updates the clock phases and adjusts the fine delay until the difference between the pulse widths of the EVEN and ODD signal is less than the resolution of the variable delay provided by the FDU. When the comparison result has changed three times, the QEC holds the corrected clock phase and advances to the next step to repeat the correction procedure for the next pair of clock signals. After all three steps are completed, the QEC enters standby state to save power and then resumes the correction procedure by an external

command.

CHAPTER 6

EXPERIMENTAL RESULTS

6.1 CONTROLLER PHY

The MDS controller chip shown in Figure 6.1.1 was fabricated in a 55nm CMOS process, and it occupies 118.6mm^2. The controller was mounted on an MDS DIMM and interfaces with the 40 DRAM packages, in which 20 are on the front side, and 20 are on the backside, as shown in Figure 6.1.2. The MDS DIMM is mounted on the test board and the overall measurement setup is shown in Figure 6.1.2 and Figure 6.1.3.

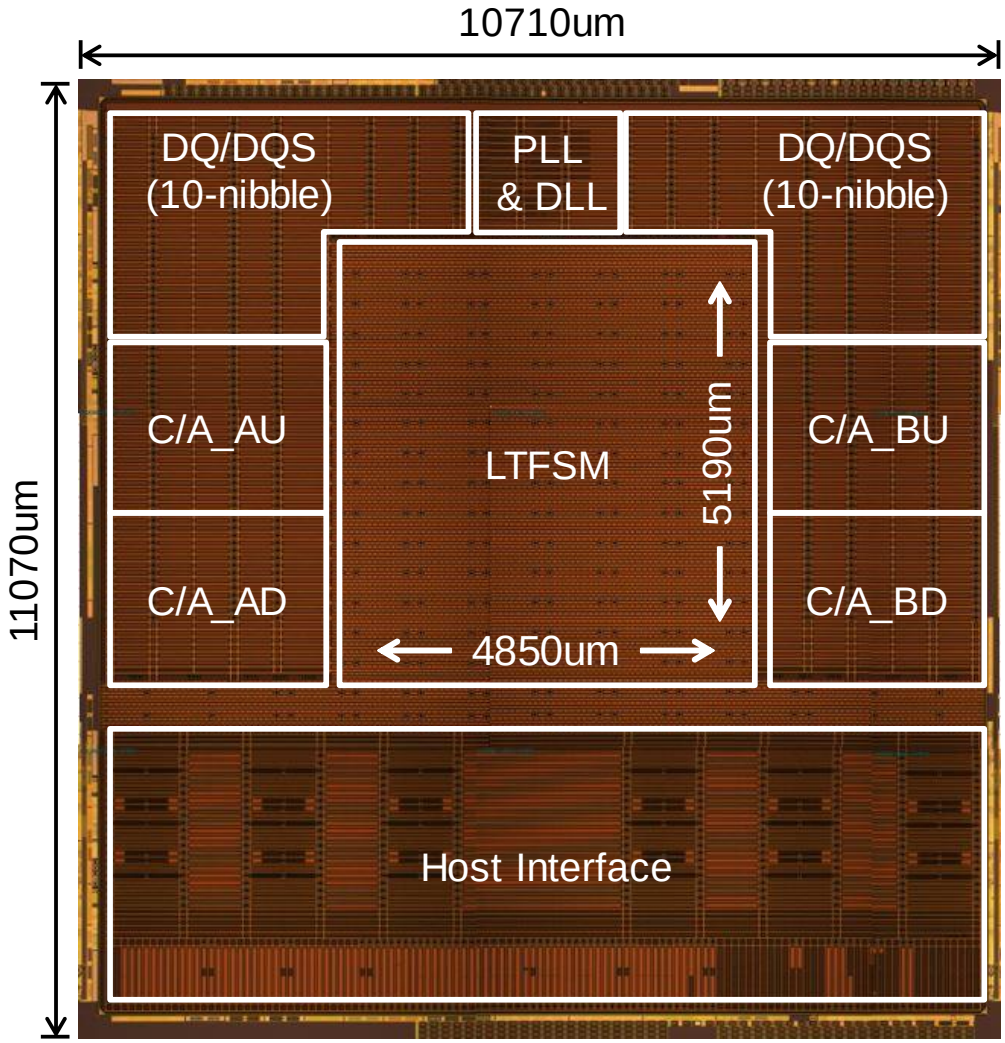


Figure 6.1.1 Die micrograph of the MDS controller chip.

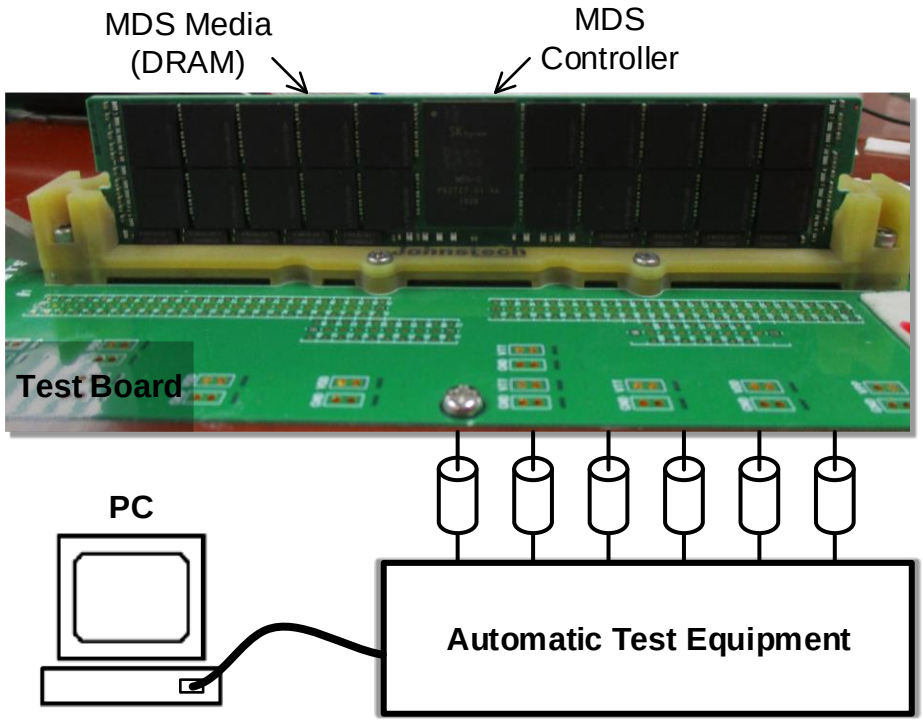


Figure 6.1.2 Measurement setup

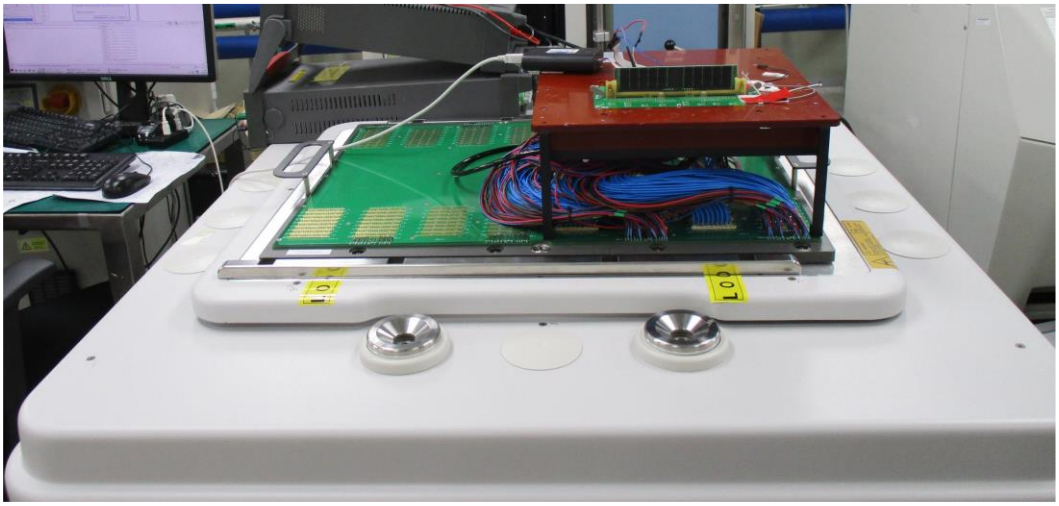
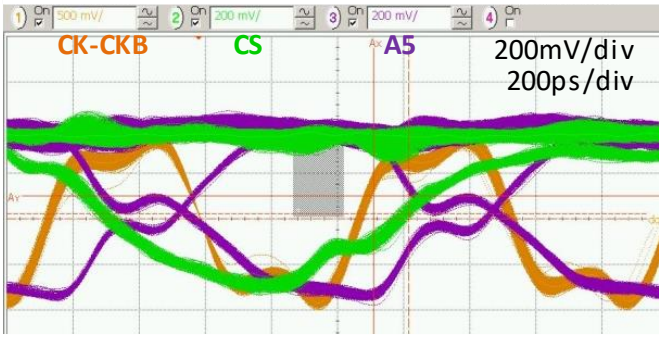


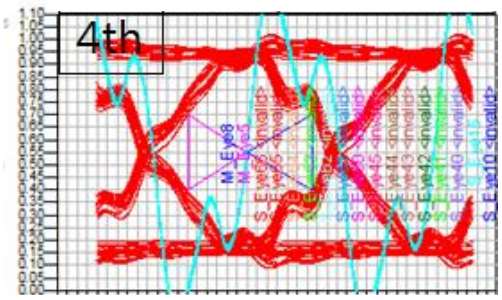
Figure 6.1.3 Photograph of the measurement setup.

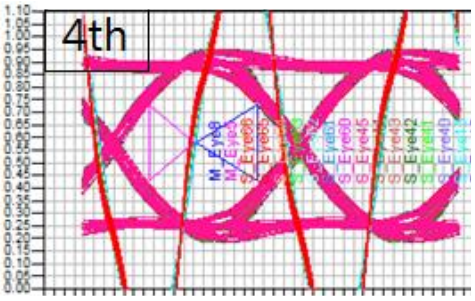
The C/A_AU group of the controller drives the upper right 10 DRAM packages, five are on the front, and five are on the back of the DIMM. The C/A_AD group drives the lower right 10 DRAM packages. The C/A_BU and C/A_BD groups drive the upper left and lower left 10 DRAM packages, respectively. The DIMM is mounted on the test board, and the control signals and data signals are connected to the test equipment below the test board. The automatic test equipment (ATE) accesses the MDS controller through SMBus, exchanges data with the host interface of the controller, and transmits the measurement results to the PC. When power is on, the controller performs initialization sequences including media power-up, MRS initialization, ZQ calibration on both the DRAM and controller side, CA training, read preamble training, read training, write leveling, and write training.

Figure 6.1.4 (a) shows measured waveforms of CK-CKB, CS, and one of C/A signals (A5). During CA training, the CS signal is trained to have 75% setup time and 25% hold time whereas the other C/A signals are trained to have both setup and hold time of 50%. As a result of the CA training, CS signal is faster than A5 signal. The simulation results of Figure 6.1.4 (b) and (c) show that the distorted waveform of Figure 6.1.4 (a), which is measured at DRAM package ball is not the real input to the receiver at the DRAM, and the real input signal has less reflection.



(a)





(b) (c)

Figure 6.1.4 (a) Measured waveforms of CK-CKB, CS, and CA, and simulation of the CA signal at (b) the package ball and (c) the receiver buffer.

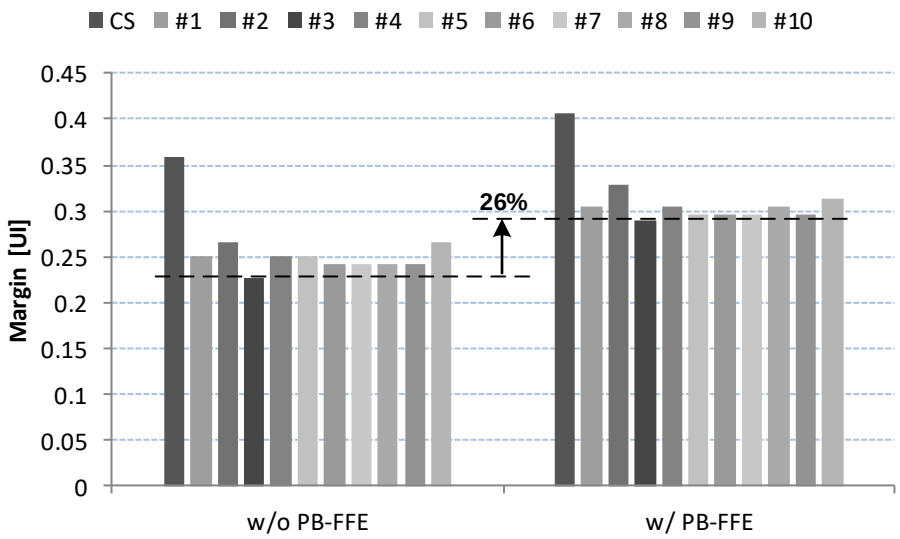


Figure 6.1.5 Measured C/A margin with and without PB-FFE

Figure 6.1.5 shows the timing margin on the C/A lines, measured by the DCDL in the C/A transmitter with a resolution of T_{CK}/128. Because the CS channel has less DRAM load than the other C/A channels, the timing margin of the CS channel is larger than the others. The minimum timing margin without FFE is 0.23UI and the use of PB-FFE increases this margin to 0.29UI, at 1067Mbps.

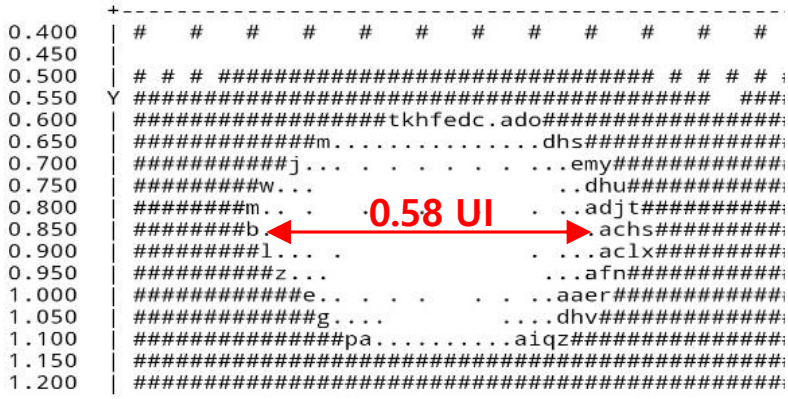


Figure 6.1.6 Read margin measured by ATE.

Figure 6.1.6 shows a shmoo plot of the read operation with data driven by ATE, and the read margin of the controller is 0.58UI. However, the read margin measured on the MDS DIMM is reduced to 0.53UI due to the duty-cycle error and output jitter of the DRAM. The internal reference voltage step of the controller PHY is 4.3mV and the measured voltage margin is 49 steps or 211mV as shown in Figure 6.1.7.

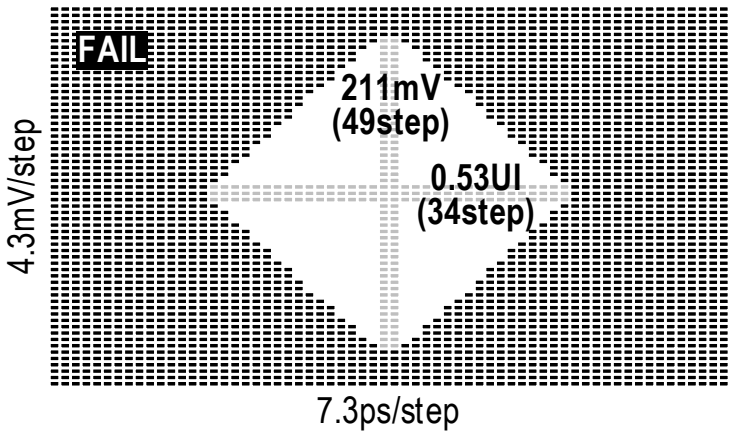


Figure 6.1.7 Measured read timing and voltage margin on the DIMM.

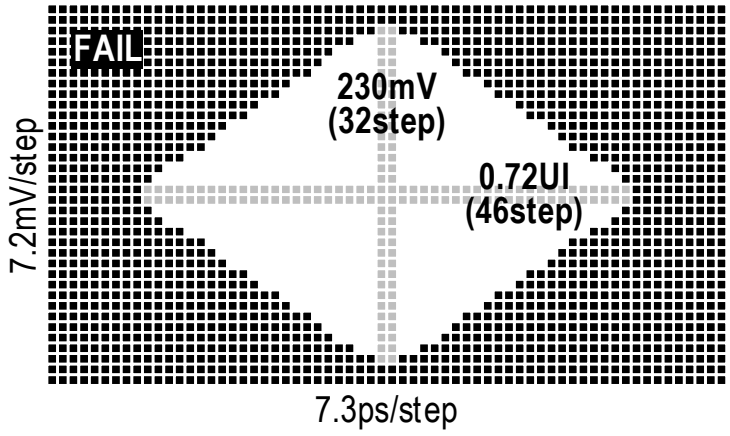
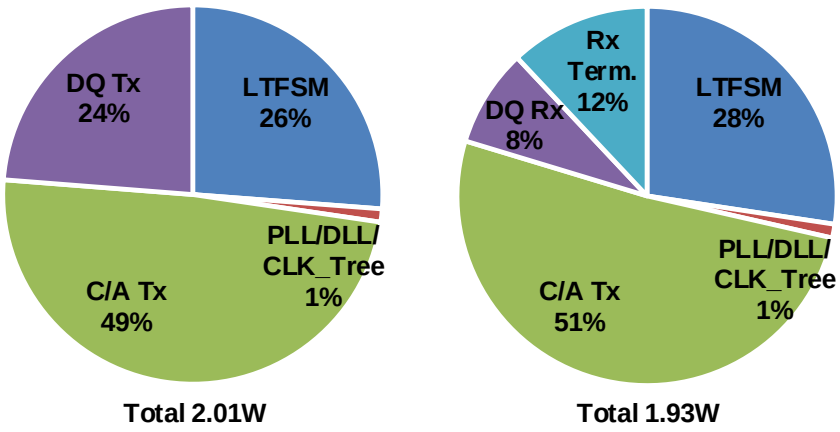


Figure 6.1.8 Measured write timing and voltage margin on the DIMM.

The write timing margin is measured by varying the DQ timing with a step size of T_{CK}/128 or 1/64 UI, and read back the previous written data. The write voltage margin is measured by varying the reference voltage of the DRAM with a step size of 0.65% of the VDDQ as the DDR4 standard. The measured write timing and voltage margins are 0.72UI and 230mV as shown in Figure 6.1.8.



(a) (b)

Figure 6.1.9 Power breakdown of (a) write and (b) read operation.

Figure 6.1.9 shows the power breakdown of the burst write and read operation, respectively. The total power consumption of the controller during burst write and read operation is 1.97W, which satisfying the requirement for an MDS DIMM [1.1.6]. Table 6.1.1 compares this controller PHY to other DRAM interfaces.

Table 6.1.1 PERFORMANCE SUMMARY AND COMPARISON WITH OTHER DRAM INTERFACES

		This Work	[6.1.1] ISSCC'15	[6.1.2] TCAS2'18	[6.1.3] ESSCIRC'17
Process		55nm CMOS	25nm CMOS	65nm CMOS	14nm SOI CMOS
Supply		1.1V	1.2V	-	1.2V
Data-rate		2133Mbps	2133Mbps	4266Mbps	2400Mbps
Interface		DDR4-MDS	DDR4-3DS	LPDDR4	DDR4
DRAM Stack		8-H Wire-bond	4-H TSV	N/A	N/A
C/A topology		Fly-by 80-load	Point-to-Point	Point-to-Point	N/A
Number of I/O		x80	-	x16	X4
Equalizing features		PB-FFE (@C/A Tx)	-	AF-CTLE (@DQ Rx)	AC boosting (@DQ Tx)
C/A margin	w/o PB-FFE	0.23UI (@ 1067Mbps)	N/A	N/A	N/A
	w/ PB-FFE	0.29UI (@ 1067Mbps)	N/A	N/A	N/A
Write (DQ Tx) margin		0.72UI / 230mV	0.58UI	0.36UI / 148mV	0.52UI / 155mV
Read (DQ Rx) margin		0.53UI / 211mV	-	0.47UI / 80mV	
Power efficiency (mW/Gb/s/pin)	C/A Tx	7.0	N/A	-	N/A
	DQ Tx	2.8	-	5.68	2.8
	DQ Rx	2.3	-	1.86	-

6.2 PROTOTYPE QEC

The prototype quarter-rate transmitter with a 4-tap feed-forward equalizer (FFE), shown in Figure 6.2.1, was fabricated in a 65nm CMOS process to assess the proposed QEC. The clock buffer (CKBUF) receives an external differential clock input (CLKP/CLKN), and the four-phase generator generates the quarter-rate clock signals from the received clock signals. A PRBS generator is included on the chip for test purposes. It produces 32-bit parallel data, which is serialized to 4-bit sequences by a 32:4 SER. The 4-bit output of the SER is re-timed and rearranged into 4-groups of 4-bit sequences by an FFE retimer. When the TAP_SIGN signal for each FFE tap is asserted, the tap coefficient becomes negative and the output of the corresponding FFE retimer is inverted. The output stage has 40 segments, each of which is composed of a 4:1 SER and a voltage-mode source-series terminated (SST) driver. The FFE multiplexer (FFE MUX) allocates the selected FFE tap data to each of the 40 output segments under the control of the TAP_SEL1 and TAP_SEL2 signals.

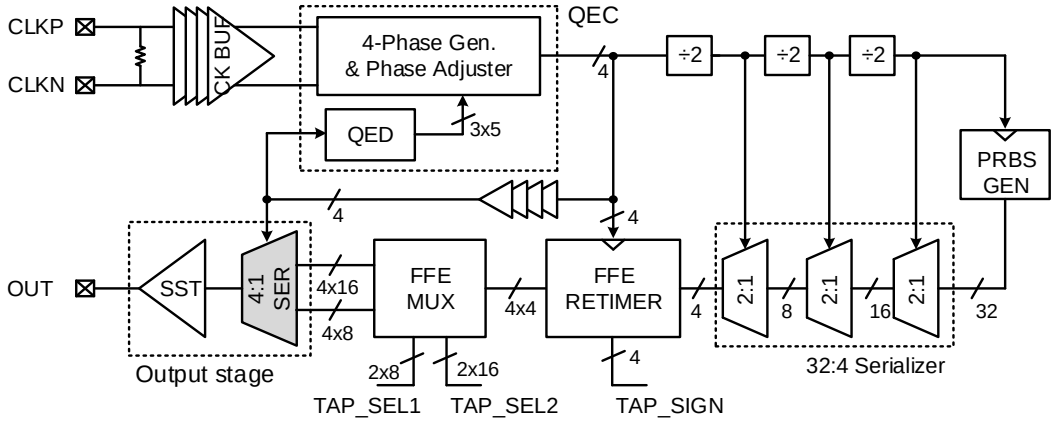


Figure 6.2.1 Block diagram of the prototype transmitter.

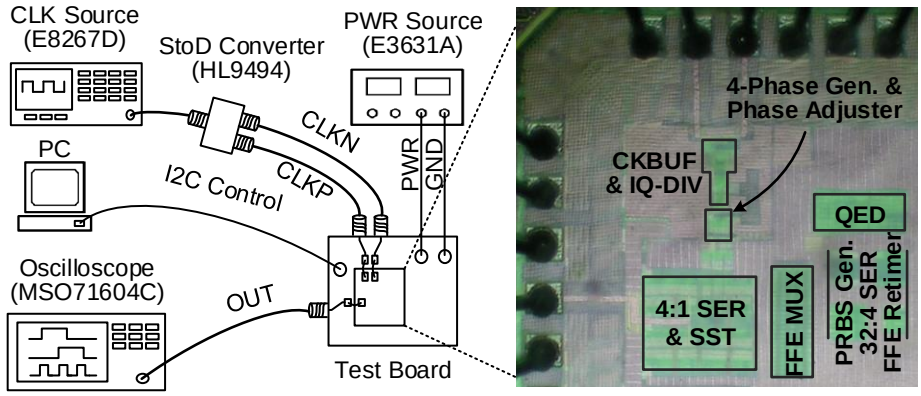
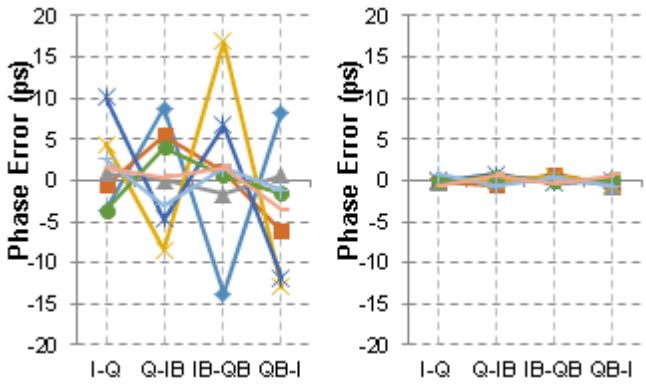


Figure 6.2.2 Measurement setup and die micrograph.

Figure 6.2.2 shows the measurement setup and a die micrograph. Half-rate differential input clock signals are applied through an external single-to-differential converter, from which four-phase quarter-rate clock signals are generated inside the chip. To evaluate the performance of the QEC, phase errors were introduced into the quarter-rate clock signals

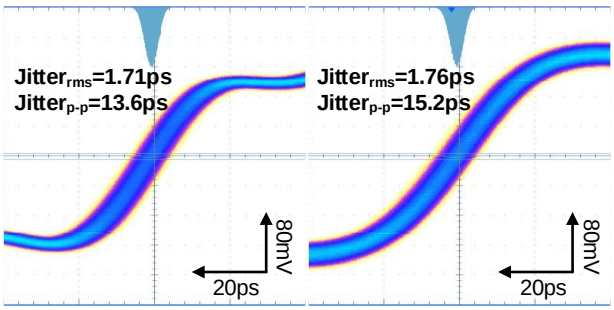
during their generation.

Figure 6.2.3 shows the measured output phase error for eight cases of quadrature clock signals with uncorrected phase errors varying from -13.7ps to 16.9ps. The QEC corrects these phase errors to within 0.8ps.



(a) (b)

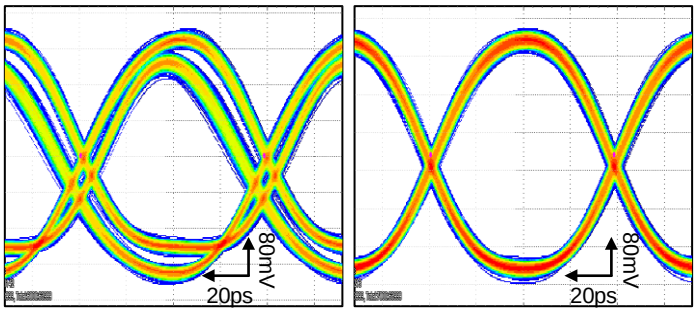
Figure 6.2.3 Measurements of (a) uncorrected and (b) corrected output phase error.



(a) (b)

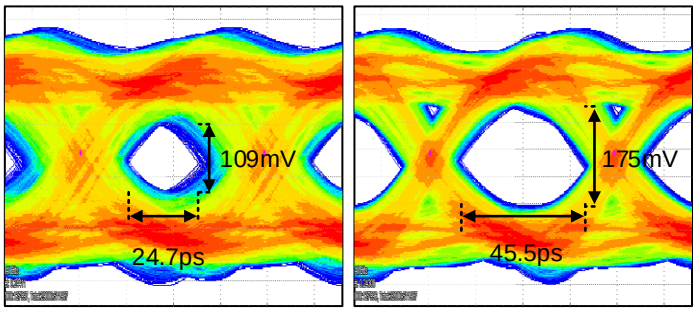
Figure 6.2.4 Measured (a) input and (b) output clock jitter at 6.4 GHz.

As shown in Figure 6.2.4, the measured jitter of the half-rate 6.4GHz input clock signal is 1.71ps$_{rms}$ or 13.6ps$_{p-p}$. The measured output jitter of the 6.4GHz clock pattern is 1.76ps$_{rms}$ or 15.2ps$_{p-p}$, and the jitters were increased slightly due to the residual phase errors between ICK and IBCK. Figure 6.2.5 and Figure 6.2.6 show measured eye diagrams of the output signals of the prototype TX with clock pattern and PRBS pattern, respectively, both at 12.8Gbps.



(a) (b)

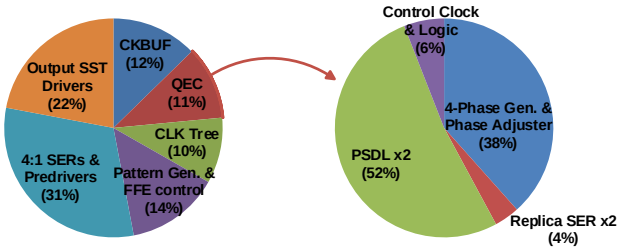
Figure 6.2.5 Measured TX output eye diagram of clock pattern at 12.8 Gbps (a) without and (b) with quadrature error correction.



(a) (b)

Figure 6.2.6 Measured TX output eye diagram of PRBS pattern at 12.8 Gbps (a) without and (b) with quadrature error correction.

The eye diagram in Figure 6.2.5 (a) shows two distinctive waveforms due to the quadrature phase skew, and the waveform of quadrature errors are corrected is shown in Figure 6.2.5 (b). In Figure 6.2.6, 512k cycles of eyes were accumulated through the FR4 PCB channel with insertion loss of 7.8dB, and we can see that the width and height of the output eye are increased by 84% and 61% respectively after the correction of quadrature error. The power breakdowns of the prototype Tx and the QEC are shown in Figure 6.2.7 and the total power consumption of the QEC is 6.7mW at 3.2GHz.



(a) (b)

Figure 6.2.7 Power breakdown of (a) the prototype Tx and (b) the QEC

Table 6.2.1 compares the performance of other recent designs with that of our QEC, which has the lowest residual phase error with highest operating frequency. Another significant feature which distinguishes our design from the others listed in Table 6.2.1 is that it corrects errors in the serializer of the output stage.

Table 6.2.1 PERFORMANCE SUMMARY AND COMPARISON WITH OTHER QEC DESIGNS

	[1.1.18]	[1.1.17]	[1.1.20]	[6.2.1]	This Work
Process	65nm	130nm	65nm	90nm	65nm
Supply	1.0V	1.2V	1.2V	1.0V	1.0V
Frequency	1.25GHz	2GHz	1GHz	2.7GHz	3.2GHz
Error Detector	Delay Line & BBPD	Clock Doubler & Integrator	TDC	TDC	SER & PSDL
Correction Range	10.4ps	63ps	40ps	-	16.9ps
Residual Phase Error	1.1ps	4.2ps	5ps	3.71ps	0.8ps
Jitter(rms)	2.53ps	-	2.98ps	-	1.76ps
Jitter(p-p)	15ps	-	19.4ps	-	15.2ps
Area	$0.01mm^2$	$0.01mm^2$	$0.09mm^2$	$0.09mm^2$	$0.01mm^2$
FoM (mW/GHz)	1.82	3.24	2.6	18.3	2.1

CHAPTER 7

CONCLUSION

A controller PHY for a high-capacity DRAM solution was presented. It was mounted on an MDS DIMM and interfaced with 40 DRAM packages. This controller supports all the training sequences specified in the DDR4 standard, including link trainings for C/A, read, and write operation. A glitch-free DCDL reduces training time. To improve the ISI due to the heavy load of many DRAM chips on a C/A channel, PB-FFE is used in the C/A transmitter. Adjustment of the input parasitic resistance and capacitance of each DRAM chip also reduces ISI and reflections. The controller was fabricated in 55nm CMOS and occupied 118.6mm^2. Its C/A timing margin at 1067Mbps is improved from 0.23UI to 0.29UI by applying PB-FFE. At 2133Mbps, the measured read timing and voltage margins are 0.53UI and 211mV after read training, and the write margins are 0.72UI and 230mV after write training. The power consumption of the controller during burst write and read operation is 1.97W, which satisfies the requirement of MDS DIMM.

For the quarter-rate transmitters, which are adopted to reduce the output latency in DQ, a quadrature error corrector (QEC) is also presented. Quadrature phase errors can occur during the generation and distribution of four-phase clock signals. In addition to

reducing the residual quadrature phase error, signal integrity can also be improved by compensating for the distortion introduced by the output stage. We have presented a QEC that corrects quadrature-phase errors, including distortions in the serializer of the output stage. Pulse-shrinking delay lines are used to detect the phase errors, and a fine delay unit improves accuracy. A prototype quarter-rate transmitter, including this QEC, was fabricated to another chip in 65nm CMOS. At a data-rate of 12.8Gbps, it reduces phase errors in the transmitter to a residual error of 0.8ps. The output eye width and height are improved by 84% and 61%, respectively.

BIBLIOGRAPHY

[1.1.1] A. M. Ionescu, "Energy efficient computing and sensing in the Zettabyte era: From silicon to the cloud," in *IEEE Int. Electron Devices Meeting (IEDM)*, San Francisco, CA, 2017, pp. 1.2.1-1.2.8.

[1.1.2] Reinsel, David, John Gantz, and John Rydning. "Data age 2025: the digitization of the world from edge to core." Seagate, https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf (2018).

[1.1.3] DDR4 SDRAM UDIMM Design Specification, JEDEC standard No. 21C. Page 4.20.26-1, Aug. 2019.

[1.1.4] DDR4 SDRAM Registered DIMM Design Specification, JEDEC standard No. 21C. Page 4.20.28-1, May. 2019.

[1.1.5] DDR4 SDRAM Load Reduced DIMM Design Specification, JEDEC standard No. 21C. Page 4.20.27-1, Aug. 2015.

[1.1.6] S. Lee *et al.*, "A 512GB 1.1V managed DRAM solution with 16GB ODP and media controller," in *IEEE Int. Solid-State Circuits Conf. (ISSCC)*, San Francisco, CA, 2019, pp. 384-386.

[1.1.7] W. Shin *et al.*, "A 4.8Gb/s impedance-matched bidirectional multi-drop transceiver for high-capacity memory interface," in *IEEE Int. Solid-State Circuits Conf. (ISSCC)*, San Francisco, CA, 2011, pp. 494-496.

[1.1.8] W. Lee *et al.*, "Parallel branching of two 2-DIMM sections with write-direction impedance matching for an 8-Drop 6.4-Gb/s SDRAM interface," *IEEE Trans. on Components, Packaging and Manufacturing Technology*, vol. 9, no. 2, pp. 336-342, Feb. 2019.

[1.1.9] J. Seo *et al.*, "A 7.8-Gb/s 2.9-pJ/b single-ended receiver with 20-tap DFE for highly reflective channels," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 3, pp. 818-822, Mar. 2020.

[1.1.10] H. Chi *et al.*, "A single-loop SS-LMS algorithm with single-ended integrating

DFE receiver for multi-drop DRAM interface," *IEEE J. Solid-State Circuits*, vol. 46, no. 9, pp. 2053-2063, Sep. 2011.

[1.1.11] S. -J. Bae *et al.*, "A 2Gb/s 2-tap DFE receiver for multi-drop single-ended signaling systems with reduced noise," in *IEEE Int. Solid-State Circuits Conf. (ISSCC)*, San Francisco, CA, 2004, pp. 244-525.

[1.1.12] J. Ren *et al.*, "Precursor ISI reduction in high-speed I/O," in *IEEE Symposium on VLSI Circuits*, Kyoto, Japan, 2007, pp. 134-135.

[1.1.13] Y. Wang and W. Gai, "Power-efficient pre-emphasis method for transmitters with LVDS drivers," *Electronics Letters*, vol. 50, no. 24, pp. 1811-1813, Nov. 2014.

[1.1.14] C. Chien, C. Wang, C. Lin, T. Hsieh, Y. Chu and J. Guo, "A Low Latency Memory Controller for Video Coding Systems," in *IEEE International Conference on Multimedia and Expo*, Beijing, China, 2007, pp. 1211-1214.

[1.1.15] H. Partovi *et al*., "Single-ended transceiver design techniques for 5.33Gb/s graphics applications," in *IEEE Int. Solid-State Circuits Conf. (ISSCC)*, San Francisco, CA, USA, 2009, pp. 136-137,137a.

[1.1.16] DDR4 SDRAM, JEDEC standard JESD79-4C, Jan. 2020.

[1.1.17] I. Raja, V. Khatri, Z. Zahir and G. Banerjee, "A 0.1–2-GHz quadrature correction loop for digital multiphase clock generation circuits in 130-nm CMOS," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 3, pp. 1044-1053, Mar. 2017.

[1.1.18] Y. Kim, K. Song, D. Kim and S. Cho, "A 2.3-mW 0.01-mm^2 1.25-GHz quadrature signal corrector with 1.1-ps error for mobile DRAM interface in 65-nm CMOS," *IEEE Trans. on Circuits and Systems II: Express Briefs*, vol. 64, no. 4, pp. 397-401, Apr. 2017.

[1.1.19] J. Kim *et al*., "A 16-to-40Gb/s quarter-rate NRZ/PAM4 dual-mode transmitter in 14nm CMOS," in *IEEE Int. Solid-State Circuits Conf. (ISSCC)*, San Francisco, CA, USA, 2015, pp. 1-3.

[1.1.20] J. Cho and Y. Min, “An all-digital duty-cycle and phase-skew correction circuit for QDR DRAMs,” *IEICE Electronics Express*, vol. 15, no. 9, pp. 1-6, May. 2018.

[2.1.1] J. Liu *et al.*, "RAIDR: Retention-aware intelligent DRAM refresh," in *International Symposium on Computer Architecture*, 2012, pp. 1-12.

[2.1.2] 3D Stacked DDR4 SDRAM, JEDEC standard JESD79-4-1A, Mar. 2020.

[2.2.1] DDR4 Registering Clock Driver (DDR4RCD02), JEDEC standard JESD82-31A, Aug. 2019.

[3.2.1] H. Chuang *et al.*, "Signal/power integrity modeling of high-speed memory modules using chip-package-board coanalysis," *IEEE Trans. on Electromagnetic Compatibility*, vol. 52, no. 2, pp. 381-391, May 2010.

[4.1.1] J. -T. Kwak *et al.*, "A low cost high performance register-controlled digital DLL for 1 Gbps x32 DDR SDRAM," in *IEEE Symposium on VLSI Circuits*. *Digest of Technical Papers*, Kyoto, Japan, 2003, pp. 283-284.

[4.1.2] D. De Caro, "Glitch-free NAND-based digitally controlled delay-lines," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 21, no. 1, pp. 55-66, Jan. 2013.

[4.3.1] G. D. Robinson, "NAND trees accurately diagnose board-level pin faults," *Proceedings., International Test Conference*, Washington, DC, USA, 1994, pp. 811-816, doi: 10.1109/TEST.1994.528028.

[5.2.1] J. Yin, P. Mak, F. Maloberti and R. P. Martins, "A Time-Interleaved Ring-VCO with Reduced $1/f^3$ Phase Noise Corner, Extended Tuning Range and Inherent Divided Output," *IEEE J. Solid-State Circuits*, vol. 51, no. 12, pp. 2979-2991, Dec. 2016.

[6.1.1] W. Yun *et al.*, "A digital DLL with hybrid DCC using 2-step duty error extraction and 180° phase aligner for 2.67Gb/S/pin 16Gb 4-H stack DDR4 SDRAM with TSVs," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Digest of Technical Papers*, San Francisco, CA, 2015, pp. 1-3.

[6.1.2] M. Kim *et al*., "A 4266 Mb/s/pin LPDDR4 interface with an asynchronous feedback CTLE and an adaptive 3-step eye detection algorithm for memory controller," *IEEE Trans. Circuits and Systems II: Express Briefs*, vol. 65, no. 12, pp. 1894-1898, Dec. 2018.

[6.1.3] M. Kossel *et al*., "DDR4 transmitter with AC-boost equalization and wide-band voltage regulators for thin-oxide protection in 14-nm SOI CMOS technology," in *IEEE European Solid State Circuits Conf. (ESSCIRC)*, Leuven, Belgium, 2017, pp. 115-118.

[6.2.1] C. Tsai, Y. Chiu, Y. Tu and K. Cheng, "A wide-range all-digital delay-locked loop for double data rate synchronous dynamic random access memory application," in *IEEE Int. Symposium on Circuits and Systems (ISCAS)*, Florence, Italy, 2018, pp. 1-4.

한글 초록

본 연구에서 용량을 최대화하면서도 리프레시 전력을 최소화할 수 있는 새로운 메모리 구조인 관리형 DRAM 솔루션을 위한 컨트롤러 PHY를 제시하였다. 이와 같은 고용량 DRAM 인터페이스에서는 많은 DRAM 칩이 명령 / 주소 (C/A) 채널을 공유하고 있어서 심볼 간 간섭이 발생한다. 본 연구에서는 이러한 C/A 채널에서의 심볼 간 간섭을 줄이기 위해 펄스 기반 피드 포워드 이퀄라이저 (PB-FFE)를 채택하였다. 또한 본 연구의 컨트롤러 PHY는 DDR4 표준에 지정된 모든 트레이닝 시퀀스를 지원한다. 링크 트레이닝을 효율적으로 수행하고 트레이닝 시간을 줄이기 위해 글리치가 발생하지 않는 디지털 제어 지연 라인 (DCDL)을 채택하였다.

컨트롤러 PHY의 DQ 송신기는 출력 대기 시간을 줄이기 위해 쿼터 레이트 구조를 채택하였다. 쿼터 레이트 송신기의 경우에는 직교 클럭 간 위상 오류가 출력 신호의 무결성에 영향을 주게 된다. 이러한 영향을 최소화하기 위해 본 연구에서는 출력 단의 4 : 1 직렬 변환기의 두 복제본을 사용하여 클록 신호 위상 오류를 수정하는 QEC (Quadrature Error Corrector)를 제안하였다. 복제된 2개의 직렬 변환기의 출력을 비교하고 균등화하기 위해 펄스 수축 지연 라인이 사용되었다.

컨트롤러 PHY는 55nm CMOS 공정으로 제조되었다. PB-FFE는

1067Mbps에서 C/A 채널 타이밍 마진을 0.23UI에서 0.29UI로 증가시킨다. 읽기 트레이닝 후 읽기 타이밍 및 전압 마진은 2133Mbps에서 0.53UI 및 211mV이고, 쓰기 트레이닝 후 쓰기 마진은 0.72UI 및 230mV이다.

QEC의 효과를 검증하기 위해 QEC를 포함한 프로토 타입 쿼터 레이트 송신기를 65nm CMOS의 다른 칩으로 제작하였다. QEC를 적용한 실험 결과, 송신기의 출력 위상 오류가 0.8ps의 잔류 오류로 감소하고, 출력 데이터 눈의 폭과 높이가 12.8Gbps의 데이터 속도에서 각각 84 %와 61 % 개선되었음을 보여준다.

주요어 : DRAM 인터페이스, 피드 포워드 이퀄라이저, 메모리 컨트롤러, 펄스 수축 지연 라인, 복제 직렬 변환기, 직교 오차 보정기

학 번 : 2014-21669