0.11-2.5 GHz All-digital DLL for Mobile Memory Interface with Phase Sampling Window Adaptation to Reduce Jitter Accumulation

Joo-Hyung Chae¹, Mino Kim¹, Gi-Moon Hong², Jihwan Park¹, Hyeongjun Ko², Woo-Yeol Shin, Hankyu Chi², Deog-Kyoon Jeong¹, and Suhwan Kim¹

Abstract—An all-digital delay-locked loop (DLL) for a mobile memory interface, which runs at 0.11-2.5 GHz with a phase-shift capability of 180°, has two internal DLLs: a global DLL which uses a time-to-digital converter to assist fast locking, and shuts down after locking to save power; and a local DLL which uses a phase detector with an adaptive phase sampling window (WPD) to reduce jitter accumulation. The WPD in the local DLL adjusts the width of its sampling window adaptively to control the loop bandwidth, thus reducing jitter induced by UP/DN dithering, input clock jitter, and supply/ground noise. Implemented in a 65 nm CMOS process, the DLL operates over 0.11-2.5 GHz. It locks within 6 clock cycles at 0.11 GHz, and within 17 clock cycles at 2.5 GHz. At 2.5 GHz, the integrated jitter is 954fsrms, and the long-term jitter is 2.33psrms/23.10pp. The ratio of the RMS jitter at the output to that at the input is about 1.17 at 2.5 GHz, when the sampling window of the WPD is being adjusted adaptively. The DLL consumes 1.77 mW/GHz and occupies 0.075 mm².

Index Terms—Delay-locked loop (DLL), phase detector, sampling window, fast locking, jitter accumulation, mobile memory, wide frequency range

I. INTRODUCTION

The use of mobile devices such as smartphones, tablet PCs, and high-end laptop computers is growing rapidly, and these devices now offer video processing and other computing-intensive capabilities on high-definition displays [1]. So the demand for high-bandwidth mobile memory is also increasing. However, the users of mobile devices also want a long battery life, so a mobile memory and its controller need to have a good power efficiency [2].

A multiphase clock (CK) generator and a phase-shift block is required to support various margin testing and training operations for a mobile memory interface. A phase-locked loop (PLL), which is widely used in the CK generator, can easily generate multiphase CK signals; but when the number of phases is large, the CK distribution tree that takes the CK signals to each path becomes complicated, consumes a lot of power, and occupies a large area. This CK distribution tree can be simplified by introducing a delay-locked loop (DLL) into each path, saving power and area.

To be suitable for the mobile memory interface, the DLL must be able to operate over a wide frequency range and consume low power. As the data-rate increases, the process technology shrinks, and the supply voltage scales down, the design of the mobile memory interface poses a formidable challenge, which include coping with increased process, voltage, and temperature (PVT) variations, circuit noise, and supply/ground noise. In order to improve a timing margin for data sampling,
DLL in the mobile memory interface has to offer good jitter performance in the presence of such uncertainties. A conventional DLL with a bang-bang phase detector (PD) is unlikely to be suitable for the mobile memory interface because the jitter performance is degraded by the UP/DN dithering and high sensitivity to supply/ground noise. In recent designs [3-6], jitter in the DLL has been reduced using various techniques, all of which have some problems: an analog window PD [3] has a narrow frequency range and is sensitive to PVT variations, the locking range of an ILO [4] only offers a narrow operating frequency range, a DLL which uses a voltage regulator [5] relies on analog circuits with high process sensitivity, and a dither-free PD with a region accumulator [6] requires a lot of chip area and too sensitive to PVT variations.

To overcome these various drawbacks, we recently proposed a 180° phase-shift digital DLL with a window phase detector [7]. In this present paper, we introduce additional features to this DLL, and make further contributions to the design of a robust mobile memory interface, as follows: (1) a streamlined architecture of the mobile memory interface, (2) detailed circuit implementation and operation of the DLL as a low-power circuit, (3) extensive measurement and analysis of a PD with an adaptively changing sampling window to demonstrate the advantage of our design, and (4) comparison with other recent designs of the DLL which are also suitable for the mobile memory interface.

II. MOBILE MEMORY INTERFACE ARCHITECTURE

Fig. 1 shows the architecture of a mobile memory interface. Its digital block manages training operations, memory boot-up, data (DQ), data strobe (DQS), command address (CA), and control signals. A PLL generates the CK signal, and this signal is distributed to a global DLL located near the PLL, as well as to each CK, CA, DQS, and DQ path through the CK distribution tree.

A phase-shift block, including a local DLL and a phase interpolator (PI), is required in the CK and CA paths to adjust the CK phase to support CA training operation. The CK and CA signals are transmitted to the mobile memory through a low-voltage swing terminated logic (LVSTL) driver.

A DQS generating unit in the DQS WRITE path generates the DQS signal, which has a burst-mode pattern. The DQS signal is shifted by the phase-shift block to support WRITE training, and transmitted to the mobile memory by the LVSTL driver. In the DQS READ path, the DQS signal is received from the mobile memory, and this signal is transmitted to each DQ path and shifted for READ training.

Each DQ WRITE path is composed of a DQ generating unit, a replica local DLL, a replica PI, a 16:1 serializer, and the LVSTL driver. The replica local DLL and PI are needed to compensate for the difference of
propagation delay between the DQ and DQS path. Each DQ READ path consists of a continuous-time linear equalizer (EQ), the replica local DLL, the replica PI, and a 1:16 deserializer.

In the phase-shift block, the local DLL generates multiphase CK signals. The PI receives these signals and shifts their phases within 1 UI of the signal, with a resolution of 64 steps, to search for the center or the edge of the signal.

III. ALL-DIGITAL DELAY-LOCKED LOOP ARCHITECTURE AND IMPLEMENTATION

The overall architecture of the proposed all-digital DLL for the mobile memory interface with a 2.5 GHz CK frequency is shown in Fig. 2. It consists of an input duty cycle corrector (DCC), a global DLL, local DLLs, output DCCs, and a PI. The global DLL consists of a coarse and a fine time-to-digital converter (TDC), each based on a Vernier delay line architecture, a digital block to correct the bubble in the delay code transmitted from both TDCs, and a delay line. A local DLL has a phase detector with an adaptive sampling window (WPD) which uses a replica fine delay line (RFDL) and a lock detector, a digital loop filter (DLF), and the delay line of the same type as that used in the global DLL.

I. Implementation of the Global and Local DLLs

Several DLL architectures which are designed to lock quickly have recently been introduced: TDC-based DLLs [4, 8, 9], SAR-based DLLs [10, 11], as well as other designs [12, 13]; and they all lock within a few CK cycles. The TDC-based DLL has the advantages that the lock time is short, and there is no need for further operations and power consumption after locking is complete. We use a TDC-based DLL as a global DLL because it makes the lock time of the DLL shorter.

The number of delay units in the delay line is changed depending on the operating frequency. When the CK frequency is 2.5 GHz, two chains of a delay unit (DU) consisting of a coarse delay unit (CDU) based on NAND gates, and a fine delay unit (FDU) based on MOS capacitors are used. These delay units generate the three multiphase CK signals, CK₀, CK₁₀₀ and CK₁₈₀. At a CK frequency of 0.11 GHz, eight chains of delay unit are used, and it generates nine multiphase CK signals, CK₀, CK₂₂.₅, CK₄₅, CK₆₇.₅, CK₉₀, CK₁₁₂.₅, CK₁₃₅, CK₁₅₇.₅, and CK₁₈₀. Because the various operations in the mobile memory interface are based on 1 UI of the DQ signal, which is half the CK period, it is sufficient for the DLL to have a 180° locking range. Using a 180° phase-shift DLL halves the range of delay that needs to be provided by the delay line and the TDC, reducing both power and area consumption. In order to prevent harmonic locking, the delay line code of the global DLL should be initialized to its minimum value whenever the global DLL starts operating, or the operating frequency of the DLL is changed.

The coarse TDC and the fine TDC have the same architecture, which is shown in Fig. 3(a). These TDCs are composed of a MUX, a dummy DU, a replica DU, and a D-flipflop (DFF). The dummy DUs have a delay of tₙ₉₉, and the replica DUs have a delay of t₁₈. Fig. 3(b) shows the block diagram and the delay of the DUs in the coarse and fine TDC. The delay of the dummy DU and the replica DU in the coarse TDC are 2tₙ₉₉ and 4tₙ₉₉, and the delay of the dummy DU and the replica DU in
the fine TDC are \(2t_{\text{inv}}\) and \(2t_{\text{inv}} + t_{\text{up}}\). Fig. 3(c) shows the timing diagram of the TDC. The coarse TDC starts operating after the chip power is on and CK signal is entered. When the STOP signal catches up with the START signal in the coarse TDC, the coarse delay code is transmitted to the digital block. The coarse TDC Lock Flag is then transmitted to the fine TDC, which starts fine TDC operation. The fine TDC begins operations in the same way, and then issues a fine TDC Lock Flag. This Lock Flag, together with the coarse and fine delay codes, are transmitted to a DLF in the local DLL which is in each path. When these operations are complete, the TDC stops operating and all the circuits in the global DLL are powered down, mitigating the high power consumption associated with a TDC-based DLL. The locking operation of the global DLL can restart between mobile memory interface operations.

The delay code and the Lock Flag are received from the global DLL, and then the local DLL locks immediately. Subsequently, the local DLL tracks the phase of the input CK signal contiguously so as to compensate for the mismatches between the global DLL and the local DLLs induced by local voltage and temperature variations. To achieve a 180° locking and continuous phase tracking, the WPD compares the falling edge of the input CK of the DLL with the rising edge of the output CK of the DLL. To prevent false locking, this delay line in the local DLL must have a range as follows:

\[
0.5 \times T_{\text{CK}} < T_{\text{DL, min}} < T_{\text{CK}} \\
T_{\text{CK}} < T_{\text{DL, max}} < 1.5 \times T_{\text{CK}}
\]

\[
\text{Max}(T_{\text{DL, min}}, 2 \times T_{\text{DL, max}}) < T_{\text{CK}} < \text{Min}(2 \times T_{\text{DL, min}}, T_{\text{DL, max}})
\]

where \(T_{\text{CK}}\) is the period of the CK, \(T_{\text{DL, min}}\) is the minimum range, and \(T_{\text{DL, max}}\) is the maximum range of the delay line. To achieve the range of delays needed for a wide frequency range, the number of DUs in the delay line of the local DLL is controlled depending on the operating frequency.

2. Implementation of the Phase Detector with an Adaptive Sampling Window in the Local DLL

Our WPD, shown in Fig. 4(a), consists of replica fine delay lines (RFDLs), DFFs, a lock detector to make a judgment on the 180° phase-shift lock, a MUX, and a frequency divider \((/N)\). The delay resolution of the RFDLs is the same as that of the FDL in the delay line of
the local DLL. The RFDLs in the WPD issue the three delayed CK phases, $d\text{CK}_{\text{IN}}+n\Delta t$, $d\text{CK}_{\text{DLL}}$, and $d\text{CK}_{\text{DLL}}+2n\Delta t$, where $n$ is the value of the delay code sent to the RFDLs, and $\Delta t$ is the resolution of the delay. This WPD can be in one of three states: UP, DN, and HOLD, determined by the value of PD<1:0>, as shown in Fig. 4(b).

Fig. 5 shows the timing diagram and operation of the WPD. The rising edges of $d\text{CK}_{\text{DLL}}$ and $d\text{CK}_{\text{DLL}}+2n\Delta t$ form a sampling window with a width of $2n\Delta t$. If the falling edge of $d\text{CK}_{\text{IN}}+n\Delta t$ is caught by this sampling window, then PD<1:0> becomes ‘10’ and the local DLL enters its 180° locking state. This is detected by the lock detector, which sets the Lock Flag to ‘1’. After the 180° locking state has been entered, the value of the delay code sent to the RFDLs is reduced to narrow the sampling window, and the WPD is operated by the $\text{CK}_{\text{IN}}/N$ and $\text{CK}_{\text{DLL}}/N$ signals from the frequency divider, so as to reduce the loop bandwidth and dynamic power consumption of the WPD. If some combination of $\text{CK}_{\text{IN}}$ jitter, supply/ground noise, supply voltage droop due to large dynamic current consumption, and PVT variations causes the falling edge of $d\text{CK}_{\text{IN}}+n\Delta t$ to be escaped from the sampling window, then the lock detector determines that the lock is broken, and the value of the delay code sent to the RFDLs is increased to widen the sampling window. In this unlock state, the WPD is operated by $\text{CK}_{\text{IN}}$ and $\text{CK}_{\text{DLL}}$, and the local DLL re-enters the lock state quickly. Through this repetitive locking process, the local DLL using the WPD tracks the input CK phase of the DLL continuously. This method of adjusting the width of the sampling window controls the loop bandwidth, suppressing the UP/DN dithering phenomenon, increasing the probability of the lock state, and minimizing jitter in the local DLL.

When the frequency dividers of the $\text{CK}_{\text{IN}}$ and $\text{CK}_{\text{DLL}}$ are operated, the falling edge of the $\text{CK}_{\text{IN}}/N$ signal and the rising edge of the $\text{CK}_{\text{DLL}}/N$ signal can be misaligned due to the operation start timing of these frequency dividers. In order to avoid this phase misalignment issue in the $\text{CK}_{\text{IN}}$ and $\text{CK}_{\text{DLL}}$ frequency dividers, a Lock Flag signal is used as the reset signal in the frequency dividers. Fig. 6 shows the timing diagram and operation of these frequency dividers to resolve phase misalignment issue. When the Lock Flag signal goes high, the $\text{CK}_{\text{IN}}$ frequency divider is operated by $i\text{CK}_{\text{IN}}$, which is inverting signal of $\text{CK}_{\text{IN}}$, and the $\text{CK}_{\text{DLL}}$ frequency divider is operated by $\text{CK}_{\text{DLL}}$. Thus, the output signal of the $\text{CK}_{\text{IN}}$ frequency divider, $\text{CK}_{\text{IN}}/N$, is triggered by the falling edge of the $\text{CK}_{\text{IN}}$ signal and the output signal of the $\text{CK}_{\text{DLL}}$ frequency divider, $i\text{CK}_{\text{DLL}}$, is triggered by the rising edge of the $\text{CK}_{\text{DLL}}$ signal. Finally, the falling edge
of the CK_{IN}/N signal and the rising edge of the CK_{DLL}/N signal, which is the inverting signal of iCK_{DLL}, are aligned in both the lock and unlock state.

3. Analysis of the Phase Detector with an Adaptive Sampling Window

The UP/DN dithering phenomenon in the digital DLL with the BBPD enlarges the output CK jitter. A window-based PD, shown in Fig. 7, suppresses this UP/DN dithering phenomenon [3, 14, 15, 19]. It has 3 states: UP, DN, and HOLD. When the target phase edge is captured in the sampling window, the HOLD state is entered. In this state, the DLF stops operating and the delay line code is fixed. This suppresses the UP/DN dithering phenomenon, and the total jitter is now only the sum of the DLL input CK jitter, DLL circuit noise, and supply/ground noise.

Since the jitter performance is affected by the loop bandwidth of the DLL, it is important to optimize the loop bandwidth. If the window-based PD adapts the width of its sampling window according to the amount of the jitter and noise, it can control the loop bandwidth. When an analog window-based PD is used, the probabilities of the lock and unlock state are determined by the UP and DN current of the charge pump in the window control loop (WCL) [3]. As shown in Fig. 8(a), the UP current (I_{UP}) decreases the width of the window in the lock state, and the DN current (I_{DN}) increases the width of the window in the unlock state. Since the width of the window is fixed at the specific value in the steady state, the probabilities of the lock and unlock state can be expressed as follows:

\[ I_{UP} \times P_{Unlock} = I_{DN} \times P_{Lock}, \]  

where \( P_{Lock} \) and \( P_{Unlock} \) are the probabilities of the lock and unlock state, and \( I_{UP} \) and \( I_{DN} \) are the amount of the UP and DN current of the charge pump.

Fig. 8(b) shows that our proposed WPD decreases the width of the window by the resolution of the RFDL (\( t_{res} \))
at 1/N of the CK frequency in the lock state, and the width of the window is increased by \( t_{\text{res}} \) at CK frequency in the unlock state. Thus, the probabilities of the lock and unlock state in the steady state are determined by both the resolution of the RFDL and the operating frequency of the WPD. In this case, Eq. (4) can be converted as follows:

\[
\begin{align*}
  t_{\text{res}} \times f_{\text{CK}} \times P_{\text{Unlock}} &= t_{\text{res}} \times \frac{1}{N} \times f_{\text{CK}} \times P_{\text{Lock}} \\
  P_{\text{Unlock}} &= \frac{1}{N} \cdot P_{\text{Lock}},
\end{align*}
\]

where \( P_{\text{Lock}} \) and \( P_{\text{Unlock}} \) are the probabilities of the lock and unlock state, respectively, \( f_{\text{CK}} \) is the frequency of the CK, and \( N \) is the factor by which the CK is divided.

Since \( P_{\text{Lock}} + P_{\text{Unlock}} = 1 \), we can rewrite Eq. (6) as follows:

\[
P_{\text{Unlock}} = \frac{1}{N} \left( 1 - P_{\text{Unlock}} \right).
\]

And the probability of the unlock state can be expressed in terms of \( N \) alone:

\[
P_{\text{Unlock}} = \frac{1}{N+1}.
\]

The transfer function of the DLL can be described with the probability of the unlock state where the DLL loop tracks for input phase error. By changing its sampling window at the different frequency in the lock and unlock state, the WPD reduces the effective bandwidth, \( BW_{\text{eff}} \), of the DLL. This \( BW_{\text{eff}} \) is proportional to the probability of the unlock state [3] and is given by

\[
BW_{\text{eff}} = P_{\text{Unlock}} \times BW = \frac{1}{N+1} \cdot BW,
\]

where \( BW \) is the bandwidth of a conventional DLL loop. If the frequency-dividing factor \( N \) is 8, the probability of the lock and unlock states are 8/9 and 1/9, respectively. Thus the loop bandwidth of the DLL is adaptively controlled during DLL operation and the effective bandwidth is 1/9 of the original bandwidth. This reduction in loop bandwidth reduces phase noise and jitter in the output of the DLL.

We used behavioral verification to assess the phase noise and jitter performance of the WPD. We also compared the performance of DLLs using 3 different types of PD: a bang-bang PD (BBPD), a BBPD with 2-bit majority voter, and a WPD. The verification setup is shown in Fig. 9(a)-(c). In Fig. 9(c), the dividing factor \( N \) is 8, and the resolution of the width of the window is 10 ps. Each DLL is fed with a \( CK_{\text{IN}} \) signal operating between 0.5 GHz and 2.5 GHz, and it contains white Gaussian noise. The \( CK_{\text{OUT}} \) signals of these DLLs are then sampled. Fig. 10(a) and (b) are phase noise plots for the DLL with the BBPD and with the WPD at 2.5 GHz and 0.5 GHz. These figures show that the WPD reduces the effective bandwidth and has good phase noise performance. Fig. 11(a) and (b) are phase noise plots for the DLL using the BBPD with the 2-bit majority voter and the WPD, at 2.5 GHz and 0.5 GHz. The WPD also outperforms the BBPD with the 2-bit majority voter.

The reduction in effective bandwidth improves jitter...
performance, as shown in Fig. 12. The DLL with the WPD minimizes the jitter accumulation, induced by the supply/ground noise of the delay line, the UP/DN dithering, and the internal circuit noise, compared to the DLL with the BBPD and the BBPD with majority voter. We also compare these DLLs in terms of the ratio between the RMS jitter at the output and the input. The results are shown in Fig. 13 over the range from 0.5 GHz to 2.5 GHz. We observe that the jitter ratio of the DLL using the BBPD is 1.60, the jitter ratio of the DLL using the BBPD with 2-bit majority voter is 1.40, and the jitter ratio of the DLL with the WPD is 1.29. These results also indicate that the best jitter performance is obtained from the DLL with the WPD.

4. Implementation of the Duty-cycle Corrector and Phase Interpolator

The mobile memory interface is a half-rate clocking system that samples the data signal at both the rising and
the falling edge of the CK signal. If the duty-cycle distortion occurs in the PLL, the CK distribution tree, the DLL, and the PI, the data sampling margin deteriorates [16, 17]. It is also important to maintain a 50% duty-cycle ratio in the DLL, because our 180° locking DLL compares the falling edge of the DLL input CK and the rising edge of the DLL output CK. Thus a DCC is required in the input of the DLL and the output of the DLL, to ensure a 50% duty-cycle ratio. The DCC, shown in Fig. 14, is based on a CMOS inverter, which is simple and economical in terms of both power and area. This simple type of DCC is suitable for the mobile memory interface. By appropriate selection of PMOS and NMOS with different widths, these relative strengths are balanced to achieve a 50% duty-cycle ratio, under ±40% distortion of the input CK signal [18].

Fig. 13 shows the ratio of the CK RMS jitter at the output to that at the input using the bang-bang PD, the bang-bang PD with the majority voter, and the PD with an adaptive sampling window.

Fig. 14. Implementation of the DCC.

Fig. 15. Implementation the digital phase interpolator.

Fig. 16. Die micrograph and layout.

IV. EXPERIMENTAL RESULTS

We implemented an all-digital DLL for the mobile memory interface in a 65 nm CMOS process, together with a conventional DLL with a BBPD, for comparison. Fig. 16 shows a die micrograph and a layout of the all-digital DLL: the global DLL occupies 0.047 mm², and the local DLL with output DCCs and the PI occupies 0.027 mm². A programmable capacitor array is therefore inserted in the PI to adjust the slew-rate by turning this array on and off, which maintains a high slew-rate over a wide frequency range. This simple form of PI has the further advantage that it consumes no power when the CK signal is not asserted, whereas an analog PI consumes power continuously. This characteristic is appropriate for the mobile memory interface.
The phase noise is measured at 2.5 GHz, with the results shown in Fig. 17. At a frequency offset of 1 MHz, the phase noise of CK\textsubscript{DLL} in the DLL with the WPD is 1.19 dBc/Hz. The integrated jitter (10kHz-100 MHz) of CK\textsubscript{DLL} in the same DLL is 954 fs\textsubscript{rms} at 2.5 GHz.

We measured the long-term jitter of the input and output CK at 2.5 GHz with the results shown in Fig. 18(a)-(c). The jitter in CK\textsubscript{IN} is 1.98 ps\textsubscript{rms} and 27.32 ps\textsubscript{pp}. In the DLL with the BBPD, the jitter in CK\textsubscript{DLL} is 2.96 ps\textsubscript{rms} and 25.32 ps\textsubscript{pp}, and with the WPD, the jitter in CK\textsubscript{DLL} is 2.33 ps\textsubscript{rms} and 23.10 ps\textsubscript{pp}. Fig. 18(d) shows the ratio between the RMS jitter at the output and input of the DLL with the BBPD is 1.49, and the jitter ratio of the DLL with the WPD is 1.17. We note that these ratios are similar to those obtained by behavioral verification, and shown in Fig. 13.

The waveforms shown in Fig. 19(a) and (b) demonstrate that the use of a TDC in the global DLL promotes fast locking: the DLL locks within 6 CK cycles at 0.11 GHz and within 17 CK cycles at 2.5 GHz. The local DLL achieves 180° phase-shift lock and 50% duty-cycle ratio at these frequencies.

Fig. 20(a) and (b) shows the measured DNL curves of the PI at 0.11 GHz and 2.5 GHz, and we see that measured DNL ranges from -0.80 to 1.81 LSB and from -0.84 to 1.40 LSB at these frequencies.

Fig. 21(a) shows the power efficiency in the locking and tracking operation. When locking operation of the DLL begins, a lot of power is consumed by the TDCs in the global DLL. After locking ends, the global DLL is powered down. A breakdown of power usage within the
The performance of our DLL is summarized and compared with other designs suitable for the mobile memory interface in Table 1.
V. Conclusions

We have implemented an all-digital DLL with a phase-shift capability of 180°, which is suitable for the mobile memory interface with an operating frequency range of 0.11-2.5 GHz. This DLL has both global and local DLLs. The global DLL uses a TDC to help it lock quickly. After the global DLL is locked, none of its circuits consume any power. The local DLL uses a PD with an adaptive sampling window, and the width of this window is varied to control the loop bandwidth and reduce jitter accumulation. This improves the timing margin of the data signal.

References

Jihwan Park received the B.S. and M.S. degree in electrical engineering from Seoul National University, Seoul, Korea, in 2010 and 2012, respectively. He is currently working towards the Ph.D. degree in electrical engineering at Seoul National University. His research interests include high-speed I/O circuits and architectures for signal integrity.

Hyeongjun Ko received the B.S. degree in electrical and electronics engineering from Korea University, Seoul, Korea, in 2005. He joined the Hynix Semiconductor (now SK Hynix), Icheon, Korea, 2005. Since then, he has been engaged in I/O circuit design and failure analysis of high-speed DRAM such as DDR2, DDR3, and DDR4 and low-power DRAM such as LPDDR2, LPDDR3 and LPDDR4. His research interests include high-speed and low-power I/O interface and signal integrity.

Woo-Yeol Shin received the B.S., M.S., and Ph.D degrees in electrical engineering from Seoul National University, Seoul, Korea, in 2005, 2007, and 2013, respectively. In 2013, he joined SK Hynix Semiconductor, Icheon, Korea, where he has been working in the area of high-speed and low-power memory design. His research interests are high-speed I/O, clocking circuits, and memory design.

Hankyu Chi received the B.S., M.S., and Ph.D degree in electronics engineering from Seoul National University, Seoul, Korea, in 2005, 2007, 2013 respectively. He joined SK hynix Inc., Icheon in 2013, where he has been involved in the high-speed interface design team. His interests include high speed interface, equalizer, CDR, and silicon photonics. He is currently focusing on post-DDR5 I/O design.

Joo-Hyung Chae received the B.S. degree in electrical engineering, in 2012, from Seoul National University, Seoul, Korea, where he is currently working toward the Ph.D. degree. His research interests are the design of high-speed I/O circuits, clock generation circuits, and memory interface.

Mino Kim received the B.S. degree in electrical engineering from Seoul National University, Seoul, Korea, in 2010. He is currently working toward the Ph.D. degree in electrical engineering at Seoul National University. His research interests include high-speed I/O circuits, clock generation circuits and high-speed memory interfaces.

Gi-Moon Hong received the B.S., M.S. and Ph.D degrees in electrical engineering from Seoul National University, Seoul, Korea, in 2009, 2011, 2016, respectively. In 2016, he joined at SK Hynix, Korea. His research interests include low-power and high-speed serial link circuits and memory interface.

Jihwan Park received the B.S. and M.S. degree in electrical engineering from Seoul National University, Seoul, Korea, in 2010 and 2012, respectively. He is currently working towards the Ph.D. degree in electrical engineering at Seoul National University. His research interests include high-speed I/O circuits and architectures for signal integrity.

Hyeongjun Ko received the B.S. degree in electrical and electronics engineering from Korea University, Seoul, Korea, in 2005. He joined the Hynix Semiconductor (now SK Hynix), Icheon, Korea, 2005. Since then, he has been engaged in I/O circuit design and failure analysis of high-speed DRAM such as DDR2, DDR3, and DDR4 and low-power DRAM such as LPDDR2, LPDDR3 and LPDDR4. His research interests include high-speed and low-power I/O interface and signal integrity.

Woo-Yeol Shin received the B.S., M.S., and Ph.D degrees in electrical engineering from Seoul National University, Seoul, Korea, in 2005, 2007, and 2013, respectively. In 2013, he joined SK Hynix Semiconductor, Icheon, Korea, where he has been working in the area of high-speed and low-power memory design. His research interests are high-speed I/O, clocking circuits, and memory design.

Hankyu Chi received the B.S., M.S., and Ph.D degree in electronics engineering from Seoul National University, Seoul, Korea, in 2005, 2007, 2013 respectively. He joined SK hynix Inc., Icheon in 2013, where he has been involved in the high-speed interface design team. His interests include high speed interface, equalizer, CDR, and silicon photonics. He is currently focusing on post-DDR5 I/O design.
Deog-Kyoon Jeong (S’85–M’89–SM’09) received the B.S. and M.S. degrees in electronics engineering from Seoul National University, Seoul, Korea, in 1981 and 1984, respectively, and the Ph.D. degree in electrical engineering and computer sciences from the University of California at Berkeley, Berkeley, CA, USA, in 1989. He was with Texas Instruments, Dallas, TX, USA, as a member of the Technical Staff, from 1989 to 1991, where he was involved in the modeling and design of BiCMOS gates and the single-chip implementation of the SPARC architecture. Then, he joined as a Faculty Member with the Department of Electronics Engineering and the Inter-University Semiconductor Research Center, Seoul National University, where he is currently a Professor. He was one of the co-founders of Lattice Semiconductor, Hillsboro, OR, USA, which specialized in digital interface circuits for video displays, such as Digital Visual Interface and HDMI. His current research interests include the design of high-speed I/O circuits, phase-locked loops, and memory system architecture. Dr. Jeong was one of the recipients of the ISSCC Takuo Sugano Award for Outstanding Far-East Paper in 2005.

Suhwan Kim (S’97–M’01–SM’07) received the B.S. and M.S. degrees in electrical engineering and computer science from Korea University, Seoul, South Korea, in 1990 and 1992, respectively, and the Ph.D. degree in electrical engineering and computer science from the University of Michigan, Ann Arbor, MI, USA, in 2001. From 1993 to 1999, he was with LG Electronics, Seoul, South Korea. From 2001 to 2004, he was a Research Staff Member with IBM T. J. Watson Research Center, Yorktown Heights, NY, USA. In 2004, he joined Seoul National University, Seoul, South Korea, where he is currently a Professor of Electrical Engineering. His research interests include high-performance and low-power analog and mixed-signal integrated circuits, high-speed I/O circuits, and power electronics. Dr. Kim served as a Guest Editor for the IEEE Journal of Solid-State Circuits special issue on the IEEE Asian Solid-State Circuits Conference. He has also served as the General Cochair and Technical Program Chair for the IEEE International System-on-Chip (SoC) Conference. He has participated multiple times on the Technical Program Committee of the IEEE International SOC Conference, the International Symposium on Low-Power Electronics and Design, the IEEE Asian Solid-State Circuits Conference, and the IEEE International Solid-State Circuits Conference. He received the 1991 Best Student Paper Award from the IEEE Korea Section and the First Prize (Operational Category) in the VLSI Design Contest of the 2001 ACM/IEEE Design Automation Conference.