A 1.74mW/GHz 0.11-2.5GHz Fast-Locking, Jitter-Reducing, 180° Phase-Shift Digital DLL with a Window Phase Detector for LPDDR4 Memory Controllers

Joo-Hyung Chae[1], Gi-Moon Hong[1], Jihwan Park[1], Mino Kim[1], Hyeongjun Ko[1,2], Woo-Yeol Shin[2], Hankyu Chi[2], Deog-Kyoon Jeong[1], and Suhwan Kim[1]
[1]Department of Electrical and Computer Engineering, Seoul National University, Seoul, Korea
[2]SK hynix, Icheon, Korea
E-mail: joo-hyung.chae@amic.snu.ac.kr, suhwan@snu.ac.kr

***Abstract* — A 180° phase-shift digital delay-locked loop (DLL) for LPDDR4 memory controllers is composed of a global DLL and a local DLL for each channel. The global DLL uses a time-to-digital converter to achieve fast-locking, and then shuts down to reduce power consumption. The local DLL, locking based on delay codes from the global DLL, uses a digital window phase detector (PD) and tracks the input clock phase to compensate for process, voltage, and temperature variations. Repeatedly controlled window size of the digital window PD in this local DLL reduces the high-frequency jitter compared to the DLL using bang-bang PD. Implemented in 65nm CMOS process, proposed digital DLL dissipates 1.74mW/GHz and occupies 0.074mm^2. It operates over a frequency range of 0.11-2.5GHz, and locks within 6 cycles at 0.11GHz and within 17 cycles at 2.5GHz. At 2.5GHz, the integrated jitter of the DLL output clock with the digital window PD is 953fs$_{rms}$ and the long-term jitter of it is 2.64ps$_{rms}$ and 20.6ps$_{pp}$.**

Keywords—DLL, LPDDR4, fast-locking, jitter-reducing, wide frequency range, digital window phase detector

I. INTRODUCTION

In an LPDDR4 memory controller, a delay-locked loop (DLL) is required for command bus training, write leveling, read training, write training, and normal operation over a frequency range of 266-2133MHz. During its operation, the DLL is expected to operate over a wide frequency range, and exhibit fast-locking with low power consumption and low jitter. The use of the conventional DLL architecture, having inherent all-pass filter noise transfer characteristics, is likely to reduce the performance of LPDDR4 memory controllers, because the dithering jitter and the power/ground jitter of the delay line increase the output clock jitter. Therefore, the jitter-reducing technique should be applied to the DLL of LPDDR4 memory controllers.

The incorporation of an injection-locked oscillator (ILO) into the DLL filters out the high-frequency jitter [1], but this approach also tends to reduce the frequency range, due to the narrow locking range of the ILO. Alternatively, an analog window phase detector (PD) in the DLL will reduce the jitter induced by the supply noise [2], but this requires a quadrature phase generator, which has a narrow operating frequency range, and is sensitive to process, voltage, and temperature (PVT) variations. Yet another approach is to use an adaptive phase-interval detector to eliminate the dithering jitter [3], but the lattice delay line in this DLL can produce remarkably large phase offsets at high frequencies, impeding accurate locking.

In this paper, we present a 180° phase-shift digital DLL, consisting of a single global DLL and a local DLL for each channel, for LPDDR4 memory controllers with an operating frequency range of 0.11-2.5GHz. The global DLL uses a time-to-digital converter (TDC) for fast-locking and the TDC stops operating when locking is complete. The local DLL receives locking delay codes from the global DLL and then tracks the input clock phase to compensate for PVT variations of each channel. A digital window PD, which controls window size automatically to adjust loop gain, and a frequency divider in the local DLL prevent the dithering phenomenon and reduce the high-frequency jitter without introducing unnecessary dynamic power consumption.

II. DLL ARCHITECTURE AND IMPLEMENTATION

Fig. 1 shows the simplified architecture of the LPDDR4 memory controller. A DLL consists of a global DLL, located

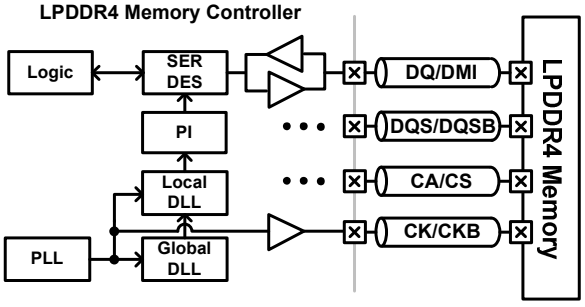


Fig. 1. Simplified architecture of the LPDDR4 memory controller.

near the PLL, for fast-locking and a local DLL in each channel to compensate for PVT variations and reduce the high-frequency jitter. A phase interpolator (PI) is coupled to the local DLL to support each training operations and evaluate the performance of the LPDDR4 memory chipsets by generating 1UI/64step clock signals. Clock signal after PI is transmitted to serializer and deserializer (SERDES) for data (DQ) serializing and deserializing.

The overall architecture of our digital DLL is described in Fig. 2. The global DLL is composed of a coarse TDC and a fine TDC for fast-locking, a digital block to handle delay codes transmitted from the TDC, and a delay line which is made up of a coarse delay line (CDL) and a fine delay line (FDL). The local DLL of each channel has a digital window PD to reduce the high-frequency jitter, a digital loop filter (DLF) and its own delay line.

A. Fast Locking Sequence in the Global DLL

Several DLL architectures with fast-locking have recently been introduced: TDC-based DLLs [1], [4], [5], SAR-based DLLs [6], [7] and other designs [8], which all lock within a few clock cycles. Time-to-digital converters have the disadvantages of large area and high power consumption, but they offer the benefit of not requiring further operations after locking [1], and thus do not consume any power during normal operation. Combining a TDC-based DLL with a conventional PD-based DLL within a multi-channel interface, such as an LPDDR4 memory controller, yields the advantages and minimizes the disadvantages of each architecture.

In our design, the global DLL uses a coarse TDC and a fine TDC to lock quickly when the chip starts up. When the PLL locks, the coarse TDC locks coarse delay lines (CDLs), and when this is accomplished, the coarse TDC issues a lock detection signal that triggers the fine TDC. After fine delay lines (FDLs) have locked, the fine TDC issues its own lock detection signal. All the circuits in the global DLL are powered down, and then delay codes, generated by the TDC, are transmitted to the DLF in the local DLL. Powering down all the circuits in the global DLL when locking is complete, mitigates the impact of the high power consumption associated with the TDC-based DLL.

B. Jitter-Reducing and PVT Tracking using a Digital Window Phase Detector in the Local DLL

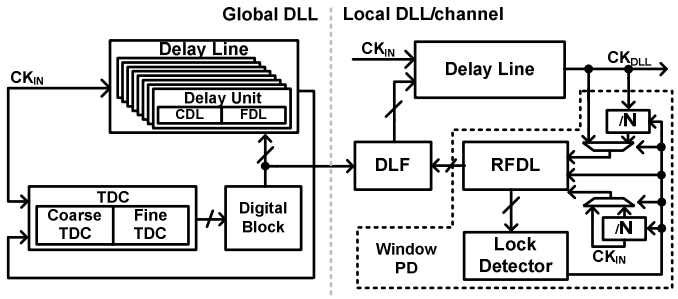


Fig. 2. Overall architecture of the proposed DLL.

The PD-based local DLL responds to the delay codes issued by the global DLL to compensate for the PVT variations in each strobe (DQS) channel. This local DLL uses a window PD, as shown in Fig. 3, rather than a bang-bang PD, to reduce the dithering and high-frequency jitter. The window PD is composed of replica fine delay lines (RFDLs), a lock detector that judges the success of the 180° phase-shift lock, and a frequency divider (/N) that provides a low-frequency clock signal to activate and drive the window PD after the lock flag has been received.

Fig. 4 shows how the window PD operates. The RFDLs change delay codes to control the values of the signals $dCK_{IN}+n\Delta t$, dCK_{DLL} and $dCK_{DLL}+2n\Delta t$, where n is the value of delay codes sent to the RFDLs, and Δt is their delay resolution. The rising edges of dCK_{DLL} and $dCK_{DLL}+2n\Delta t$ form a window with a width of $2n\Delta t$. If the falling edge of $dCK_{IN}+n\Delta t$ is caught by this window, then the PD<1:0> signals have different values and a locking state is entered: The lock detector sets the lock flag to '1' and sends it to the RFDLs, frequency divider, and MUX. When the lock flag is transmitted to the RFDLs, delay codes of the RFDLs are decreased so as to narrow the window, and the window PD is now operated by the CK_{IN}/N and CK_{DLL}/N signals from the frequency divider to lower dynamic power consumption. If some combination of CK_{IN} jitter, supply and ground noise, and PVT variations cause the local DLL to break the locking state, then the window PD increases delay codes of the RFDLs to widen the width of its window, and it is again operated by CK_{IN} and CK_{DLL} instead of CK_{IN}/N and CK_{DLL}/N; and then it is re-entered the locking

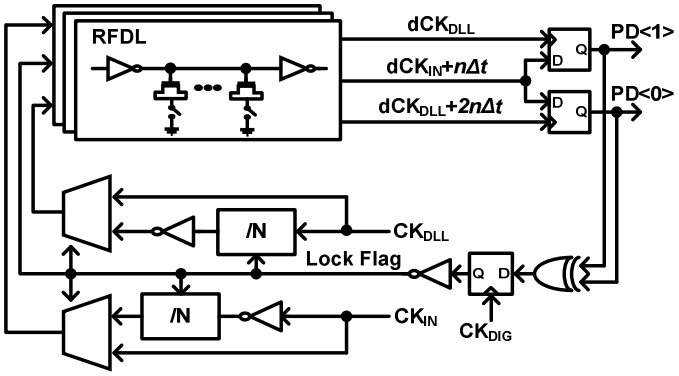


Fig. 3. Block diagram of the digital window phase detector in the local DLL.

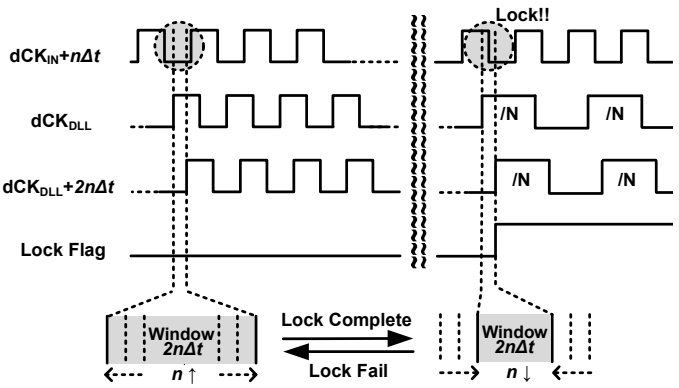


Fig. 4. Timing diagram and operation of the digital window phase detector.

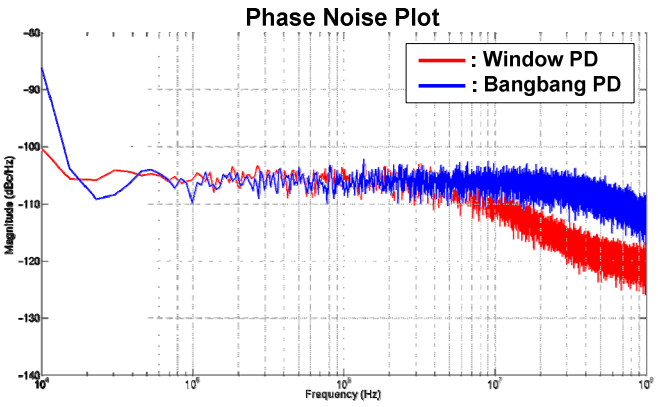
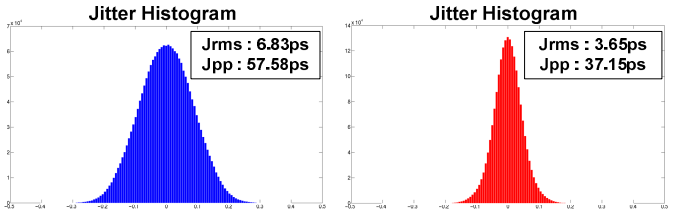


Fig. 5. Phase noise plot of DLLs using a window PD and a bang-bang PD.



(a) (b)

Fig. 6. Jitter histograms of (a) a DLL using a bang-bang PD and (b) a DLL using a window PD.

state. Repeatedly controlling window size to adjust loop gain of the local DLL prevents the dithering phenomenon and reduces both the high-frequency jitter and the dynamic power consumption of the digital window PD which is proportional to the clock frequency.

For comparing and analyzing the performance of our design with that of the conventional design using the bang-bang PD, verilog modeling and matlab simulation are performed. 2GHz clock signal having white Gaussian noise is entered into the input of DLLs, and then the output clock signals are sampled and observed. Fig. 5 shows the phase noise of these DLLs' output. Due to the dithering jitter of the bang-bang PD, high-frequency noise is increased. For the DLL with the proposed digital window PD, high-frequency noise beyond 1MHz frequency offset is reduced and the jitter performance is improved as shown in Fig. 6. RMS jitter (J_{rms}) and peak-to-peak jitter (J_{pp}) of the DLL using the bang-bang PD are 6.83ps and 57.58ps, respectively, and J_{rms} and J_{pp} of the DLL using the digital window PD are 3.65ps and 37.15ps, respectively.

III. MEASUREMENT RESULTS

The proposed digital DLL is implemented in a 65nm CMOS process. Fig. 7 shows a die micrograph and a layout, in which the global DLL and local DLL occupy areas of 0.047mm^2 and 0.027mm^2 respectively.

In order to verify jitter-reducing effect of our architecture using the window PD, both conventional bang-bang PD mode and window PD mode are employed in our DLL and the phase noise and jitter measurement are performed. At 10MHz

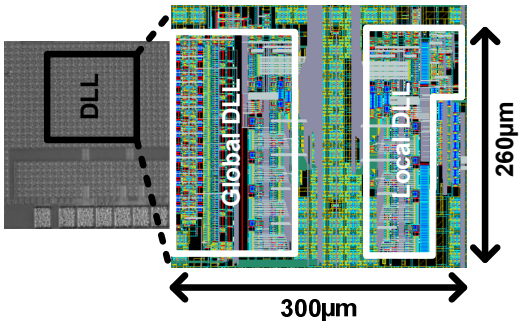


Fig. 7. Die micrograph and layout.

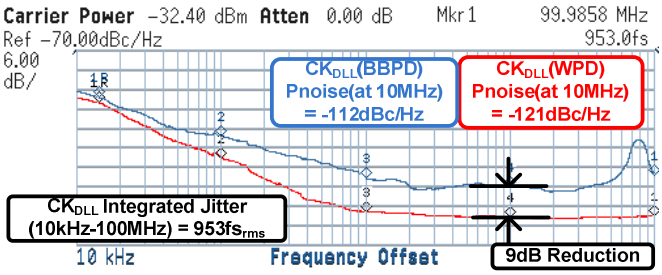


Fig. 8. Measured phase noise plot of the DLLs using bang-bang PD and the proposed DLL at 2.5GHz.

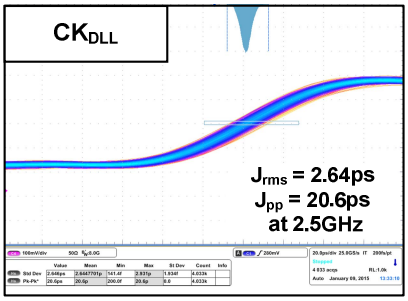


Fig. 9. Measured long-term jitter performance of the proposed DLL at 2.5GHz

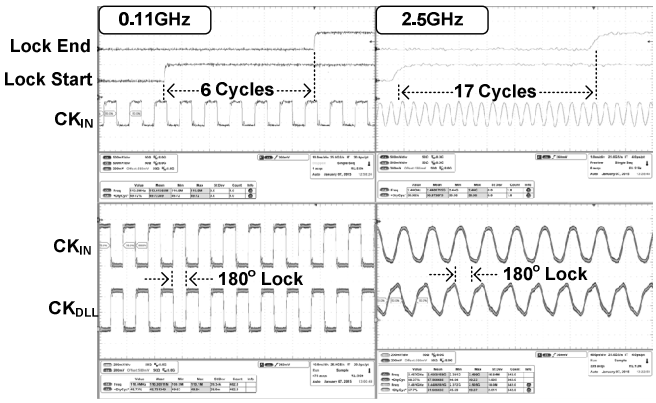


Fig. 10. Measured waveforms illustrating DLL locking behavior at (a) 0.11GHz and (b) 2.5GHz.

frequency offset, with the bang-bang PD the phase noise of CK_{DLL} is -112dBc/Hz, and with the digital window PD it is -121dBc/Hz, as shown in Fig. 8. Thus the phase noise of CK_{DLL} with the digital window PD is 9dB better than CK_{DLL} with the bang-bang PD at 10MHz frequency offset; and the integrated jitter (10kHz-100MHz) of CK_{DLL} with the digital window PD at 2.5GHz is 953fs$_{rms}$. Fig. 9 shows the long-term jitter

performance of the proposed DLL. At 2.5GHz, clock frequency, J_{rms} and J_{pp} are 2.64ps and 20.6ps respectively.

The waveforms in Fig. 10 show that the use of a TDC in the global DLL achieves fast-locking from 0.11GHz to 2.5GHz by observing Lock Start, Lock End, CK_{IN}, and CK_{DLL}. This scheme allows the 180° phase-shift digital DLL to lock within 6 clock periods at 0.11GHz and within 17 clock periods at 2.5GHz.

IV. Conclusion

In this paper, we propose a 180° phase-shift digital DLL with an operating frequency range of 0.11-2.5GHz for LPDDR4 memory controllers.

The performance of our proposed DLL is summarized in Fig. 11: The use of the TDC in the global DLL allows the DLL to lock within 6 clock cycles at 0.11GHz and within 17 clock cycles at 2.5GHz. The use of the digital window PD, repeatedly controlling window size to adjust loop gain, and the frequency divider prevent the dithering phenomenon that the bang-bang PD inherently has while reducing the high-frequency jitter and the dynamic power consumption. Thus, the timing margin of SERDES in LPDDR4 memory controllers is improved. The phase noise plot indicates that the phase noise of CK_{DLL} with the digital window PD is 9dB better than the phase noise of CK_{DLL} with the bang-bang PD at 10MHz frequency offset. The integrated jitter (10kHz-100MHz) of CK_{DLL} using the digital window PD running at 2.5GHz is $953fs_{rms}$ and the long-term J_{rms} and J_{pp} are $2.64ps_{rms}$ and $20.6ps_{pp}$, respectively. The normalized power consumption of the proposed DLL is 1.74mW/GHz.

Technology	**65nm CMOS process**
Frequency Range (Ratio)	**0.11-2.5GHz (x23)**
Lock Time	**6 cycles (at 0.11GHz)** **17 cycles (at 2.5GHz)**
Integrated Jitter (at 2.5GHz)	**$953fs_{rms}$ (10kHZ-100MHz)**
Long-term Jitter (at 2.5GHz)	**$2.64ps_{rms}$** **$20.6ps_{pp}$**
Area	**$0.047mm^2$ (Global DLL)** **$0.027mm^2$ (Local DLL)**
Normalized Power	**1.74mW/GHz**

Fig. 11. Performance summary.

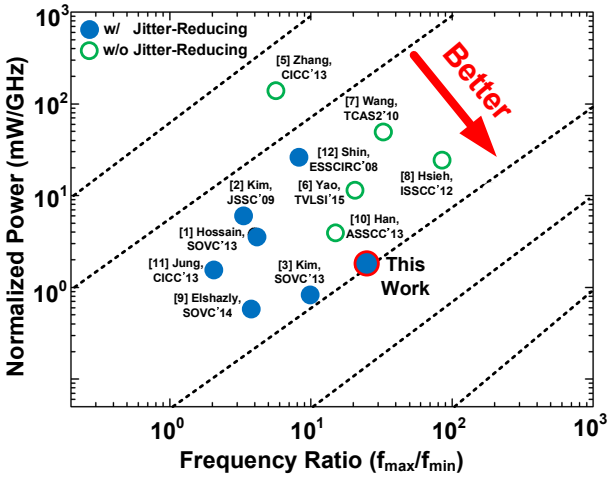


Fig. 12. Comparison of frequency ratio and normalized power for recently published designs of fast-locking and wide frequency range DLLs with closed-loop architecture.

Fig. 12 shows that our digital DLL outperforms the other recently published fast-locking designs operating over a wide frequency range and having a closed-loop architecture suitable for use in a memory controller interface.

Acknowledgment

This research was supported in part by Inter-university Semiconductor Research Center (ISRC) of Seoul National University.

References

[1] M. Hossain et al, "A 400MHz-1.6GHz fast lock, jitter filtering ADDLL based burst mode memory interface," *IEEE Symp. on VLSI Circuits*, pp. 244-245, June 2013.

[2] B.-G. Kim, L.-S. Kim, K.-I. Park, Y.-H. Jun, and S.-I. Cho, "A DLL with jitter reduction techniques and quadrature phase generation for DRAM interface," *IEEE J. of Solid-State Circuits*, vol. 44, pp. 1522-1530, May 2009.

[3] K. Kim et al, "A 1.3mW, 1.6GHz digital delay-locked loop with two-cycle locking time and dither-free tracking," *IEEE Symp. on VLSI Circuits*, pp. 158-159, June 2013.

[4] T. Saeki et al, "2.5ns clock access, 250MHz, 256-Mb SDRAM with synchronous mirror delay," *IEEE J. of Solid-State Circuits*, vol. 31, pp. 1656-1665, Nov 1996.

[5] D. Zhang et al, "A fast-locking digital DLL with a high resolution time-to-digital converter," *IEEE Custom Integrated Circuit Conference*, Sept 2013.

[6] C.-Y.Yao et al, "Designing a SAR-based all-digital delay-locked loop with constant acquisition sycles using a resettable delay line," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 23, pp. 567-574, Mar 2015.

[7] L. Wang, L. Liu, and H. Chen, "An implementation of fast-locking and wide-range 11bit reversible SAR DLL," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 57, no, 6, pp. 421-425, June 2010.

[8] M.-H.Hsieh et al, "A 6.7MHz-to-1.24GHz $0.0318mm^2$ fast-locking all-digital DLL in 90nm CMOS," *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, pp. 244-246, Feb 2012.

[9] A. Elshazly et al, "A 2GHz-to-7.5GHz quadrature clock-generator using digital delay locked loops for multi-standard I/Os in 14nm CMOS," *IEEE Symp. on VLSI Circuits*, June 2014.

[10] S. Han, T. Kim, and J. Kim, "A 0.1-1.5GHz all-digital phase inversion delay-locked loop," *IEEE Asian Solid-State Circuits Conf.*, pp. 341-344, Nov 2013.

[11] D.-H.Jung et al, "All-digital 90° phase-shift DLL with a dithering jitter suppression scheme," *IEEE Custom Integrated Circuit Conference*, Sept 2013.

[12] D. Shin et al, "A 0.17-1.4GHz low-jitter all digital DLL with TDC-based DCC using pulse width detection scheme," *IEEE European Solid-State Circuits Conf.*, pp. 82-85, Sept 2008.