A fast-locking harmonic-free digital DLL for DDR3 and DDR4 SDRAMs

Junsub Yoon, Seo Weon Heo, and Jongsun Kim^{a)}

Electronic and Electrical Engineering, Hongik University,

94 Wausan-ro, Mapo-gu, Seoul 121–791, Korea

a) js.kim@hongik.ac.kr

Abstract: A new digital delay-locked loop (DLL) for DDR3/DDR4 SDRAM is presented. The proposed digital DLL employs a new noise-tolerant triple (MSB-interval + binary + sequential) search algorithm for implementing a harmonic-free, fast-locking capability while retaining low jitter, low power performance, and a wide operating frequency range. The proposed DLL with duty-cycle correction is designed using a 38-nm CMOS process and occupies an active area of just 0.02 mm². The DLL operates over a frequency range of 0.3–2.0 GHz and achieves a peak-to-peak jitter of 7.78 ps and dissipates 3.48 mW from a 1.1 V supply at 1 GHz.

Keywords: delay-locked loop, DDR3, DDR4, SDRAM, harmonic-free, DLL

Classification: Integrated circuits

References

- [1] K. Sohn, *et al.*: “A 1.2 V 30 nm 3.2 Gb/s/pin 4 Gb DDR4 SDRAM with dual-error detection and PVT-tolerant data-fetch scheme,” *IEEE J. Solid-State Circuits* **48** (2013) 168 (DOI: [10.1109/JSSC.2012.2213512](https://doi.org/10.1109/JSSC.2012.2213512)).
- [2] W.-J. Yun, *et al.*: “A digital DLL with hybrid DCC using 2-step duty error extraction and 180° phase aligner for 2.67 Gb/S/pin 16 Gb 4-H stack DDR4 SDRAM with TSVs,” *IEEE ISSCC Dig. Tech. Papers* (2015) 396 (DOI: [10.1109/ISSCC.2015.7063056](https://doi.org/10.1109/ISSCC.2015.7063056)).
- [3] J.-H. Lim, *et al.*: “A delay locked loop with a feedback edge combiner of duty-cycle corrector with a 20%–80% input duty cycle for SDRAMs,” *IEEE Trans. Circuits Syst. II, Exp. Briefs* **63** (2016) 141 (DOI: [10.1109/TCSII.2015.2468911](https://doi.org/10.1109/TCSII.2015.2468911)).
- [4] H. Kang, *et al.*: “Process variation tolerant all-digital 90 phase shift DLL for DDR3 interface,” *IEEE Trans. Circuits Syst. I, Reg. Papers* **59** (2012) 2186 (DOI: [10.1109/TCSI.2012.2188943](https://doi.org/10.1109/TCSI.2012.2188943)).
- [5] G. K. Dehng, *et al.*: “Clock-deskew buffer using a SAR-controlled delay-locked loop,” *IEEE J. Solid-State Circuits* **35** (2000) 1128 (DOI: [10.1109/4.859501](https://doi.org/10.1109/4.859501)).
- [6] L. Wang, *et al.*: “An implementation of fast-locking and wide-range 11-bit reversible SAR DLL,” *IEEE Trans. Circuits Syst. II, Exp. Briefs* **57** (2010) 421 (DOI: [10.1109/TCSII.2010.2048379](https://doi.org/10.1109/TCSII.2010.2048379)).
- [7] R. Yang and S. Liu: “A 40–550 MHz harmonic-free all-digital delay locked loop using a variable SAR algorithm,” *IEEE J. Solid-State Circuits* **42** (2007) 361 (DOI: [10.1109/JSSC.2006.889381](https://doi.org/10.1109/JSSC.2006.889381)).

- [8] S. Han and J. Kim: “A high-resolution wide-range dual-loop digital delay-locked loop using a hybrid search algorithm,” IEEE Asian Solid-State Circuits Conference (2012) 293 (DOI: [10.1109/IPEC.2012.6522683](https://doi.org/10.1109/IPEC.2012.6522683)).
- [9] S. Han, *et al.*: “A 0.1–1.5 GHz all-digital phase inversion delay-locked loop,” IEEE Asian Solid-State Circuits Conference (2013) 341 (DOI: [10.1109/ASSCC.2013.6691052](https://doi.org/10.1109/ASSCC.2013.6691052)).
- [10] D.-H. Jung, *et al.*: “All-digital fast-locking delay-locked loop using a cyclic-locking loop for DRAM,” IEEE Trans. Circuits Syst. II, Exp. Briefs **62** (2015) 1023 (DOI: [10.1109/TCSII.2015.2456111](https://doi.org/10.1109/TCSII.2015.2456111)).
- [11] Y. Jeon, *et al.*: “A 66–333-MHz 12-mW register-controlled DLL with a single delay line and adaptive-duty-cycle clock dividers for production DDR SDRAM,” IEEE J. Solid-State Circuits **39** (2004) 2087 (DOI: [10.1109/JSSC.2004.835809](https://doi.org/10.1109/JSSC.2004.835809)).
- [12] C.-Y. Yao, *et al.*: “Designing a SAR-Based all-digital delay-locked loop with constant acquisition cycles using a resettable delay line,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **23** (2015) 567 (DOI: [10.1109/TVLSI.2014.2313131](https://doi.org/10.1109/TVLSI.2014.2313131)).
- [13] DDR3 SDRAM Standard: JEDSD79-3F (2012) <https://www.jedec.org/standards-documents/docs/jesd-79-3d>.
- [14] DDR4 SDRAM Standard: JEDSD79-4A (2013) <https://www.jedec.org/standards-documents/results/jesd79-4%20ddr4>.

1 Introduction

Synchronous dynamic random access memory (SDRAM) has served as an important low-cost main memory solution for the personal computer (PC) and other cost-sensitive consumer electronics markets. Traditional SDRAM provides a maximum memory bus data rate of only 166 Mbps/pin with a maximum clock rate of 166 MHz. As processor speeds continue to increase, the PC becomes more reliant on low cost, higher bandwidth memory solutions. The double data rate (DDR) SDRAM was introduced to the public market to meet these requirements. DDR SDRAM, also called DDR1 SDRAM, continues to evolve and has been superseded by various DDR-x SDRAMs (i.e., DDR2, DDR3, and DDR4) that provide better performance and consume lower power. DDR3 was introduced in 2007 and DDR4 in 2013 [1]. Currently, the market is moving swiftly from DDR3 to DDR4 because of the lower power consumption and higher speed advantages of DDR4.

In order to achieve a memory bus data rate of over 400 Mbps/pin, DDR-x SDRAMs must incorporate an on-chip delay-locked loop (DLL) [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12] that can eliminate skew problems and achieve higher timing margin at high frequencies. To design a DLL that can support both DDR3 and DDR4 specifications at the same time [13, 14], the DLL should be locked within 512 clock cycles and operate over a frequency range from 300 MHz to 1.6 GHz using an internal supply voltage of less than 1.2 V. Also, the DLL must be capable of correcting the duty cycle of the distorted input clock so that the data-valid window (tDV) could be widened [2].

Currently, most DDR3/DDR4 SDRAMs use a digital DLL [1, 2, 3, 4, 10, 11]. One of the reasons for using digital architectures is because DDR3/DDR4 SDRAMs require fast recovery times for various power mode transitions. To

achieve a fast locking time, a successive approximation register (SAR)-based binary search algorithm was adopted in DLL designs [5, 6, 7, 8, 9]. However, this introduced harmonic lock problem [6, 7, 12]. Harmonic locking may occur when the most significant bit (MSB) of the SAR code is changed in the beginning of the SAR operation, since the delay of the DLL is increased to 50% of the total delay line. To eliminate the harmonic locking problem in a SAR-based digital DLL, a variable SAR (VSAR) algorithm was introduced [7]. However, [7] requires a very complex and timing sensitive fail-to-lock detection circuit which could easily cause logic failures in the presence of noise in the power supply or ground.

In this paper, a digital DLL for DDR3/DDR4 SDRAMs is proposed which relies on a new noise-tolerant triple (MSB-interval + binary + sequential) search algorithm for achieving a harmonic-free, fast-locking capability while maintaining low jitter, low power consumption, and a wide operating frequency range.

2 Circuit design

Fig. 1(a) shows a block diagram of the proposed digital DLL for DDR3/DDR4 SDRAMs. It consists of a clock buffer, a digitally-controlled delay line (DCDL), a duty-cycle corrector (DCC), a DCDL controller, a divider, a phase detector (PD), a clock tree, and a replica delay line (RDL). The DCDL comprises a coarse delay line

Fig. 1. (a) Proposed digital DLL architecture (b) Flow chart of the proposed triple search algorithm with three operating modes

Fig. 2. Proposed CDL with eight MSB-interval periods

(CDL) and a fine delay line (FDL). The DCDL controller consists of a control logic, a 10-bit multi-mode register (MMR) and two 5-to-32 thermometer decoders. The clock tree, which introduces a delay of t_2 , is a clock distribution network connected to the output drivers (DQs) of the SDRAM. The RDL is a dummy delay that replicates, with a smaller area and lower power overhead, the propagation delay of the clock tree. To attain phase aligned locking between the CLK/CLKB and the CLK_{DQ} without clock skews, the delay of the RDL (t_3) is given by the equation $t_3 = t_1 + t_2$, where t_1 is the CLK buffer delay. Many of the previously reported fast-locking DLL schemes cannot be implemented in DDR-x SDRAMs due to the existence of the RDL which affects the locking time and the harmonic locking issue.

The PD takes CLK_{IN} and CLK_{FB} as inputs and generates a Comp signal that is used by the DCDL controller to adjust the DCDL delay. M[4:0], the five most-significant-bits (MSBs) of the 10-bit MMR, are converted to the thermometer codes, C[31:0]/Cb[31:0], by the 5-to-32 decoder. The codes C[31:0]/Cb[31:0] are used for controlling the CDL of the DCDL. L[4:0], the five least-significant bits (LSBs) of the MMR, are converted to the thermometer codes, F[31:0]/Fb[31:0]. The codes F[31:0]/Fb[31:0] are used for adjusting the magnitude of the delay in the FDL.

This digital DLL architecture has three operating modes (MSB, SAR, and Counter mode) and utilizes a triple (MSB-interval + binary + sequential) search algorithm for achieving fast locking without harmonic locking, as shown in Fig. 1(b).

The MSB mode relies on eight evenly spaced delay intervals (1'st~8th interval) to control the 5-bit CDL, which provides high noise tolerance in the DCDL controller design.

Fig. 2 shows the proposed CDL which is made up of 32 cascaded digital delay elements (DEs). The DE consists of four NAND gates [7] and the delay step in a

DE is equivalent to two NAND gates. Therefore the maximum total delay of the CDL is $32 \times t_D$, where t_D is the propagation delay of the DE.

The delay of the CDL is controlled by the 32-bit thermometer code $C[31:0]/Cb[31:0]$. As the operating frequency decreases, the required number of activated DEs increases. The fine delay line (FDL) is based on feedback delay element (FDE)-based inverters [9] and the tunable delay range of the FDL is equal to one DE delay, t_D . Since the FDL delay is incremented or decremented by the 32-bit $F[31:0]/Fb[31:0]$ code, the delay resolution of the proposed digital DLL is about $t_D/2^5$ ($= 135 \text{ ps}/32 = 4.2 \text{ ps}$).

The propagation delay of the DLL can be represented as follows

$$t_{\text{propagation delay}} = t_{\text{variable}} + t_{\text{fixed}} = N \times t_{\text{CK}} \quad (1)$$

where t_{variable} is the tunable delay of the DCDL, t_{fixed} is the initial fixed delay of the DLL when the t_{variable} equals to zero, and t_{CK} is the cycle time of the input clock. For ideal phase lock locking without the harmonic lock problem, N needs to be one. Therefore, t_{variable} can be represented as follows.

$$t_{\text{variable}} = 1 \times t_{\text{CK}} - t_{\text{fixed}} \quad (2)$$

Since the CDL consists of 32 DEs and the tunable delay range of the FDL is equal to one t_D , t_{variable} becomes as follows.

$$t_{\text{variable}} = (32 + 1) \times t_D = 33 \times t_D \quad (3)$$

In order to achieve harmonic-free operation at the minimum frequency of 0.3 GHz ($t_{\text{CK}} = 3.33 \text{ ns}$) with any values of t_{fixed} and t_3 , the maximum variable delay of the DCDL ($t_{\text{variable_max}}$) should be at least 3.33 ns. Therefore, the proper value of t_D should be larger than 100 ps ($= t_{\text{variable_max}}/33 = 3.33 \text{ ns}/33$) even in the fastest process corners. Consequently, by choosing a typical t_D of 135 ps, the proposed DLL can achieve harmonic-free wide-range operation from 300 MHz to 2.0 GHz.

Referring to Fig. 1(b) and Fig. 2, the DLL starts in the MSB mode with $M[4:0] = [00010]$, which enables the two DEs (#1 and #2) of the CDL and subsequently the MSB-interval search algorithm is executed. Therefore, at the beginning, the DLL starts in the 1'st interval period and then moves to the 2'nd interval period regardless of the Comp signal by enabling four DEs (#1~#4) of the CDL. Starting from the 2'nd interval period, the DLL moves to the next interval period if the Comp signal is logic high. For instance, $M[4:0] = [11100]$ represents the 8'th interval period, in which twenty eight DEs (#1~#28) are enabled in the CDL. When the Comp signal is changed from logic high to low, the MSB mode is completed and the DLL enters the SAR mode and the binary search algorithm begins.

Fig. 3(a) illustrates the triple search locking process along with the DCDL control bits and the three operating modes. The MSB mode is used to set the DLL output clock near the locking point within eight CLK_{CTRL} cycles, where CLK_{CTRL} is an output of the divide-by- N divider with $N = 4$. Thus the delay range of the DCDL is separated by eight interval periods, which produces a maximum delay step change of only $4 \times t_D$ in the MSB mode. When the MSB mode is completed, the DLL enters the SAR mode and the binary search algorithm is applied to the MMR.

Fig. 3. (a) Example of the triple search locking process which shows the DCDL control bits and the three operating modes (b) 10-bit Multi-Mode Register (MMR)

Fig. 3(b) shows the 10-bit MMR. In order to increase the DCDL delay rapidly without incurring any harmonic locking problem, only the 3 MSB bits, M[4:2], are controlled in the MSB mode. Then the rest 7 LSB bits are controlled in the SAR mode for binary search. The SAR-based binary search requires a maximum 7 CLK_{REG} cycles, where CLK_{REG} is an output of the divide-by-M divider with $M = 8$.

After the binary search is completed in the SAR mode, the 10-bit MMR is transformed into a 10-bit counter and the DLL starts the sequential search in the Counter mode, maintaining a closed loop to track process, voltage, and temperature (PVT) variations. As a consequence, the use of triple search algorithm results in a relatively fast locking time with no harmonic locking problems. The worst case locking time of the proposed DLL is 88 ($8 \times \text{CLK}_{\text{CTRL}} + 7 \times \text{CLK}_{\text{REG}} = 8 \times 4 + 7 \times 8$) input clock cycles.

Fig. 4(a) and (b) show the proposed control logic and the 10-bit MMR of the DCDL controller. Fig. 4(c) shows the truth table in the MSB mode and describes the relationship between K[3:0] and M[4:1] and the number of active DEs. In the MSB mode, M[4:2], the three MSBs of the 10-bit MMR and representing the eight interval sequences from 000 to 111, is the same as K[3:1]. Also, the bits M[1:0] are set to [00] and L[4:0], the five LSBs of the MMR, are also set to [00000]. Therefore, M[4:0], the five MSBs, are converted to the thermometer codes, C[31:0]/Cb[31:0], using the 5-to-32 decoder. The signal C[31:0]/Cb[31:0] is used for controlling the number of active delay elements of the CDL. To avoid harmonic locking problems, the number of active DEs in the MSB mode is increased almost linearly (from 2 to 28) with a maximum delay step change of only $4 \times t_D$.

Fig. 4. Proposed DCDL controller (a) control logic (b) 10-bit multi-mode register (MMR) (c) truth table in the MSB mode

3 Experiment results

The proposed digital DLL was designed in a 38-nm Powerchip CMOS process. Fig. 5(a) shows a chip layout which has an active area of $153\mu\text{m} \times 137\mu\text{m}$ ($= 0.02\text{mm}^2$). To minimize process variations and device mismatch problems, 70 nm was used as the minimum channel length for the simulation and layout in this design. Fig. 5(b) shows a layout of the 2 Gb DDR3 SDRAM which incorporates the proposed DLL.

The timing diagram of Fig. 6 shows the simulated locking process of the proposed digital DLL at 300 MHz. The DLL starts with $M[4:0] = [00010]$ and $L[4:0] = [00000]$ ($= 64$ in decimal). It can be seen that the MSB mode is completed at 481 ns with $M[4:0] = [11000]$ and $L[4:0] = [00000]$ ($= 768$ in decimal) and that the SAR mode is completed at 681 ns with $M[4:0] = [10110]$ and $L[4:0] = [10001]$ ($= 721$ in decimal). In Counter mode, the input clock (CLK_{IN}) and output clock (CLK_{OUT}) of the DLL are precisely aligned with each other with no clock skews. It is assumed that the clock buffer and the replica path are ideal and therefore they are omitted in this simulation.

Fig. 7 shows the simulated peak-to-peak (p-p) jitter of the output clock. The proposed DLL achieves a simulated p-p jitter of 12.03 ps and 9.71 ps at 300 MHz

Fig. 5. Chip layout of the (a) proposed digital DLL (b) DDR3 SDRAM with the proposed DLL.

Fig. 6. Simulated locking process of the proposed digital DLL.

Fig. 7. Simulated peak-to-peak jitter (a) 300 MHz (b) 2 GHz.

and 2.0 GHz, respectively. Fig. 8 shows the simulated p-p jitter of the output clock when there exists a 1 GHz supply noise of 50 mV. Although the p-p jitter has been increased to 46 ps with this injected supply noise, the DLL shows good noise-

Fig. 8. Simulated peak-to-peak jitter at 2 GHz with a 1 GHz supply noise of 50 mV.

Fig. 9. Simulated DCC-equipped DLL operations with distorted input clock duty-cycles (a) Input duty-cycle of 55% at 300 MHz (b) Input duty-cycle of 60% at 2 GHz.

tolerant operation. The DLL operates over a wide frequency range of 0.3–2.0 GHz, which encompasses the operating frequency ranges of both DDR3 and DDR4. The proposed DCC-equipped DLL dissipates 3.48 mW at 1 GHz from a supply voltage of 1.1 V.

In DDR memory systems, the input clock of the DLL may have a duty-cycle distortion (DCD). Therefore, the DLL should be able to lock properly regardless of the input clock's duty-cycle variation. In this design, the PD compares only the rising edges of the input and feedback clocks, and the DCDL controller has no relationship with the input clock's duty-cycle ratio. Therefore, the phase locking operation of the proposed DLL is immune to DCD. Since the output of the DLL clock signal can be distorted due to device mismatches, the proposed DLL adopts a digital DCC [9] to improve performance at high frequencies. The digital DCC achieves a fast locking time of less than 24 input clock cycles and it can be turned off during the power-down mode. The DCC achieves a duty-cycle correction range of 30–70% at 1.0 GHz. Fig. 9 shows the simulated input and output clocks of the proposed DCC-equipped DLL with distorted input clock duty-cycles. As shown in Fig. 9(a), the DLL achieves a corrected output clock duty-cycle of 50.38% from a 55% input duty-cycle at 300 MHz. Fig. 9(b) shows an output duty-cycle of 50.48% from a 60% input duty-cycle at 2 GHz.

A performance comparison between the proposed digital DLL and other state-of-the-art DDR3/DDR4 digital DLLs is given in Table I.

Table I. Performance Summary and Comparison with state-of-the-art DDR3/DDR4 digital DLLs

	[3]	[4]	[5]	[6]	[7]	This work
Process	65 nm	45 nm	0.25 μ m	0.13 μ m	0.18 μ m	38 nm
Architecture	Digital	Digital	Digital	Digital	Digital	Digital
Supply (V)	1.0–1.5	1.1	1.1	1.2	1.8	1.1
Frequency range (GHz)	0.12–2	0.4–0.8	0.1	0.03–1	0.04–0.55	0.3–2.0
Pk-to-pk jitter (ps)	14 @2 GHz	20.4 @800 MHz	95 @100 MHz	30 @1 GHz	16.9 @200 MHz	9.71 @2 Hz
Duty-cycle Correction	O	O	X	X	O	O
Locking time	-	-	30 cycles	42 cycles	134 cycles	88 cycles
Power	6.6 mW @1.2 V, 2 GHz	3.3 mW @800 MHz	3.3 mW @100 MHz	3.6 mW @1 GHz	9 mW @200 MHz	3.48 mW @1 GHz
Active area (mm ²)	0.059	0.01	0.136	0.2	0.2	0.02

4 Conclusion

This Letter presents a new digital DLL architecture which is capable of duty-cycle correction and harmonic-free, fast-locking in DDR3 and DDR4 SDRAMs. The proposed DLL achieves these capabilities by adopting a new noise-tolerant triple (MSB-interval + binary + sequential) search algorithm. Implemented in a 38-nm Powerchip CMOS DRAM process, the proposed DLL operates over a frequency range of 0.3–2.0 GHz, achieves a peak-to-peak jitter of 7.78 ps, and dissipates 3.48 mW from a 1.1 V supply at 1 GHz. And the proposed DCC-equipped DLL occupies an active area of just 0.02 mm².

Acknowledgments

This work (C0396252) was supported by Business for Cooperative R&D between Industry, Academy, and Research Institute funded Korea Small and Medium Business Administration in 2016. The EDA tools were provided by IDEC, Korea. This work was also supported by the Korea Institute for Advancement of Technology (KIAT) grant funded by the Korean government (Motie: Ministry of Trade, Industry & Energy, HRD Program for Software-SoC convergence, No. N0001883).