analog

MDPI

Article

Delay Locked Loop Based on Sawtooth Waveforms †

Andres Ayes * and Eby G. Friedman

Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY 14627, USA; friedman@ece.rochester.edu

* Correspondence: aayes@ece.rochester.edu

† This paper is an extended version of our paper published in the 2023 IEEE 36th International System-on-Chip Conference (SOCC), Santa Clara, CA, USA, 5–8 September 2023: "Dual Sawtooth-Based Delay Locked Loops for Heterogeneous 3-D Clock Networks", pp. 246–250.

Abstract

Reliable timing is a crucial issue in all synchronous systems. Delay locked loops are capable of dynamically synchronizing clock signals; the increasing speed of deeply scaled technologies however leads to long and complex delay lines. In this paper, a sawtooth-based delay locked loop is proposed to address the increasing difficulty of delay generation in high speed systems. The proposed architecture replaces a conventional delay line with a sawtooth waveform-based mechanism for delay generation, reducing the need for numerous delay elements. The timing offset is mapped to a voltage level on the sawtooth waveform, where the required delay is the time to cross this voltage level. The architecture, evaluated using a 7 nm device model, achieves a locking speed as low as four cycles for a 1 GHz clock signal. The DLL achieves a full period locking range, lowers the clock skew to 16 ps at room temperature, and exhibits 65 ps clock skew variations over extreme temperature corners.

Keywords: synchronization; clocking; delay locked loop; sawtooth waveforms; timing

1. Introduction

Synchronous systems are dependent on accurate and reliable distribution of the clock signal. Different topologies and design methodologies for clock distribution networks have been explored [1–6]. The introduction of 3-D integration and multi-clock domains, however, demand dynamic timing solutions [7–9]. Phase compensation circuits are used to reduce the effects of variations on timing. These circuits can be used to compare and align two clock signals. Delay locked loops (DLLs) are a type of compensation circuit that has recently attracted significant research attention [10–20].

DLLs compare two clock signals and compensate for any accumulated clock skew caused by process, voltage, and temperature (PVT) variations. This circuit function enables precise point-to-point temporal alignment since these circuits synchronize the arrival time of the clock signal at a clock leaf rather than the source of the clock, the clock root [1,3]. A distributed DLL strategy where the DLLs are spread throughout the system can guarantee precise synchronization at the local level. An important component of DLLs is controllable delay lines, which require multiple delay elements. These delay elements increase in speed as technology scales. The higher speed of these elements leads to either a small delay range of the DLL or an excessive number of delay elements. To address this concern, a DLL with sawtooth waveforms is proposed to replace a traditional delay line. The DLL aligns two

check for updates

Received: 10 September 2025
Revised: 5 November 2025
Accepted: 7 November 2025
Published: 24 November 2025

Citation: Ayes, A.; Friedman, E.G. Delay Locked Loop Based on Sawtooth Waveforms. *Analog* **2026**, *1*, 1. https://doi.org/10.3390/analog1010001

clock signals after a few cycles, reducing the clock skew despite a wide range of initial skew. The DLL also includes a correction circuit to prevent output errors.

This paper is organized as follows: a brief overview of DLLs is presented in Section 2. The operating principle of the proposed DLL is described in Section 3. The circuit architecture is described in Section 4. The performance is reviewed in Section 5. A strategy for achieving programmable delays is discussed in Section 6. Some conclusions are offered in Section 7.

2. Background on Delay Locked Loops

Phase compensation circuits can be used for dynamic synchronization. Delay locked loops are a desirable candidate for phase compensation due to high speed and good linearity [3]. An illustration of a DLL synchronizing a clock leaf with respect to a reference clock is shown in Figure 1. DLLs align a reference clock signal with a feedback clock signal by introducing delay into the clock path. The locking time is the time required to determine the appropriate delay produced by the DLL to synchronize the input and feedback signals. DLLs are particularly useful in 3-D systems where the short vertical interconnect path between clock signals can be exploited to accurately compensate for inter-layer variations [16,17,21,22].

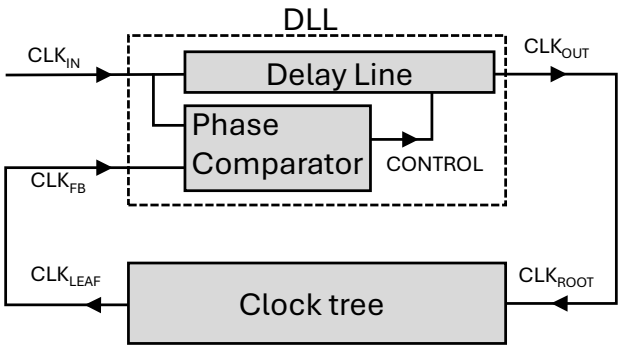

Figure 1. Delay locked loop synchronizing a clock leaf and input clock signal.

DLLs typically consist of a phase detector, charge pump, loop filter, and controllable delay line [3]. In analog DLLs, a control voltage is used to determine the delay of the delay line. Digital DLLs use a digital code to extract the appropriate delayed output signal [10–14,16,17]. Techniques to enhance the locking time have been proposed, such as successive approximation register (SAR) algorithms and flash time-to-digital conversion (TDC) [10–14].

As mentioned in Section 1, a challenge in DLLs is the temporal range, i.e., the delay provided by the delay line. Deeply scaled technologies typically increase the speed of the individual delay elements. DLLs therefore require additional circuitry to achieve a target delay. A DLL that uses a sawtooth waveform in the delay line was proposed in [15]. While the DLL described in [15] addresses temporal range challenges, the method of locking delay relies on gradually changing the slope and control voltage. This method can lead to long locking times, depending upon the initial clock skew. In this work, a DLL architecture that delays the input clock signal based on differential sawtooth waveforms rather than traditional delay lines is presented. This DLL exploits the inherent timing relationship between a clock input and a sawtooth signal to reduce the locking time to a few cycles.

3. Operating Principles of the Sawtooth DLL

The objective of the proposed DLL is to rapidly extract the required timing information to achieve near zero clock skew using sawtooth waveforms. The operating principle of this

approach is described in this section. Understanding how to extract the timing relationship between an input clock signal and a sawtooth waveform motivates the circuit components described in Section 4.

To illustrate this behavior, assume an input clock signal CLK_{IN} with a 50% duty cycle and period T_{period}. The clock signal also has an associated sawtooth waveform. The sawtooth waveform rises when CLK_{IN} is high and falls when CLK_{IN} is low, as illustrated in Figure 2. Assume that the clock skew between the input and feedback signals is T_1. To successfully attenuate the clock skew, the DLL must introduce a delay of $T_{period} - T_1$ into the clock path. This delay ensures that the feedback signal arrives at a multiple of T_{period}, effectively synchronizing the clock signals. The sawtooth signal provides the timing information to generate this delay. To extract this information, the initial clock skew is encoded as a voltage level, referred to as V_{offset}. At T_1, the DLL saves the voltage of the sawtooth waveform. A method for saving this voltage is described in Section 4.1. In the example shown in Figure 2, the sawtooth waveform is rising at T_1. The time when the sawtooth waveform crosses the same voltage level in the opposite direction (falling edge) is referred to as T_2. The relationship between T_1 and T_2 is described by the following expressions,

$$T_2 = T_1 + 2(T_{period}/2 - T_1) = T_{period} - T_1, \quad for\ T_1 < T_{period}/2, \tag{1}$$

$$T_2 = T_1 + 2(T_{period} - T_1) = 2T_{period} - T_1, \quad for\ T_1 > T_{period}/2. \tag{2}$$

In both cases, the crossing occurs at the same time as the delay needed to compensate for the initial clock skew. Note that this sawtooth waveform only provides the delay of the rising edge. A second sawtooth waveform is required to align the falling edge while maintaining the duty cycle of the input clock signal.

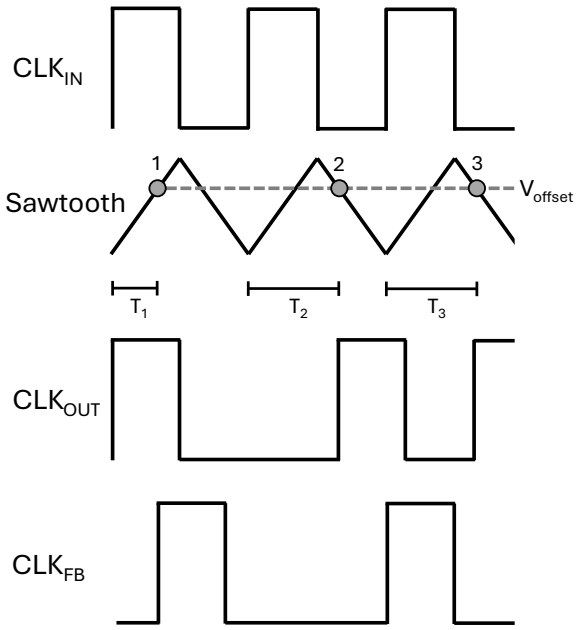

Figure 2. Sawtooth DLL waveform depicting the ideal timing relationship between the reference clock signal, DLL output (CLK_{OUT}), and feedback clock signal. Note that the sawtooth waveform that provides the delay information of the rising edge is shown.

Three steps are involved in this operation: (1) generation of the sawtooth waveform, (2) conversion of the initial clock skew into a voltage level, and (3) production of an output signal when the sawtooth waveform crosses the voltage level in the opposite direction. The

possibility that the voltage level is too high or too low must also be considered. In these cases, the DLL does not detect a crossing. A correction stage identifies these occurrences and produces an appropriate signal that minimizes the clock skew. The circuit architecture and components used to realize this operation are discussed in the following section.

4. Circuit Architecture of Sawtooth DLL

The architecture and components of the sawtooth DLL to realize the behavior described in Section 3 are discussed in this section [21]. A block diagram of the DLL is shown in Figure 3. In the sawtooth generation stage, a pair of sawtooth waveforms is produced to extract the timing information from the rising edge and falling edge waveforms. The phase comparator monitors the phase difference between the input and feedback signals. The waveforms are passed to the crossing detection stage, which uses differential amplifiers to detect a crossing between the sawtooth waveform and the offset voltage. The output generation stage produces the output signal based on detection of a crossing. If the offset voltages are too high or too low, the correction stage senses the lack of a crossing and determines the appropriate output of the DLL that minimizes the clock skew. A more complete description of each stage is described in the following subsections.

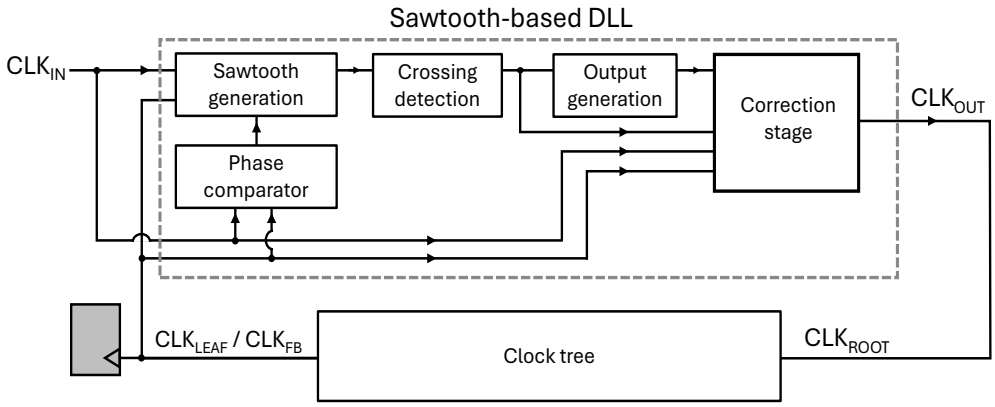

Figure 3. Block diagram of the proposed sawtooth DLL.

4.1. Sawtooth Generation Stage

The relationship between the sawtooth waveforms and the input clock signal is crucial to the operation of the DLL. The timing information is encoded as an offset voltage within the sawtooth voltage waveform. To achieve the relationship illustrated in Figure 2, a pair of differential integrators [23] is used. These circuits charge and discharge two capacitors. The voltage across each capacitor linearly increases and decreases depending upon the clock phase.

The integrators are also responsible for generating the offset voltage V_{offset}. As described in Section 3, V_{offset} is a measure of the initial clock skew. The circuits generate this voltage by turning off those transistors that supply charge to one of the capacitors upon arrival of the first feedback pulse. When the transistors are turned off, the capacitors are isolated and maintain the voltage. Capacitor C_2 is illustrated in Figure 4a, where the voltage is related to the arrival time of the falling edge. C_3 is depicted in Figure 4b, where the voltage is related to the arrival time of the rising edge.

The timing relationship between the input, output, feedback, and sawtooth signals is illustrated in Figure 5. C_3 is isolated when the first rising edge of the feedback signal arrives, and C_2 is isolated when the first falling edge of the feedback signal arrives. Note that the voltage across C_1 and C_3 is used to extract the timing information of the rising edge, and the voltage across C_2 and C_4 is used to extract the timing information of the falling edge since these signals have the same polarity (C_1 and C_3 both discharge when

the clock is high). The timing of the output signals is set by the crossing of the sawtooth waveforms and V_{offset}. This method is achieved by the crossing detection and output generation stages.

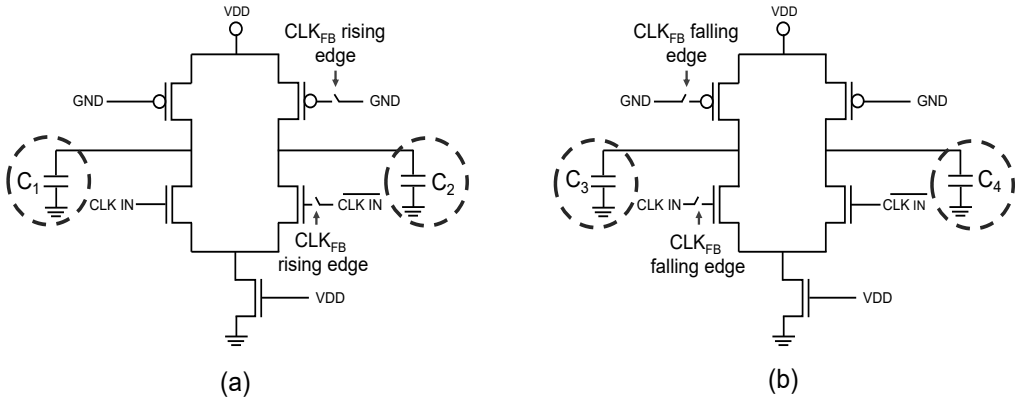

Figure 4. Integrators with differential output, (**a**) when the rising edge of CLK_{FB} arrives, capacitor C_2 is no longer charged or discharged by the integrator, maintaining the offset voltage, and (**b**) when the falling edge of CLK_{FB} arrives, capacitor C_3 is no longer charged or discharged, maintaining the offset voltage.

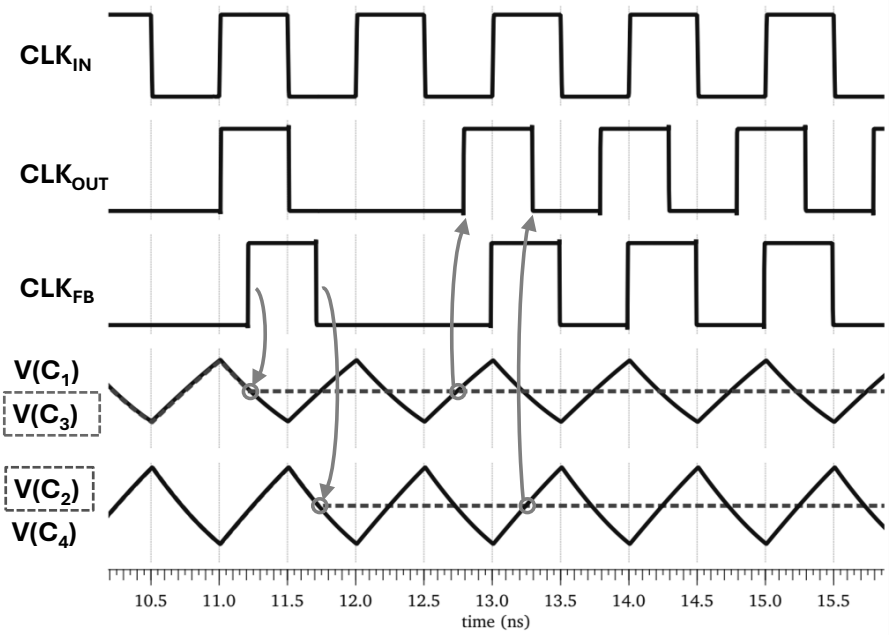

Figure 5. Timing diagram of relevant waveforms. The clock skew between CLK_{IN} and CLK_{FB} is stored as a pair of offset voltages, $V(C_3)$ and $V(C_2)$. $V(C_3)$ corresponds to the rising edge clock skew, and $V(C_2)$ corresponds to the falling edge clock skew. When the sawtooth waveform crosses the offset voltage in the opposite direction, the output signal is generated with an appropriate delay to compensate for the initial clock skew.

4.2. Phase Comparator

Since V_{offset} is stored as charge on a capacitor, the voltage leaks over time, degrading the accuracy of the sawtooth DLL. The effect of leakage current on the accuracy is quantified in Section 5. Duty cycle distortion on the input can also affect the accuracy of V_{offset}. Once the clock skew between the input and feedback signals reaches a timing threshold (70 ps in this example), the DLL must resynchronize the input and feedback signals by repeating the locking process, as illustrated in Figure 5. The threshold is selected to exceed the worst case skew. If frequent relocking events occur due to persistent values of clock skew

close to the timing threshold, the threshold of the phase comparator can be increased to reduce the relocking rate at the cost of a larger clock skew margin. The phase comparator, shown in Figure 6, detects a phase difference between the input and output signals. The D flip flop requires a 70 ps pulse at the clock edge to latch a high state, setting the timing threshold. This latching event occurs if the input signal is either leading or lagging behind the feedback signal by 70 ps. Once the flip flop latches a high state, the sawtooth DLL repeats the locking process, as illustrated in Figure 5.

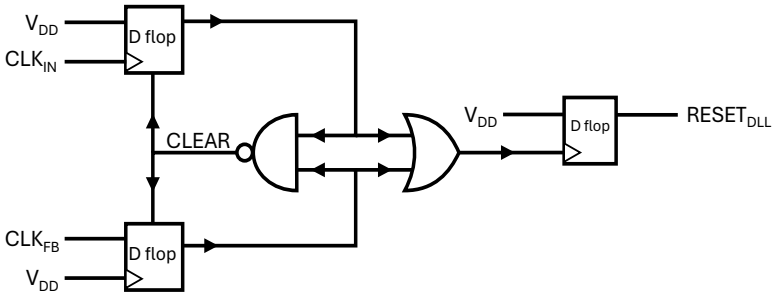

Figure 6. Phase comparator.

4.3. Crossing Detection and Output Generation Stages

The DLL generates a delayed version of the input clock signal based on the timing information provided by the sawtooth waveform. These stages, illustrated in Figure 7, ensure that the output of the DLL is generated at the correct time (i.e., T_2, see Figure 2). If the sawtooth waveform of the rising edge crosses the offset voltage, the output of the DLL transitions high. Alternatively, if the sawtooth waveform of the falling edge crosses the offset voltage, the output of the DLL transitions low. The crossing detection stage is composed of a differential amplifier and a pulse generator. The output generation stage is composed of a D flip flop. The differential amplifiers determine when the sawtooth waveform crosses V_{offset}. Once a crossing has been detected, a small pulse is generated to provide sufficient hold time for the flip flop. This flip flop functions as the output generation stage. A pulse originating from the rising sawtooth causes the output of the flip flop to transition high. A pulse originating from the falling edge clears the flip flop, causing the output to transition low. Note that the output of this stage is OUT and is not the same signal as CLK_{OUT} shown in Figures 3 and 5. The DLL must consider the possibility that no crossing is detected, which is the purpose of the correction stage.

4.4. Correction Stage

Since the sawtooth DLL generates a delay based on the signal crossing V_{offset}, a potential hazard arises when the level detectors are unable to sense this crossing. This hazard occurs when the offset voltage V_{offset} is close to one of the peaks of the sawtooth waveform. These peaks are a result of CLK_{IN} transitioning between high and low. This information is exploited to deduce the initial clock skew when a crossing is not detected.

An offset voltage close to the maximum peak voltage ensures that the initial clock skew is approximately T_{period}, while an offset voltage close to the minimum peak voltage ensures that the clock skew is approximately $T_{period}/2$. To interpret the initial clock skew, this stage must (1) sense that at least one of the crossing detectors is not generating a pulse, (2) deduce whether the clock skew is T_{period} or $T_{period}/2$, and (3) pass either CLK_{IN} or $\overline{CLK_{IN}}$ to the output. The rest of this section describes how the circuit depicted in Figure 8 achieves these objectives.

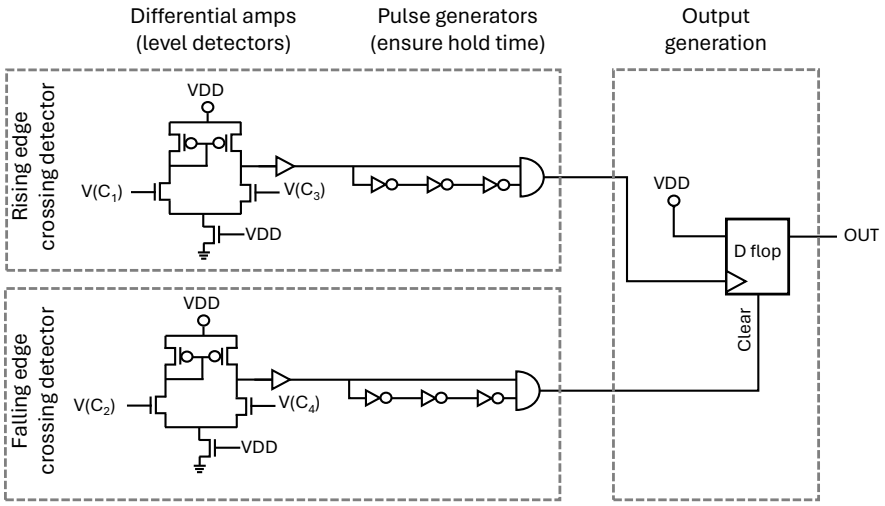

Figure 7. Crossing detection and output generation stages.

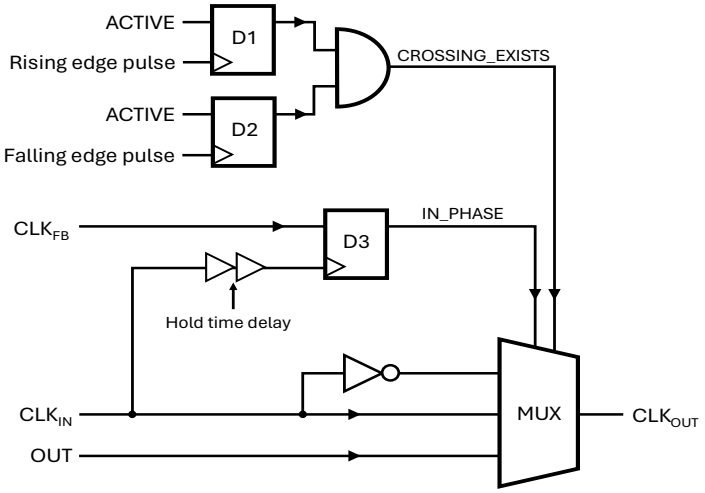

Figure 8. Correction stage circuit. This stage evaluates the possibility that no crossing is detected and selects the appropriate output.

The first step is to sense whether a pulse is produced by the crossing detectors. This step is achieved using a pair of flip flops, D1 and D2, shown in Figure 8, that evaluates the output of the crossing detectors. If either flip flop fails to detect a pulse, V_{offset} is assumed to be near one of the voltage peaks, and the initial clock skew is considered to be either T_{period} or $T_{period}/2$. To avoid a potential glitch when the DLL starts, an ACTIVE signal rather than V_{DD} is used as the input of D1 and D2. The ACTIVE signal is logically high within a time window that maintains a fast locking time while avoiding miscellaneous pulses that may occur during start-up.

The second step is to determine the correct clock skew. A separate flip flop D3 compares CLK_{IN} and CLK_{FB}. The input clock signal is sufficiently delayed to satisfy any hold time constraints. If the output of D3 transitions high, CLK_{IN} and CLK_{FB} are in phase, and the clock skew is approximately T_{period}. Otherwise, the clock signals have opposite phase, and the clock skew is approximately $T_{period}/2$.

The third step is to produce the appropriate output signal. A three input multiplexer selects the output based on the results of the previous steps. If the crossings are successfully detected, the OUT signal from the output generation stage is selected. If, however, the crossings are not detected, the multiplexer selects CLK_{IN} if the input and feedback signals are in phase or $\overline{CLK_{IN}}$ if the signals are out of phase.

5. Performance of Sawtooth DLL

The locking range, performance at extreme temperatures, jitter, and accuracy over time of the proposed sawtooth DLL are discussed in this section. The sawtooth DLL is evaluated with a 7 nm predictive technology model [24]. The period of the input clock signal is assumed to be 1 ns. Power noise characteristics are considered, where the following power and ground network impedances are assumed: $R_{PG} = 1.3\ \Omega$, $L_{PG} = 20$ nH, and $C_{PG} = 88$ pF [25].

The reduction in clock skew for different clock tree delays is shown in Figure 9. The DLL successfully attenuates the clock skew between the input and feedback path across a delay range that spans the clock period. When the initial clock skew is within ±25 ps of $T_{period}/2$ (475 to 525 ps) or within ±25 ps of T_{period} (above 975 ps or below 25 ns), the crossing detection stages do not produce a valid pulse. In these cases, the correction stage determines the clock skew and passes the appropriate signal, CLK_{IN} or $\overline{CLK_{IN}}$, to the output.

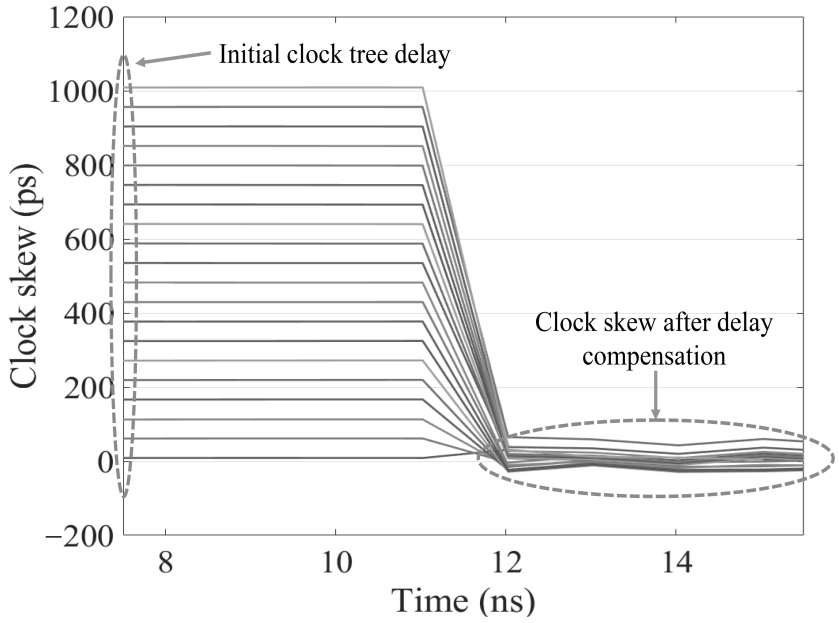

Figure 9. Attenuation of clock skew within range of the clock period (1 ns).

The performance of the DLL at mitigating an initial clock skew of 250 ps is evaluated at −50 °C, 27 °C, and 125 °C. The input and feedback clock signals are shown in Figure 10 for the three temperatures. Despite extreme temperatures and power supply noise, the timing relationship described in Section 3 is maintained, and the clock skew between the input and feedback signals is reduced. The DLL lowers the clock skew to 16 ps at room temperature. A 65 ps variation in clock skew exists between the extreme temperature corners. The circuit exhibits an average power dissipation of 512 µW at −55 °C, 618 µW at 27 °C, and 935 µW at 125 °C.

The jitter characteristics of the feedback signal are illustrated in the eye diagram shown in Figure 11. The feedback signal exhibits a 59 ps peak-to-peak jitter for a 1 GHz clock signal.

Due to leakage current from the capacitors, the offset voltage degrades over time. This loss of charge affects the accuracy of the DLL. The clock skew between the input clock signal and the feedback clock signal accumulates at a rate that depends upon the type of capacitor. To compensate for this skew accumulation, the DLL periodically repeats the locking process to synchronize CLK_{IN} and CLK_{FB}. This process is akin to the replenishment of charge for DRAM cells [26]. The locking process is repeated when the phase comparator, described

in Section 4.2, detects a significant skew between CLK_{IN} and CLK_{FB}. The frequency of repeating the locking process depends upon the temperature and type of capacitor. The relationship between the accumulated clock skew and leakage current is shown in Figure 12. High-K trench capacitors exhibit a low leakage current, below 10 nA [27], while a through silicon via with a 3-D metal-insulator-metal (MIM) capacitor can exhibit a leakage current above 10 nA [28]. These characteristics are illustrated in Figure 12 over the same temperature range. The accumulation of clock skew at low temperatures is almost negligible with high-K trench capacitors. The accumulation is, however, more significant at higher temperatures and with 3-D MIM capacitors. Since the accumulation of clock skew is around tens of femtoseconds per cycle at room temperature for high-k trench capacitors, thousands of cycles are required before the phase comparator detects a significant clock skew of around 70 ps.

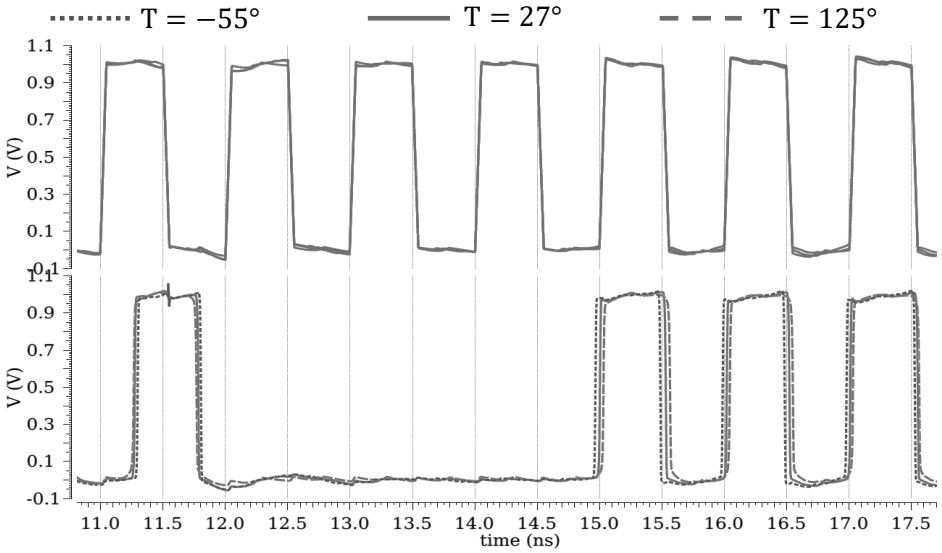

Figure 10. Performance of the sawtooth DLL at extreme temperature corners.

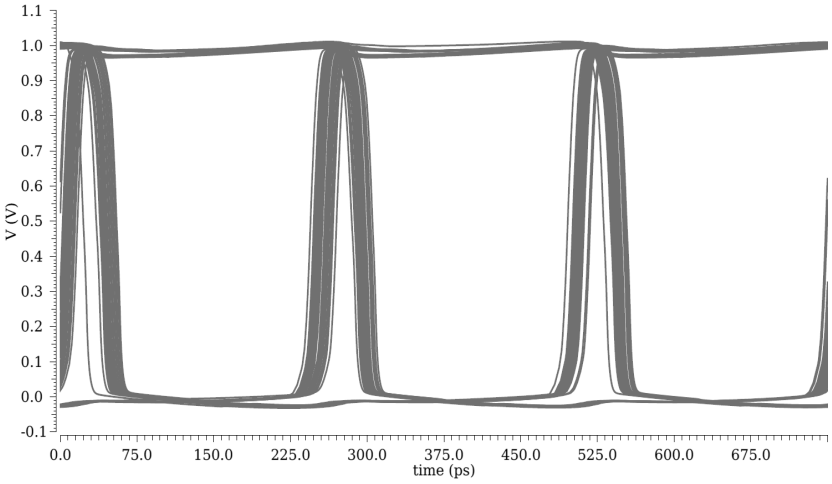

Figure 11. Eye diagram of the feedback clock signal.

A comparison of the proposed architecture with other DLLs is provided in Table 1. The sawtooth DLL locks the delay within four cycles. To the authors' knowledge, the fewest cycles achieved in published work is three cycles [12]. The power consumption is also relatively low at 1 GHz. This advantage, however, is due to the advanced technology node. The architecture of the delay lines employed in each DLL is also listed. The architectures are categorized into digitally controlled delay lines (DCDLs), voltage controlled

delay lines (VCDLs), digitally controlled oscillators (DCOs), and sawtooth signals. These architectures require multiple cascaded delay elements to maintain phase accuracy and increase the number of delay stages to extend the locking range. The proposed DLL circuit topology achieves a full period locking range using two integrators and four capacitors to generate a continuous delay. Note that the performance of the sawtooth DLL is based on a 7 nm technology, while published DLLs are based on 40 nm to 130 nm technology nodes. This technology difference affects the reported power metrics; however, the underlying operating principle is technology independent, and the performance results provide a basis for comparison.

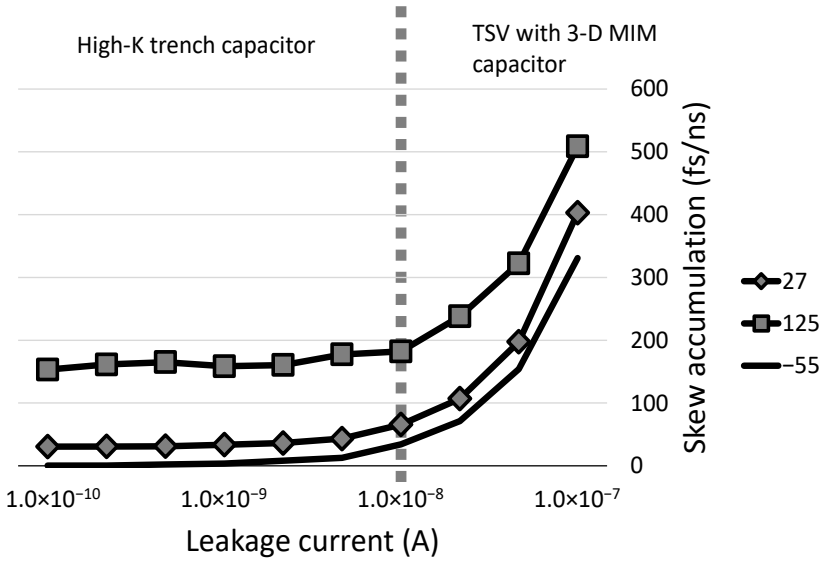

Figure 12. Effect of leakage current on skew accumulation.

Table 1. Comparison among the proposed DLL and other published DLL examples.

	[10]	[11]	[12]	[13]	[15]	[18]	[19]	[20]	**This Work**
CMOS Technology (nm)	130	65	40	65	180	45	90	40	7
Supply voltage (V)	1.2	1.0	N/A	N/A	1.5	N/A	1.0	1.1	1.0
Delay line architecture	DCDL	DCDL	DCDL	DCDL	Single Sawtooth	DCDL	Hybrid DCDL/VCDL	DCO	Dual Sawtooth
All-digital	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	No
Power (mW)	1.15	7.1	12	0.19	6	17	49.4	3.56	0.62
Frequency (GHz)	1.5	7.0	3.2	0.11	0.05–0.5	4.25	0.1–2.7	1.6–3.2	1.0
Locking time (clock cycles)	32	6	3	6	N/A	100	143	6	4
Experiment (E) or simulation (S)	S	E	E	E	E	E	E	E	S

A key disadvantage of the proposed DLL is the need to repeat the locking process when the phase comparator detects a significant phase difference. The DLL has a fast locking time, but the need to repeat the process more often than other DLLs is an aspect

that can be improved. A potential solution to mitigate this disadvantage is to incorporate V_{offset} into a more stable bandgap reference voltage [29]. If the bandgap circuit supports a controllable reference voltage, programmable delay can be introduced into the system.

6. Programmable Delay

Programmable delay is a useful capability for clock network design and synthesis since certain data paths can benefit from useful skew [3,4]. The DLL presented in this paper locks the delay with the objective of near zero clock skew. To enable dynamic control of the skew between the input and feedback signals, delay information, stored on the capacitor as V_{offset}, must be controlled. This approach can be realized by a programmable bandgap reference voltage. The circuit copies V_{offset} from the capacitor after completion of the locking process. The bandgap voltage sets a suitable delay by increasing or decreasing the voltage, as illustrated in Figure 13. The voltage range of the sawtooth signals should be sufficiently large to support fine tune control of V_{offset}. In the 7 nm predictive technology, the voltage range is between 200 and 400 mV, which poses a challenge on the bandgap voltage. More mature technologies can exploit a larger supply voltage and lower current requirements to achieve a wider voltage range for the sawtooth DLL.

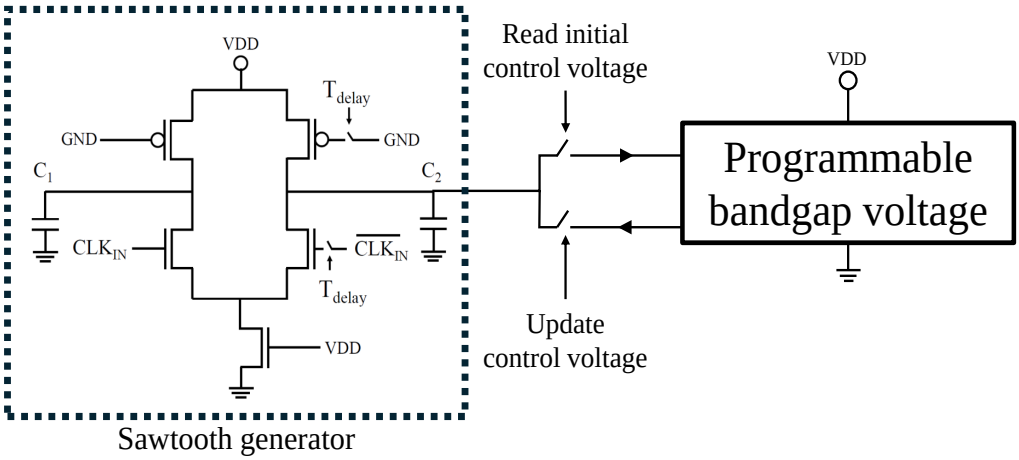

Figure 13. Dynamic control of V_{offset} with a programmable bandgap voltage.

7. Conclusions

A differential sawtooth DLL is described in this paper. The DLL replaces a traditional delay line with a pair of sawtooth waveforms. The DLL compensates for clock skew based on a voltage level generated upon arrival of a feedback signal. The DLL achieves a full period locking range, exhibiting a range of 16 ps clock skew at room temperature and a 65 ps variation in clock skew at extreme temperatures. The input and feedback signals are synchronized after four cycles, assuming the initial clock skew is less than the period. Due to leakage current, the DLL must repeat the locking process after thousands of cycles. The frequency of repeating the locking process depends upon the type of capacitors used to store the voltage level. Future work includes reducing the effects of leakage current on the accuracy of the DLL and the introduction of a programmable bandgap voltage. The bandgap voltage provides a more stable voltage reference, reduces the frequency of repeating the locking process, and enables programmable delay to support useful skew between the input and feedback clock signals.

Author Contributions: A.A. wrote the main manuscript text and prepared all of the figures. E.G.F. supervised and advised on the development of the work and reviewed the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported in part by the National Science Foundation under Grant No. 2124453 (DISCOVER Expeditions) and Grant No. 2308863 and the Department of Energy under FOA number DE-FOA-0002950.

Data Availability Statement: All data supporting the findings of this work are included within the manuscript.

Acknowledgments: The authors thank the reviewers for their valuable comments and suggestions, which improved the quality of the paper.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Friedman, E.G. Clock Distribution Design in VLSI Circuits—An Overview. In Proceedings of the IEEE International Symposium on Circuits and Systems, Chicago, IL, USA, 3–6 May 1993; pp. 1475–1478.
2. Friedman, E.G. Clock Distribution Networks in Synchronous Digital Integrated Circuits. *Proc. IEEE* **2001**, *89*, 665–692. [CrossRef]
3. Salman, E.; Friedman, E. *High Performance Integrated Circuit Design*; McGraw-Hill Professional: Columbus, OH, USA, 2012.
4. Kourtev, I.S.; Taskin, B.; Friedman, E.G. *Timing Optimization Through Clock Skew Scheduling*, 2nd ed.; Springer Science+Business Media: Berlin/Heidelberg, Germany, 2009.
5. Inoue, K.; Kaneko, M. Reliable and Low-Power Clock Distribution using Pre- and Post-Silicon Delay Adaptation in High-Level Synthesis. In Proceedings of the IEEE International Symposium on Circuits and Systems, Seoul, Republic of Korea, 20–23 May 2012; pp. 1664–1667.
6. Pavlidis, V.F.; Savidis, I.; Friedman, E.G. Clock Distribution Networks for 3-D Integrated Circuits. *IEEE Trans. Very Large Scale Integr. VLSI Syst.* **2011**, *19*, 2256–2268. [CrossRef]
7. Krstic, M.; Grass, E.; Gürkaynak, F.K.; Vivet, P. Globally Asynchronous, Locally Synchronous Circuits: Overview and Outlook. *IEEE Des. Test Comput.* **2007**, *24*, 430–441. [CrossRef]
8. Xu, H.; Pavlidis, V.F.; De Micheli, G. Skew Variability in 3-D ICs with Multiple Clock Domains. In Proceedings of the IEEE International Symposium of Circuits and Systems, Rio de Janeiro, Brazil, 15–18 May 2011; pp. 2221–2224.
9. Pavlidis, V.F.; Savidis, I.; Friedman, E.G. *Three-Dimensional Integrated Circuit Design*, 2nd ed.; Morgan Kaufmann: Burlington, MA, USA, 2017.
10. Quchani, M.E.; Maymandi-Nejad, M. Design of a Low-Power Linear SAR-Based All-Digital Delay-Locked Loop. In Proceedings of the Iranian Conference on Electrical Engineering, Yazd, Iran, 30 April–2 May 2019; pp. 118–124.
11. Park, D.; Kim, J. A 7-GHz Fast-Lock 2-Step TDC-Based All-Digital DLL for Post-DDR4 SDRAMs. In Proceedings of the IEEE International Symposium on Circuits and Systems, Florence, Italy, 27–30 May 2018; pp. 1–4.
12. Hossain, M.; Aquil, F.; Chau, P.S.; Tsang, B.; Le, P.; Wei, J.; Stone, T.; Daly, B.; Tran, C.; Eble, J.C.; et al. A Fast-Lock, Jitter Filtering All-Digital DLL Based Burst-Mode Memory Interface. *IEEE J. Solid-State Circuits* **2014**, *49*, 1048–1062. [CrossRef]
13. Chae, J.H.; Hong, G.M.; Park, J.; Kim, M.; Ko, H.; Shin, W.Y.; Chi, H.; Jeong, D.K.; Kim, S. A 1.74 mW/GHz 0.11–2.5 GHz Fast-Locking, Jitter-Reducing, 180 Phase-Shift Digital DLL with a Window Phase Detector for LPDDR4 Memory Controllers. In Proceedings of the IEEE Asian Solid-State Circuits Conference, Xia'men, China, 9–11 November 2015; pp. 1–4.
14. Zhang, D.; Yang, H.G.; Zhu, W.; Li, W.; Huang, Z.; Li, L.; Li, T. A Multiphase DLL with a Novel Fast-Locking Fine-Code Time-to-Digital Converter. *IEEE Trans. Very Large Scale Integr. VLSI Syst.* **2014**, *23*, 2680–2684. [CrossRef]
15. Chen, C.C.; Liu, S.I. An Infinite Phase Shift Delay-Locked Loop with Voltage-Controlled Sawtooth Delay Line. *IEEE J. Solid-State Circuits* **2008**, *43*, 2413–2421. [CrossRef]
16. Chung, C.C.; Hou, C.Y. All-Digital Delay-Locked Loop for 3D-IC Die-to-Die Clock Synchronization. In Proceedings of the IEEE International Symposium on VLSI Design, Automation and Test, Hsinchu, Taiwan, 28–30 April 2014; pp. 1–4.
17. Sadi, M.; Kannan, S.; England, L.; Tehranipoor, M. Design of a Digital IP for 3D-IC Die-to-Die Clock Synchronization. In Proceedings of the IEEE International Symposium on Circuits and Systems, Baltimore, MD, USA, 28–31 May 2017; pp. 1–4.
18. Rehman, S.U.; Khafaji, M.M.; Ferschischi, A.; Carta, C.; Ellinger, F. A 0.2-1.3 ns Range Delay-Control Scheme for a 25 Gb/s Data-Receiver Using a Replica Delay-Line-Based Delay-Locked-Loop in 45-nm CMOS. *IEEE Trans. Circuits Syst. II Express Briefs* **2020**, *67*, 806–810. [CrossRef]
19. Tsai, C.W.; Chiu, Y.T.; Tu, Y.H.; Cheng, K.H. A Wide-Range All-Digital Delay-Locked Loop for DDR1–DDR5 Applications. *IEEE Trans. Very Large Scale Integr. VLSI Syst.* **2021**, *29*, 1720–1729. [CrossRef]
20. Jin, J.; Kim, S.; Kim, J. A Fast-Lock All-Digital Clock Generator for Energy Efficient Chiplet-Based Systems. *IEEE Access* **2022**, *10*, 124217–124226. [CrossRef]
21. Ayes, A.; Friedman, E.G. Dual Sawtooth-Based Delay Locked Loops for Heterogeneous 3-D Clock Networks. In Proceedings of the IEEE International System-On-Chip Conference, Santa Clara, CA, USA, 5–8 September 2023; pp. 246–250.

22. Ayes, A.; Friedman, E.G. Linear Clock Tree Topology for Dynamic Source Synchronous and Fully Synchronous 3-D Interfaces. *Integr. VLSI J.* **2023**, *93*, 102066. [CrossRef]
23. Smilkstein, T.H. Jitter Reduction on High-Speed Clock Signals. Ph.D. Thesis, University of California, Berkley, CA, USA, 2007.
24. Clark, L.T.; Vashishtha, V.; Harris, D.M.; Dietrich, S.; Wang, Z. Design Flows and Collateral for the ASAP7 7nm FinFET Predictive Process Design Kit. In Proceedings of the IEEE International Conference on Microelectronic Systems Education, Lake Louise, AB, Canada, 11–12 May 2017; pp. 1–4.
25. Lu, T.C.; Fu, C.M.; Liao, C.C.; Lin, Y.T.; Chang, C.H.; Hsieh, K. A Cost-Effective On-Chip Power Impedance Measurement (PIM) System in 7nm FinFET for HPC Applications. In Proceedings of the Symposium on VLSI Circuits, Virtual, 13–19 June 2021; pp. 1–2.
26. Hoffman, W.K.; Kalter, H.L. An 8K b Random-Access Memory Chip using the One-Device FET Cell. *IEEE J. Solid-State Circuits* **1973**, *8*, 298–305. [CrossRef]
27. Burke, M.; Blake, A.; Djara, V.; O'Connell, D.; Povey, I.M.; Cherkaoui, K.; Monaghan, S.; Scully, J.; Murphy, R.; Hurley, P.K.; et al. High Aspect Ratio Iridescent Three-Dimensional Metal–Insulator–Metal Capacitors using Atomic Layer Deposition. *J. Vac. Sci. Technol. A* **2015**, *33*, 01A103. [CrossRef]
28. Lin, Y.; Tan, C.S. Through-Substrate Via (TSV) with Embedded Capacitor as an On-Chip Energy Storage Element. In Proceedings of the IEEE International 3D Systems Integration Conference, San Francisco, CA, USA, 8–11 November 2016; pp. 1–4.
29. Brokaw, A.P. A Simple Three-Terminal IC Bandgap Reference. *IEEE J. Solid-State Circuits* **1974**, *9*, 388–393. [CrossRef]