

UC Davis

UC Davis Electronic Theses and Dissertations

Title

A Front-End Sampling ASIC with On-Chip DLL in 65 nm CMOS to Facilitate Low-Jitter Waveform Digitization for XFEL Research

Permalink

<https://escholarship.org/uc/item/4gh866pv>

ISBN

9798189225703

Author

McHale, Sean Patrick

Publication Date

2026-06-25

Peer reviewed|Thesis/dissertation

A Front-End Sampling ASIC with On-Chip DLL in 65 nm CMOS to Facilitate Low-Jitter Waveform
Digitization for XFEL Research

By

SEAN MCHALE
THESIS

Submitted in partial satisfaction of the requirements for the degree of

MASTER OF SCIENCE

in

Electrical and Computer Engineering

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Rajeevan Amirtharajah, Chair

Stephen Lewis

Carl Grace

Committee in Charge

2026

Abstract

X-ray free electron lasers (XFELs) have enabled cutting-edge research across disciplines such as structural biology, ultrafast chemistry, condensed matter, plasma materials science, and nuclear physics. Cavity-based XFELs (CBXFELs) represent the next generation of these systems, offering significant performance improvements but requiring extreme precision in beam positioning. This imposes stringent demands on diagnostic platforms in terms of spatial and temporal resolution.

Diamond-based quadrant sensors have emerged as a promising solution for high bandwidth positional beam sensing. However, fully leveraging these capabilities requires a precise and coherent method for timing the measurement of each element in the quadrant array.

This thesis outlines the design of an application-specific integrated circuit (ASIC) frontend for digitization of the outputs of a diamond quadrant sensor. Fabricated in TSMC 65 nm CMOS, it incorporates an on-chip delay-locked loop (DLL) to generate multi-phase clock signals, enabling channelized sampling with picosecond-scale resolution while minimizing the distortion induced by time-domain jitter and layout mismatches. Additionally, a complementary design for a reference signal generator implemented on the carrier printed circuit board (PCB) provides a properly conditioned 1GHz low-jitter reference signal for driving the on-chip DLL.

The combined ASIC and diamond sensor platform facilitates high-precision timing for beam position measurements, supporting the diagnostic requirements of CBXFEL systems and enabling state-of-the-art experimental capabilities.

Contents:

Introduction and Project Motivation	1
FPS2 Schematic Overview	7
FPS2 Layout Overview	21
Reference Generation and Offset Regulation.....	31
Simulated Results.....	35
Future Improvements.....	48
Conclusion	58
References	60

Introduction and Project Motivation

As the product of decades of experimentation and theoretical development, the emergence of practical X-ray free electron lasers (XFELs) has been the genesis of a staggering number of scientific discoveries. The highly coherent and narrow-wavelength photons they emit enable researchers to not only create extreme states of matter, but also to observe physical processes with spatial and temporal resolution that were unprecedented prior to their invention. The techniques they have enabled are the basis for a vast array of exploration in diverse fields such as structural biology, ultra-fast chemistry, condensed matter and plasma physics, molecular imaging and coherent scattering, and even atomic and nuclear testing. Intriguingly, the pace and variety of the research enabled by this technology appears poised to increase with the advent of contemporary XFEL generation techniques.

The first practical XFEL was originally implemented in the 1970s by John Madey at Stanford University's Linear Accelerator National Laboratory, commonly referred to as SLAC, according to an overview by Claudio Pellegrini and Joachim Stöhr [1]. Madey and his collaborators produced X-rays by accelerating electrons through a device called an undulator. Undulators, essentially a group of magnets lined up with alternating polarities, serve to "vibrate" the accelerated electrons as they pass through the magnetic fields to create groupings of electrons with wave functions that are perfectly in phase with one another. As these groupings continue to move through the undulator, they produce highly coherent X-rays through a process termed self-amplified spontaneous emission (SASE). Although groundbreaking at the time, Madey's apparatus was limited in the length of X-ray pulses it could produce and the rate at which it could produce them, primarily due to the stability of the electron beam [1].

Fast forward many decades and the state-of-the-art has progressed considerably since Madey's demonstration. While still fundamentally reliant on SASE to produce X-rays, modern interpretations of the setup, known as cavity-based XFELs or CBXFELs, have improved significantly upon these shortcomings. As illustrated in the figure below, CBXFELs capitalize upon diamond-based optics to cyclically reflect the resulting beam through the undulator [2].

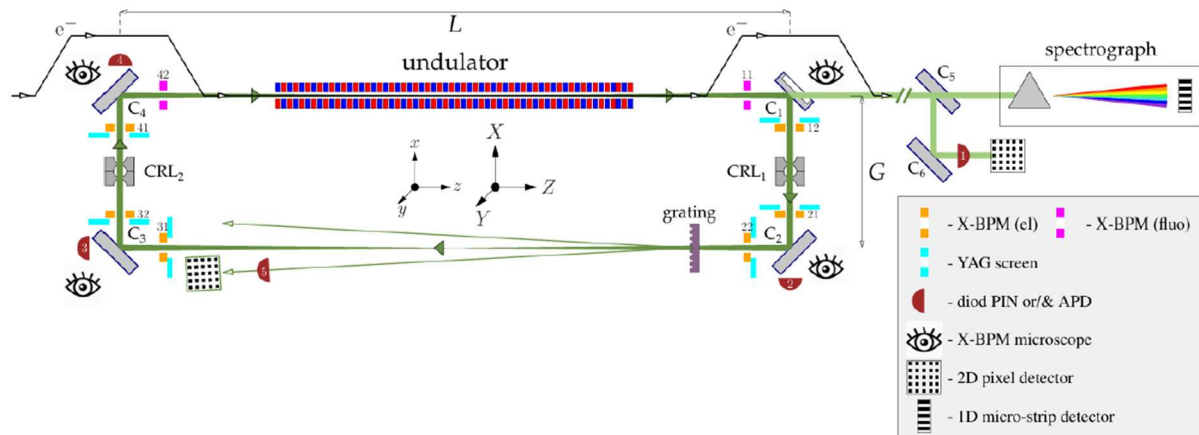


Figure 1.1: A rendering of the cavity-based X-ray free electron laser (CBXFEL) proposed in [2].

An experimental setup for exploration of the concept from Figure 1.1 has been constructed at SLAC to verify its practicality. A portion of the apparatus is shown in Figure 1.2, which pictures a section of the equipment corresponding to the lefthand side of the diagram in Figure 1.1. During operation, containment vessels must create a vacuum to prevent atmospheric absorption of the X-rays as they propagate around the system.

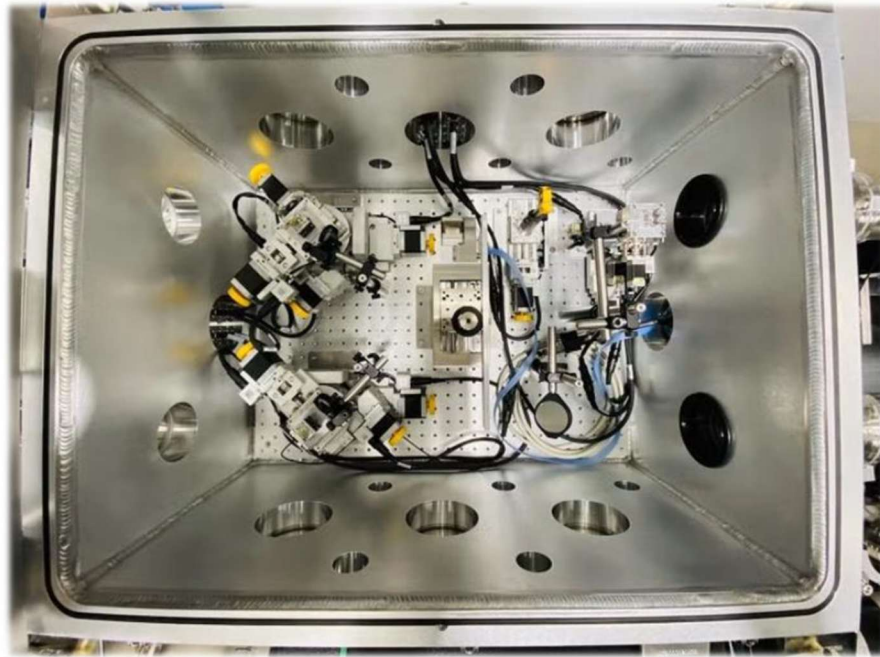


Figure 1.2: A portion of the experimental setup described in [3].

This approach sacrifices the peak brightness achieved by Madey to produce a beam with greater stability and photon density at a much higher repetition rate. However, the complexity of CBXFELs necessitates an extreme degree of precision for optimal functionality. According to one of the leaders of the team that demonstrated one of the first practical examples photographed above, Diling Zhu, “the question was how to produce diamond mirrors of high enough quality and how to line them up with enough precision to steer the X-rays around the cavity” [3].

To enable the level of precision Zhu describes, the AAD or Advanced Accelerator Diagnostics collaboration consisting of researchers from Lawrence Berkeley National Laboratory, Los Alamos National Laboratory, UC Santa Cruz, and UC Davis have proposed a novel diamond sensor platform as a solution. An article authored by the collaboration for the *Nuclear Instruments and Methods in Physics* journal goes into detail on how these sensors work [4]. Biasing a thin sheet of synthetic diamond with a large DC voltage and striking it with an incident X-ray pulse produces a burst of free charge carriers that can be detected in the form of a current pulse through a pick-off

resistor, with the amount of current proportional to the power of the incident X-ray. The saturated drift velocity of the diamond sensor substrate lends itself well to high bandwidth sensing, and the combination of its radiation hardness and near-transparency to the X-ray spectrum means it is well-suited to current and envisioned applications with CBXFEL systems as stated in [4]. Four of these diamonds can be configured in the form of a quadrant sensor to produce a position-sensitive diagnostic platform with the requisite precision to optimize the mirror positioning in a CBXFEL setup. A rendering of the quadrant sensor is shown below for reference in Figure 1.3.

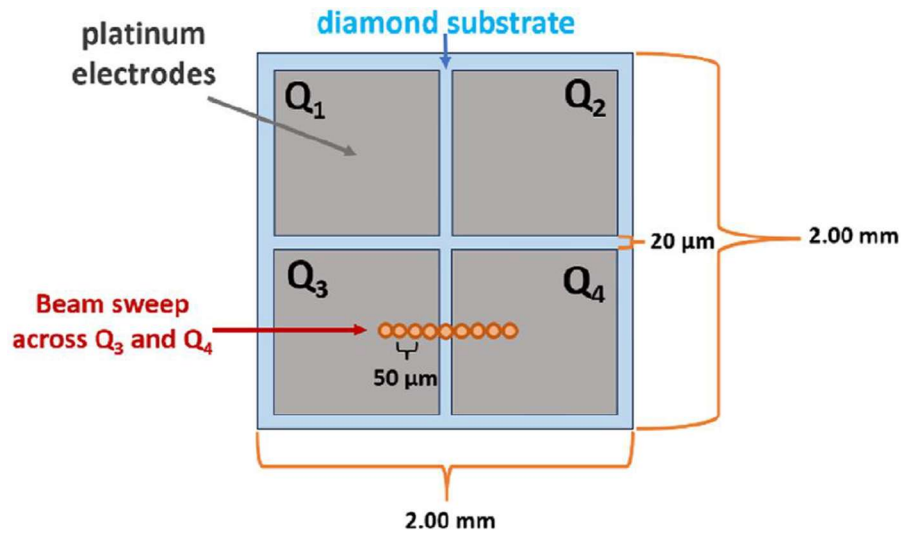


Figure 1.3: A rendering of the experimental diamond quadrant sensor reported in [4].

By measuring the resultant current spikes from each sensor generated from a passing X-ray beam, it is possible to deduce the position of the beam in space with a high degree of accuracy [4]. Downstream systems can utilize the output of this quadrant sensor to reposition the CBXFEL mirrors as needed to optimize system performance. However, given the time scales and high bandwidth signals involved with CBXFELs, interfacing with these sensors is not a trivial task.

The AAD team interfaced with their diamond sensors using hardware in the form of a custom sampling ASIC fabricated in the TSMC 65nm process node. Colloquially referred to as the

Fast Pulse Sampler, or FPS chip, this IC operates by storing the resulting signals from the diamond sensors in an analog memory implemented as a switched capacitor array (SCA). As shown in the block diagram for the FPS in Figure 1.4, the outputs from the four diamonds comprising the quadrant sensor are routed into four identical channels.

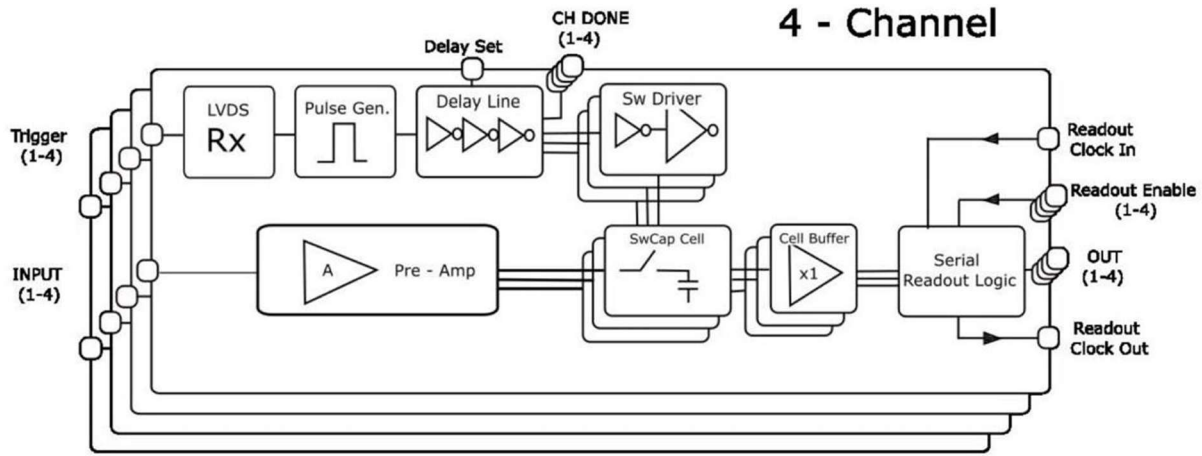


Figure 1.4: The proposed FPS ASIC block diagram from [5].

The signals are first amplified by a set of preamplifiers, the outputs of which are to be stored on the aforementioned SCAs. The capacitors are gated by a travelling pulse, with the associated delay set using an open-loop control scheme modulated by an external voltage source. The charge output by the preamplifiers is deposited accordingly on the capacitors as the pulse travels down the delay line and is in turn read out to an off-chip analog to digital converter (ADC). The backend data conversion is clocked at a comparatively slower rate than the read in signals in a “fast-in, slow-out” arrangement [5]. Initial testing of the FPS chip indicates it is effective as a proof-of-concept, though the bandwidth performance of the preamplifier requires improvement to reach its initial design goal of 10GHz. With the planned revisions in store for the preamplifier, the AAD team has opted to pursue an additional design iteration to improve the sampling distortion at the inputs of the capacitor arrays. The resulting design revision has been termed FPS2.

Given that CBXFELs like the one pioneered at SLAC will be used to study subatomic interactions, they demand picosecond-level timing accuracy. It is imperative to keep all sources of timing error to a minimum. Even small amounts of uncertainty can have massive consequences that limit the validity of experimental results. That said, there is room for improvement on the first iteration of the FPS in this respect. For example, the travelling pulse gating method employed in the prototype is susceptible to undesirable convolution of sampled sensor outputs due to unwanted overlap in the sampling signals applied to the switch drivers.

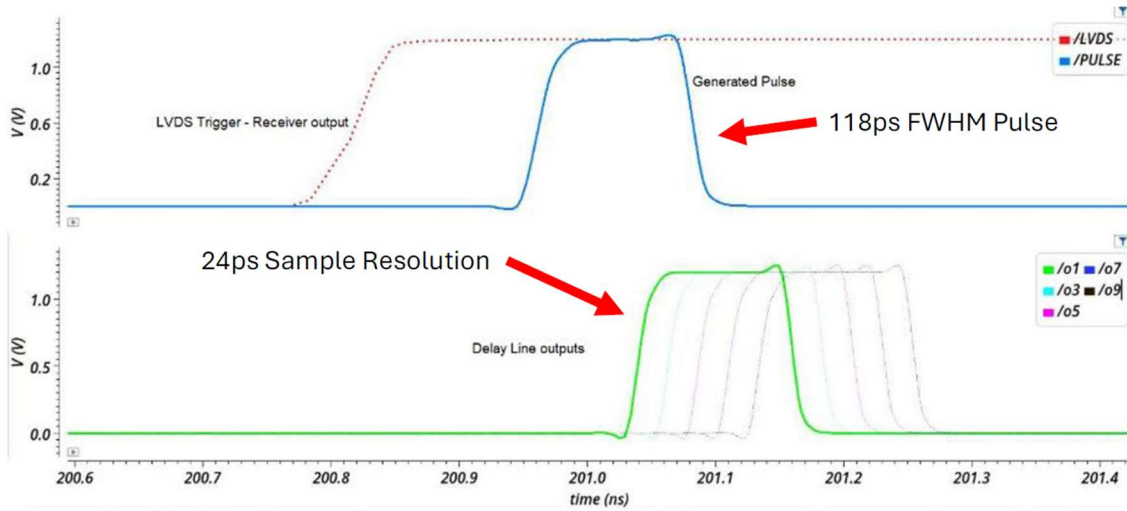


Figure 1.5: A simulated timing diagram published in [5].

This error is clearly visible in the timing diagram shown in Figure 1.5. With the pulse opening several capacitor switches as it travels across the delay line, it is possible for charge to be deposited improperly on adjacent capacitors in the array for each channel. While it may be possible to digitally correct for this source of error with post-processing deconvolution, this work seeks to explore hardware correction techniques to render such correction unnecessary. Furthermore, the open-loop delay control method is well-known to be vulnerable to variation in process, supply, and temperature. While conceding its advantages in simplicity and speed to design and implement, it lacks robustness, accuracy, and disturbance rejection due to its inability to observe and correct for

its own errors. Adapting the FPS architecture to make use of closed-loop negative feedback in the form of a delay-locked loop (DLL) will allow for error correction that will greatly improve its time-domain jitter performance.

FPS2 Schematic Overview

Because FPS2 is a direct progression of the FPS, henceforth referred to as FPS1 to highlight the distinction with this work, much of the basic design philosophy is either wholly or partially leveraged from the original ASIC design. This is evident in the high-level block diagram shown in Figure 2.1, noting that the sampling line has been heavily truncated for visual simplicity. In actuality, the 3-stage delay line in the diagram consists of 91 stages.

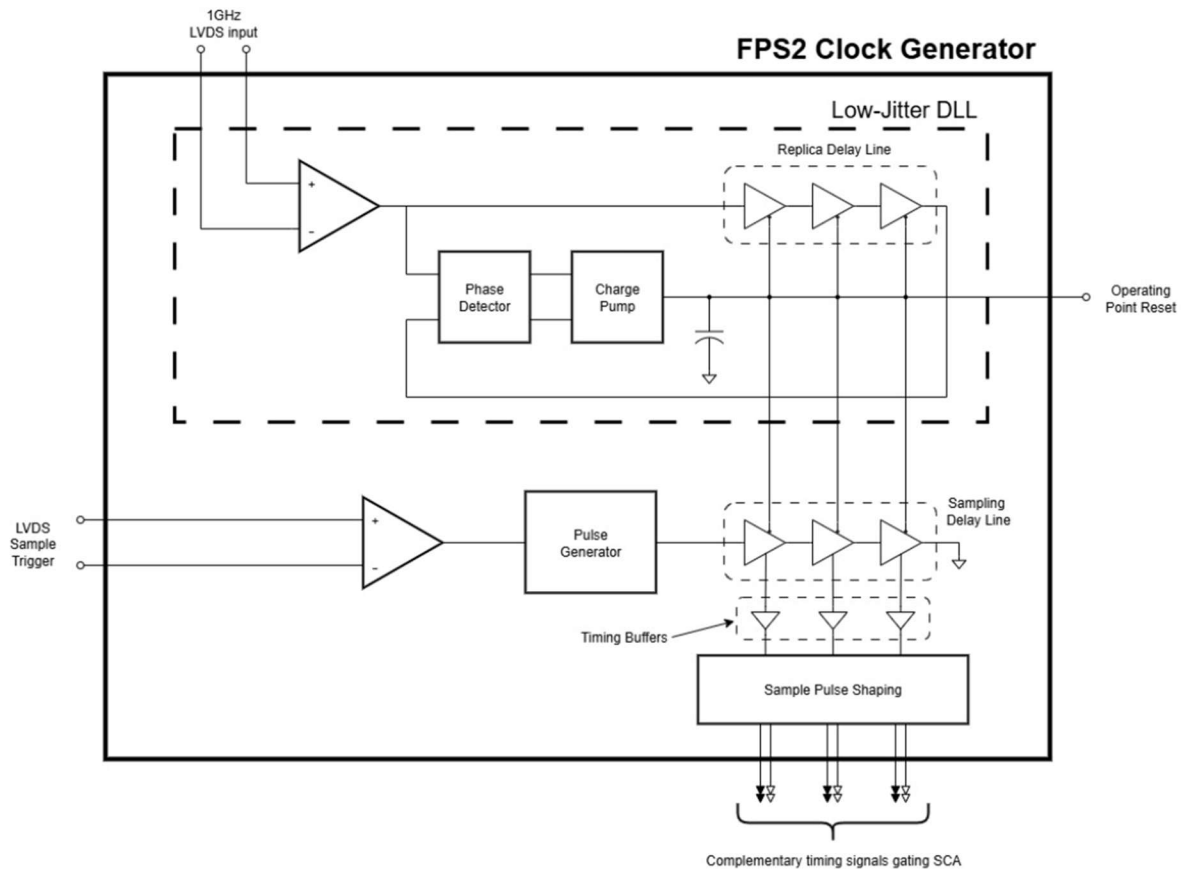


Figure 2.1: A simplified block diagram of the proposed FPS2 ASIC architecture.

The major changes relative to the FPS1 design are shown in the top half and bottom right corner of Figure 2.1 in the form of the on-chip DLL and the non-overlapping pulse generators, labelled “Sample Pulse Shaping,” respectively. The travelling pulse sample gating is preserved from FPS1, but rather than controlling the delay with an externally provided control signal, the control voltage is generated using the DLL as alluded to in the previous section. The cornerstone of this revised approach is a replica biasing technique that ensures the low-jitter performance of the DLL is conferred to the centralized sampling delay line. To ensure a consistent and repeatable operating mode is available, an external operating point reset signal may tie the control voltage to V_{DD} . This effectively disables the loop and will allow a restart sequence to be initiated when the reset signal is removed. This is a quick and easy way around the difficulties imposed by the phenomenon of false lock described in [6]. However, this approach was unexpectedly complicated by duty cycle distortion, which will be addressed in a later section of this document.

The voltage-controlled delay line, or VCDL for short, can be thought of as the foundational structure for any DLL design. Its role is to throttle the speed at which the reference signal propagates through the system, working in concert with the other circuit blocks to match the input and output reference phases. As such, it intrinsically dictates the overall performance of the DLL. The fundamental building block of a VCDL is the individual voltage-controlled delay cell, a cascade of which forms a delay line using an arbitrary number of elements. The FPS1 delay cells used in the open-loop sampling line represent a possible starting point design. These are formed using “off-the-shelf” inverter cells from the TSMC 65 nm Process Design Kit (PDK). Delay modulation is achieved by varying the supply voltage applied to each cell from 0.8-1.5 V, with group delay decreasing at higher supply voltages. While this approach facilitates rapid prototyping, it produces a delay line with poor tuning linearity and limits system-to-system repeatability due to process, voltage, and temperature variation. This drawback is visualized in the simulated delay tuning curve

shown below in Figure 2.2. It also requires an additional voltage supply that exceeds the 1.2 V supply for the ASIC [5].

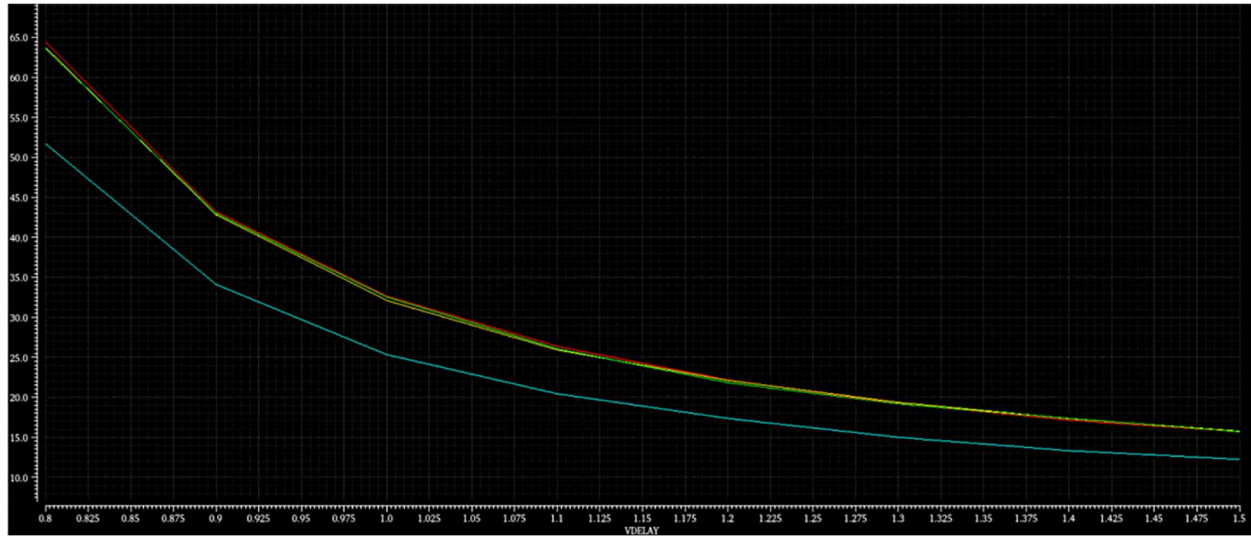


Figure 2.2: Simulated delay tuning curve of CMOS inverters provided by the TSMC 65nm PDK.

Custom topologies were explored to correct this nonlinear characteristic without relying on an additional voltage supply. Two well-known delay cells that provide delay modulation capability without varying the supply voltage of the circuit are the current-starved inverter and the NMOS pseudoinverter. The former improves process, voltage, and temperature variation by its topology, which uses a negative-feedback current mirror structure to provide the bias current for each stage. In the configuration investigated for FPS2, shown in Figure 2.3, decreasing the control voltage causes the bias network to inject additional bias current, thereby increasing the transconductance of the stage. By extension, this increases the speed (decreases the delay) of the stage.

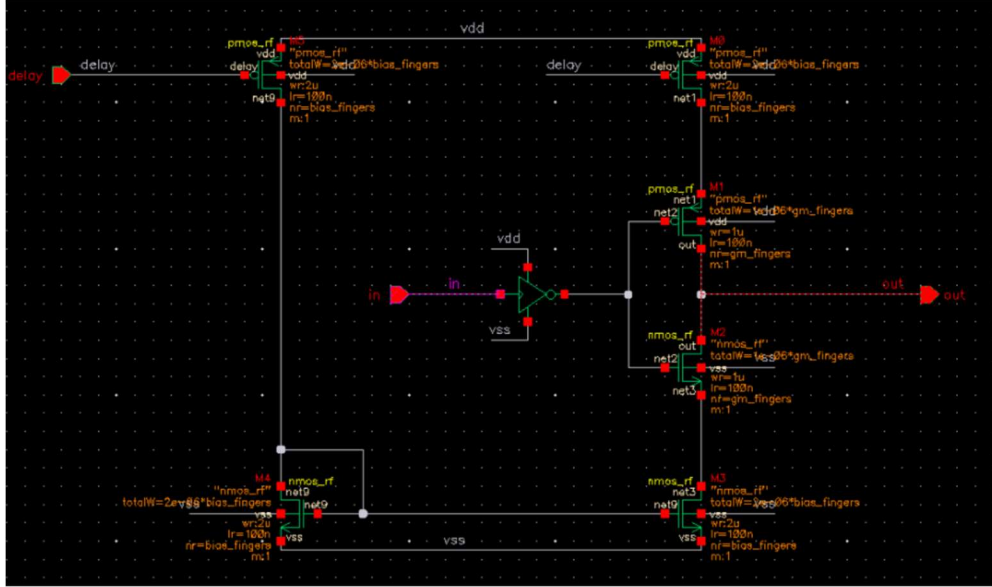


Figure 2.3: Current-starved inverter delay cell schematic.

This effect can be visualized through transient simulation, depicted in Figure 2.4 showing the response of incrementally increasing control voltages ranging from 0-1.2 V in steps of 0.1 V superimposed over one another.

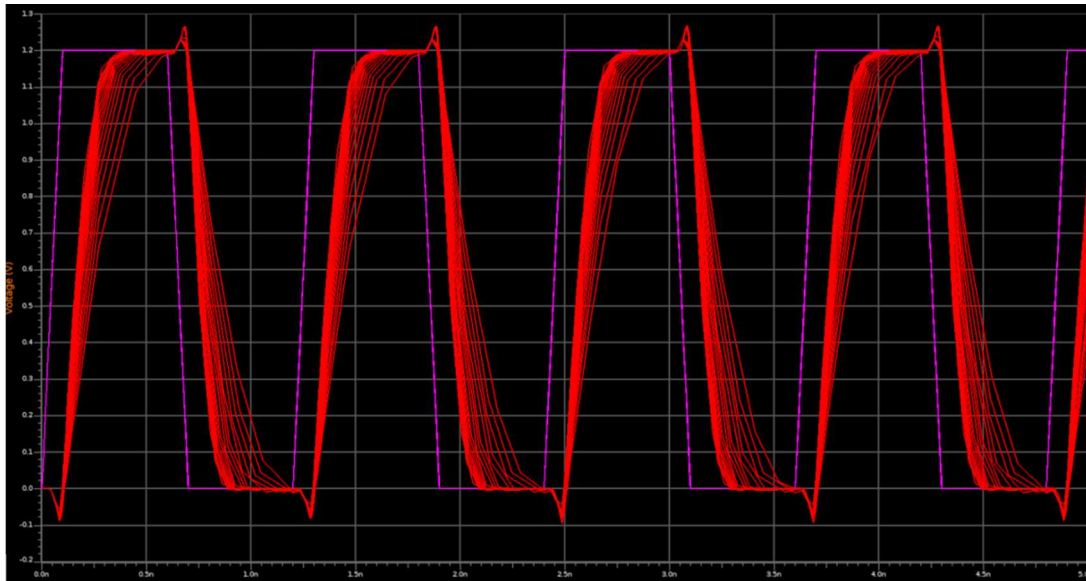


Figure 2.4: Time-domain illustration of the variable delay of current-starved inverter from Figure 2.3, showing the response of a control voltage sweep between 0-1.2 V in steps of 0.1 V. The purple trace is the input signal.

This delay-cell topology has several drawbacks. Most significantly, it fails to improve the tuning linearity of the delay cells used in the FPS1 design. It also uses comparatively more die area per cell due to the additional transistors needed to form the current mirror biasing network. Lastly, its minimum delay was found to be much slower than FPS1, limiting the sample resolution capability of the sampling circuitry.

Having discounted the viability of a current-starved inverter delay circuit, the second topology evaluated for FPS2 is called an NMOS pseudoinverter. Requiring only three devices for proper operation as depicted in the schematic from Figure 2.5, the pseudoinverter is a much more compact solution for the delay cell.

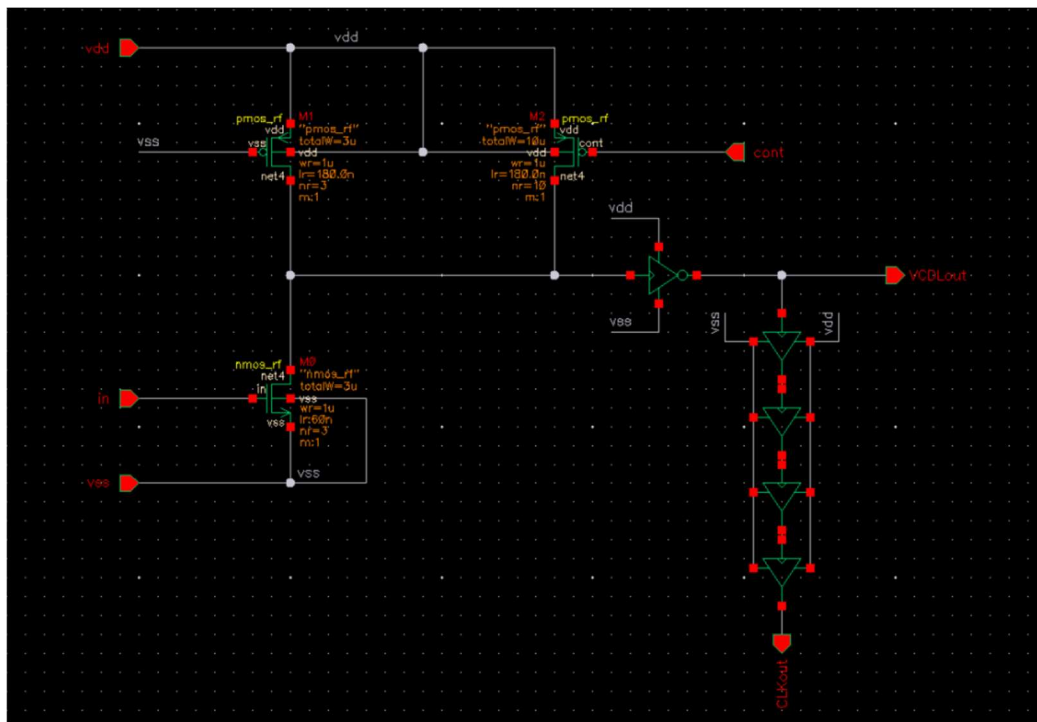


Figure 2.5: Circuit schematic of the pseudoinverter delay cell.

Not only does this topology save considerable die area when aggregated into a lengthy cascade, it also exhibits a highly linear tuning curve shown in Figure 2.6 to improve upon FPS1. This linearity is clearly shown in Figure 2.6.

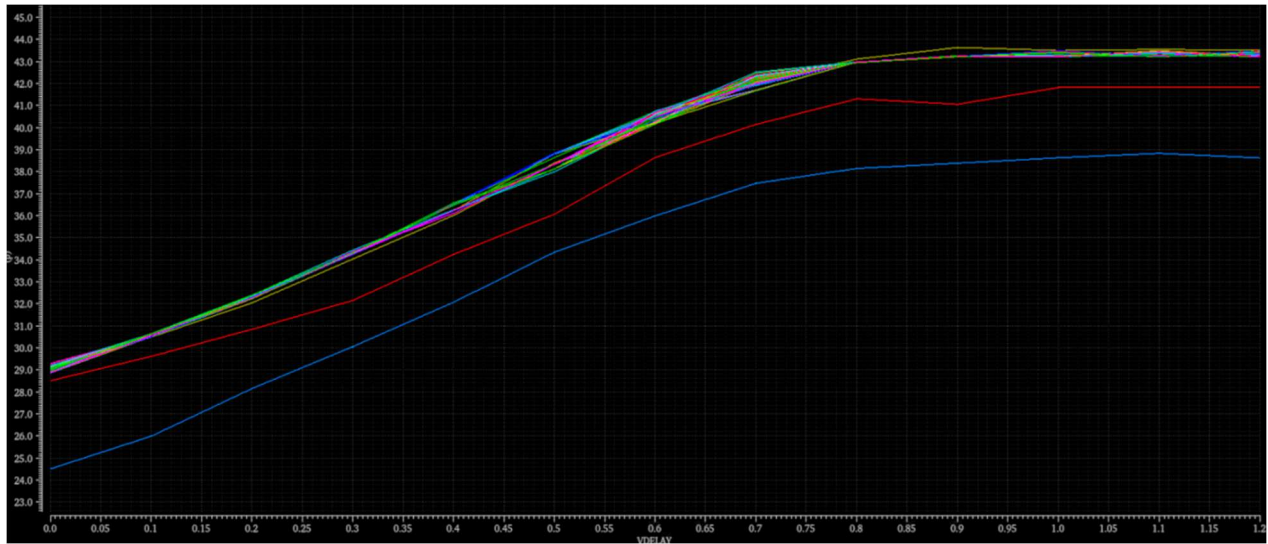


Figure 2.6: Variable NMOS pseudoinverter time delay as a function of control voltage.

In addition to its improved linearity and minimal footprint, the pseudoinverter carries the additional benefit of being much faster in comparison to the current-starved delay cell since it has significantly less parasitic capacitance due to the reduced transistor count. Any additional capacitance contributed from parasitics will undesirably increase the charging time constant of the input and output nodes of the delay cell, producing a slower delay stage as outlined in [7].

The operation of a pseudoinverter is deceptively simple, but the schematic presented previously has several added features that warrant explanation. Three MOSFETs (M0-M2) form the classical pseudoinverter structure, and each one serves a specific purpose in its operation. The NMOS device M0 forms the active part of the delay cell and, analogously to a common source amplifier stage, performs the action of inverting the input signal at its gate to the output at the drain. The PMOS device on the left (M1) has its gate tied to ground to form a baseline current source, ensuring continuous operation in a situation where the control PMOS M2 is operating in the cutoff region. The control transistor, opposite the biasing device, modulates the group delay of the stage by sourcing bias current in accordance with the control voltage applied at the gate. Because the

control PMOS is active low, the lower the control voltage applied at the gate, the faster the pseudoinverter will propagate the input signal. Beyond these three FETs, each stage has an inverting clock buffer inserted between the drains of each transistor and the output node. This provides rail-to-rail voltage swing capability and limits static power draw of the integrated VCDL by placing each stage in cutoff when no signal is applied to reduce DC current flow. Finally, each cell's output node is tapped by a telescoping set of clock buffers to produce a sharp-edged output clock signal associated with each delay cell in the line. Each buffer in the sequential cascade increases in drive level to create clock signals powerful enough to drive the switched capacitor array for each of the four parallel channels in the quadrant sensor without excessively loading each pseudoinverter.

A cascade of 91 pseudoinverters forms the composite VCDL used in FPS2. Three additional dummy stages are added to the input and output stages to improve the uniformity of the loading of the active stages. It is a well-known issue with DLLs used for multi-phase clock generation that discontinuity in loading at the boundaries of the delay line are a common source of distortion as detailed in [6]. The schematic for the cascaded VCDL can be seen in Figure 2.7 at the top of the next page.

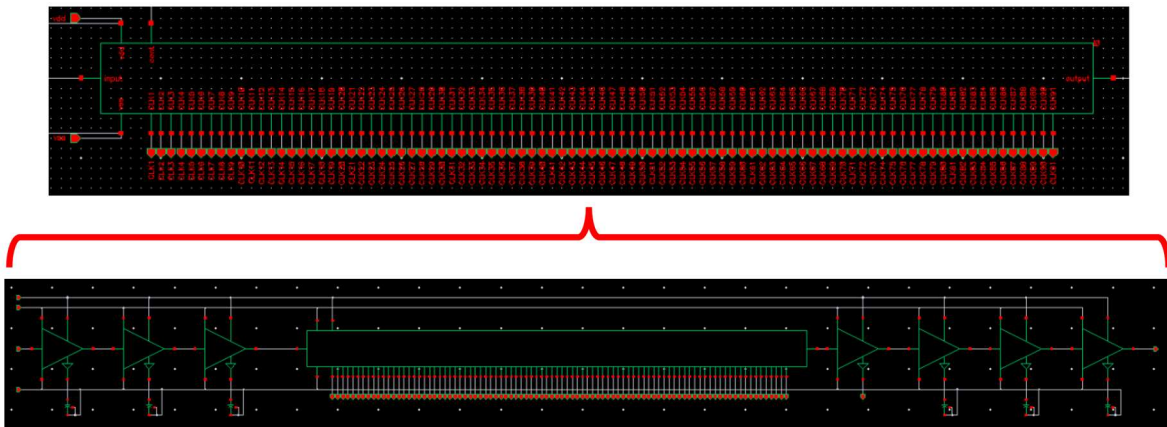


Figure 2.7: The circuit block corresponding to the VCDL in the high-level DLL schematic (top), along with the integration including the sacrificial dummy delay elements for uniform multi-phase clocking (bottom).

The number of elements in the delay line was selected to give an adequate capture window for the quadrant sensor. At steady-state equilibrium, the group delay for each delay cell is on the order of about 30 picoseconds, meaning a 90-phase clock generator will span a range of about 2.7 nanoseconds in total. The additional 91st stage is necessary to produce 90 sampling pulses when paired with a non-overlapping lead-lag pulse generator. The pulse generator circuitry will be discussed later in this section.

The next link in the chain forming the DLL feedback loop is the phase/frequency detector, abbreviated as PFD. This block takes the reference input and VDCL output as inputs and measures the phase mismatch between the two signals. In turn, it generates output signals to adjust the control voltage of the VCDL to counteract any errors.

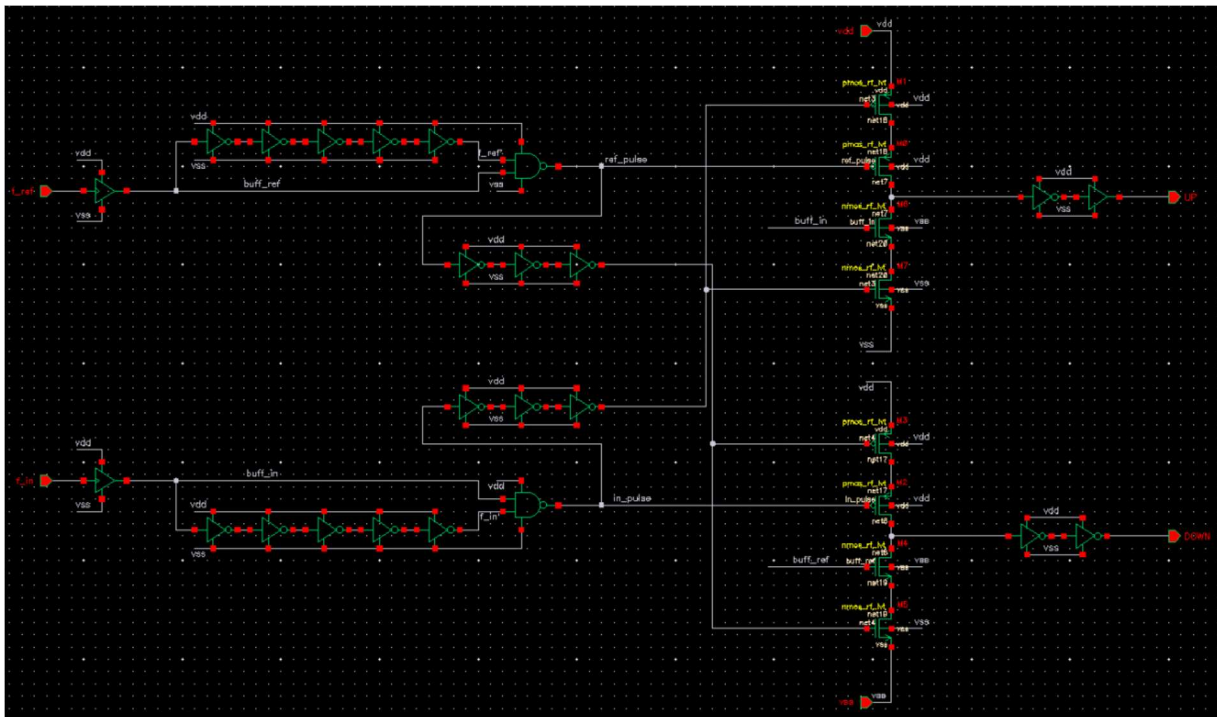


Figure 2.8: Schematic for the “bang-bang” PFD circuit used in FPS2.

As depicted in the accompanying schematic from Figure 2.8, the PFD for FPS2 uses a cross-coupled latch architecture, sometimes called a “bang-bang” topology. While traditional PFDs

usually incorporate D-flip flops with reset functionality, these designs are limited in their operating speed, are prone to dead zones, and can exhibit undesirable output glitches from their use of feedback. The bang-bang PFD omits the need for feedback and solves all these problems as discussed in [7]. The operational flow for this PFD is straightforward: the reference signal and the VCDL output are each tied to both the NMOS data input of a dynamic latch and to an inverting pulse generator. Then each inverted pulse is routed to the clocking input of the opposing latch. Connecting the latches in this way will latch the up or down signals high or low as necessary until the rising edge of the opposite signal resets the branch. When the reference signals are perfectly in phase, the up and down signals are both held low to maintain equilibrium until a phase error is detected and corrected for downstream by the charge pump driven by the up and down signals. Note that the up signal is inverted relative to the down signal in Figure 2.8 to correspond with the logic polarity of the charge pump, as will soon become clear.

Like the name implies, the charge pump either sources or sinks current from the control node of the DLL to “pump” charge in and out of the system and vary the control voltage applied to the VCDL. By performing this task in accordance with the control signals output by the PFD, the control voltage is modulated in opposition to any phase error detected between the two reference signals. The varying control voltage either speeds up or slows down the VCDL, forcing the output to track the phase of the reference signal. The charge pump schematic from this design is included in Figure 2.9.

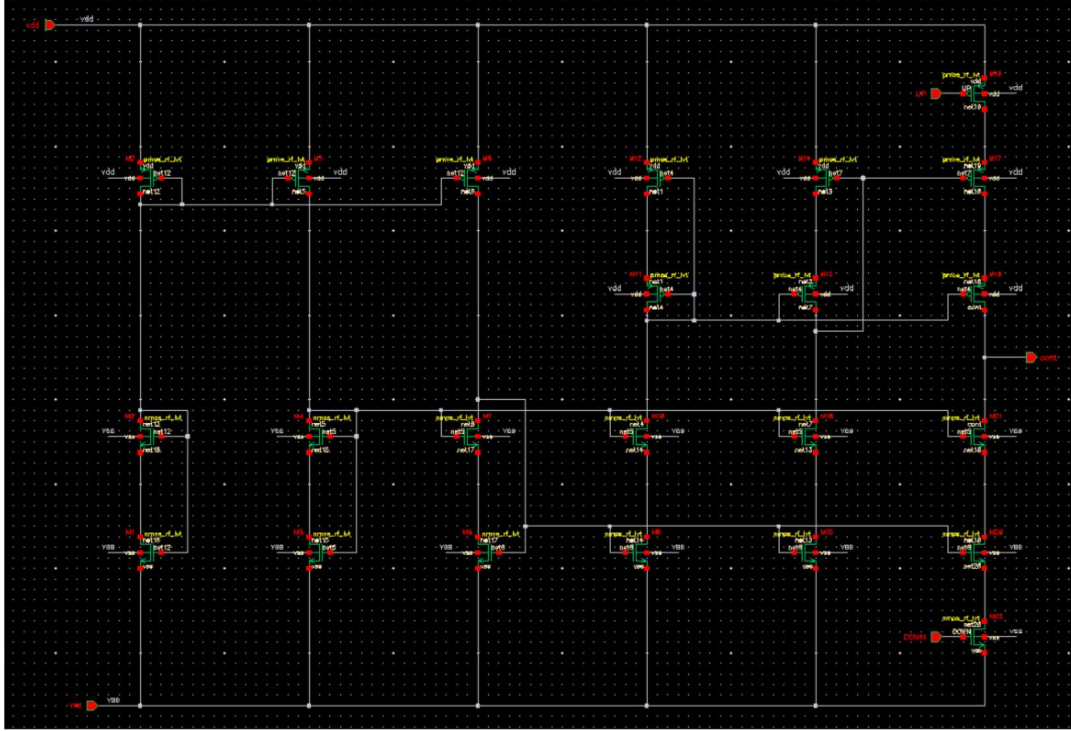


Figure 2.9: Schematic for the FPS2 charge pump featuring a self-biasing architecture.

A key feature of this topology is the cascode current source structure. This dramatically increases the output impedance of the circuit, promoting stability of the control voltage node. An additional advantageous feature of the design is the self-biasing architecture. This reduces susceptibility to process, supply, and temperature variation. To improve the balance of the up and down current sources, the PMOS transistors of the output branch are enlarged slightly to overcome the disparity in transconductance between them and the NMOS devices sourcing the down current. Four unit devices are used for each device in the schematic to increase the charge pump's drive strength so it can drive the control voltage of the delay line more effectively. The switching transistors interface directly with the PFD, using those signals to open and close the up and down currents as dictated by phase discrimination by the PFD between the VCDL input and output to maintain steady-state lock. Inspection of this portion of the circuit clarifies the need for inversion on the 'UP' signal from the PFD. Because the 'UP' current is gated by a PMOS switch in this design, it is an enable-low

signal. With the incorporation of the controlled delay line, the PFD, and the charge pump all accounted for, the main loop of the FPS2 DLL is very nearly complete.

The loop is closed at the control voltage node by which the correction signals are fed back to the delay line. A caveat that must be accounted for here is that the charge pump output will contain a frequency component at the reference frequency. Similarly to PLL applications, a DLL requires a loop filter to extract the DC component to be applied to the control input of the VCDL. For FPS2, this design opts for simplicity and makes use of a single capacitor for its loop filter.

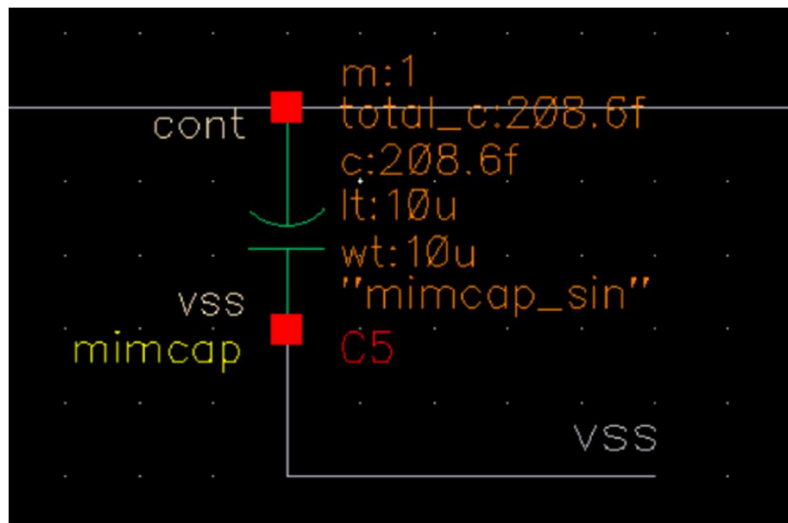


Figure 2.10: Close up image of the series capacitor loop filter used in FPS2.

In many PLL applications, this may be an approach that risks loop instability. For such designs it is common to design higher-order feedback loops that incorporate more sophisticated loop filters. PLL loop filters often insert a zero to increase the phase margin of the loop to improve stability should it encounter a phase or frequency step. Fortunately for FPS2, because DLLs achieve phase lock only by varying the time delay of the reference signal as opposed to PLLs that must modulate the frequency of a VCO, they are comparatively simpler to stabilize in most cases. In fact, according to [8], it is somewhat unorthodox for higher order loop filters to be used in DLLs, unless phase

discontinuities are expected in the application. A loop filter in the form of a capacitor to ground is an effective choice for FPS2 for several reasons. A standalone capacitor has a single pole at DC and rolls off quickly at the rate of 20dB per decade, illustrated by the bode plot in Figure 2.11.

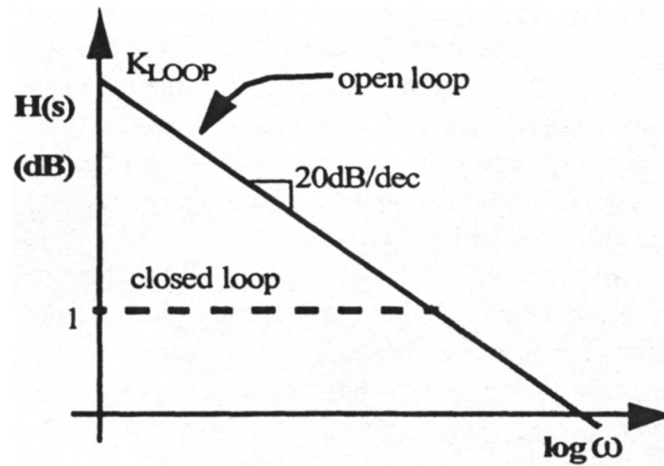


Figure 2.11: Bode plot of the frequency response of a series capacitor loop filter.

As will be discussed later, the reference generator for the DLL is a high-performance temperature-compensated crystal oscillator (TCXO) with a nominal output frequency of 1 GHz. At this frequency the filter capacitor will present a short circuit to the residual frequency content, causing it to be sunk directly to ground while passing the DC content of the signal. Furthermore, the TCXO will deliver a highly stable reference signal without any expected phase or frequency discontinuity. It is therefore relatively straightforward to achieve and maintain phase lock in FPS2, rendering higher order loop filters unnecessary.

Although not part of the DLL feedback loop, the final piece of circuitry that differentiates FPS2 from FPS1 is the additional non-overlapping pulse generator. This block acts as a buffering switch driver and is tied to the output of the sampling delay line. It consists of a cascade of 90 lead-lag timing stages that produce extremely fast pulses with as short rise and fall times as possible. Each element is constructed according to the schematic in Figure 2.12.

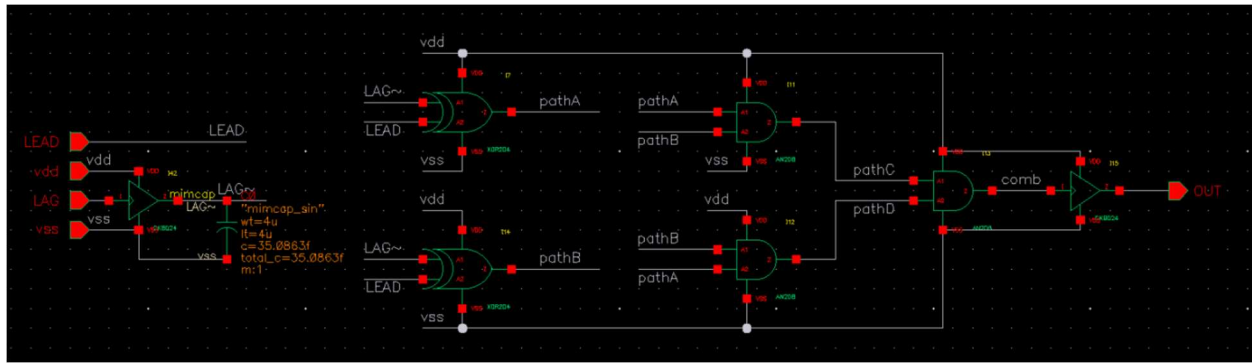


Figure 2.12: Circuit schematic of an individual element of the non-overlapping pulse generator.

This design approach closes the complementary switches for an individual capacitor in the SCA upon the rising edge of the leading clock signal and opens the switches with the rising edge of the lagging clock. It has been specifically tailored to be insensitive to the falling edge of the sampling pulse to prevent the sampled data from being overwritten. A capacitively loaded clock buffer is used to exaggerate the time offset between the lead and lag clocks so the logic gates have ample time to respond to their inputs. Each cell uses a balanced cross-coupled structure of the TSMC PDK's generic XOR and AND logic gates to produce a pulse that is highly symmetrical in the time domain. The cross-coupled routing equalizes the rise and fall times of each pulse by correcting for the difference in intrinsic carrier mobility between the NMOS and PMOS transistors used in the logic gates. The two balanced paths are then recombined by a final AND gate and subsequently edge-sharpened by an output clock buffer with sufficient drive level to gate the associated capacitor in each of the four SCA channels. Cascading 90 of these pulse generator stages yields a set of sequential timing pulses when the sample pulse is applied to the sampling delay line, as depicted in Figure 2.13.

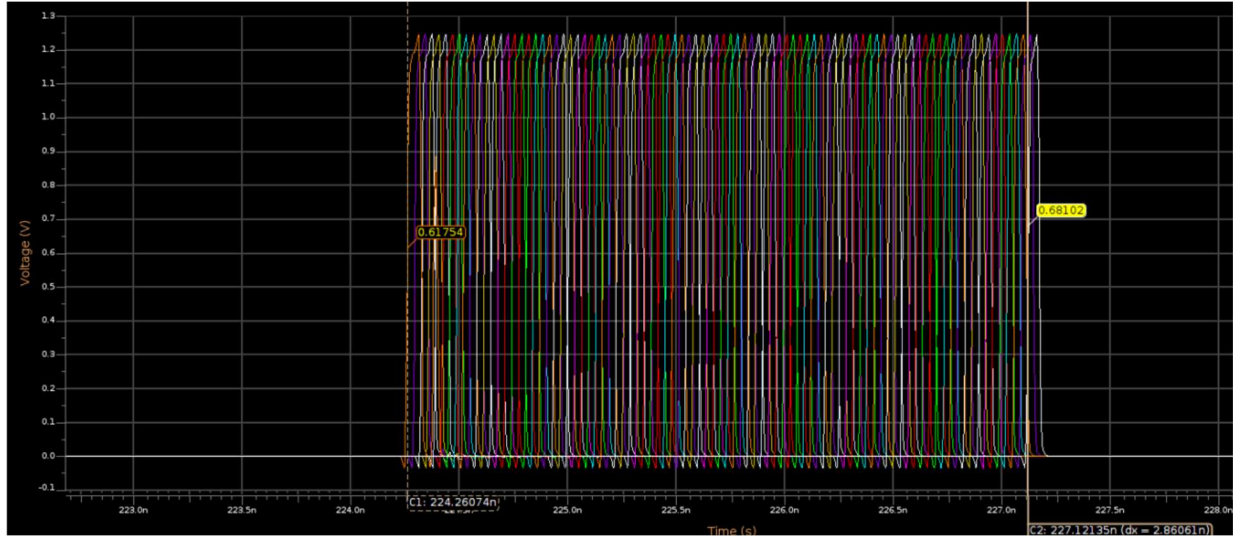


Figure 2.13: Transient simulation of a sampling event showing the 90 cascaded sampling pulses.

Varying the capacitance that loads the lagging clock buffer allows for the fine-tuning of the output pulse characteristics for each pulse stage. A capacitance of 35 fF produces optimal sample pulses in terms of the full-width half-max (FWHM) pulse duration. In simulation, the output of which is pictured in Figure 2.14, this configuration generates pulses with about 30 ps FWHM duration. The resolution between adjacent pulses is dictated by the delay associated with each delay cell in the locked DLL. In the case of FPS2, this is about 28 ps when sampling at steady state. Even at this sampling resolution, the overlap between neighboring pulses is limited by design, obviating the need for digital deconvolution of the sampled data.

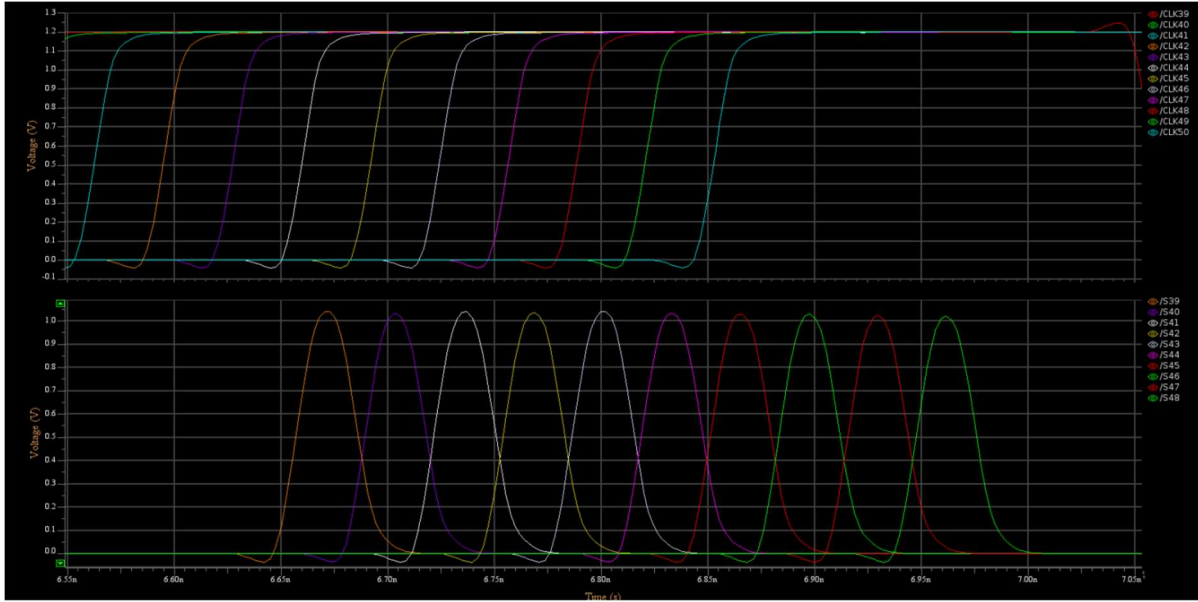


Figure 2.14: A segment of the resulting simulated sampling pulse train after optimization, zoomed in for emphasis.

As discussed in the introduction of this work, this non-overlapping pulse generator provides a hardware-based solution to correct any unwanted convolution of the sampling signals as the gating pulse traverses the sampling line.

FPS2 Layout Overview

It goes without saying that layout is a critical step in the development of any chip. In many cases, it can be even more influential on final performance than the schematic itself, since a poor layout can undermine even the most carefully designed circuit. Beyond limiting peak performance, inadequate layout can reduce reliability and, in extreme cases, compromise basic functionality. The semiconductor and metallic structures that compose practical integrated circuits all have parasitic resistance, inductance, and capacitance that impose limitations and second-order effects on any design. With modern process nodes producing increasingly smaller device geometries and operating at elevated frequencies, the proportional impact these parasitics have on

performance of contemporary ICs is bound to become progressively more important to consider throughout the design process. For example, bandwidth can be negatively impacted by unexpected resistance or capacitance, or line inductance of traces can degrade rise and fall times and cause signal ringing. Device matching, noise resistance, and crosstalk reduction are all achieved through thoughtful layout as opposed to clever schematics. Good layout practice is critical to the DLL on FPS2 given the length of the delay line involved and its foundational reliance on replica biasing to regulate the control voltage for low-jitter performance. Furthermore, the correlation of the data read into each channel from the quadrant sensor requires a highly compact layout architecture.

The efficient use of die area is paramount in the layout approach for the pseudoinverter delay cell. Counting the six dummy elements used to buffer both the input and output of the VCDL, there are 97 elements in the whole delay line. Considering that there are two VCDLs, one for the DLL and a replica for sampling the sensor outputs, it would be easy for these lengthy structures to consume an excessive amount of die space if care is not taken to condense the layout of each element as much as possible.

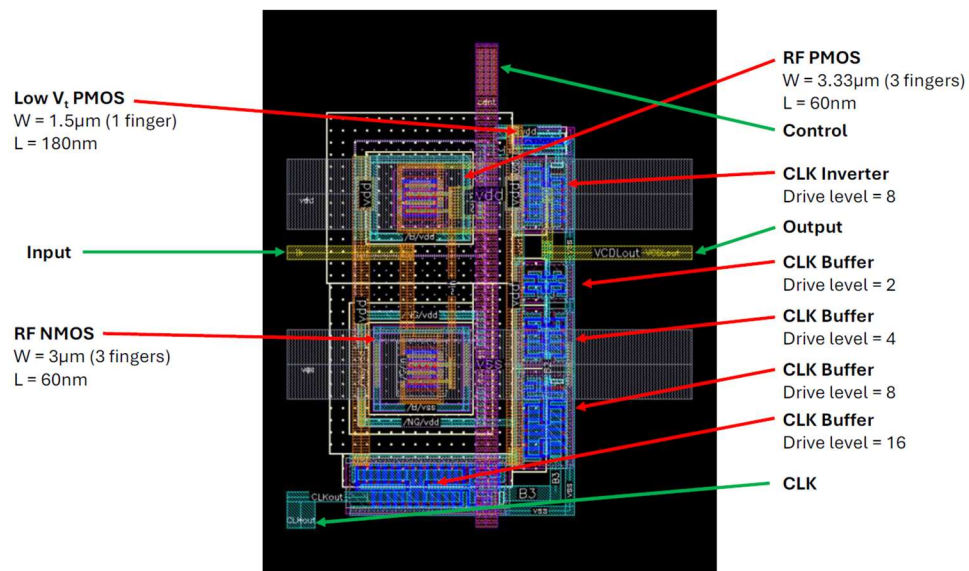


Figure 3.1: Close-up annotated layout of a pseudoinverter delay cell from the FPS2 DLL.

Shown in the annotated snippet in Figure 3.1, each pseudoinverter is laid out to use die space as efficiently as possible. The various input and output interfaces are oriented to promote ease of integration with the rest of the circuit elements including neighboring delay cells, the other blocks of the DLL, and the replica sampling line. The control voltage is fed in at the top of the element with the trace extending out of the bottom to allow the control voltage to be tied to the corresponding delay element of the replica line while minimizing any associated loss. The input and output nodes are aligned with each other so the structure can be easily cascaded with the adjoining delay elements. The telescoping clock buffers wrap around the perimeter of each pseudoinverter and output the clock signal in the bottom left corner of the layout. Lastly, the supply rails run over the top of the entire stage to connect the supply voltage and ground plane with wide traces to avoid loss when the stages are lined up together. This ensures each stage is supplied with a consistent voltage that is well-matched with the rest of the stages in the cascade.

Once the individual pseudoinverter cell has been successfully laid out, the layout of the delay lines is less complicated in comparison. Instantiations of each pseudoinverter cell can be placed and tied together without much difficulty using relatively short traces.

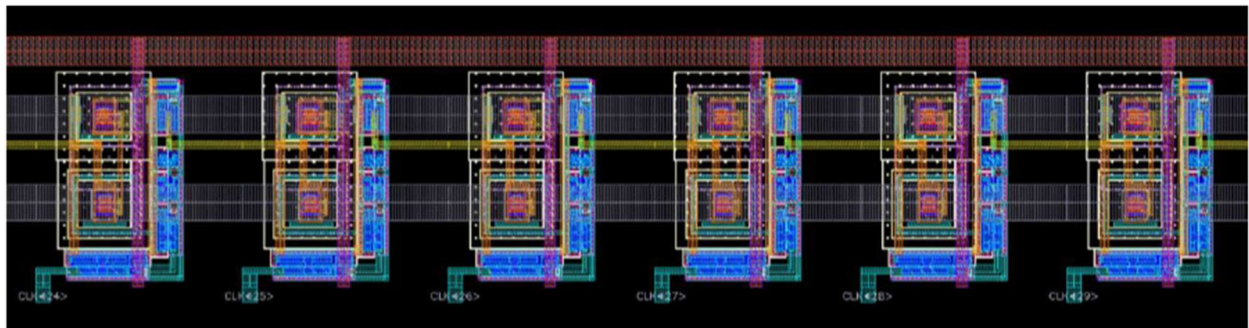


Figure 3.2: An excerpt of the layout for the cascaded pseudoinverters making up the FPS2 VCDL.

Ideally, each delay cell should be placed as closely together as possible. However, as shown in the image of the layout in Figure 3.2, this is not the case for the VCDL in FPS2. In this design, the pitch

between the delay cells has been dictated by the spacing between the capacitors in the SCAs. Each cell in the line is laid out such that its buffered clock output can be tied directly to the complementary switch drivers of each capacitor without the need for complicated fanout routing.

Relative to the pseudoinverter cells and the VCDL cascade, the PFD was more challenging to layout in a condensed fashion, mostly because of the higher device count. As depicted in Figure 3.3 at the top of the next page, the different sections of the PFD have been folded in the layout to reduce the amount of die area they consume. On the left are the two signal input ports from the reference input and the output of the VCDL. These feed directly into clock buffers to drive the inverting pulse generators and downstream latch gates without introducing too much additional load to the rest of the DLL. The latches themselves can be seen on the right side of the layout, with the buffered control signal output terminals located at the top and bottom corners. These port locations allow the PFD to be placed as close as possible to the downstream charge pump and drive it with minimal routing required to connect the two blocks.

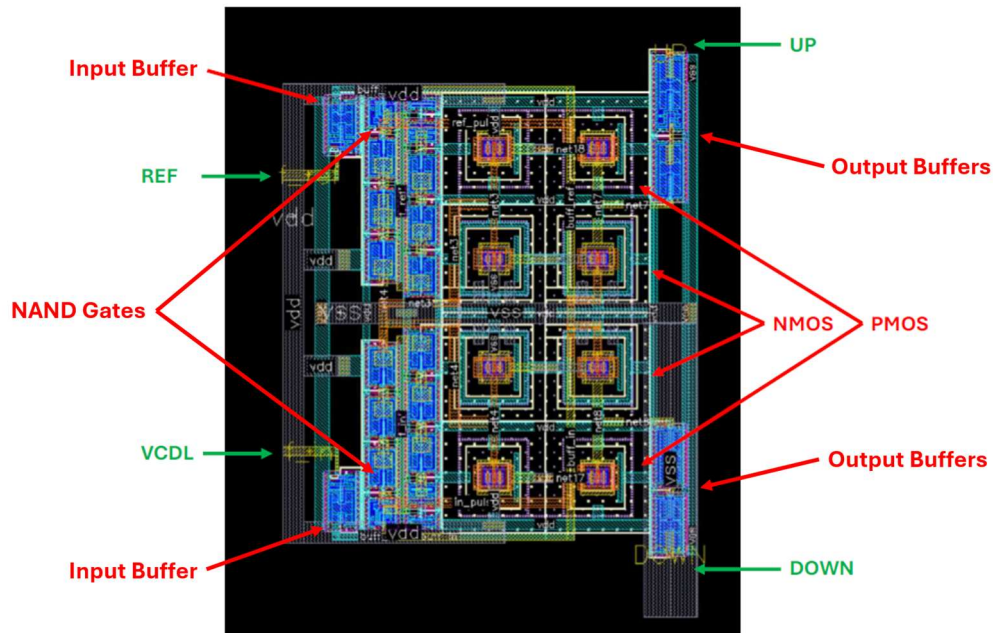


Figure 3.3: An annotated layout for the PFD used in FPS2.

Most of the space required for the charge pump layout is dedicated to the biasing network as seen in Figure 3.4. Each transistor in the layout is composed of four unit devices as discussed in the schematic section, which have been grouped together in a quadrant formation to allow them to operate functionally as one large transistor.

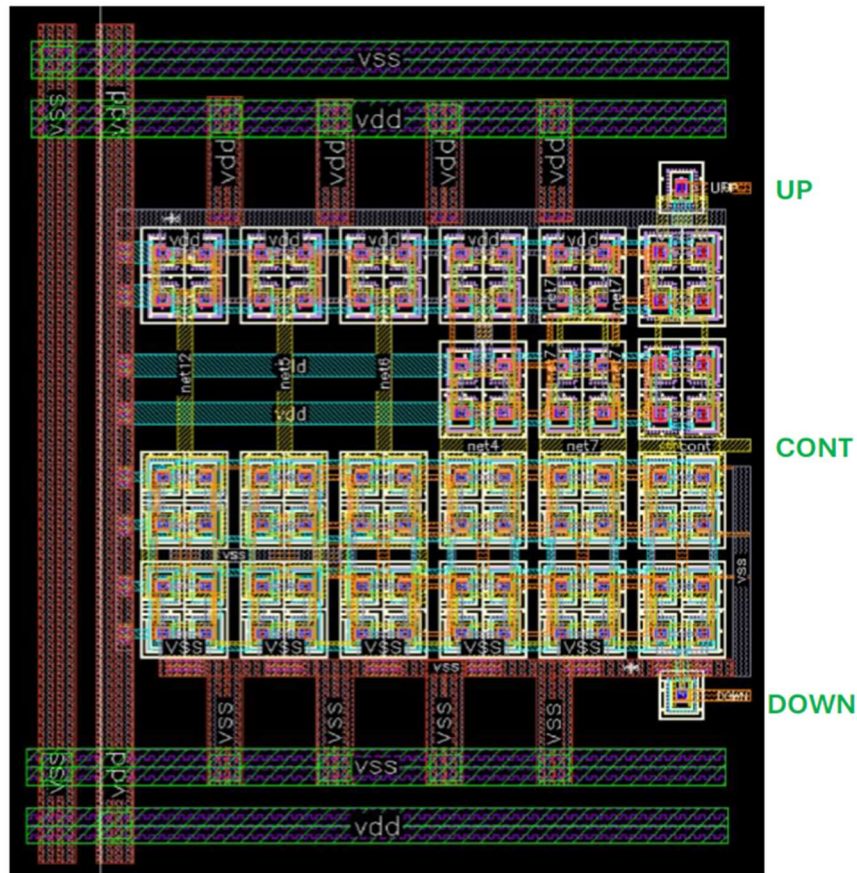


Figure 3.4: Layout of the charge pump for the FPS2 DLL.

The control signal input ports have comparable pitch to the corresponding outputs from the PFD, simplifying the task of tying them together in the combined layout. The control signal output, used to drive the loop filter, is centrally located so the control voltage can be distributed to the VCDL while preserving symmetry in the high-level design. The loop filter, a capacitor to ground as discussed previously, takes the form of a metal-insulator-metal (MIM) capacitor fabricated in the

topmost layers of the 65 nm process. This feature is placed as close as possible to the charge pump structure so it can be connected with negligible loss.

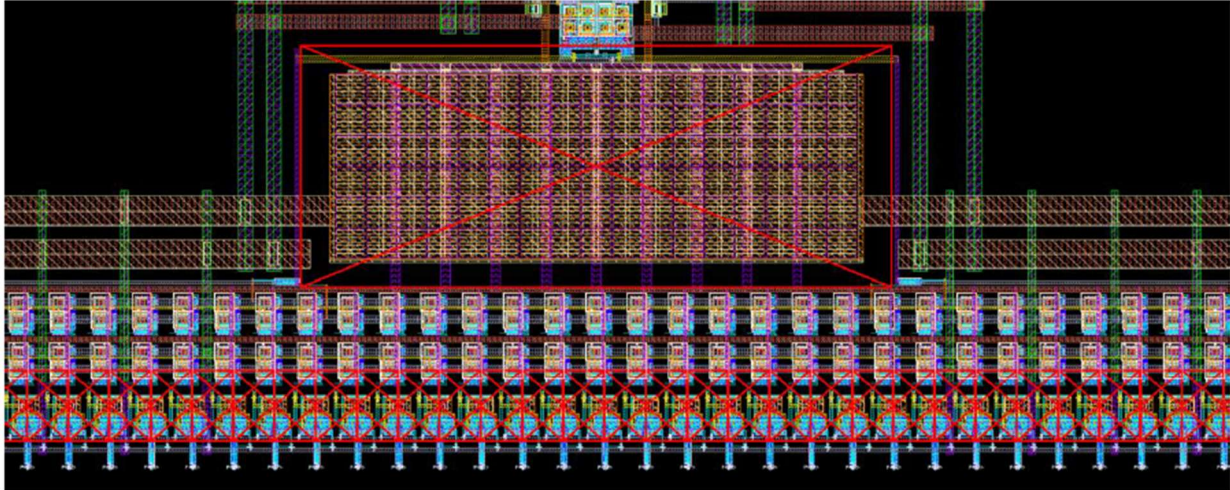


Figure 3.5: A zoomed-in view of the FPS2 layout illustrating the intersection of multiple circuit blocks, specifically a portion of the charge pump (top), loop filter (center), and the stacked cascades corresponding to the DLL delay line, replica sampling line, and non-overlapping pulse generator (bottom, top-to-bottom).

From this point, visualized above in Figure 3.5, the control voltage is modulated by the charge pump, stored on the MIM capacitor, and distributed to the DLL and sampling line by way of a symmetrical fanout network. To ensure there is no voltage droop at this critical point in the architecture and guarantee that every delay cell therefore receives an identical control voltage, the fanout structure uses wide traces at multiple metal layers in the stackup to minimize resistive losses as much as possible.

In a similar fashion to the approach outlined for the pseudoinverter, the non-overlapping pulse generator is laid out using a folding approach shown in Figure 3.6.

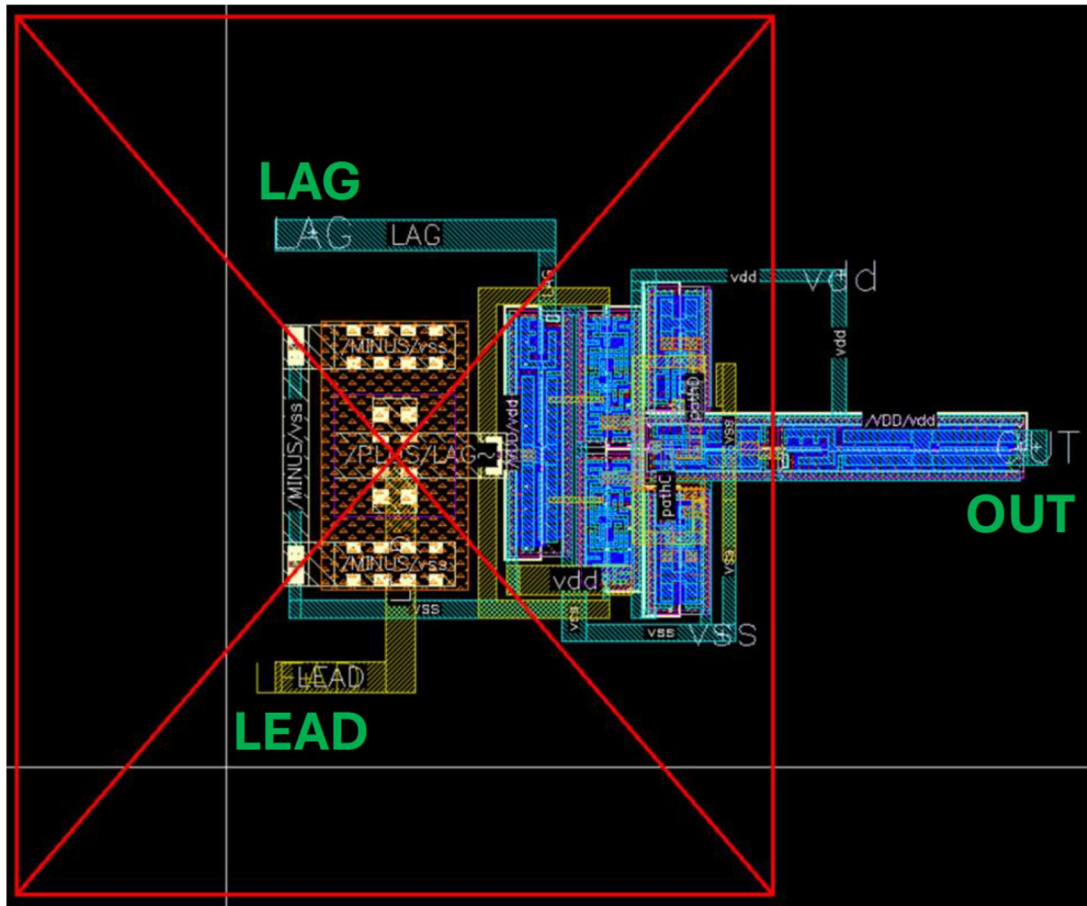


Figure 3.6: Layout for an individual non-overlapping pulse generator cell, part of a 90-element array.

The throttling capacitor that spaces out the rising edges of the leading and lagging clock inputs is shown on the lefthand side of the layout. This capacitor uses the same MIM capacitor structure used for the loop filter described previously. The cross-coupled XOR and AND gates sit in the middle of the block with both branches feeding into the recombinational AND gate near the centroid of each pulse generator cell. The buffered pulse is then output from the center of the righthand side to be routed to the switch drivers for the SCAs of each channel. An additional feature to promote miniaturization of the integrated layout after cascading the pulse generator array is that the lead and lag input ports are placed at metal 2 and metal 1 of the stackup, respectively.

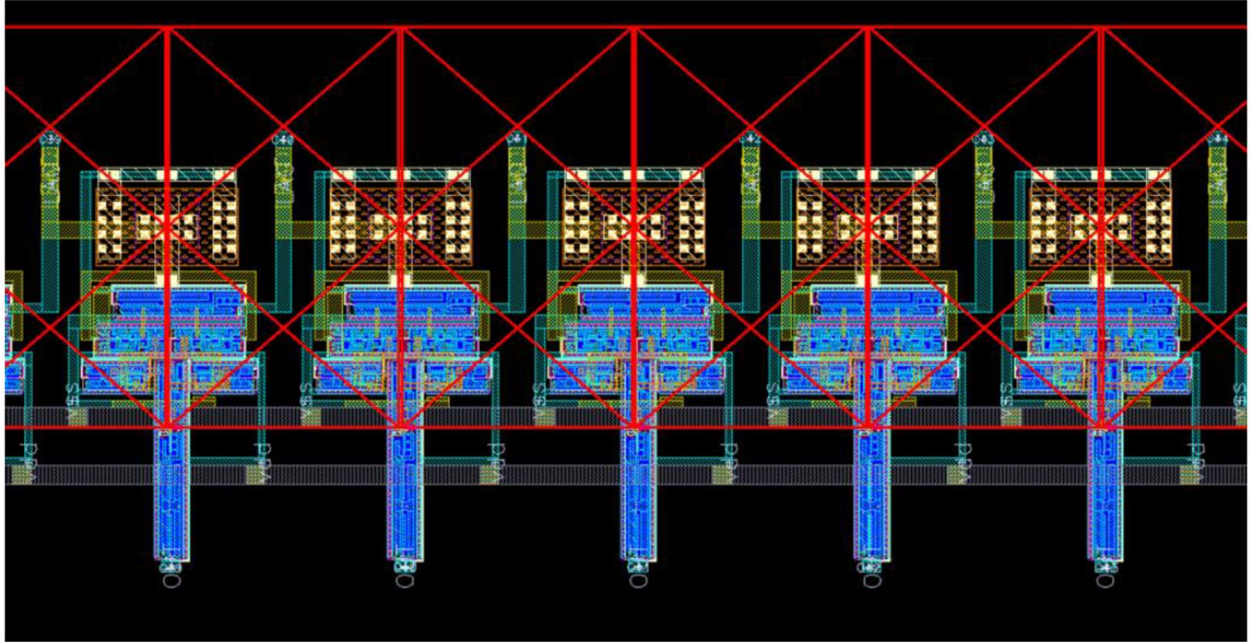


Figure 3.7: Close up of the layout for the cascaded non-overlapping pulse generators.

This approach allows the leading and lagging clock inputs of neighboring pulse generators to be overlapped with one another and be driven in common with the output signal from the corresponding element from the sampling line as illustrated in Figure 3.7. This is a highly advantageous layout because it requires essentially zero routing from the outputs of the sampling line to the inputs of the pulse generator elements. By extension, this will keep interstage mismatch to a minimum and ensure the delays between each of the output pulses stay as uniform as possible. As a result, our team can be confident that sampling jitter is limited by the design of the layout and that any data gathered using FPS2 is highly reliable for leading-edge scientific experimentation.

After finalizing the layout of each block, they must be carefully integrated such that the undesirable effects of layout parasitics and process gradients are mitigated to the greatest extent possible. A visualization of the layout of the system-level DLL is included in Figure 3.8, to help illustrate how each circuit block fits together in the high-level design. The whole design, including

the core DLL loop, the replica sampling line, pulse generators, and reference and trigger signal receivers has been combined using a linear layout approach for ease of integration with the rest of the ASIC.

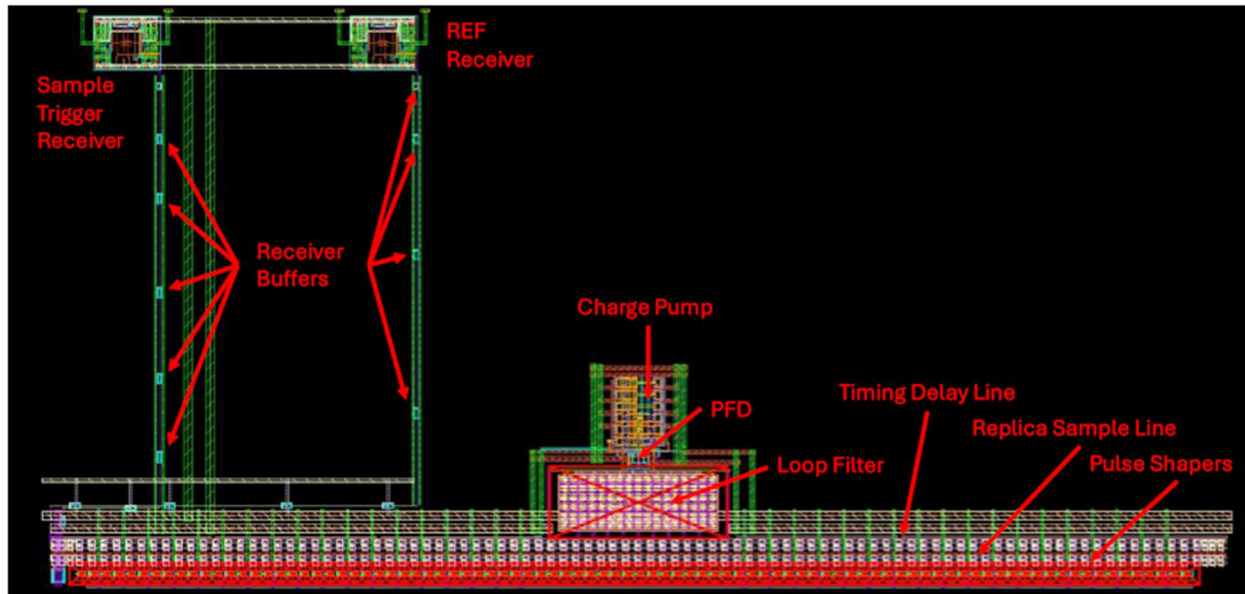


Figure 3.8: An annotated consolidation of all the layout blocks outlined in this section, visualizing how they fit together.

In many IC applications, it is generally advisable to avoid laying out large circuits like this in a linear fashion. Many process variations tend to follow linear gradients, so there is a risk in this approach that delay elements at either extremity of the line may impart timing mismatches because of process-induced error. Ultimately though, the threat of linear process deviation was outweighed by the convenience and practicality of this approach when it came to marrying the DLL timing circuitry with the SCAs and the rest of the FPS2 layout. With details such as the pitch and locations of the SCAs predetermined by prototyping work done previously by other engineers on FPS1, any attempt to combat linear gradients through creative folding or a common centroid layout would complicate the higher-level chip layout too much to be a realistic possibility. This fact is clearly visible in the chip-level layout included in Figure 3.9.

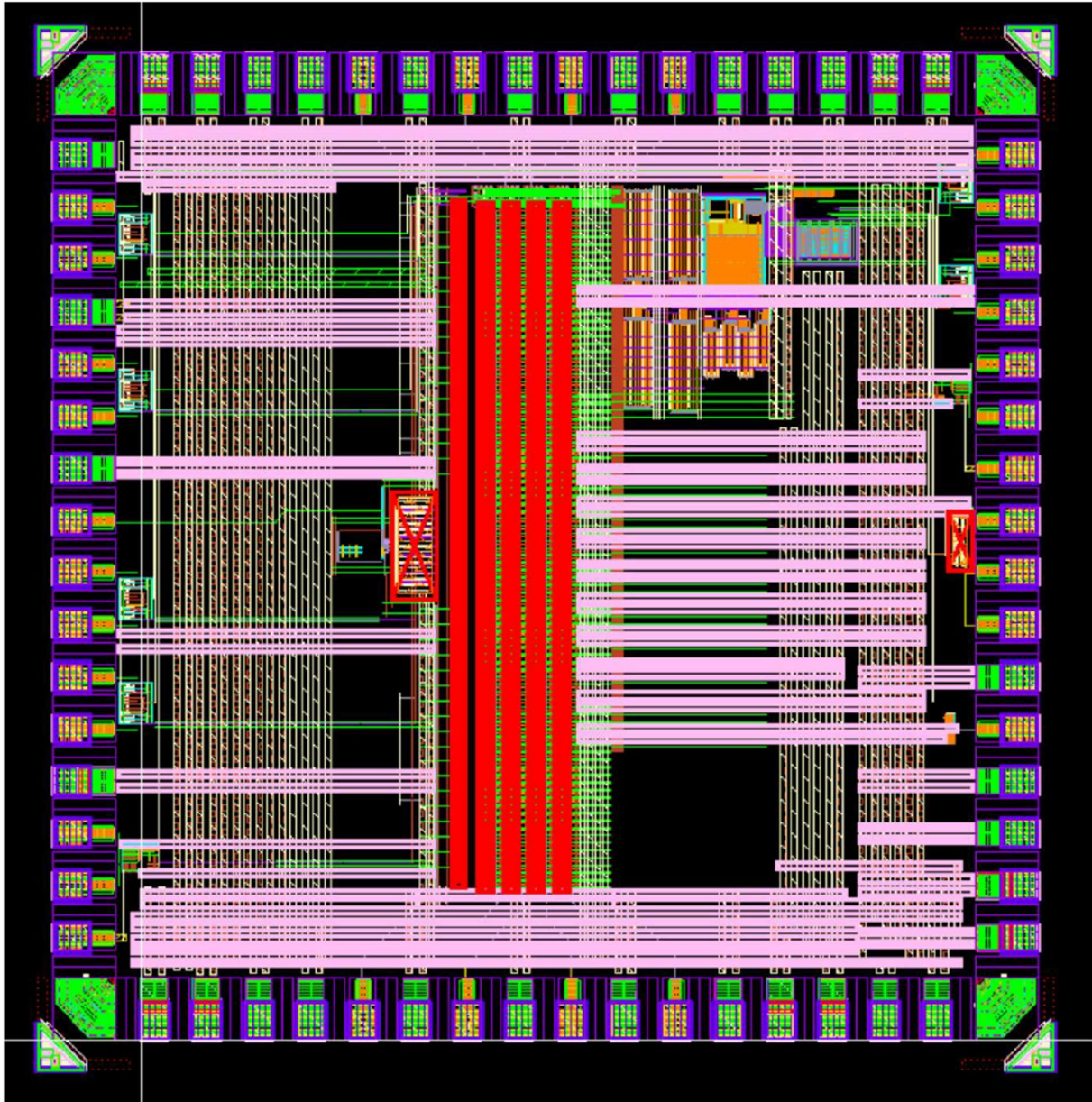


Figure 3.9: Complete FPS2 layout including the DLL and sampling line, associated receivers, four SCA channels, bias control block, and the pad ring boundary for interfacing with the carrier board.

Simulation of this linear approach that accounted for layout parasitics assuaged any remaining concerns on this topic as the simulation indicated the DLL performance agreed sufficiently with the schematic simulations performed earlier in the development process.

While the linear layout approach simplified things from the standpoint of layout integration with the rest of FPS2, a shortcoming introduced as a result was excessive loss incurred on the long signal paths. This came into play in two areas of the chip and required some changes to address the issue in both cases. First, the reference signal and the sample trigger input paths needed to be routed in from the LVDS receivers near the terminals on the border of the chip. To make sure the signal integrity is not disrupted by the lengthy traces needed here, a series of regeneration buffers are added to offset their loss. These buffers are pointed out by the annotations of the top-level DLL layout presented in Figure 3.8. The same issue is also a concern when routing the VCDL input and output signals to the input ports of the PFD. Regeneration buffers are also employed here to guarantee the DLL will function as intended, with great care taken to space them equidistantly and symmetrically so as not to introduce systematic timing offsets that may hinder the DLL's ability to achieve phase lock. With these additional features, the DLL is expected to function well with low-jitter clock signals delivered simultaneously to all four sensing channels of the quadrant diamond sensor. All simulations indicate FPS2 will perform well as the front-end interface for the diamond sensors described in the introduction after fabrication on practical silicon.

Reference Generation and Offset Regulation

Recalling from previous sections, DLLs function similarly to PLLs in that they lock the phase of two signals to one another. Although DLLs are often easier to lock because they lock a signal to a time-delayed copy of itself as opposed to varying the frequency of a VCO, this comparative simplicity is often a double-edged sword. One of the foremost drawbacks in this contrast between the two is the fact that DLL performance is fundamentally limited by the quality of the reference signal that it locks to. As outlined in [6], while PLLs exhibit low-pass loop dynamics that allow them to reject time-domain jitter error, DLLs are said to have an “all-pass” response and are therefore

incapable of filtering errors caused by jitter. Seeing as the performance of FPS2 is therefore at the mercy of the performance of the reference signal, quite a bit of attention was paid to the design of the reference generator. The cornerstone of this part of the system is a low-phase noise 1 GHz temperature compensated crystal oscillator, or TCXO, from Abracon [9]. Figure 4.1 is an excerpted image from [10], the distribution resource where the project was able to procure samples for the eventual turn-on of the FPS2 ASIC.



Figure 4.1: Webpage header for Digi-Key supply of ASGTX TCXO used in the reference generator [10].

A TCXO is a sophisticated type of crystal oscillator that incorporates temperature sensing and feedback to produce an output signal with improved stability and greater spectral purity than a generic crystal oscillator without temperature compensation. The Abracon TCXO will provide a practical reference signal that the FPS2 DLL will be easily capable of locking to.

Integrating the TCXO into the system with the on-chip DLL was not a trivial task. To begin with, it needed to be mounted on a PCB close by the FPS2 ASIC to properly drive the circuitry on the IC. Due to the reduced supply voltage of the 65 nm process, additional signal conditioning is required for the ASGTX TCXO to properly interface with the DLL. FPS2 is powered by a 1.2 V supply and incorporates LVDS receivers that have been designed to operate with a 600 mV common-mode (CM) input voltage. Unfortunately, the TCXO configuration that was available to the team uses the low-voltage differential signaling (LVDS) protocol. In this configuration, the output voltage is a

differential signal with a 1.25 V DC offset. To operate as intended for the FPS2 design, this offset voltage required a CM level shift down to the required 600 mV for the on-chip LVDS receivers. The challenge with this approach was to accomplish this shift without injecting additional noise that would degrade the performance of the reference generator. To overcome this, as demonstrated in the simplified schematic in Figure 4.2, the TCXO is tied to the reference input with AC coupling. Described in [11], this method of terminating the TCXO output will block the DC offset voltage while passing the differential output signal with as little attenuation as possible.

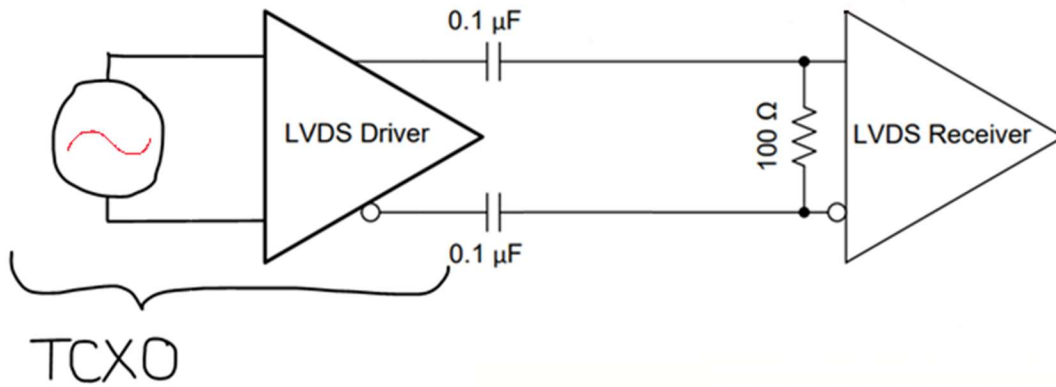


Figure 4.2: The AC-coupled LVDS termination from [11], annotated to point out the location of the TCXO.

This design does not by itself solve the level shifting problem, because there is not a DC common-mode voltage applied for proper operation of the LVDS receiver. Without the correct 600 mV applied, the reference generator will be unable to correctly drive the DLL loop. The solution here was to condition the CM input voltage using a low-noise, low-dropout linear regulator or LDO. The TPS7A94 from Texas Instruments was selected for this part of the design. An expanded schematic is shown in Figure 4.3, indicating how the LDO is integrated into the LVDS termination. The values for the peripheral passives have been derived from [12] and have been selected to optimize for low-noise and high PSRR while trading off startup time, which is not critical for the FPS2 application. On

the carrier board, the regulated offset is injected between two 50 Ω resistors to create a balanced termination with a 600 mV CM voltage.

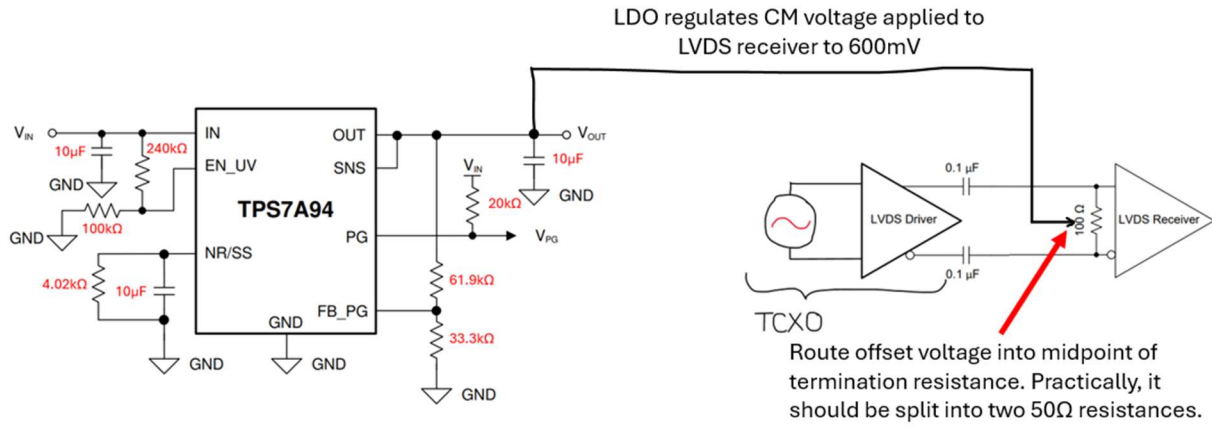


Figure 4.3: Complete reference generator schematic showing the LVDS TCXO with offset voltage level shift.

The AC coupled interface ensures that the LVDS offset voltage of the TCXO is bifurcated from the 600 mV output from the LDO while passing the 1 GHz signal of interest that will serve as the DLL reference. The TPS7A94 is an ideal choice to regulate the offset voltage of the reference receiver for two reasons. Firstly, it is designed to operate down to voltages as low as 400 mV. Many commercial LDOs cannot output a stable DC voltage down to the required 600 mV. Moreover, it has the lowest noise contribution of all LDOs in the TI portfolio: 0.46 μ V_{RMS} between 10 Hz and 100 kHz per [12]. This allows it to provide a sufficiently low DC offset voltage, doing so while contaminating the TCXO output with as little additional noise as possible.

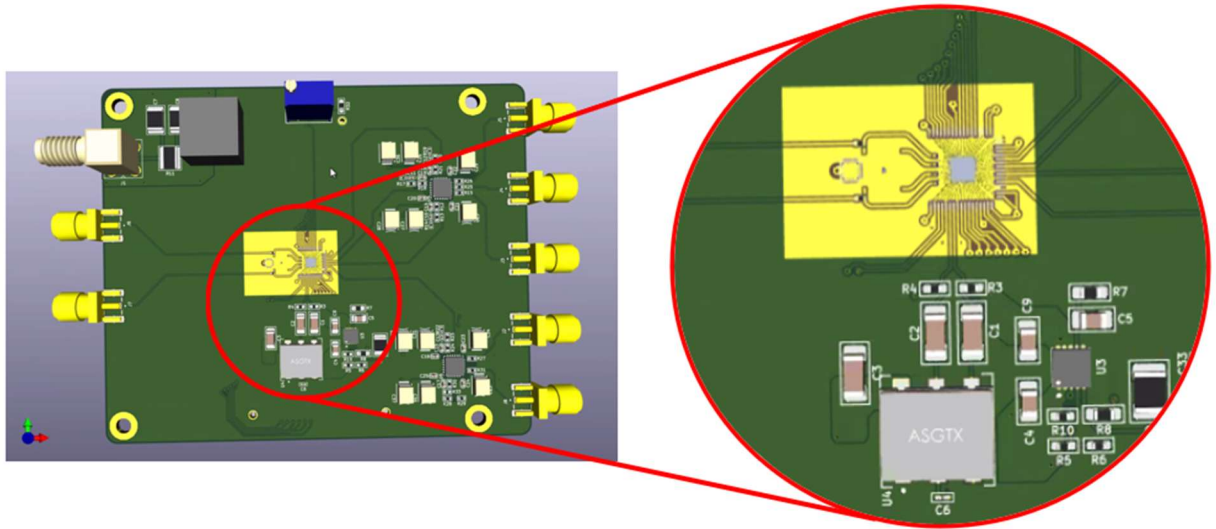


Figure 4.4: Mechanical model of the carrier board detailing the intersection of the 1GHz reference source and FPS2 ASIC.

Putting all this together as seen in the 3D mechanical rendering of the printed circuit board from Figure 4.4, the DLL on FPS2 will be driven by a high-performance reference circuit that closely approximates the ideal 1 GHz square wave used to validate the design in simulation during the development process.

Simulated Results

The primary motivation for adding a DLL to the design for FPS2 is to reduce sampling distortion resulting from time-domain clocking jitter. While the practical ASIC has yet to be tested in the application, Siemens Eldo simulations appear to be indicative that this goal has largely been achieved [13]. There are two types of jitter that have been analyzed with this design. The first, referred to in this work as sampling jitter, is a direct product of the DLL and is reminiscent of the classical conception of timing jitter that comes to mind for most engineers. This quantity describes the variation of the delays between the rising edges of adjacent phases of the multi-phase DLL outputs. For reference, a plot of these delays for each pair of clocks is included below in Figure 5.1.

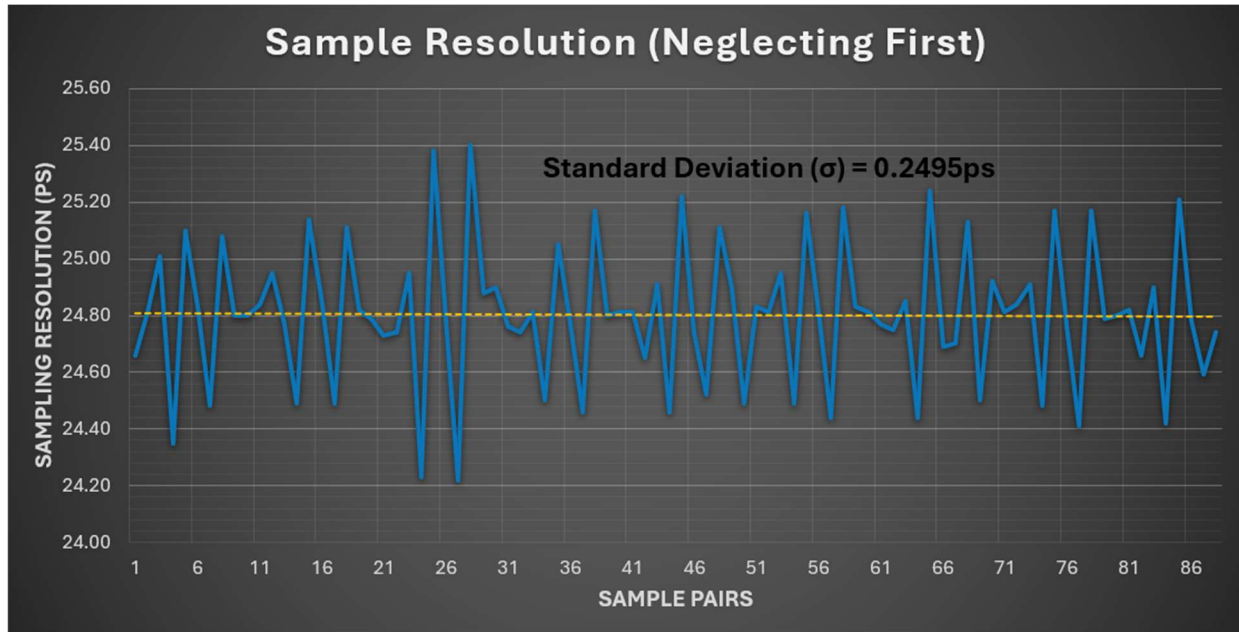


Figure 5.1: Plotted simulation data of the time delays between rising edges of each clock signal from the DLL, including the calculated standard deviation of each time delay.

The data in this plot is an excellent result to provide useful sampling capability to the AAD collaboration. The simulation that produced this plot is a single transient simulation performed at room temperature (27°C). It is accounting for the parasitic capacitance and resistances contributed by the layout interconnects. The variation in the circuit timing is likely the result of fluctuations of the reference signal off the gate terminal of each successive pseudoinverter delay element in the VCDL due to capacitive coupling. A full finite element model accounting for process corners and temperature variation would provide more robust prediction of actual chip performance. That said, even after accounting for a small amount of additional degradation from practical nonidealities there is high confidence that FPS2 will be capable of delivering <30 ps sampling resolution with excellent deterministic jitter performance. Minimizing jitter here is critical to ensure consistent resolution from sample to sample for FPS2. There are two factors that intuitively contribute to the sampling jitter. First is the timing error contributed by the DLL loop itself, imparted from sources such as mismatches in devices within the DLL. These sources of error are accounted for in the

simulation outputs plotted above. The second source of practical timing jitter comes from the intrinsic errors injected by the reference source. This jitter, on the other hand, is not factored into the plot. However, it can still be analyzed by inspection of the TCXO specification. According to [14], the ASGTX TCXO contributes a measured jitter of 911 fs_{RMS} in the 1 GHz LVDS configuration. Because the intrinsic jitter of the reference generator and the additive jitter contributed by the DLL are assumed to be uncorrelated, they can be combined by adding them in quadrature. Using the standard deviation calculation reported in Figure 5.1, an analysis approximating the combined jitter performance of the DLL and TCXO (neglecting random jitter of the DLL) is presented in detail in Figure 5.2.

$$Var(\Delta_t) = Var(t_{n+1}) + Var(t_n)$$

$$\sigma_{\Delta}^2 = \sigma_t^2 + \sigma_t^2 = 2\sigma_t^2$$

$$\sigma_t = \frac{\sigma_{\Delta}}{\sqrt{2}}$$

Additive sample jitter

$$RMS\ jitter = \frac{\sigma_{\Delta}}{\sqrt{2}} = \frac{0.2495ps}{\sqrt{2}} \approx \mathbf{176fs}$$

Carrier	RF Output	rms Phase Jitter
10.00MHz	LVDS	992 fs
10.00MHz	LVC MOS	906 fs
25.00MHz	LVDS	774 fs
25.00MHz	LVC MOS	754 fs
50.00MHz	LVDS	768 fs
50.00MHz	LVC MOS	999 fs
120.00MHz	LVC MOS	1.1 ps
500.00MHz	LVPECL	956 fs
1.00GHz	LVDS	911 fs
1.2890625GHz	LVDS	1.03 ps
1.50GHz	LVDS	1.55 ps

$$\sqrt{\sim(176fs)^2 + (911fs)^2} = \mathbf{927.8fs}$$

Additive DLL jitter

Intrinsic TCXO jitter

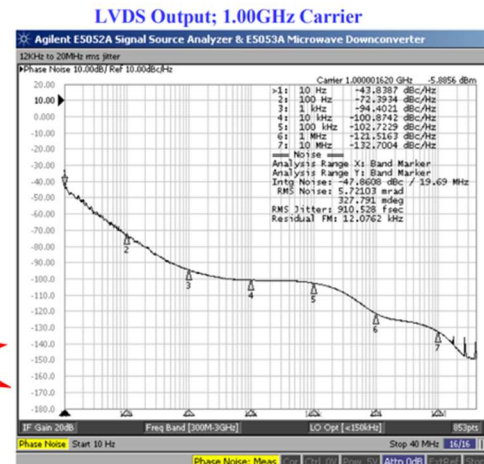


Figure 5.2: A calculated approximation for the overall FPS2 sampling jitter performance.

The conclusion here is that the combined jitter performance of the DLL and the reference is very good for a practical system. That said, as will be discussed in the next section, there is room for improvement if future design iterations are attempted. According to discussions with the physicists in the AAD collaboration, a 100 fs_{RMS} jitter figure is the desired threshold to enable leading edge research in the field of high energy physics.

The second variety of jitter investigated through simulation of the sampling circuitry is the variability in the widths of the sampling windows for each cell in the SCA, henceforth referred to as period jitter. This jitter is quantified by deriving variance observed in the simulated full-width half-max (FWHM) duration of each output pulse from the non-overlapping pulse generator as it is driven by the replica sampling line during each sampling event. The source of this error in FPS2 is likely to come from mismatches and parasitics impacting the non-overlapping pulse generation hardware, with some second order impact from the sampling jitter coming from the DLL. Like the additive DLL jitter presented previously, the period jitter is accounted for using the same parasitic extraction approach in simulation as described in the previous paragraph with Figure 5.1. The resulting plot is shown in Figure 5.3. Notably, each sample is only ~4 ps wider in simulation than the adjacent sample delays shown in the data from Figure 5.2. This implies sample overlap has been effectively minimized to about 2 ps at the start and end of successive samples.

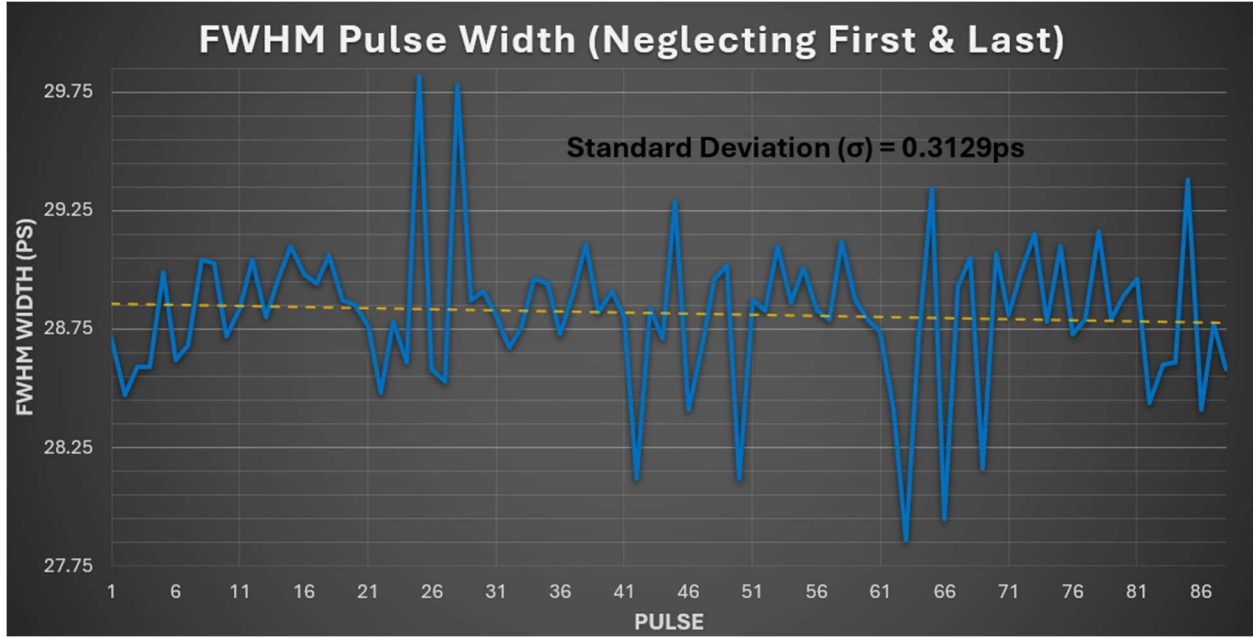


Figure 5.3: Plotted simulation data describing the FWHM duration of each sample pulse annotated with the calculated standard deviation.

It is expected that layout parasitics will increase the width of each sample in a similar manner to the sampling jitter. Through analysis of the standard deviation of the sampling periods, it is possible to derive the corresponding RMS period jitter. This derivation is included in Figure 5.4.

$$\begin{aligned}
 Var(\Delta_t) &= Var(t_{n+1}) + Var(t_n) \\
 \sigma_{\Delta}^2 &= \sigma_t^2 + \sigma_t^2 = 2\sigma_t^2 \\
 \sigma_t &= \frac{\sigma_{\Delta}}{\sqrt{2}} \\
 RMS\ jitter &= \frac{\sigma_{\Delta}}{\sqrt{2}} = \frac{0.3129ps}{\sqrt{2}} \approx \mathbf{221fs}
 \end{aligned}$$

Period jitter

Figure 5.4: Derivation of the simulated RMS period jitter in FPS2.

Granting that further mathematical analysis is required to investigate the interaction between all sources of jitter presented in this section, this is a good result. It indicates that this is a highly

functional chip design that can make repeatable measurements to gather reliable scientific data for the AAD collaboration. In this regard, the replica DLL design methodology can be considered a success.

Time-domain jitter is not the only form of error requiring consideration when designing DLLs. As discussed in [6], it is also common for them to be susceptible to duty cycle distortion, and the DLL on FPS2 appears to suffer from this phenomenon in simulation. Its cause is a disparity between the rise and fall times of the delay cells. If the rise and fall times are not equivalent, the duty cycle of the reference signal will expand or contract as each phase traverses the delay lines [6]. If the rise time exceeds the fall time for each delay cell at steady state, the signal will take longer to go from a low to high logic level. Conversely, the signal is faster to leave a logic high state, resulting in a progressive compression of the duty cycle as the signal passes through each delay cell in the DLL. If instead the opposite were true and the rise time is faster than the fall time, the result is additive distortion that expands the duty cycle. To counteract this effect, designers often include a duty cycle correction block in DLL applications. The DLL on FPS2, however, does not incorporate any duty cycle correction, and because of its relatively long line length of 97 delay cells including the buffering elements on either end of the VCDL, it suffers greatly from duty cycle distortion. In simulation, its impact is immediately apparent by visual inspection, as can be seen clearly in the overlaid plots of the VCDL input and output traces at equilibrium in Figure 5.5. The duty cycle of the VCDL output is clearly much greater in comparison to the input trace.

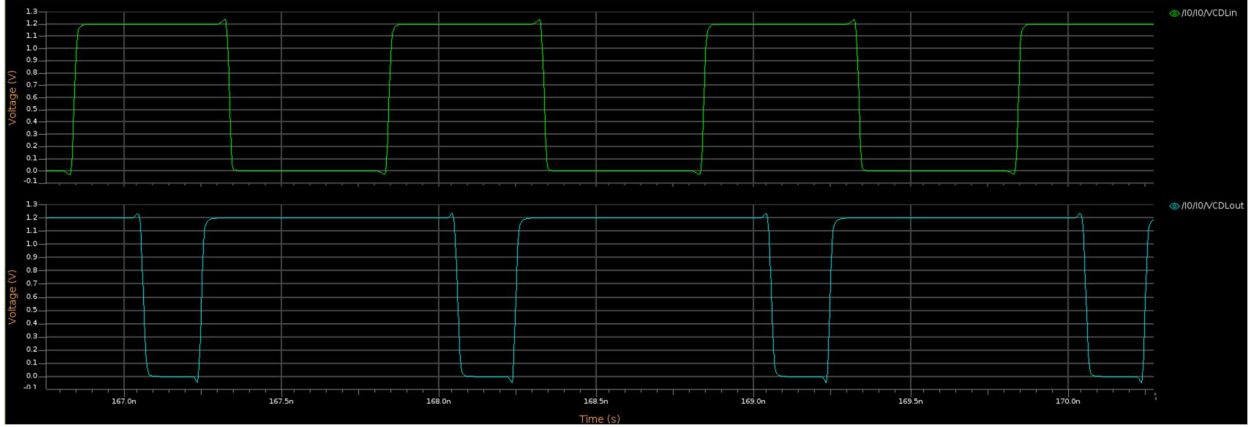


Figure 5.5: Steady state transient simulation of the locked FPS2 DLL showing the differing duty cycles of the VCDL input (green) and the VCDL output (cyan).

As described previously, the cause of this distortion is an imbalance in the rise and fall times of the delay cells. It follows that, at equilibrium with the DLL's control voltage level, the transconductance of the NMOS device must be greater than the combined G_m of the two PMOS devices in each pseudoinverter. Consequently, at the shared node between the transistors the voltage fall time is less than the rise time. Recall that this node drives an inverting clock buffer that provides the output of each delay cell. Incorporating the effect of this inverter, each delay cell exhibits the inverse characteristics of the transistors alone. Each delay cell therefore exhibits faster rise time and a slower fall time, resulting in large positive duty cycle distortion as this effect is integrated across the entirety of the VCDL.

The duty cycle distortion impacting the FPS2 DLL is problematic for a variety of reasons. One of the most apparent issues is the impact that it has on the stability of the control voltage when the loop is operating at a steady-state operating point. Even in a locked state, small periodic voltage swings are still evident on the control voltage in simulation that can be attributed to the difference between the reference duty cycle seen at the input and output of the VCDL. Instability can be observed in the plot included in Figure 5.6. Inspection of these traces indicates that the regions of

different slope in the control voltage trace overlap with the characteristics of the superimposed reference input and output signals when phase locked.

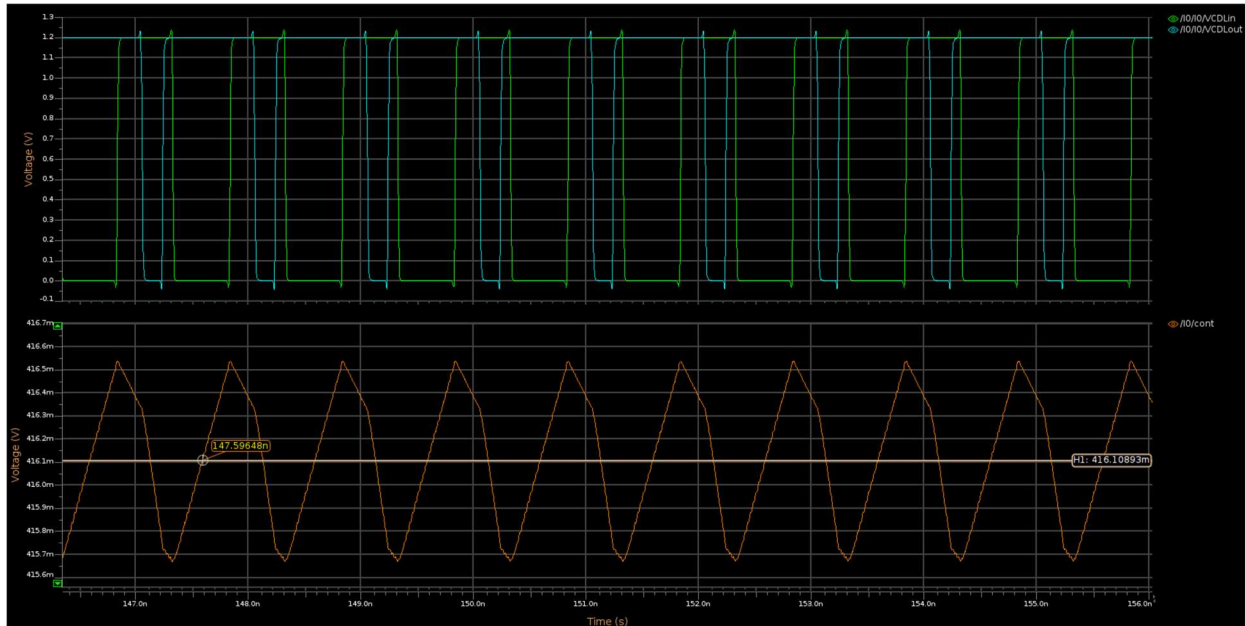


Figure 5.6: Transient simulation highlighting periodic swing in the orange control voltage trace.

The root cause for this small instability is suspected to be reference duty cycle distortion in combination with a PFD optimized for high-speed operation. The PFD circuit described in the previous section has been designed to phase lock two signals with equivalent duty cycles. Phase locking two signals with drastically different duty cycles is not an application it has been designed for. In this case, the high-speed capability of the PFD causes it to be highly reactive to the duty cycle error, preventing the DLL from performing optimally. Luckily, this is not a showstopper for the FPS2 application. It must be emphasized that the simulated voltage swing is quite small at about 0.8 mV_{Pk-Pk}. The average value of the control voltage equates to a DC level of about 416 mV. The loop evidently achieves and maintains phase lock, allowing for consistent functionality as intended by design. The small swings on the control voltage will have a minimal impact on the speed of the reference signal's propagation through the delay line. Regardless, adjustments were made to the

DLL to reduce the instability as much as reasonably possible. Careful balancing of the charge pump control currents and enlargement of the capacitance used on the loop filter both serve to smooth the control voltage swings. Additionally, the transient simulation presented in Figure 5.6 does not account for the contributions of layout parasitics. The additional capacitance from the physical devices and their interconnects will work in favor of the design here by further filtering the periodic voltage swings being fed back to the DLL on the control signal.

Another complication caused by the duty cycle distortion of the reference signal that is worth analyzing here is how it potentially interferes with the sampling circuitry. By virtue of FPS2's reliance on replica biasing to condition the control voltage for low-jitter sampling, the same duty cycle distortion observed on the DLL will be conveyed to the sampling line. Indeed, this result is borne out by transient simulation of the travelling sample pulse shown in the plot presented in Figure 5.7.

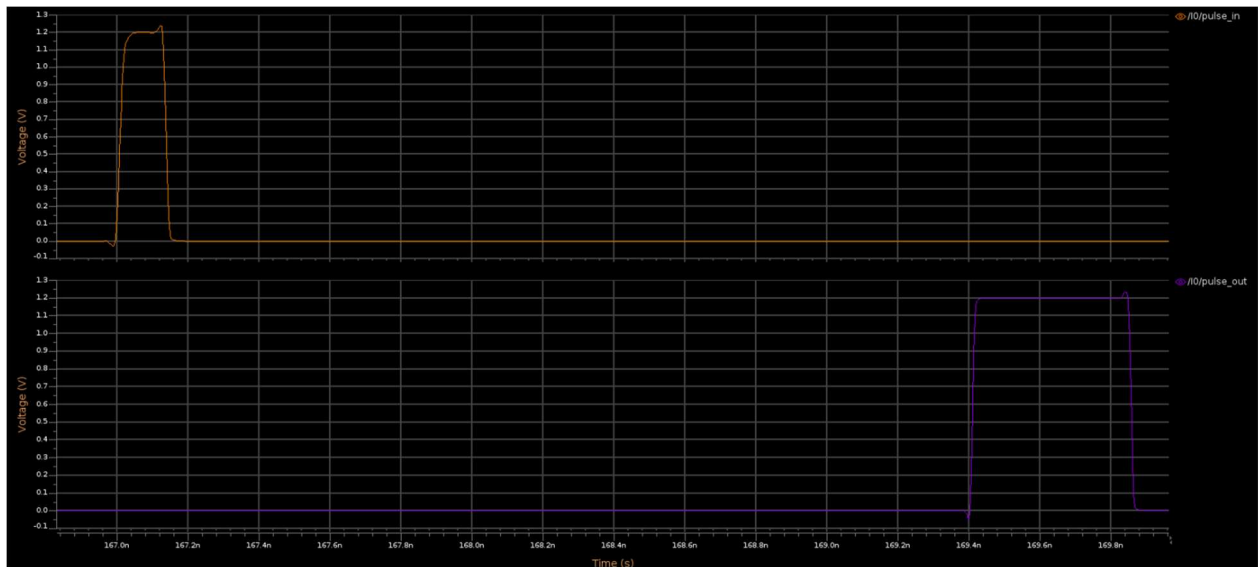


Figure 5.7: A simulated visualization of the duty cycle distortion effecting the sampling pulse. The input pulse is shown in orange, and the heavily distorted output pulse can be seen in purple.

The output pulse has clearly widened from the corresponding input pulse. In one sense, this result is encouraging because it validates the design philosophy of the replica biasing approach with the on-chip DLL. On the other hand, though, it is less than ideal for uniform sampling of diamond quadrant sensors with the FPS2. As the pulse travels down the line and gradually expands in width, it would introduce systematic distortion in the samples toward the rear of the capacitor array because their gating switches would be closed for a greater duration of time in comparison to the beginning of the array. This makes the decision early in the design process to add a non-overlapping pulse generator quite fortuitous since the duty cycle distortion on the sampling line makes it a necessity for FPS2 to function as intended. The pulse generator winds up acting as a buffer to eliminate the sampling distortion imparted by this duty cycle error. Because each element in the pulse generator array has been tuned to be sensitive to only the rising edge of each pulse, it ensures that each sample will store sampled charge from the sensor preamplifiers in equivalent time increments. While the original motivation to use this approach was a hardware solution to correct sampling overlap observed in FPS1, it is useful here to insulate the sampling arrays from the DLL's duty cycle error.

Even though it is fortunate that the non-overlapping pulse generator array mitigates the impact to the multi-phase sample timing by the expansion of the sampling pulse, this is not the only adverse effect brought on by the duty cycle distortion contributed by the replica DLL. As the travelling pulse expands over the sampling period, the increasing width exerts a progressively stronger deflection on the control voltage. This is ostensibly caused by the sample pulse coupling onto the control voltage node of the DLL over the drain-gate capacitance of the sampling pseudoinverter elements as pictured below in the annotated schematic presented in Figure 5.8.

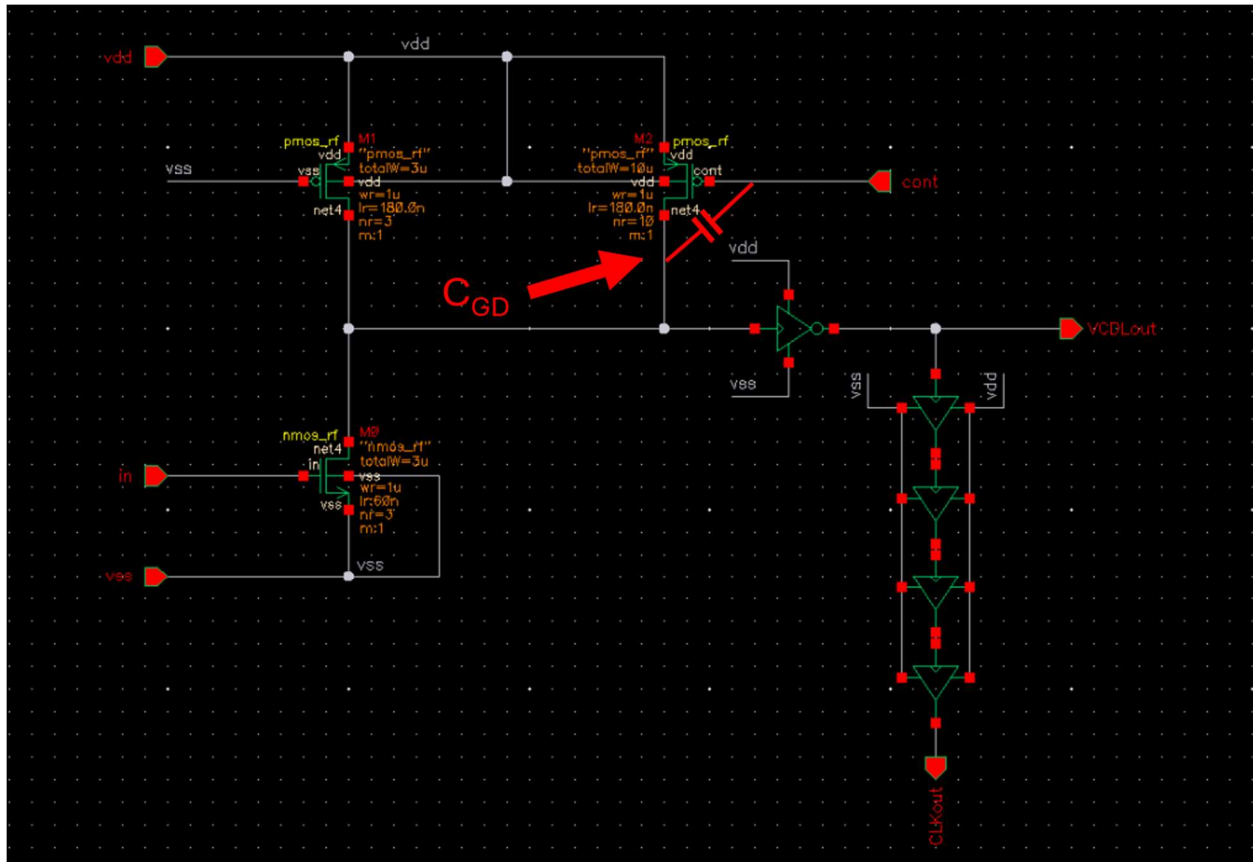


Figure 5.8: The pseudoinverter schematic described in the schematic section, annotated to show a parasitic gate-drain capacitance, C_{GD} .

As each pulse travels down the sampling line, the inverting action of the NMOS transistors causes the node voltage on one end of the parasitic capacitance of the PMOS control device to be pulled low. It follows that this voltage will couple onto the control voltage line for the DLL across the parasitic C_{GD} , resulting in a reduction of the control voltage regulating the entire system. This effect is confirmed in Figure 5.9, showing the transient plot of a simulated sampling event.

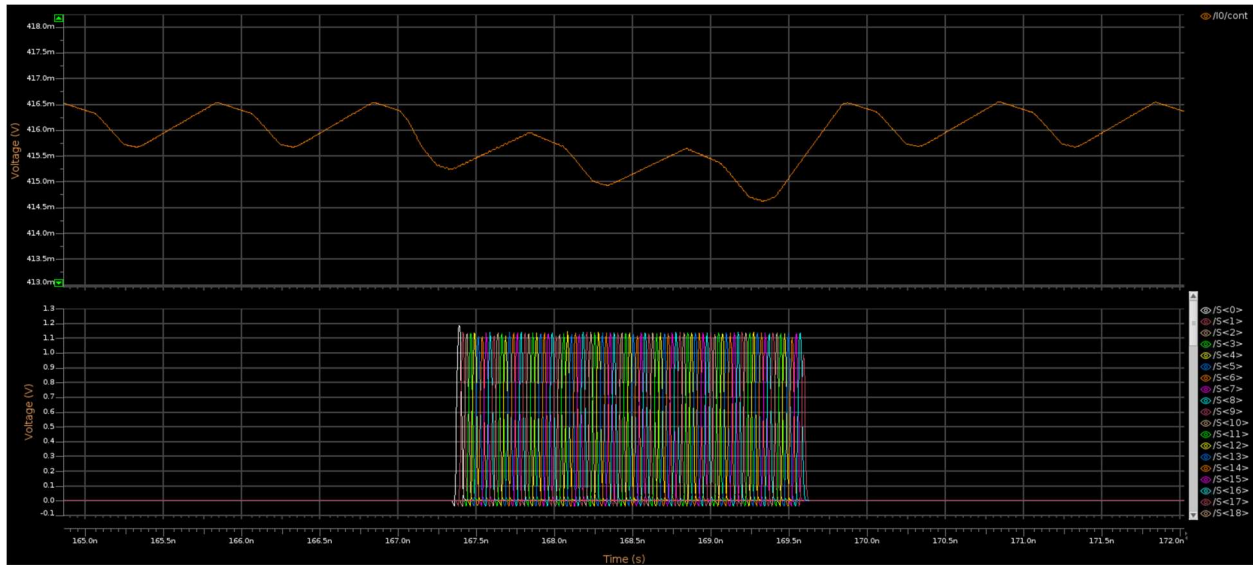


Figure 5.9: A simulated FPS2 sampling event. The non-overlapping pulses are shown to be well-shielded from duty cycle expansion of the sample pulse, but the impact of the coupling effect is evident on the control voltage trace in orange.

The gradual increase of this pulling effect over time is readily apparent looking at a narrow voltage range on the topmost Y-axis above. Since the control devices in each pseudoinverter element are PMOS transistors, lowering this voltage increases the gate-source voltage for every delay cell in both the sampling line and the replica DLL alike. In turn, every element will source an increased bias current, causing the entire system to speed up slightly. Crucially this effect is shown to be relatively small, no more than a millivolt from inspection of Figure 5.9. Therefore, the functionality of the sampling mechanism is only minimally impacted. Similarly to the small voltage swings in control voltage imparted by the duty cycle distortion, layout parasitics are assumed to dampen the influence this issue has on the operating point of the system overall. Nonetheless, the coupling effect identified here is a vulnerability for the pseudoinverter delay cells in the FPS2 application, at least to some extent.

Another well-known deficiency in some DLL designs is their propensity to settle in a state referred to as false lock. This is a state in which a DLL will stabilize having locked to an integer fraction or multiple of its input reference frequency, according to an analysis in [6]. The architecture of the FPS2 insulates it from false lock by design using independent pulse generators at the reference and VCDL inputs of the bang-bang PFD. VCDL outputs corresponding to integer multiples of the reference period will produce fewer pulses over time and the PFD will consequently drive the loop through these points where false lock may occur. Reset capability has been added to the ASIC to ensure that a consistently repeatable operating point can be achieved between interruptions of the power supply. The reset circuitry uses a single PMOS transistor with a reset-low input signal tied to its gate as depicted in Figure 5.10.

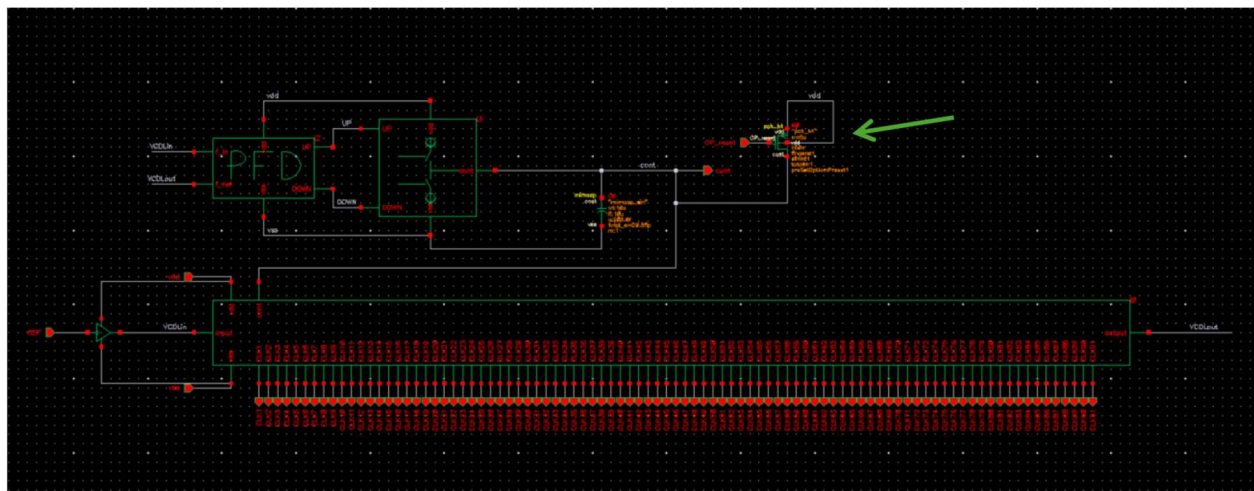


Figure 5.10: Circuit schematic showing the inclusion of a PMOS switch to pull the DLL control voltage up to V_{DD} to reset the loop as needed.

Once the reset signal is asserted by pulling it low, the control voltage will be pulled up to V_{DD} by the action of the PMOS reset transistor with its source connected to the power supply. With the control voltage set to V_{DD} the reference signal will be fully attenuated by the delay line, and the loop will remain inert in a reset state. When the reset signal is pulled high the PMOS switch cuts off and the

control voltage will be adjusted by the usual operation of the PFD and charge pump, gradually decreasing until the loop locks at steady state. This is a relatively crude solution to set the operating point of the DLL in comparison to other methods outlined in technical literature like [6], such as an integrated replica PLL with control voltage interpolation to avoid false lock. The proposed approach has the added benefit of effectively shutting off the DLL loop when the reset state is asserted. Accordingly, it can put the system into standby with minimal power consumption if necessary.

Future Improvements

The addition of a DLL to regulate the sampling control voltage on FPS2 provides a variety of improvements over the open-loop control used in FPS1. At the same time, as is the case with anything designed in an iterative approach, there are a handful of aspects of this DLL-based ASIC that can be improved if additional revisions are pursued in the future. Specifically, there are two areas of the design that will be identified as candidates for future refinement in this work. The first, and perhaps most obvious, would be to update the core DLL itself to increase its overall stability while reducing its jitter contribution and reference signal distortion. The second is to adjust the reference generator to improve the baseline performance of the system by reducing jitter inherent in the reference signal. Notably, the upgrades to the reference source proposed in this work would incur a large amount of additional cost and complexity, but these would undoubtedly be offset by the performance upside that can be realized with the proposed changes. According to the leadership team of the AAD collaboration, the magnitude of the cost increases brought by a higher performance oscillator is relatively minor juxtaposed with the expense of operating an XFEL beam. If the previously noted improvement opportunities are pursued in combination, the theoretical enhancement in system performance would be significant.

Considering the difficulties imposed by the duty cycle distortion outlined in the previous section, inserting a duty cycle correction block in the DLL architecture would be an instinctive corrective action that would help reduce the small periodic variability of the DLL control voltage it causes. According to [8], it is most common for designers to simplify the task of duty cycle correction through the addition of a dedicated block at either end of the VCDL as opposed to addressing the problem with an overly complicated delay line design. Duty cycle correction (DCC) circuits operate using a negative feedback principle comparable to that used to achieve and maintain phase lock in the DLL loop. One example published by a collaboration of researchers in [15] uses a differential architecture and is implemented in a 130 nm CMOS process. In it, pulse width modification cells (PMCs) alter the duty cycle of each input individually. The PMCs are controlled by a differential amplifier, which is driven by a duty cycle detector as depicted in the block diagram in Figure 6.1.

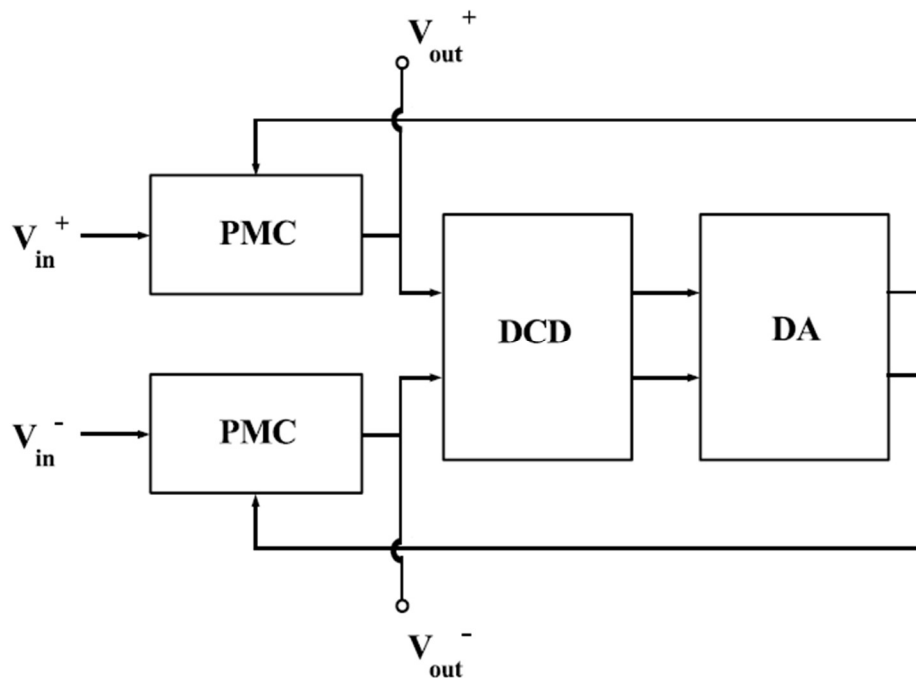


Figure 6.1: The block diagram of a duty cycle correction circuit proposed in [15].

Each PMC is essentially a buffer with integrated current starvation to skew the threshold in accordance with the detected duty cycle error. This equalizes the rise and fall times and counteracts the duty cycle error. Another publication, [16], demonstrates how a duty cycle correction block fits into a high-speed DLL system. The block diagram from that publication can be seen in Figure 6.2.

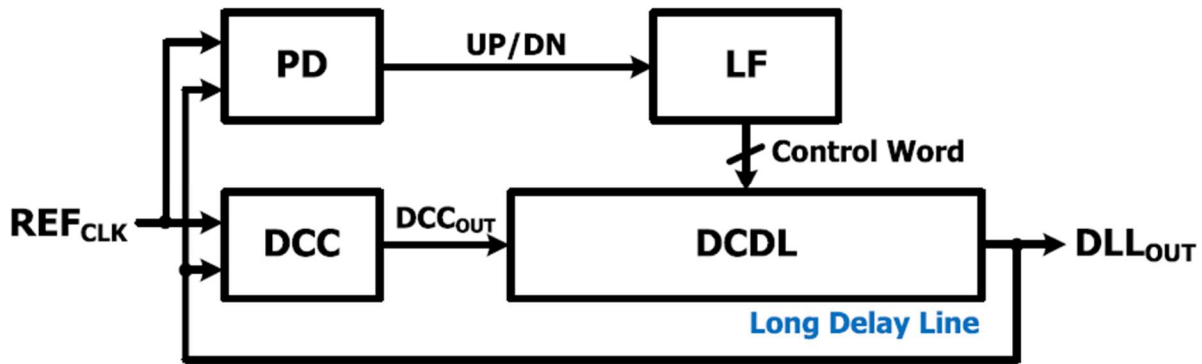


Figure 6.2: A system block diagram for a DLL with integrated duty cycle correction described in [16].

This diagram is applied in a digital DLL whereas FPS2 is an analog system, but the diagram is still useful to illustrate how a DCC block would be incorporated into a practical DLL. Here a DCC block is used to precondition a low duty-cycle input reference signal before phase locking to the corrected output. It follows that a similar approach for the FPS2 can be applied with the DCC block at the output of the pseudoinverter chain to facilitate a more robust phase lock using a 50% duty cycle output. Theoretically, the control voltage would exhibit much smoother steady state behavior with the reduction in duty cycle distortion between the reference input and output in FPS2.

Alternatively, reworking the topology of the FPS2 delay cells has been shown to insulate the design against duty cycle distortion without the use of a DCC block. This would also have beneficial effects for other areas of the FPS2 design. For instance, a source-coupled differential pair delay element can be used. In [8], Yang notes that single-ended delay lines can be susceptible to

common-mode noise and in many cases differential implementations are therefore preferable.

Retrospective simulations explored after the FPS2 tape out taking this fact into account used the differential delay element shown below in Figure 6.3.

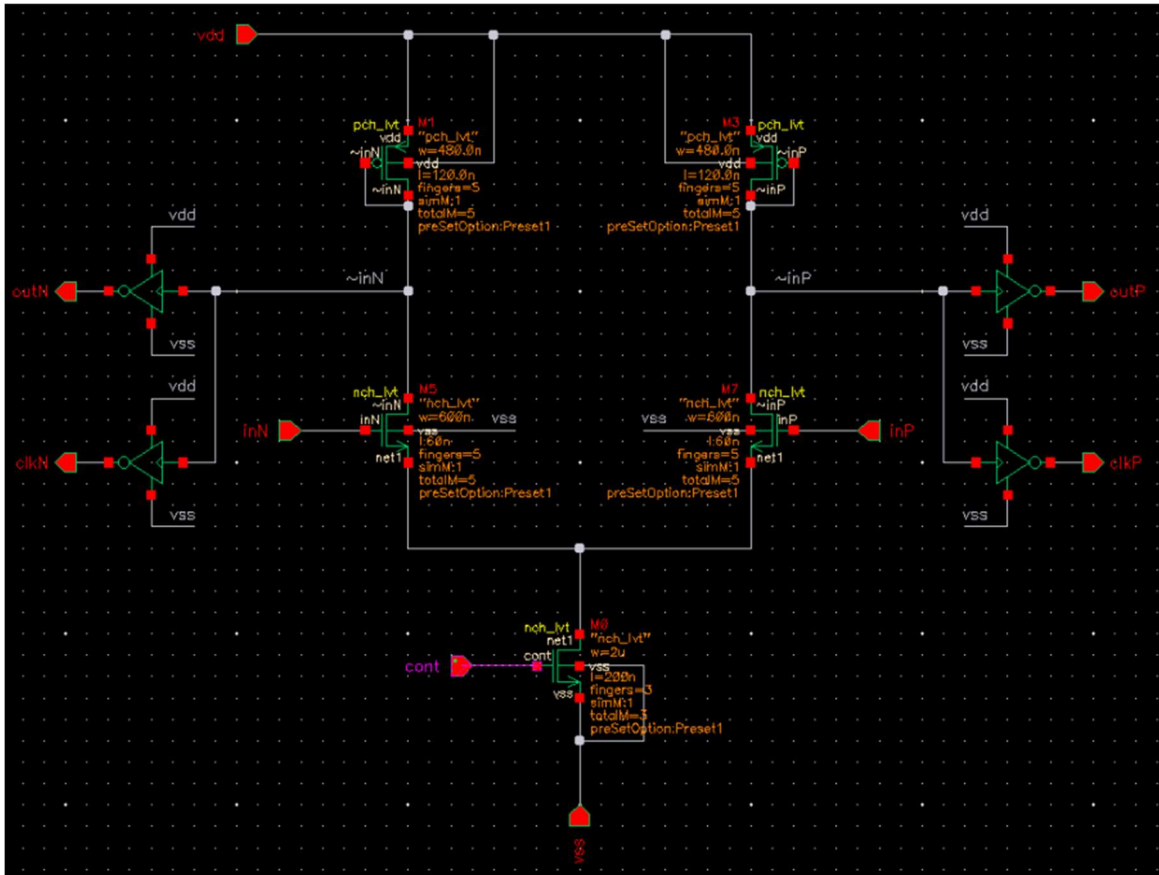


Figure 6.3: A differential delay line topology used to replace the FPS2 pseudoinverter delay cells in retrospective simulations.

The associated delay in this approach is modulated by varying the gate bias of the tail current source. Increasing the voltage here causes the delay to decrease because of the increase in bias current sunk through the source-coupled node. While the pseudoinverter demonstrated in FPS2 varies the delay through a similar mechanism, the balanced structure afforded by this differential version offers a variety of advantages. Chief among these is a drastic reduction in duty cycle distortion as indicated by the simulated results shown in Figure 6.4.

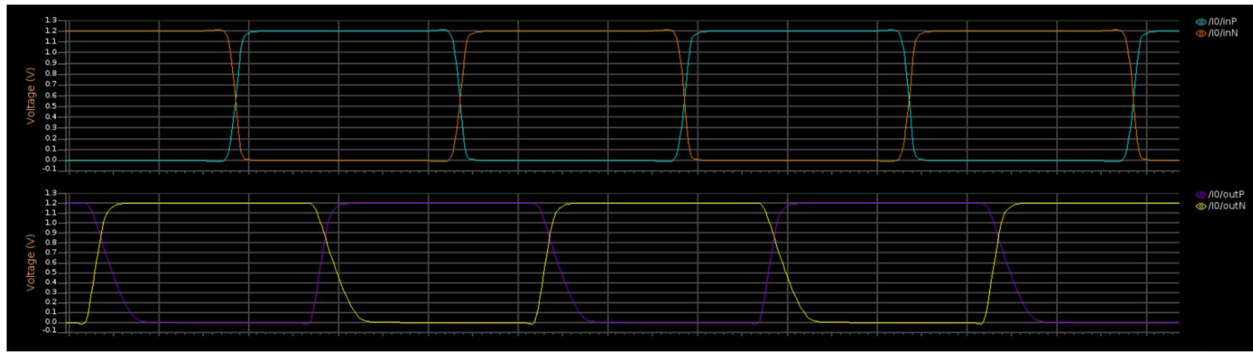


Figure 6.4: A transient simulation output of a steady-state DLL incorporating the differential delay line topology shown in Figure 6.3 showing the reference input pair (top) overlaid on the reference output (bottom).

Other than swapping the pseudoinverters for differential topology delay cells, the circuit used to produce the simulation in figure 6.4 is completely identical to that of the FPS2 DLL. The improvement in duty cycle distortion observed here is so profound that a duty cycle correction block likely is not necessary in this configuration. Consequently, the pulse width expansion in the replica sampling line will experience a similar improvement. Although it was out of scope for this after-the-fact analysis, it is likely the control voltage coupling observed with the pseudoinverters will also be heavily attenuated by a differential VCDL like the one in Figure 6.3. The presence of the NMOS transistors between the nodes carrying the sample pulse and the delay control devices of each element will shield the control voltage from a large part of the control voltage deflection shown to be a weakness of the pseudoinverters during each sampling event. Furthermore, scrutinizing the control voltage behavior of the differential DLL in a locked state in Figure 6.5 shows more than an order of magnitude reduction in the peak-to-peak amplitude of the periodic swings of the control voltage. Where the pseudoinverter showed about 800 μV of peak-to-peak swing on the control voltage, the differential system has only 14 μV .

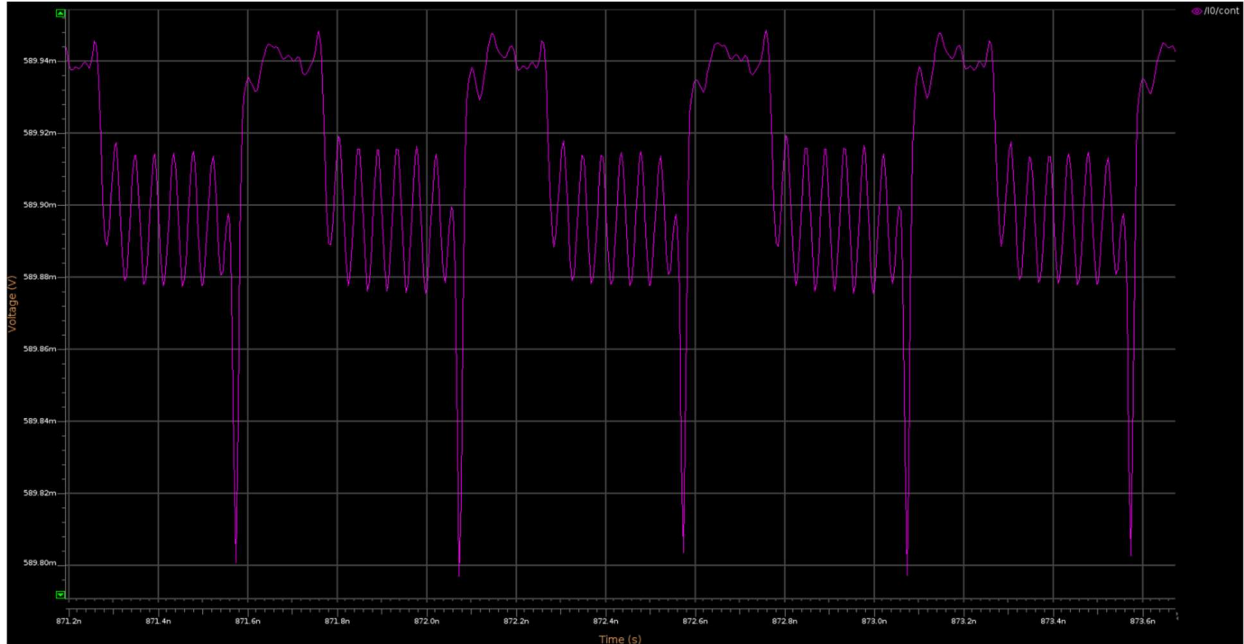


Figure 6.5: The steady state control voltage of the differential DLL in the discussed retrospective analysis.

This is clearly a more stable system in comparison to the behavior of the single-ended system shown in Figure 5.6. It is also possible that a differential DLL will contribute less additive jitter than a single-ended approach due to its insulation from common-mode noise, although further simulation will be required in the future to confirm if this is the case. Yet one more advantage of the differential delay line will be its ability to directly drive the complementary switches gating each capacitor in the sampling array. Detailed in [5], complementary switches are used in FPS2 to reduce the impact of pedestal error on the sampled charge, and the current design requires both inverting and non-inverting clock buffers to actuate these gate voltages from a single-ended timing signal. While this is admittedly more of a peripheral benefit in comparison to the reduced duty cycle distortion and possible jitter reduction, it is noteworthy to be able to eliminate the additional circuitry to minimize sources of error from fabrication offsets and free up a small amount of die area.

Independently of the DLL improvements that have been identified, performance gains over the current FPS2 design can also be obtained through adjustments to the reference generator. As

established in the Reference Generation and Offset Regulation section, most of the cumulative jitter impacting the sample timing is the result of jitter contributed by the practical reference signal delivered by the TCXO. While the ASGTX TCXO is by no means a low-performance selection for an oscillator, there are competing options available commercially that would provide considerable performance improvement. The TCXO used in the proposed reference generator could feasibly be replaced by an oven-controlled crystal oscillator, or OCXO. This type of oscillator incorporates a miniaturized oven enclosure specifically designed to hold their quartz crystal resonators at a temperature where they are optimized for their lowest noise performance, referred to as the crystal's turnover point. This temperature is typically on the order of about 90 °C for most crystals. The lowest phase noise OCXOs utilize SC-cut quartz crystals, which trade off frequency pull range for phase noise jitter performance. Because phase noise in the frequency domain corresponds to jitter in the time domain, SC-cut OCXOs will offer better performance for the FPS2 application. The limitation with this approach is that quartz crystal oscillators do not generally operate up to the required 1 GHz reference frequency needed to drive the current DLL. Additionally, high-performance OCXOs usually provide a sinewave output waveform, so additional circuitry will be needed to generate a square wave from a high purity sinewave. All this said, redesigning the reference source to use a 1 GHz OCXO will surely result in a substantial jitter reduction in comparison to the $911 \text{ fs}_{\text{RMS}}$ specified for the 1 GHz LVDS ASGTX in [14]. As discussed, this is an unusually high frequency for an OCXO, but there are suitable options available on the market. One attractive option is the O-CEGM 1 GHz OCXO supplied by NEL Frequency Controls, a subsidiary of Abracon specializing in ultralow phase-noise oscillators. This product is marketed to customers focused on low phase-noise applications such as high-performance radar systems or commercial test and measurement equipment. As such, [17] does not specify time-domain jitter but rather frequency-domain phase noise. As a result, mathematical conversion is needed to determine the

equivalent jitter performance. This derivation is detailed in Figure 6.6, using a linear approximation of the anticipated phase noise performance across each decade of offsets and integrating to find the composite jitter performance this corresponds to.

1GHz OCXO RMS jitter:

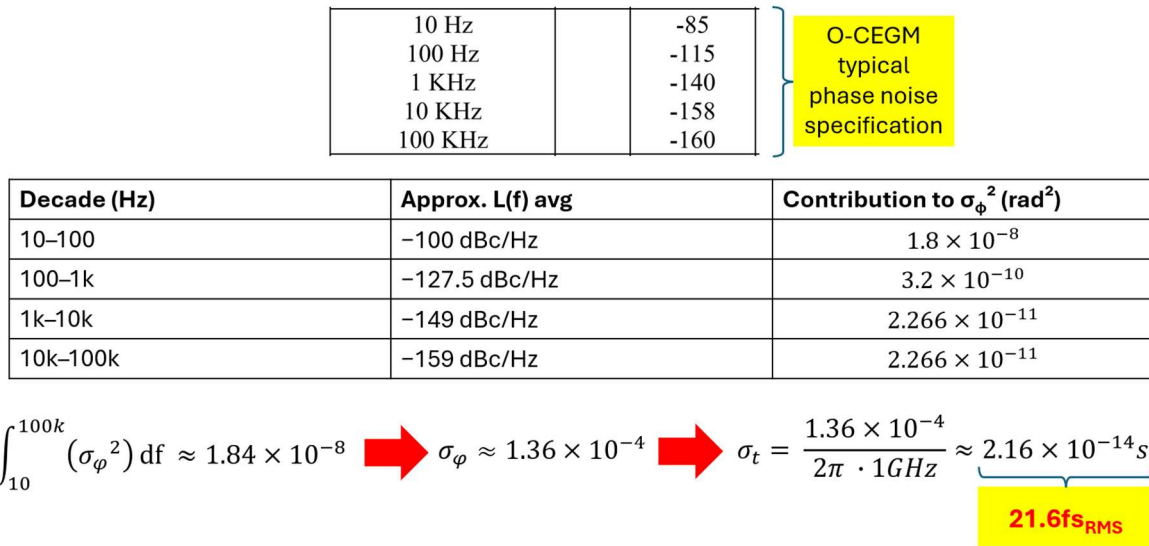


Figure 6.6: Conversion of O-CEGM 1GHz OCXO phase noise specification to approximate RMS time jitter.

Once again assuming the OCXO jitter and DLL additive jitter are random and uncorrelated, the combined jitter performance can be obtained by adding the two RMS values together in quadrature. This operation is demonstrated in Figure 6.7, illustrating the hypothetical performance of FPS2 locked to a 1 GHz OCXO reference signal.

$$\sqrt{\sim(176fs)^2 + (21.6fs)^2} = \mathbf{177.3fs}$$

↑

Additive
DLL
jitter

↑

Intrinsic
OCXO
jitter

Figure 6.7: Quadrature addition of jitter sources to estimate performance of an OCXO-based FPS2.

Based on the previous analysis of the TCXO-based reference generator using the measured jitter from the ASGTX in Figure 5.2, the estimated FPS2 system jitter is about $927.8 \text{ fs}_{\text{RMS}}$. In comparison to the O-CEGM, the DLL becomes the dominant source of jitter, resulting in an 80% reduction in clock jitter for the system-level design. For future development of the FPS2 system, this is clearly an advisable change to execute.

Although the jitter contribution of the LDO was not considered in the preceding calculations, it should be assumed to have an impact on the jitter performance of the reference source just like any other practical circuit component. Recalling that the LDO is used in this project to set the correct DC offset voltage for the reference signal input to the LVDS receiver on the ASIC, any noise on its DC output will translate to jitter in the system if it causes the rising and falling edges of the reference waveform to vary slightly over time. Fortunately, the TI TPS7A94 LDO is already a very low noise device by design, specified in [12] to be typically $0.46 \mu\text{V}_{\text{RMS}}$ over the 10 Hz-100 kHz band. What's more, if the LDO as a noise source is considered uncorrelated with both the oscillator and the DLL loop, what little noise it does inject will come out in the wash after quadrature addition at the system level. It is quite possible the jitter the LDO will add will be dominated by other parts of the system, most likely the TCXO which itself exhibits $911 \text{ fs}_{\text{RMS}}$ of jitter. Either way, the TPS7A94 datasheet identifies a method to reduce the noise of the LDO if an application demands it by using multiple regulators in parallel as pictured in the schematic in Figure 6.8.

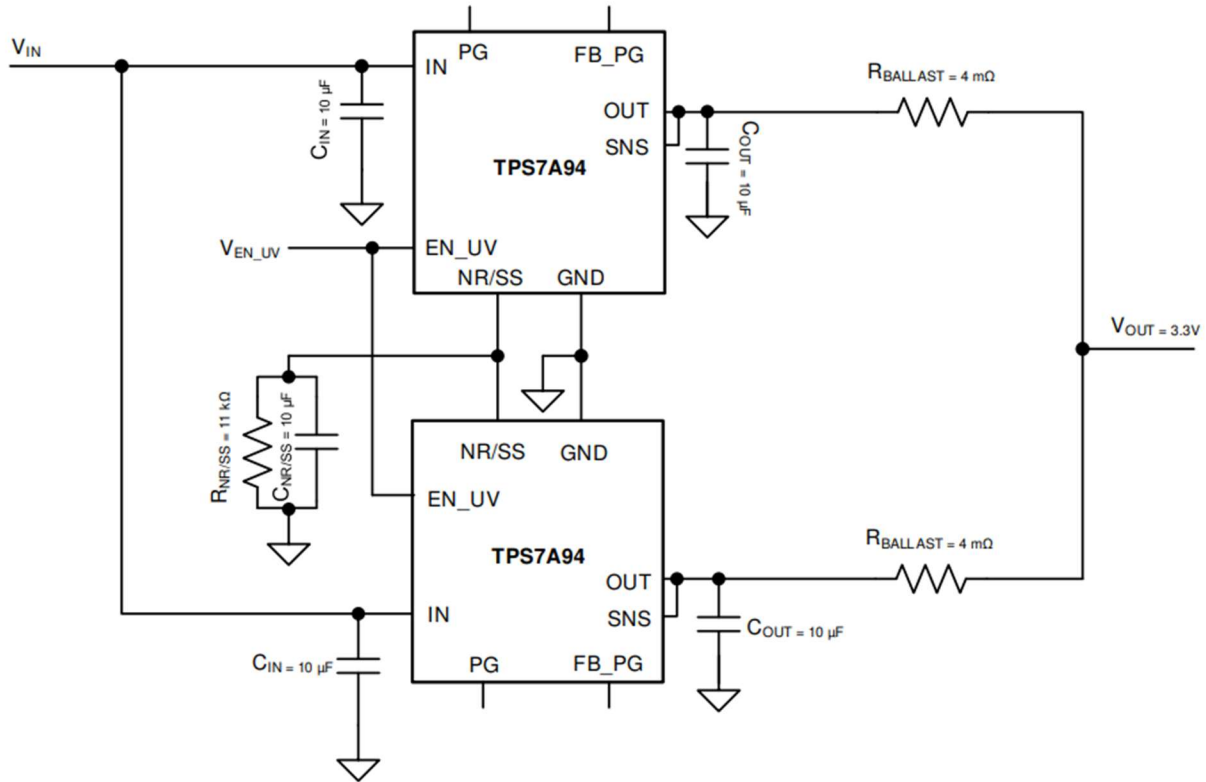


Figure 6.8: A parallelized LDO circuit suggested in [12] to reduce output voltage noise if necessary.

Using parallelized LDOs, the specification for the noise performance of a single LDO can be reduced by a factor of $1/\sqrt{2}$. If necessary, it follows that 4 could be used in parallel to pursue reduction by an additional factor of $1/\sqrt{2}$. Further analysis of the LDO is needed to evaluate the extent to which noise reduction like this will benefit the performance of the FPS2 system. If it is helpful, it can be applied to both the current TCXO-based reference generator as well as an upgraded OCXO-based circuit seeing as both will require offset voltage regulation to properly drive an ASIC designed on a 65 nm process node operating with a 1.2 V supply. Therefore, this is a change worth highlighting for consideration on any future design changes initiated for FPS2 and its associated circuitry.

Conclusion

The object of this work has been to deliver a waveform digitizing ASIC to serve as a sampling interface for a diamond quadrant sensor providing beam diagnostics for next generation cavity-based X-ray free electron lasers (CBXFELs). The data gathered from these sensors will be fed back to the control systems for the beam to direct the positioning of a set of diamond mirrors used to optimize beam recirculation in the cavity assembly. Given the incredibly short time scales involved, and the extremely narrow wavelength of the X-rays produced with modern techniques, it is vital to sample data to control these systems with minimal jitter distortion to enhance optimization of the stability and average brightness of the XFEL, thereby enabling leading-edge experimentation methods in high energy physics.

In the prototype IC designed in this work, this goal is met by iterating upon a proof-of-concept IC called the Fast Pulse Sampler 1, or FPS1, designed by a collaboration of engineers from UC Davis and Lawrence Berkeley National Laboratory. This adaptation of FPS1 introduces negative feedback from a replica DLL to control a sampling line for production of coherent multi-phase clock signals with an estimated 177 fs of RMS jitter. This new design was subsequently laid out in a TSMC 65 nm CMOS process under constraints that allow it to be seamlessly integrated with the preexisting chip architecture, reducing the time required for the redesign effort. The new design also includes a high-performance reference signal generator design. This reference source incorporates commercial off-the-shelf components, including a low-jitter temperature-compensated crystal oscillator (TCXO) and ultra-low noise low-dropout regulator (LDO), on the ASIC's carrier board to produce a 1 GHz reference signal (with approximately 911 fs of RMS jitter) that is properly conditioned to drive the on-chip DLL. Simulations and calculations on the new design predict that the combined jitter of the DLL and its reference source is approximately 928 fs_{RMS} .

The performance of the new prototype will soon be evaluated using the XFEL beam at the Stanford Linear Accelerator Center (SLAC) in Palo Alto, CA. In the interim, several minor limitations in the system functionality have been identified. Simulations indicate the single-ended design of the DLL is vulnerable to duty cycle distortion and undesirable control voltage coupling. Additionally, simulations show that the jitter of the core DLL is dominated by the intrinsic jitter of the TCXO used in the reference generator. In hindsight, these deficiencies can be improved upon if given the chance with follow-on design iteration. Duty cycle correction techniques exist that will reduce the distortion of the reference signal as it transits the DLL's lengthy voltage-controlled delay line (VCDL) of 97 delay elements. In lieu of duty cycle correction, the DLL can use a differential topology for the delay elements to reduce the impact of duty cycle distortion and control voltage coupling, all while rejecting the effects of common-mode noise to possibly reduce the additive jitter of the DLL even further. Perhaps the largest performance gains in the future can be achieved by redesigning the reference generator to use an oven-controlled crystal oscillator (OCXO) in place of the TCXO used for this work. A commercial 1 GHz OCXO has been identified that can reduce RMS reference jitter by more than an order of magnitude from 911 fs to less than 30 fs, an improvement that would lead to the DLL becoming the dominant source of jitter for future consideration.

While the purpose of the new prototype is to implement a sensor platform for physics research, it could feasibly be repurposed in a wide array of electronic applications requiring low-jitter waveform digitization such as communication systems, radar and lidar, power management, and biomedical diagnostics.

References

- [1] C. Pellegrini and J. Stohr, “X-ray free-electron lasers: Principles, properties, and applications,” Stanford Synchrotron Radiation Laboratory (SSRL), Stanford Linear Accelerator Center (SLAC), Stanford, CA, USA, Tech. Rep., 2001.
- [2] Accelerator Systems Division, “2020 Accelerator Systems Division Strategic Plan,” Advanced Photon Source, Argonne National Laboratory, Argonne, IL, USA, Jul. 2020.
- [3] G. Chui, “SLAC researchers take important step toward developing cavity-based X-ray laser technology,” SLAC National Accelerator Laboratory, Menlo Park, CA, USA, Aug. 15, 2023. [Online]. Available: SLAC National Accelerator Laboratory website
- [4] R. Padilla et al., “Development of diamond-based diagnostics for next-generation XFELs,” Nucl. Instrum. Methods Phys. Res., Sect. A, vol. 1057, Art. no. 168763, 2023, doi: 10.1016/j.nima.2023.168763.
- [5] T. Prakash and C. Rowling, “FastPulse Precision Sampler (FPS) Datasheet,” UCFEES, rev. 1.0, Jul. 31, 2023.
- [6] B. Razavi, “The delay-locked loop,” IEEE Solid-State Circuits Magazine, vol. 10, no. 3, pp. 9–15, Summer 2018, doi: 10.1109/MSSC.2018.2844615.
- [7] I. Alhousseiny, M. Ali, N. Ben-Hamida, M. Honarparvar, M. Sawan, and Y. Savaria, “Delay-locked loop based multiphase clock generator for time-interleaved ADCs,” in Proc. 28th IEEE Int. Conf. Electron., Circuits Syst. (ICECS), Dubai, UAE, Dec. 2021, pp. 1–4, doi: 10.1109/ICECS53924.2021.9665452.
- [8] C.-K. K. Yang, “Delay-locked loops—An overview,” IEEE J. Solid-State Circuits, vol. 34, no. 10, pp. 1341–1354, Oct. 1999.

- [9] Abracon, “Innovation for Tomorrow’s Design,” Abracon, Austin, TX, USA. [Online]. Available: <https://abracon.com/>. [Accessed: May 31, 2026].
- [10] Abracon LLC, “ASGTX-D-1.000GHZ-1: 1 GHz VCTCXO LVDS oscillator,” Digi-Key Electronics, Thief River Falls, MN, USA. [Online]. Available: <https://www.digikey.com/en/products/detail/abracon-llc/ASGTX-D-1-000GHZ-1/4754718>. Accessed: Apr. 25, 2026.
- [11] Y. Ibrahim, “*How to Terminate LVDS Connections*,” Texas Instruments, Application Report SLAA840, May 2018.
- [12] Texas Instruments, “TPS7A94 1-A, Ultra-Low Noise, Ultra-High PSRR, Low-Dropout Regulator,” datasheet SBVS336C, Sept. 2021, rev. June 2023.
- [13] Siemens Digital Industries Software, “*Eldo Platform: Advanced verification for analog-centric ICs*,” Fact Sheet 82842-C1, Oct. 2020.
- [14] Abracon LLC, “ASGTX Series Configurable High Performance SMD TCXO/VCTCXO,” Datasheet, Rev. F, Mar. 18, 2026. [Online]. Available: <https://abracon.com/>. [Accessed: Apr. 27, 2026].
- [15] I. Raja, G. Banerjee, M. A. Zeidan, and J. A. Abraham, “A 0.1–3.5-GHz duty-cycle measurement and correction technique in 130-nm CMOS,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 5, pp. 1975–1982, May 2016, doi: 10.1109/TVLSI.2015.2478804.
- [16] J.-H. Lim, J.-H. Bae, J. Jang, H.-K. Jung, H. Lee, Y. Kim, B. Kim, J.-Y. Sim, and H.-J. Park, “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*, vol. 63, no. 2, pp. 141–145, Feb. 2016, doi: 10.1109/TCSII.2015.2468911.

- [17] NEL Frequency Controls, Inc., “O-CEGM series low phase noise UHF OCXO in 36 × 27 mm ‘Europack,’” Datasheet no. 0907B, Rev. Y, Dec. 5, 2025. [Online]. Available: <https://www.nelfc.com>