A Digital Phase-Locked Loop for Frequency Synthesis using an Adaptive Pulse Shrinking TDC

Viktor Lewin
vi0787le-s@student.lu.se

Department of Electrical and Information Technology
Lund University

Supervisor: Mohammed Abdulaziz & Henrik Sjöland

Examiner: Pietro Andreani

May 29, 2023

Printed in Sweden
Tryckeriet i E-huset, Lund

Abstract

This thesis investigates a new type of Phase-Locked Loop (PLL) architecture which combines a phase/frequency detector (PFD) and a digital loop filter. The quantization is done by a time-to-digital converter which continuously shrinks the pulse coming from the PFD and registers how far it propagates. Based on how far into the TDC each pulse propagates, the phase error is determined. The design is focused on the PFD and TDC at a reference frequency of 4 GHz, targeting output frequencies in the mmWave range. Advantages using this type of TDC include reduced power consumption and phase jitter as well as simplified digital processing logic and high reference frequencies.
The TDC has adjustable gain with its highest resolution less than 1 pikosecond. By using an adaptive PFD, the width of the output pulses is made adjustable. Using this feature, the number of active stages in the TDC can be adjusted to account for corner variation while also reducing power consumption and phase noise. The PFD and TDC together consumes around 2.1 mW of power at 4 GHz, with phase jitter of less than 30 fs.

Popular Science Summary

Imagine one second passing by. A fairly short time. Divide that time by one thousand and we get 1 millisecond. That is about 1/100th of the time it takes for the eye to blink - in-humanely fast. Dividing the millisecond by 1000 yet again and we get 1 microsecond. Another division by 1000 gets us 1 nanosecond. During this time, light travels only 30 cm. The electronics in this thesis attempts to quantize one thousand of a nanosecond, i.e 1 picosecond!

Within the field of electronics, phase-locked loops (PLL) are heavily used for synchronization, frequency generation and unravelling messages encoded in oscillations in the electromagnetic fields that surround us. In its core, the PLL is rather simple. It works by comparing the phase of its output voltage to an input reference voltage oscillation. If the reference is oscillating at a higher pace, then the output frequency is increased to match it. Similarly if the input frequency is lower than the output, then the output frequency is reduced until the two signals are aligned. In other words, works by locking onto the phase of the input signal. The PLL typically consists of 3 components:

1. A module for comparing if the input frequency is higher or lower than the output frequency.
2. A "battery" which is charged or discharged depending on whether the output frequency is higher or lower compared to the input.
3. An oscillator which increases the output frequency if the charge in the battery is high and decreases it if the charge is low.

The aim in this thesis is to remove the battery and instead quantize the phase difference to a digital number. The oscillator will then use this number to adjust the output frequency instead of the battery charge.

Preface

This Master Thesis is the final project of an Engineering Master's degree within high frequency and nanoelectronics. The project has been carried out at Ericsson AB who have been providing the author with office space, computer, knowledge and guidance on top of an inspiring and innovative environment. A special amount of gratitude is extended toward the Ericsson project supervisor Mohammed Abdulaziz who has been a helpful and committed supervisor, providing both insightful feedback and learning opportunities. Another thank you to Simon Richter, another Master student working in the same project, and finally Henrik Sjöland, the Lund University supervisor.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

A Phase-Locked Loop (PLL) is a control systems feedback loop used for generating an output frequency which is phase-locked to the input frequency. A PLL can be used for demodulation of FM or FSK signals or to recover a noisy signal. This thesis focuses on frequency synthesis where a PLL is used to multiply an incoming reference frequency. The PLL which is investigated is a digital PLL which uses pulse shrinking, abbreviated PS-DPLL. It is to be used in a massive multiple in, multiple out (MIMO) system driven by a central PLL as shown in figure 1.1. Each of the PS-DPLLs, which might be hundreds in count, are individually addressable and the phase of the output frequency can be delayed to achieve beam steering. The frequency is first generated by a crystal oscillator, which is multiplied up to the low-GHz range by a central PLL. This then acts as the reference frequency for the MIMO array. This type of setup has two main benefits. Firstly, each of the PS-DPLL can be designed with a relatively simple architecture with integer frequency multiplication. This is because the central PLL is used for channel selection. This can then be made fractional. Secondly, the array PLLs can be designed for low power consumption whereas more power can be spent in the central PLL to suppress noise. The PS-DPLLs have a low frequency division

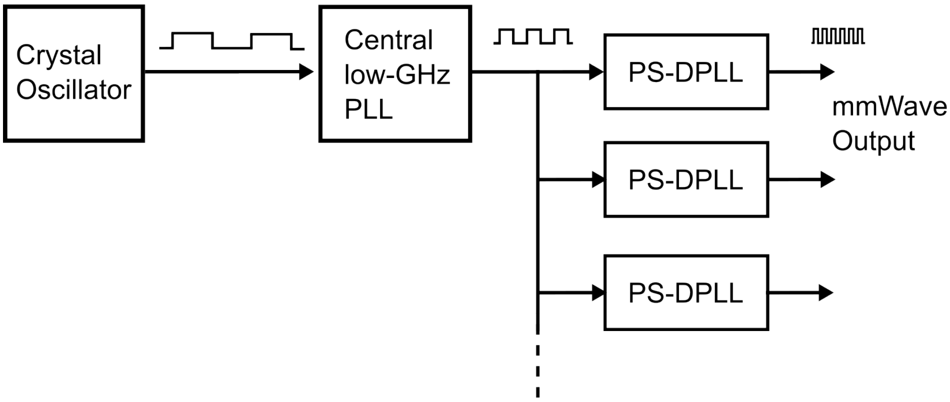


Figure 1.1: A central low-GHz PLL acting as a reference for a massive MIMO where each output frequency is synthesised using a PS-DPLL.

ratio, and therefore any reference and quantization noise is not amplified as much as for the central PLL which has a much higher division ratio. By allowing the central PLL to use more power, the constraints on the array are relaxed allowing for a total low power system.

In this thesis, the theory behind the analog and digital PLL is first discussed, before moving over to the pulse shrinking digital PLL in the theory section. This is then followed by the method of designing the components used in the PLL and subsequently the results achieved when simulating in using circuit simulation software. Finally, section 5 concludes the work and results.

Chapter 2

Theory

2.1 The Analog PLL

A charge pump PLL (CPPLL) uses a phase/frequency detector (PFD), a charge pump, loop filter, VCO and frequency divider as shown in figure 2.1. The reference frequency, typically from a crystal oscillator, is compared to the output frequency divided by a frequency divider with ratio N. At first, N is assumed to be equal to 1.

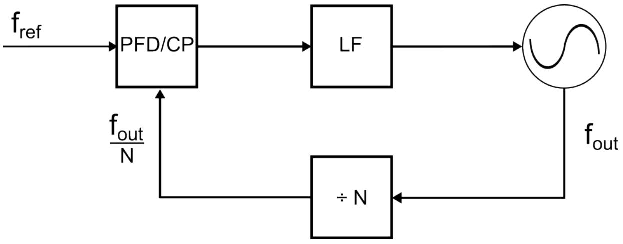
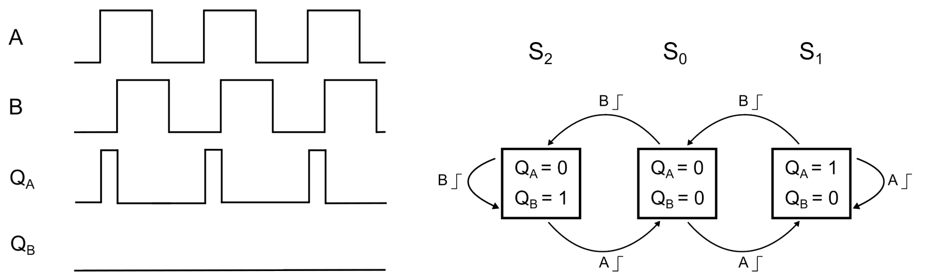


Figure 2.1: The main blocks of a CPPLL.

The phase/frequency detector takes two input signals A and B, and produces outputs Q_A or Q_B depending on which of the signals are leading. Figure 2.2a shows signal A and B having the same frequency but A leading in phase. As a result, the PFD sets Q_A high when A rises and resets it to low when B rises. The output pulse width is proportional to the phase difference between the signals. Importantly however, Q_B stays low when B is high and A low. This is represented in the state diagram in figure 2.2b showing that the state S_2 can only be reached from S_1 if B triggers twice without a trigger from A. In other words, B must be leading to reach state S_2. This state machine can be realized by feeding the inputs to two edge-triggered resettable D flip-flops. The outputs are fed into an AND-gate which has its output connected to the flip flop reset as shown in figure 2.3a. A delay buffer is added after the AND-gate to ensure pulses even when the two signals are in phase with each other. The outputs of the PFD, Q_A and Q_B, are fed into the charge pump. The idea of a charge pump is to charge or discharge a capacitor depending on which signal is high. For example, if A is leading B, the state machine should be in state S_2, setting Q_A high. The charge pump should

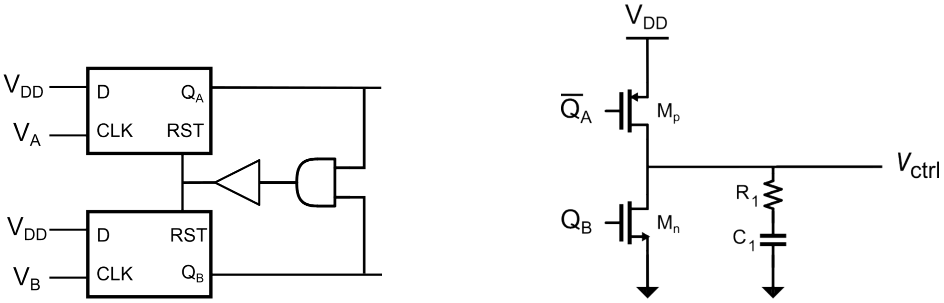


(a) Input and output.

(b) State diagram.

Figure 2.2: The response of a phase/frequency detector.

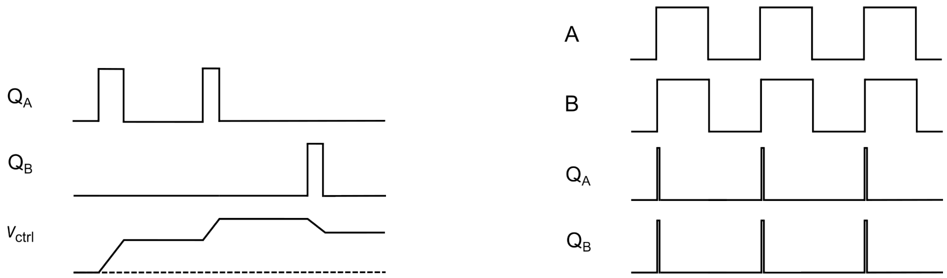
then charge up the capacitor. Inversely, if Q_B is high, then the capacitor should be discharged. The simplest implementation of a charge pump is shown in 2.3b and consists of a PMOS and an NMOS. Since M_p conducts for low signals, Q_A must be inverted before connected to the gate. The output voltage, v_{ctrl}, is the oscillator control voltage. From the charge pump action, v_{ctrl} increases if A is leading and decreases if B is leading. Figure 2.4a shows the effect on v_{ctrl} from up and down pulses Q_A and Q_B. If the two signals are perfectly in phase with each other, v_{ctrl}, ideally remains unchanged. However, due to the delay to reset the flip-flops, there will be thin pulses on both Q_A and Q_B even with zero phase difference. This is illustrated in figure 2.4b. Due to these spikes, it is essential that the charging current is equal to the discharging current. This calls for some careful balancing of the two transistors to make sure that the same current coming from M_p also exits through M_n. Furthermore, the two pulses must arrive at the charge pump simultaneously. Otherwise, if Q_B arrives before Q_A, v_{ctrl} will momentarily decrease before it rises again. This in turn will cause the output frequency to decrease shortly. Since Q_A must be inverted before reaching the charge pump it will also be delayed compared to Q_B. To combat this Q_B can be delayed using a transmission gate. When the charge- and discharge current is balanced, $I_n = I_p$. [1]



(a) An implementation of the phase/frequency detector.

(b) A basic charge pump and a second order loop filter.

Figure 2.3: Implementations of the PFD and charge pump.



(a) Charge pump action.

(b) Short pulses at zero phase difference.

Figure 2.4: Implementations of the PFD and charge pump

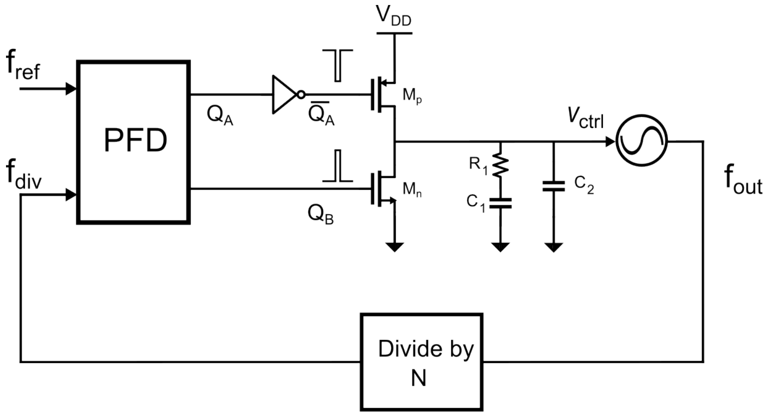
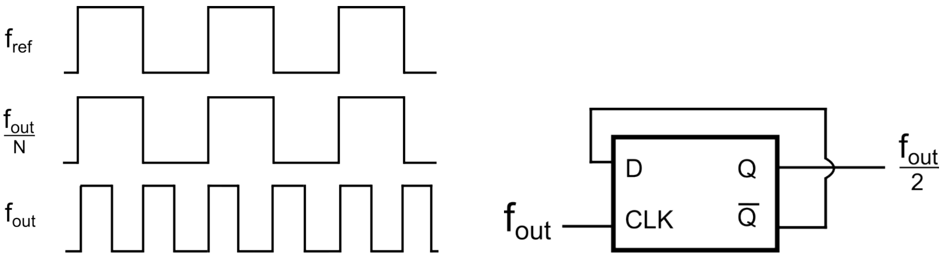


Figure 2.5: The charge pump phase-locked loop.

The control voltage is fed into the voltage controlled oscillator. A VCO is generally described by a free-running frequency along with a tuning range, the range of frequencies it can cover, and a gain, the frequency change given by a unit change in voltage. The VCO is commonly realized by an LC-oscillator where the oscillation frequency is at $f = 1/(2\pi\sqrt{LC})$. The tuning is achieved by increasing or decreasing the capacitance using varactor diodes, thus changing the oscillation frequency.

Up until now, the frequency divider has been assumed to be N=1. When a PLL is used for frequency synthesis, a divider is placed in its feedback path. The output frequency becomes $f_{out} = Nf_{ref}$, an N-multiple of the input frequency. This is because the loop will lock in on the same frequency and phase as the divided output frequency. Since f_{ref} now is compared to f_{out}/N, the loop will adjust the oscillator control voltage until f_{out}/N has the same frequency and phase as f_{ref}. The output frequency f_{out} is therefore N times the input frequency. Figure 2.6a shows an example of frequency multiplication where N=2. For power-of-two divider ratios, the divider can simply be implemented using D flip-flops where the input frequency is connected to the clock and the $\bar{Q}$-output is fed back into data port as shown in figure 2.6b. By stacking n of them, a division ratio of 2^n is achieved. [1]

It is also possible to make fractional-N dividers although not discussed here.



(a) The effect on output frequency.

(b) D flip-flop counter.

Figure 2.6: Frequency multiplying.

2.2 Phase Domain Model

In order to find the transfer function of the system, the phase domain characteristics of each module must be established. A unit step in phase, $u(t)$, is applied to the input. Looking at the PFD and charge pump, a phase step of $\Delta\phi_0$ causes a pulse in PFD-output, Q, which lasts for $\Delta\phi_0 T_{in}/(2\pi)$ seconds. During this time, current is pushed into the capacitor through the resistor from the charge pump. The increase in voltage in the capacitor during this time is $\Delta v_C = \frac{\Delta\phi_0}{2\pi}\frac{I_p}{C_1}T_{in}$. This happens every period but in the time between the pulses, there is no change in voltage. Therefore, the charge pump is active at discrete time steps. An approximation can be made to work with the system in continuous time. The increase over the whole time period is Δv_C. Thus, the average increase over the period is $\Delta v_C/T_{in}$ and the capacitor voltage is approximated continuous as

$$v_C(t) = \frac{\Delta\phi_0}{2\pi}\frac{I_p}{C_1}tu(t) \tag{2.1}$$

where the ramp function $tu(t)$ comes from the capacitor being charged up by the current I_p.

The resistor does not hold on to any charge. The voltage increase due to the resistor is simply the charge pump current multiplied with the resistance, lasting for $\Delta\phi_0 T_{in}/(2\pi)$ seconds. Applying the same approximation as to the capacitor, we instead assume that the voltage is continuous during the entire time period. Averaging the voltage over the entire period yields

$$v_R(t) = \frac{\Delta\phi_0}{2\pi}I_pR_1u(t) \tag{2.2}$$

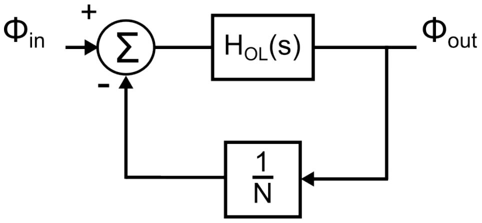


Figure 2.7: Closing the loop with feedback divider N.

The total voltage is the voltage built up in the capacitor and the voltage over R_1. The control voltage is

$$v_{ctrl}(t) = \frac{\Delta\phi_0}{2\pi}\frac{I_p}{C_1}tu(t) + \frac{\Delta\phi_0}{2\pi}I_pR_1u(t) \tag{2.3}$$

The transfer function is found by differentiating with respect to time and finding the Laplace transform. [2]

$$\frac{\partial v_{ctrl}(t)}{\partial t} = \frac{\Delta\phi_0}{2\pi}\frac{I_p}{C_1}(u(t) + t\delta(t)) + \frac{\Delta\phi_0}{2\pi}I_pR_1\delta(t) \tag{2.4}$$

$$\frac{v_{ctrl}}{\Delta\phi_0}(s) = \frac{I_p}{2\pi C_1}\frac{1}{s} + \frac{I_pR_1}{2\pi} \tag{2.5}$$

The phase of the VCO is the time integral of the VCO output frequency. In other words, it acts as a phase integrator. It has a gain, K_{vco} which is the increase in frequency for a unit increase in voltage, expressed in rad/s/V. The transfer function of VCO is thus K_{vco}/s. The open loop transfer function of the charge pump PLL becomes

$$H_{OL}(s) = \frac{K_{vco}}{s}\frac{I_p}{2\pi}(\frac{1}{sC_1} + R_1) \tag{2.6}$$

Figure 2.7 shows the open loop transfer function fed back to the input through feedback divider N.

The closed loop transfer function is

$$H_{CL}(s) = \frac{NH_{OL}(s)}{N + H_{OL}(s)} \tag{2.7}$$

Inserting the open loop transfer function from 2.6 yields the charge pump PLL transfer function in equation 2.8.

$$\frac{\phi_{out}}{\phi_{in}}(s) = \frac{\frac{NK_{vco}I_p}{2\pi}(\frac{1}{C_1} + R_1s)}{Ns^2 + \frac{K_{vco}I_p}{2\pi}(\frac{1}{C_1} + R_1s)} \tag{2.8}$$

This transfer function has low-pass characteristics, plotted as $20\log(\phi_{out}/\phi_{in})$ in figure 2.8 for some arbitrary loop parameters. For this loop example, a divider ratio of 8 was chosen, causing the low frequency magnitude to become $20\log 8 \approx 18$. This can also be seen in equation 2.8 where at low frequencies the magnitude of $s \longrightarrow 0$. Then only N is left in the numerator since $K_{vco}I_p/(2\pi C_1)$ cancels out with the same term in the denominator. [1]

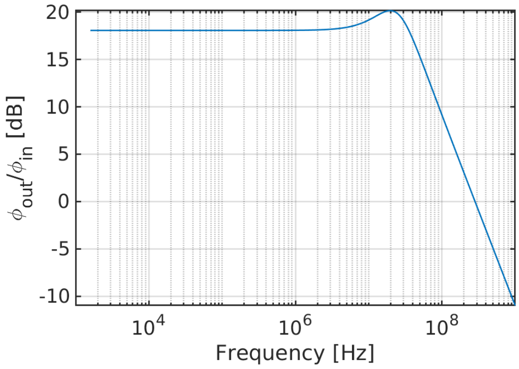


Figure 2.8: Closed loop transfer function.

2.3 The Digital PLL

When transferring to a digital PLL, the charge pump and loop filter is swapped for a time-to-digital converter (TDC) and a digital loop filter. The voltage controlled oscillator is replaced by a digitally controlled oscillator (DCO).

2.3.1 Loop filter

To achieve the same second order loop characteristics as with the analog PLL, the digital filter needs a proportional part, α, and an integral part, β. The integral part is realized by an adder and a register. The result of the addition is stored in the register which feeds back to the adder. Whatever is stored in the register is added in the next clock cycle to the error scaled by β. This is shown in figure 2.9. [2]

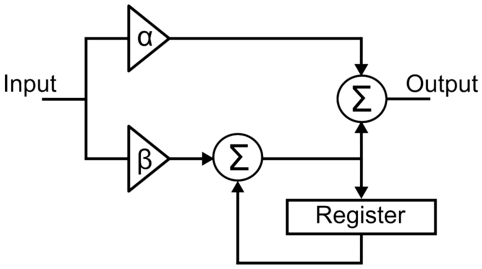


Figure 2.9: Block diagram of a digital second order loop filter.

The z-domain transfer function is presented in equation 2.9.

$$\frac{D_{out}}{D_{in}}(z) = \alpha + \frac{\beta}{1 - z^{-1}} \tag{2.9}$$

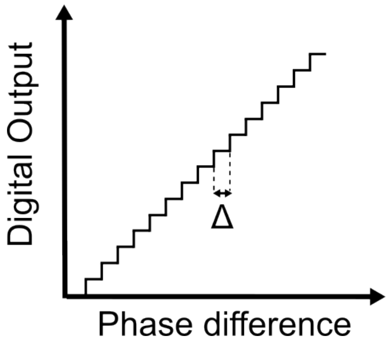


Figure 2.10: A typical graph of a TDC quantizer with step size Δ.

2.3.2 DCO

The DCO is controlled by a digital word as opposed to a voltage. The word is used to activate more or less capacitors and thereby increase or decrease the oscillation frequency. The DCO may consist of a fixed inductor, a bank of higher value capacitors and a bank of lower value capacitors. This enables course and fine tuning.

2.3.3 Time-to-Digital Converter

The time-to-digital converter, TDC, inputs a reference and a divided signal and quantizes the phase difference to a digital representation. The idea is shown in figure 2.10 where a larger phase difference corresponds to a higher digital output D_{out}. The resolution, Δ, is the stepsize and the minimum difference in phase difference that can be detected. A basic implementation of the TDC is shown in figure 2.11. A signal V_A is sent into a line of delay elements, for instance buffers. Each element delays the signal by Δ seconds before the signal propagates on to the next element and the data-port of a flip-flop. The signal V_B clocks the flip-flops such that at its rising edge, the values of all the data ports in the line are sampled and stored. This is shown in figure 2.12. If V_A has had time to propagate to a given flip-flop, a 1 will be stored otherwise a 0. [2]
As an example, the gate delay might be around 12 ps for a given technology. For a phase difference of 40 ps, the pulse propagates $\lfloor \frac{40}{12} \rfloor = 3$ stages before being clocked and the output code is three ones followed by zeros (1110000...). To be able to also detect if V_B is leading, another delay line is introduced where the input and clock is reversed.
This type of TDC has three main limitations. The first one is that the minimum step size is one full gate delay which is often too high. The second is that the driving strength of V_B must be high to act as the clock to all the flip-flops. Thirdly, V_A will unnecessarily continue to propagate until the end of the chain after it is clocked. This is unnecessary because the phase error has already been quantized and the additional propagation brings extra power consumption and adds to the phase jitter.

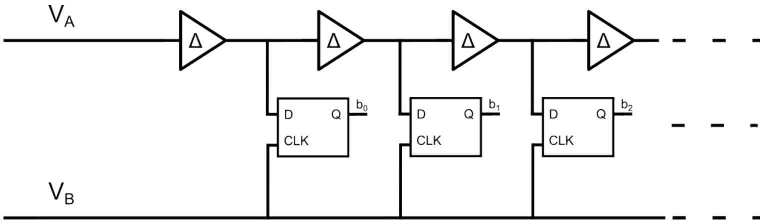


Figure 2.11: The a simple type of TDC where the flip-flops are clocked by V_B.

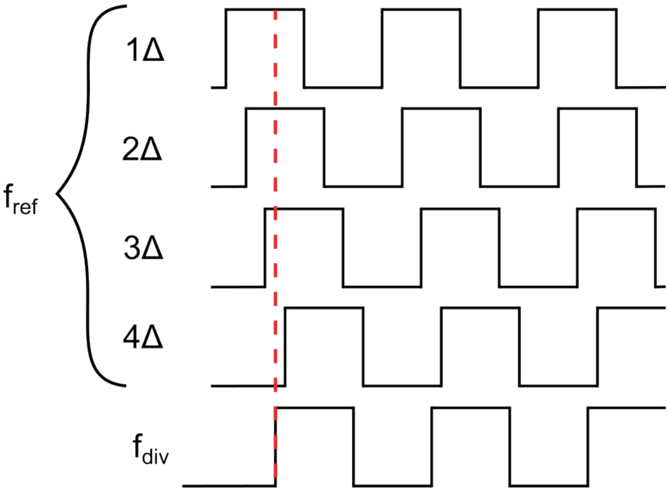


Figure 2.12: The workings of a conventional TDC.

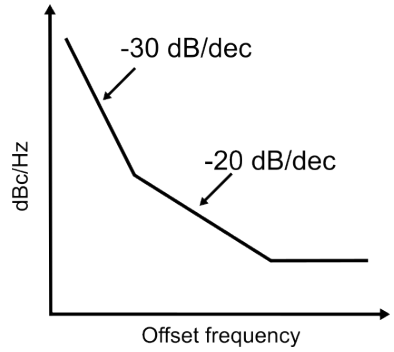


Figure 2.13: A typical noise spectrum for a VCO.

2.4 Noise

The noise in a PLL is mainly introduced from the reference and the oscillator. Commonly, this noise is treated in phase domain, referred to as phase noise and is the random fluctuations in phase from cycle to cycle. A digital PLL also has quantization noise.

2.4.1 Oscillator Noise

Ideally, an oscillator has all of its power concentrated at a single frequency ω_c and the voltage output can be described as $v(t) = A\cos(\omega_c t)$. However, taking phase noise into consideration, a function $\phi(t)$ must be added, $v(t) = A\cos(\omega_c t + \phi(t))$. If the variation in phase is small ($<< 1$), then the output voltage can be approximated as $v(t) \approx A\cos(\omega_c t) - A\phi(t)\sin(\omega_c)$. From this follows that for any noise tone ω_m, sum and difference frequencies will arise next to the main frequency component, $v(t) \approx A\cos(\omega_c t) + A\frac{\phi_p}{2}(\cos(\omega_c + \omega_m)t - \cos(\omega_c - \omega_m)t)$. [3] An oscillator generally has noise power highest near the resonance frequency and reducing with frequency offset. The noise can be divided into three regions; thermal noise, flicker noise and flat noise. These are shown in figure 2.13. Closest to the resonance frequency is the flicker noise region which is noise induced by the presence of surface states at the oxide interface in CMOS technologies. In the flicker noise region, oscillator phase noise power drops by 30 dB per decade. [1].
The thermal noise has a $1/f^2$- dependency, meaning noise power will reduce by 20 dB/decade as the offset frequency increases. Finally, there is flat noise which does not have a frequency dependence.

The noise arising from the oscillator gets shaped by a transfer function from the noise input to the output of the PLL. The VCO phase noise is added to the system after the oscillator according to figure 2.14. To derive the noise transfer

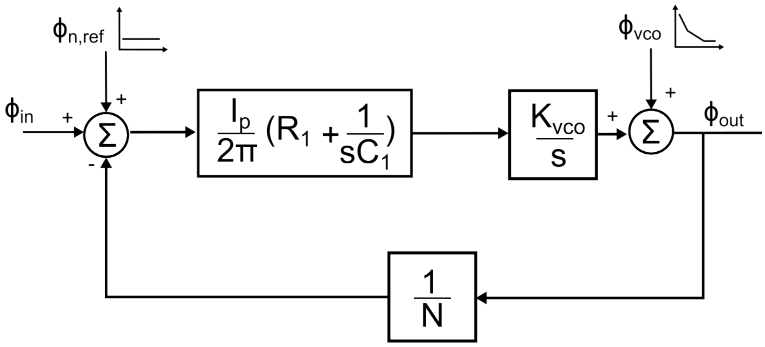


Figure 2.14: The charge pump loop phase domain model with reference noise and oscillator noise added.

function, ϕ_{in} and $\phi_{n,ref}$ is set to zero. At the output,

$$\phi_{out} = -\frac{\phi_{out}}{N}\frac{I_p}{2\pi}(R_1 + \frac{1}{sC_1})\frac{K_{vco}}{s} + \phi_{vco}$$

$$\phi_{vco} = \phi_{out}\Big(1 + \frac{K_{vco}}{sN}\frac{I_p}{2\pi}(R_1 + \frac{1}{sC_1})\Big)$$

$$\frac{\phi_{out}}{\phi_{vco}} = \frac{1}{1 + \frac{K_{vco}}{sN}\frac{I_p}{2\pi}(R_1 + \frac{1}{sC})}$$

$$\frac{\phi_{out}}{\phi_{vco}}(s) = \frac{s^2}{s^2 + \frac{I_p K_{vco} R}{2\pi N}s + \frac{I_p K_{vco}}{2\pi C_1 N}} \tag{2.10}$$

Contrary to the closed-loop transfer function ϕ_{out}/ϕ_{in}, this transfer function shows high pass characteristics. However, the poles of this transfer function are identical to that of equation 2.8.

2.4.2 Reference Noise

A PLL typically uses a crystal oscillator as a frequency reference. Such references contains phase noise which typically has a frequency independent spectrum. [2] The reference noise appears at the input of the PLL and thus experiences the transfer function ϕ_{out}/ϕ_{in}. Therefore, it passes at frequencies below the bandwidth whereas it is suppressed for high frequencies. Further more, a divider ratio of N increases reference phase noise by $20 \log N$ since this is the increase in transfer function magnitude when using dividers.

2.4.3 Quantization Noise

A digital PLL also has quantization noise arising from the discrete nature of the TDC. For a resolution of Δ, the total noise power arising from quantization is

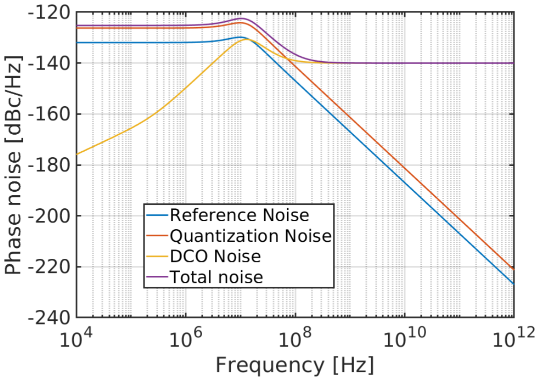


Figure 2.15: Estimates of the different noise sources using a 30 MHz bandwidth.

$\Delta^2/12$ and the sample rate is f_{ref}. The noise is present between $-f_{ref}/2$ to $f_{ref}/2$ and the total noise spectrum is calculated according to equation 2.11. [2]

$$S_Q = \frac{4\pi^2\Delta^2}{12T_{ref}} \tag{2.11}$$

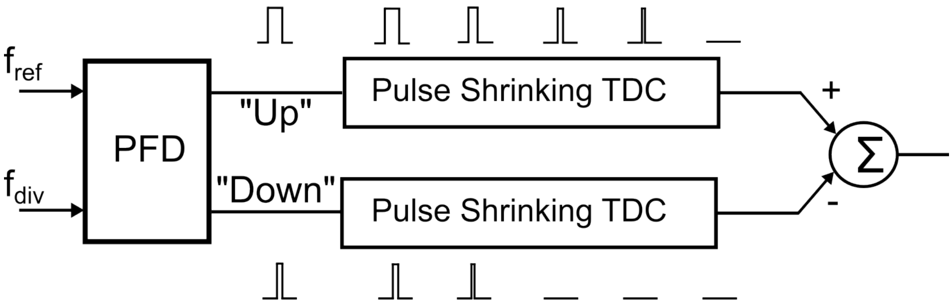
As with reference noise, quantization noise is shaped by the PLL transfer function, meaning a divider ratio N increases it by $20\log N$. It is suppressed at high frequencies above the loop bandwidth.

2.4.4 Noise estimates

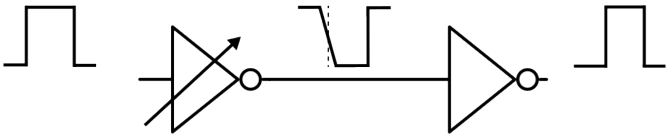
For the purpose of making some informed design decisions, some reasonable noise power numbers were estimated in together with a 30 MHz loop bandwidth. The parameters are listed below:

- Loop bandwidth of 30 MHz.
- Reference frequency of 500 MHz.
- Divider ratio 8.
- White reference phase noise of -150 dBc/Hz.
- TDC resolution of 1.5 ps.
- A DCO with flicker noise corner frequency at 250 kHz, and noise floor corner frequency at 50 MHz. Noise floor at -140 dBc/Hz.

Each of the spectra is shaped by the transfer function from their respective input to the output. The result is shown in figure 2.15. Quantization noise dominates the in-band noise and reaches a maximum of -125 dBc/Hz near the bandwidth frequency.



(a) The principle of using a PFD paired with a pulse shrinking TDC.



(b) The concept of pulse shrinking.

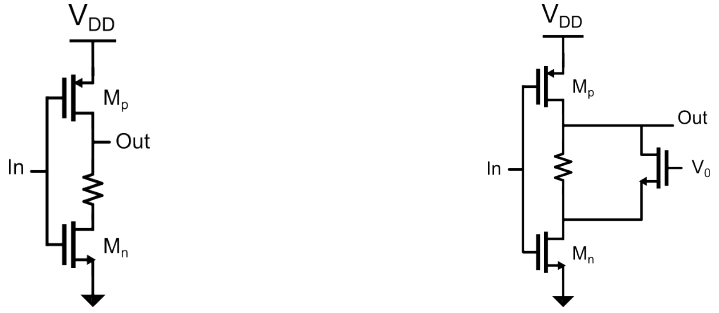
Figure 2.16: A TDC using a chain of pulse shrinking inverters to quantize the length of a pulse (a) and each pulse shrinking stage (b).

2.5 The Pulseshrinking Digital PLL

2.5.1 Quantization by Pulse Shrinking

A pulse shrinking TDC (PS-TDC) inputs the pulses from the PFD used in the analog CPPLL. This pulse then propagates through the TDC and as it does, the pulse width is shortened by Δ seconds for each stage. Each of the "Up" and "Down" pulses from the PFD are sent into the TDC which quantizes the length of the pulse. The phase error is then calculated based on the difference in length. This is shown in figure 2.16a. In the case of the figure, f_{ref} is leading, causing a long "Up"-pulse and a short "Down"-pulse from the PFD. Due to the "Up"-pulse being longer, it propagates further into the TDC and stores more ones than the short "Down"-pulse does. Assume the TDC in the figure has 5 stages. In the "Up"-TDC, the digital output code will be "11110", interpreted as temperature code to 4. The digital output in the "Down"-TDC will be "11000" interpreted as 2, giving a total phase error of $4\Delta - 2\Delta = 2\Delta$.

The idea of pulse shrinking is shown in figure 2.16b where a pulse is sent in to an inverter with a weaker NMOS than PMOS. This results in a longer fall time than rise time. The pulse is followed by a regular, balanced inverter which restores the pulse shape. Due to it taking longer time to reach the threshold voltage of the PMOS in the balanced inverter, the length of the output pulse is shorter. For each pulse shrinking stage, the pulse shrinks by Δ seconds until the pulse is no longer long enough to propagate. The pulse shrinking cell can be made adjustable by having different paths to V_{DD} and ground activated. A non-adjustable and a 1-bit adjustable example schematic of a pulse shrinking inverter is shown in figure



(a) Non-adjustable.

(b) 1-bit adjustable fall time using V_0.

Figure 2.17: Examples of pulse shrinking cells to achieve longer fall time than rise time.

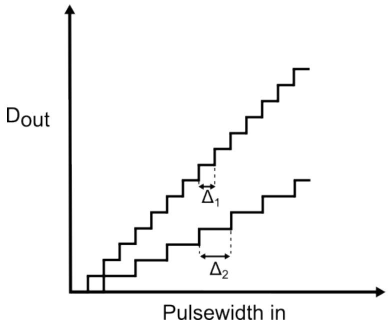


Figure 2.18: Adjustable resolution due to activation of an additional pull-down path.

2.17. In these figures, the increased fall time is achieved by a resistance in the pull-down path. For an actual implementation however, the rise and fall time may be adjusted by balancing the width/length ratio in the PMOS and NMOS. The additional path to ground is activated by setting the voltage V_0 high and allow for a different value of the resolution Δ. The effect of this is shown in figure 2.18 where the two step sizes Δ_1 and Δ_2 correspond to V_0 being set high or low respectively.

The TDC inputs the pulse coming from the PFD according to figure 2.19. Using f_{ref} and f_{div}, the PFD outputs a pulse with a pulse width proportional to the phase difference between the signals. This pulse is sent into the pulse shrinking chain. The width of the pulse, and thereby also the phase difference, is estimated in units of Δ by counting how far along the chain the pulse propagates. To be able to do this, D flip-flops are used where the pulse is the data input. To clock the flip-flops a second pulse is used which arrives slightly later, allowing the data pulse to rise to V_{DD} before rising itself. In order to get the same power consumption benefit as the data pulse itself, the clock pulse is also sent into a pulse shrinking inverter chain. Crucially, the clock pulse must propagate at least one additional stage compared to the data pulse. This is in order to store at least one zero in a

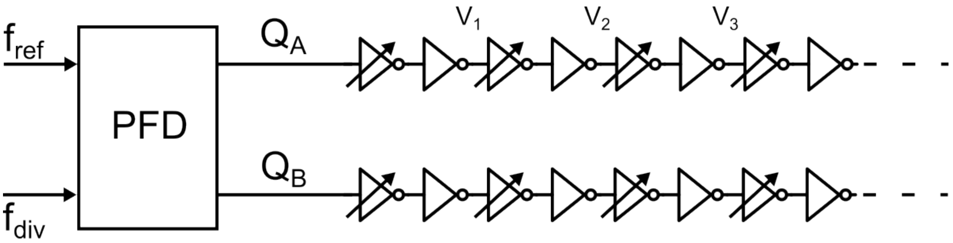


Figure 2.19: The TDC inputs the pulse from the PFD. How long it propagates is proportional to the phase difference between f_{ref} and f_{div}.

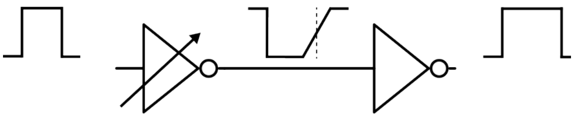


Figure 2.20: Pulse extending.

flip-flop after the string of ones. Once one zero has been found, it can be known that the data pulse has stopped propagating. One method of assuring that the clock pulse propagates longer through the chain than the data pulse, is to extend it before sending it into the chain. This is the purpose of the block preceding the clock chain along with delaying the pulse slightly compared to the data pulse. The pulse can be extended by using an inverter with a weaker pull-up path than pull-down path followed by a balanced inverter. The principle of this is shown in figure 2.20 and a schematic can be implemented similar to that of the pulse shrinking cell but balanced such that rise time is longer than the fall time. The same V_0 control bit, but inverted, can be used to achieve more or less pulse extending. The data pulse chain, the clock pulse chain and the flip-flops are shown in figure 2.21.

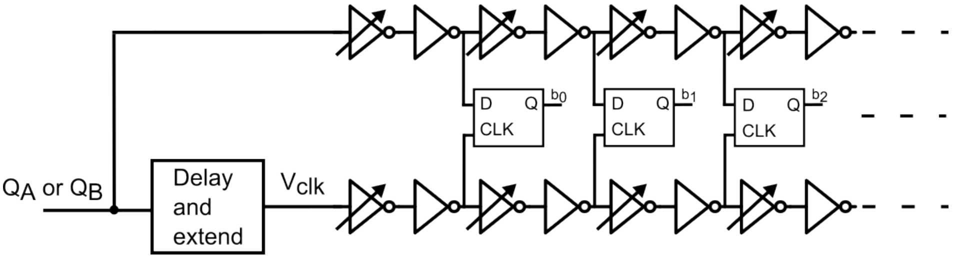


Figure 2.21: Two identical pulse shrinking inverter chains where one is delayed and extended to clock the data-pulse.

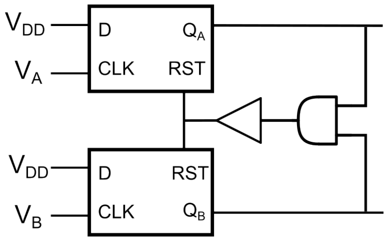


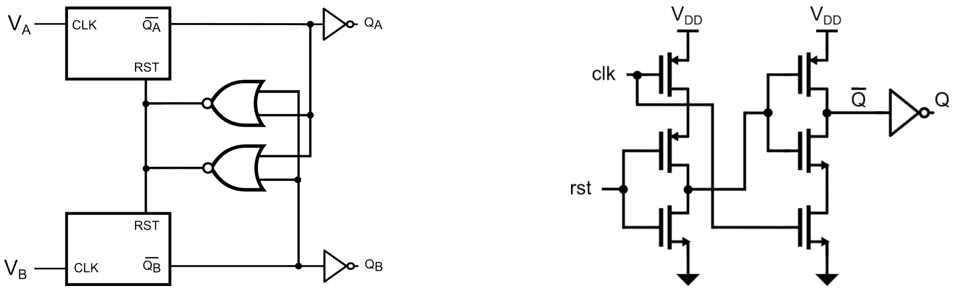
Figure 2.22: A PFD using two D flip-flops and an AND-gate.

2.5.2 PFD Adjustments

The PFD can be implemented using two D flip-flops along with an AND-gate as shown in figure 2.22. However, for the purposes of a PFD, the flip-flops can be implemented with fewer devices as shown in [5]. This schematic of the flip-flop is shown in figure 2.23b. The output of the flip-flop is the inverted phase difference $\bar{Q}$ and the output must therefore be inverted before being sent into the TDC. Q_A and Q_B can fed into an AND-gate where the output is connected back to the reset of the flip-flops. However, to achieve minimum pulse widths, $\bar{Q_A}$ and $\bar{Q_B}$ can be immediately connected to a NOR-gate which feeds back to the reset of the flip-flops. The benefit of this is to avoid the additional gate delay when inverting $\bar{Q}$.

Because the phase error will be calculated based on the difference in pulse width for Q_A and Q_B, it is of great importance that a given phase error will yield the same pulse width for Q_A and Q_B when f_{ref} is leading or lagging, respectively. However, simulations show that this is not the case when simply connecting $\bar{Q_A}$ and $\bar{Q_B}$ to a NOR-gate. The reason for this is that the propagation from V_{DD} to the reset input of the flip-flops are slightly longer depending on whether the signal arrives to the top or bottom PMOS in the NOR-gate first. This can be mitigated by connecting $\bar{Q_A}$ and $\bar{Q_B}$ to two NOR-gates with the inputs reversed as shown in 2.23a.

The design of the PFD plays an important role when paired with the TDC. To ensure that not too many stages in the TDC are active, the PFD is designed such that the pulses out of it when in lock are narrow enough that they only propagate 2 to 5 stages in the TDC. In a high resolution setting, the TDC is designed such that only a small amount of pulse shrinking occurs each stage. With variations in corners and statistical mismatch variations, the pulse might propagate somewhere between 0 and 20 stages. This may result in the TDC not working properly or having a high power consumption. It is therefore desired to be able to detect how far the pulse is propagating and then be able to adjust accordingly. For example, if the PLL is in lock and 20 stages are detected to be active every period then the minimum pulse width should be adjusted downwards until only 2 to 5 stages are active. The PFD is made adjustable such that the minimum pulse width can be reduced or increased based on the system requirements. One way to do this is to introduce an adjustable buffer after the NOR-gates. However, this will increase pulse widths by at least 6 ps which hinders a short pulse of 18 ps. Another solu-



(a) A PFD with dual NOR-gates to ensure that Q_A and Q_B have equal pulse width for a given phase error.

(b) A flip-flop with fewer devices producing an output $\bar{Q}$ which is then inverted to Q. The $\bar{Q}$ output is fed into the NOR-gates.

Figure 2.23: The PFD reset path using NOR-gates (a) and the flip-flop implementations (b).

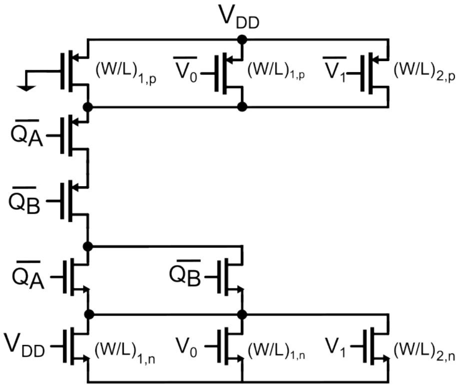


Figure 2.24: 2-bit adjustable NOR-gate schematic.

tion is therefore required. The solution proposed here is to limit the reset current coming from the NOR-gate in a controllable manner. This is achieved by partially blocking the path to V_{DD} and ground using transistors which are activated or deactivated to increase or decrease reset current and thereby also the minimum output pulse width. The schematic of the adjustable NOR-gate is shown in figure 2.24. Depending on the sizing of $(W/L)_1$ and $(W/L)_2$ more or less current is used in the reset path, making the PFD output pulses shorter or longer.

When the two input signals are in phase, the output from the PFD is a short pulse with a width that is dependant on the delay in the reset path to the flip-flops. A shorter time to reach the reset implies a shorter minimum output pulse. However, it is also important that the minimum pulse width is not too short. In that case it might not excite any stages at all and no quantization can be done. Propagating fewer stages has three main benefits. First, the power consumption is lower when fewer stages are active. Secondly, less jitter can be expected when propagating fewer stages since some jitter is added every stage. Thirdly, if the pulse

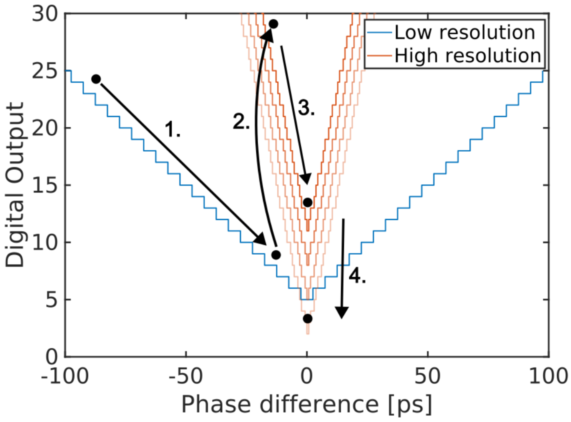


Figure 2.25: Figure showing how to use the adjustability of the PFD and TDC.

propagates fewer stages then the logic for calculating the phase error does not have to wait as long. This enables high reference frequencies and lower magnification of reference and quantization noise.

2.5.3 Acquisition Sequence

Both the TDC and the PFD are adjustable. The TDC can be switched between a high resolution mode and a low resolution mode by increasing the current to ground. The PFD can be adjusted by increasing or decreasing the reset current to the flip-flops. When the PLL operates it uses these adjustability features as shown in figure 2.25. Initially during the acquisition phase, the phase difference between the two signals is up to 360° off. At 4 GHz, this might be a phase difference of up to 250 ps. To be able to quantize that large of a pulse, the TDC is set to low resolution. In this configuration, the step size is about 5 ps. Thus, a 50 stage TDC is needed to quantize it compared to 313 stages for a non-adjustable TDC which had only a 0.8 ps resolution mode. The phase error is then worked down until equally many TDC stages are active in the up and down chain. This is shown by arrow "1." in figure 2.25. Next, the PLL switches to high resolution mode as indicated by "2.". The phase error is now small enough to be fully quantized by a high resolution TDC. The PLL decreases the phase error to zero using the "3." move. Finally, to save power and reduce jitter, the number of active stages is decrease by movement "4.". This is done by increasing the reset current in the PFD. However, the number of active stages is not reduced below 1 or 2. In that case, reset current is adjusted such that the number of active stages fall under a predefined accepted interval.

When only a few stages are active, the phase error can be calculated quickly. Initially however, such as during phase 1 to 3 in figure 2.25, the phase error might be large causing the pulses to propagate far into the TDC. Due to the limited propagation speed, the logic must wait until the phase error can be calculated.

Therefore, a divided reference is initially used to act as a clock for the logic. 1 GHz is initially used, giving plenty of time for the pulse to propagate through all stages, for the logic such as calculating the difference between the "Up" and "Down" TDCs, going through the loop filter and adjusting the DCO input. Once lock is reached, the pulses are short and only propagate a few stages. By now the fast, undivided reference is used for clocking the logic since the propagation through only a few stages is much faster. Adjusting the number of active stages using the "4"-move can further speed up the digital processes.

Chapter 3

Method

For the design process of the PFD and TDC, simulations were carried out using Cadence Virtuoso and ADE. Transient simulations were used as well as Periodic Steady State analysis for phase noise and jitter estimates. The technology node used was 22 nm fully depleted silicon on insulator. Super-low threshold devices were used. Office space, computer and model files were provided by Ericsson AB.

3.1 General Design Strategy

The first step was to achieve consistent pulse shrinking. Therefore, the project started off by comparing different cells designs where pulse shrinking could be achieved and to assess the standard deviation of the mean pulse shrinking. The cells were also compared in terms of power consumption and propagation delay.

Once the most suitable pulse shrinking cell design had been determined, the next step was to find suitable sizing of the devices as well as to investigate the trade-offs when settling on general sizing of the transistors. For instance to find the positive and negative effects of doubling the sizing of the inverters in the TDC chain. Next, the TDC was characterized in terms of linearity and the minimum width of which it starts reacting to pulses. This was done in both the high- and low resolution mode. To be able to draw some real-life conclusions, Monte Carlo simulations were done where both the differential and integral non-linearity were determined. Once the minimum detectable pulse width was known, the phase/frequency detector could be designed. The general strategy when designing the PFD was to achieve a short output pulse even if it required that some extra power was used. Recognising that the PFD had a loop from the flip-flop output to the NOR-gates and then back to the flip-flop reset, the strategy was to achieve similar rise and fall times at all loop-nodes, i.e at the flip-flop input and output as well as the NOR-gate output. This way, no transistor was unnecessarily oversized. The "loop" is shown in figure 3.1. Since the output of the PFD is loaded by 4 relatively large inverters, the inverter from $\bar{Q}$ to Q must also sized to provide a lot of current. This implies increased size of the flip-flop output stage to maintain a short rise and fall time in the $\bar{Q}$ node, which in turn increases the size of the input stage and the NOR-gate. To reduce this phenomena, the strategy was to maintain a low load in the $\bar{Q}$ node by using a small inverter to get to Q, followed by a buffer to increase the driving

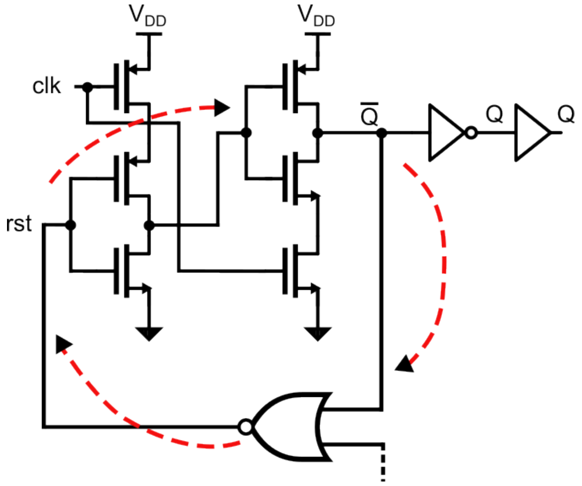


Figure 3.1: The PFD implementation which has a flip-flop with an output that feeds back to its reset.

strength for the following TDCs.

Finally, the current starving feature was added upon the NOR-gates and the transistors were sized such that fall time and rise time in the reset path was similar for all the different modes.

The PFD and TDC were then tested together by having two AC sources where the phase drifts apart by a small amount each cycle. The number of ones were then counted for each phase error. Finally some system figures such as power consumption and phase jitter were determined.

3.2 Limitations

The scope of this project was limited to the design of the PFD and TDCs as well as finding the trade-offs when choosing parameters for these. Design of the digital parts such as the loop filter, error encoders and control logic was implemented although not by the author and is thus not included here. Although the PS-DPLL may at some point be used in a MIMO system for beam steering, no implementation for phase shifting was included in this project. Furthermore, no layouts were done and no fabrication of the designs were made.

Chapter 4

Results & Discussion

4.1 Phase/Frequency Detector

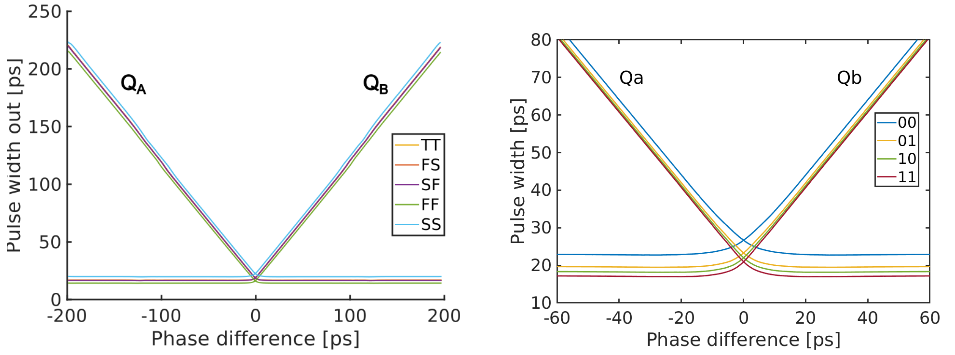
Figure 4.1a shows the PFD output pulse width as a function of input phase difference. The simulation frequency was 4 GHz and all corners were tested. The SS corner performed the worst but still covers a range from about $-300°$ to $300°$.

The minimum output pulse is around 17 ps long. As will be shown in section 4.2, this might in some cases be too short to detect. However, the short pulses is a too attractive feature to simply re-design for longer pulses. Therefore, the current limiting stage is added to the NOR-gates. Figure 4.1b shows the PFD with its 2-bit adjustable output pulse. A similar step size is desired when going from 00 to 01 to 10 etc. The minimum pulse width is 22.8, 19.5, 18.2 and 17 ps for the four settings. However, what is more important is the pulse width out when the PLL is in lock, i.e. zero phase difference. These pulse widths are 26.5, 23.2, 22.0 and 20.8 ps and are the ones sent into the TDC once every period when in lock. These widths along with the TDC resolution are crucial for the TDC power consumption and phase jitter. The PFD uses a weaker inverter at the output followed by a buffer to increase the strength of the output pulse. The benefit of this is short output pulses with short rise and fall time. The drawback is that a small amount of extra propagation delay (around 6 ps) is introduced which means that the logic for calculating the phase error must wait a short amount of additional time, however this is not an issue when the system is in lock since the pulse only propagates a few stages.

The pulses from the PFD can be further reduced by applying body bias. This is discussed in section 4.3.2 along with its power consumption.

4.2 Time-to-Digital Converter

Different types of pulse shrinking cells were investigated for the TDC. The main metrics on which they were compared were delay stage to stage and pulse shrinking standard deviation. It is desired to have a low delay stage-to-stage in order to be able to operate at high frequency. For example, at 5 GHz, a new pulse is sent into the TDC every 200 ps. If the propagation delay is 20 ps per stage, then the previous pulse has only propagated 10 stages when the next pulse arrives at the input. This might not pose a problem for the TDC while in lock, but if the pulses



(a) PFD output pulse width. TT, FS and SF is close to identical.

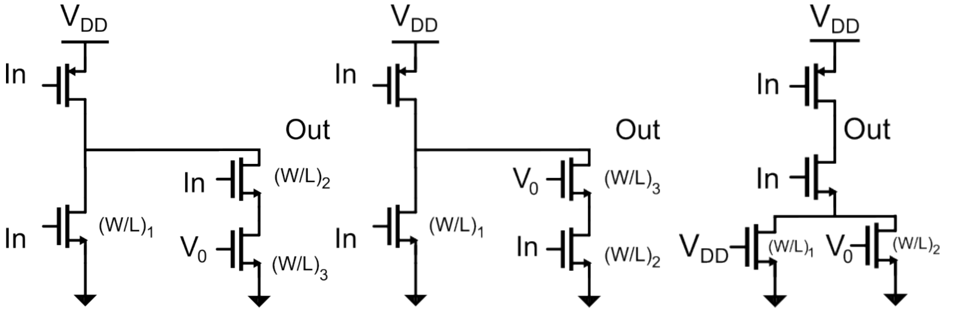
(b) The effect on PFD output pulse for the four different combinations.

Figure 4.1: Phase/Frequency Detector performance.

are long enough, as during acquisition, some unintended results might arise due to pulses not being clearly isolated. A low standard deviation is beneficial since it enables a similar step size when quantizing the pulse width. Furthermore, having a low standard deviation is crucial when implementing a high resolution PS-TDC. This is because a low amount of pulse shrinking is necessary without extending the pulse. If the pulse shrinking per stage is close to zero, then the standard deviation must also be low in order not to have too high risk of extending the pulse instead.

Figure 4.3a, 4.3b and 4.3c shows the Monte Carlo simulations with mean pulse shrinking and standard deviations for each of the pulse shrinking cells in figure 4.2a, 4.2b and 4.2c respectively. The inverters all had same size of the PMOS. The NMOSes were sized such that they achieve around 6 ps and 0.8 ps of pulse shrinking per stage when paired with the same balanced inverter. For the inverters in figure 4.2a and 4.2b, the NMOSes on the additional path to ground were sized such that $(W/L)_2 << (W/L)_3$ in order to maintain a low input capacitive load. The results are also summarized in table 4.1 where power consumption and propagation delay has been included for the TT case. The current starved pulse shrinking inverter in figure 4.2c achieves lowest standard deviation and propagation delay at 241 fs and 8.1 ps in the typical case. Furthermore, simulations show that this type of pulse shrinking inverter has lower power consumption around 3.16 mW at 4 GHz for a pulse propagating all the way through a single 60-stage TDC line. This is most likely attributable to lower dynamic power due to lower input capacitance. The cells in figure 4.2a and 4.2b consumed 3.5 and 3.7 mW of power respectively for the same test, i.e. pulses at 4 GHz propagating all the way through a single 60-stage TDC.

Figure 4.4 shows the propagation delay between stages for figure 4.2c. This type of cell showed the fastest propagation through the chain, likely also attributable to the lower input capacitance. In the corners, the propagation delay varied from 6.86 to 9.77 ps in mean time.



(a) An added path to ground where the in signal connects to the NMOS closest to the output.

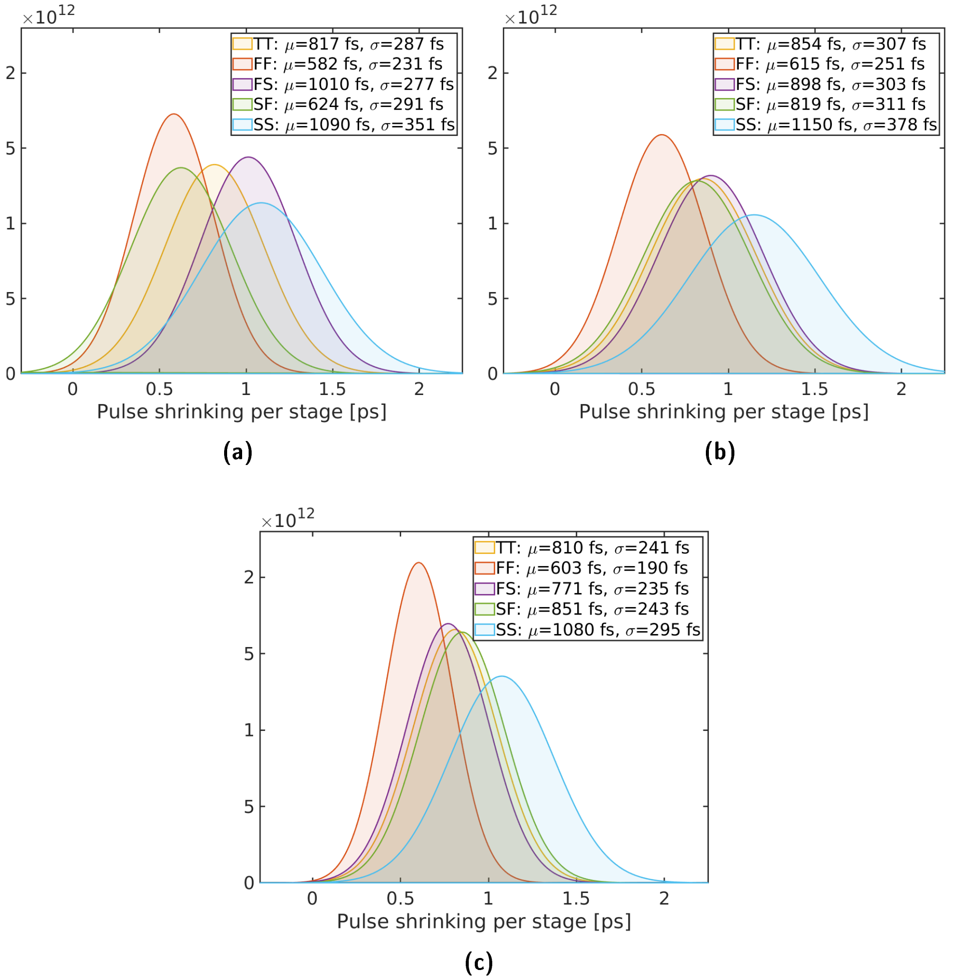
(b) An added path to ground where the in signal connects to the bottom NMOS closest to ground.

(c) A balanced inverter where the NMOS current is limited by a weaker NMOS in the pull-down path.

Figure 4.2: Three types of 1-bit adjustable pulse-shrinking inverters that were tested.

Table 4.1: Statistic parameters from the Monte Carlo simulations of three different pulse shrinking cells in figure 4.2 as well as the power consumption at 4 GHz.

	Standard deviation	Propagation delay	Power consumption
Figure 4.2a	287 fs	10.6 ps	3.5 mW
Figure 4.2b	307 fs	11.1 ps	3.7 mW
Figure 4.2c	241 fs	8.1 ps	3.16 mW



(a)

(b)

(c)

Figure 4.3: Monte Carlo simulations where the amount of pulse shrinking per stage were investigated for the three types of pulse shrinking inverters in figure 4.2a, 4.2b and 4.2c respectively.

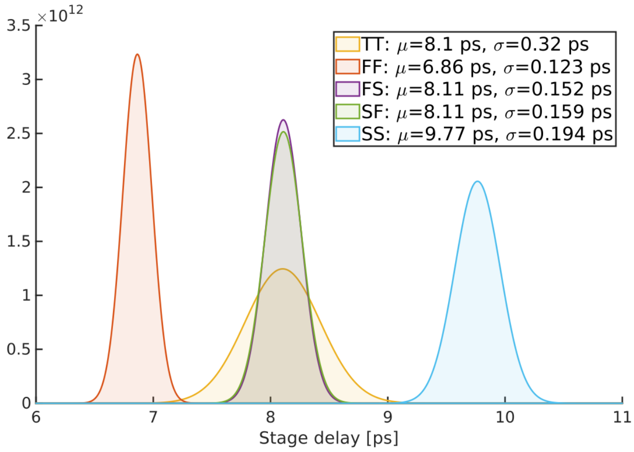


Figure 4.4: The stage propagation delay for the inverter type in figure 4.2c in the different corners.

4.2.1 High resolution mode

Continuing with the current starved PS-inverter in figure 4.2c, the next objective was to characterize the TDC in terms of linearity and how narrow pulses it can detect. This can be done by setting up a test bench with pulse width as a parameter and taking note of which pulse widths produce digital output. The results are shown in figure 4.5. Each of the lines have a linear fit of the data from 1 to 59 in digital output code. The inverse slope of the line is the average step size which has been denoted Δ in the figure. The lowest pulse width that can be detected is 13.9, 16.6, 17.1, 17.9 and 22.0 ps in the FF, FS, TT, SF and SS corner respectively, as shown in table 4.2. This effectively sets a higher limit on the PFD performance since a too short output pulse might not be detected by a TDC. The length of the pulse when the TDC is fully saturated, meaning the pulse propagates all the way through the chain, sets the limit for the low resolution mode in the TDC. If a 65 ps long pulse propagates all the way through a TDC in the high resolution setting, then the low resolution mode needs to have reduced the phase error to less than 65 ps. Ideally however, the phase error should be as close to zero as possible before switching since the signals may drift apart during the change in resolution.

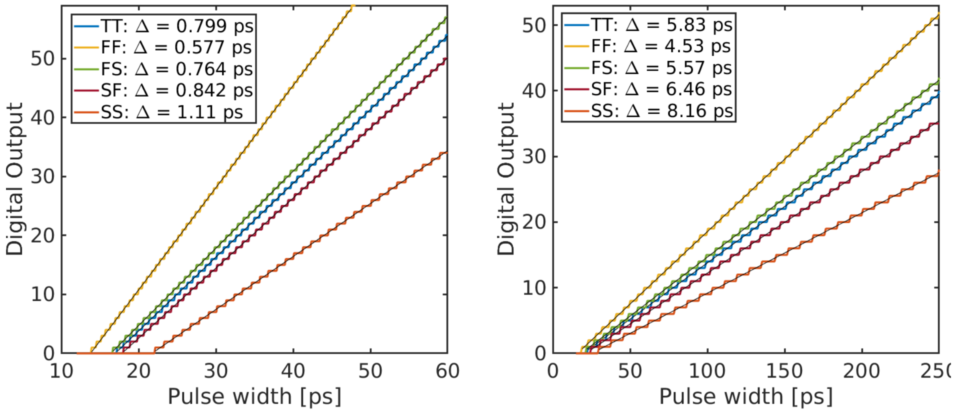
4.2.2 Low resolution mode

The step size in the low resolution mode can be chosen arbitrarily. The simulation in figure 4.5b shows a TDC quantizing pulses up to 250 ps for the different corners. The resolution ranges between 4.53 and 8.16 ps over the corners with a typical resolution of 5.8 ps.

To assess the linearity after manufacturing, 20 iterations of Monte Carlo simulations were done, shown in figure 4.6. The ideal step size (LSB) was determined from a linear fit. Each of the steps along the chain was then measured in units of ideal step size and plotted as a function of stage number (titled "DNL"). The

Table 4.2: Detectable pulse widths for a 60-stage high resolution PS-TDC.

Corner	Shortest detectable pulse width	Length of pulse when saturated
FF	13.9 ps	47.6 ps
FS	16.6 ps	62.3 ps
TT	17.1 ps	64.9 ps
SF	17.9 ps	68.3 ps
SS	22.0 ps	88.5 ps



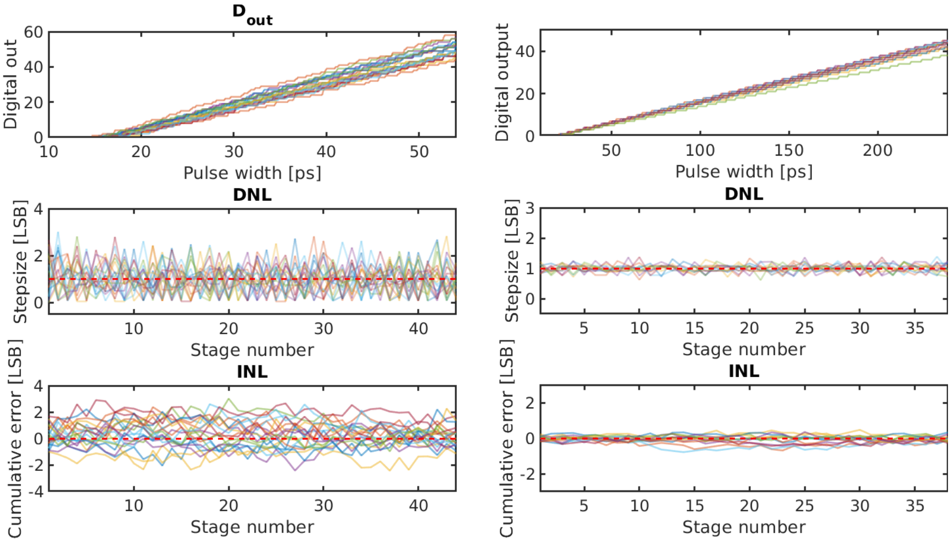
(a) High resolution mode.

(b) Low resolution mode.

Figure 4.5: Digital output for a pulse of a given width in the two different resolution modes.

ideal value for the differential non-linearity (DNL) is 1 LSB. The integral non-linearity (INL) however, was calculated as the cumulative sum of DNL but with 1 ideal LSB subtracted. This shows the built up error and should ideally remain at zero. The high resolution mode showed step size errors up to 2 LSB whereas the low resolution mode had step sizes closer to the ideal. The effect of process variation from device to device might be more reduced than the plots in figure 4.6 imply. Process variation can be divided into across-die and within-die variations. Across-die variations affect the entire die similarly. These are modelled by corners. The variations from the Monte Carlo simulation model the within-die variations. These consist of both spatially correlated and uncorrelated variations. [6]

Most importantly, the data and clock pulse shrinking lines must behave similarly as the pulses propagate through the line. If there is some extra mismatch on the data-line then it is desired to have similar mismatch on the clock line to keep the delay and extra width of the clock pulse. The spatially correlated mismatch between devices can be combatted by a layout which places the clock pulse in-



(a) High resolution mode.

(b) Low resolution mode.

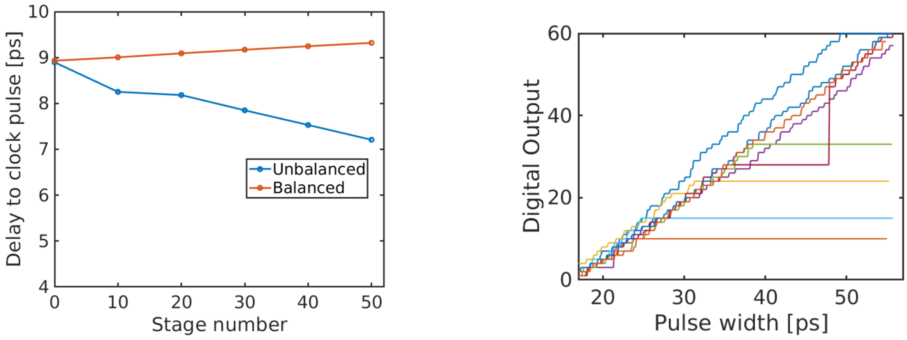
Figure 4.6: Monte Carlo simulations of the TDC chain to assess linearity when accounting for statistical variations in mismatch. Each simulation had 20 iterations of a 60 stage TDC chain.

verter line close to the data pulse inverter line, and the up and down-TDCs in a common-centroid arrangement.

4.2.3 Clock pulse

Delay

The purpose of the clock pulse is to act as clock for the flip-flops and store a one or a zero depending on whether the data pulse reached the stage or not. To be able to do this, the clock pulse must be adequately delayed compared to the data pulse. However, it must not arrive too late such that the data pulse has already passed by. This is a delicate balance although it can be relaxed by letting the clock pulse fall slightly longer behind as the pulse propagates through the chain. This can be done by careful sizing of the data and clock input stage of the flip-flops. For a typical standard cell D flip-flop, the capacitive loads in the clock and data input are not exactly the same. This causes the clock and data pulse to drift apart or together since one line has less capacitive load and therefore also faster propagation time. An example of this is shown in figure 4.7a where a clock line initially had lower flip-flop input capacitance causing it to "catch up" to the data pulse. This resulted in poor capturing of the data, shown in figure 4.7b, especially longer into the chain. The figure shows that in four cases, the clock pulse catches up such that it stops registering ones. In some cases, the clock pulse falls behind again later in the chain and starts capturing ones again.



(a) Example of the clock pulse catching up to the data pulse before balancing the load capacitance between clock and data line.

(b) Monte Carlo simulation showing issues where the clock pulse lags behind causing the flip-flop to wrongfully store a zero.

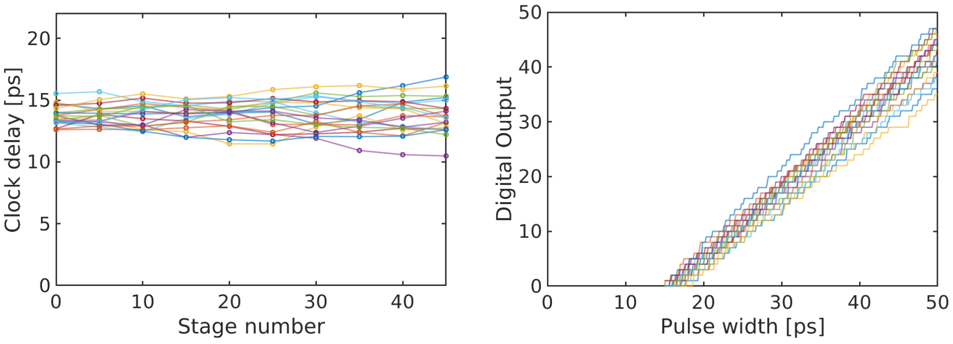
Figure 4.7: Unbalanced clock pulse "catch up" (a) and the effect of it in a Monte Carlo simulation (b).

The issue was resolved by making sure the propagation time is the same for the clock and data pulse. Since all pulse shrinking inverters are the same for the two inverter lines, the issues with propagation delay in the clock line was increased by slightly increasing the size of the flip-flop clock input buffer.

Once the delay drift had been resolved, it was still needed to determine the optimal delay. Figure 4.9a shows the digital output for four different clock delays. As expected, a longer delay to the clock corresponds to a longer minimum detectable pulse width. Another simulation, showed in figure 4.9b shows the timing requirement for the clock pulse to accurately capture the data in the flip-flop. The simulation is done for 50 Monte Carlo iterations of a 20 ps data pulse with delay to the clock pulse on the x-axis. Depending on the delay to the clock pulse, either a one or a zero is stored in the flip-flop. Ideally, the delay to the clock pulse should be such that a one is always stored if the data pulse is present. A delay to design for is therefore in the middle of the interval where ones are sampled in figure 4.9b. The figure shows that the set-up time of the flip-flop is approximately 5.5 ps but ranges up to 7 ps when the Monte Carlo simulation is applied. Similarly, the clock pulse can not arrive less than 1.5 ps before the data pulse passes. The ideal delay is therefore somewhere in between. Since 20 ps pulses are expected while in lock, the clock delay is designed to be 13 ps which is halfway between the two limits and should therefore be most resistant to variation. The clock delay is set by adding a small capacitor at the output of the clock-extender module. A 3 fF capacitor is chosen, giving around 13.8 ps of delay to the clock pulse in the typical case.

Extending

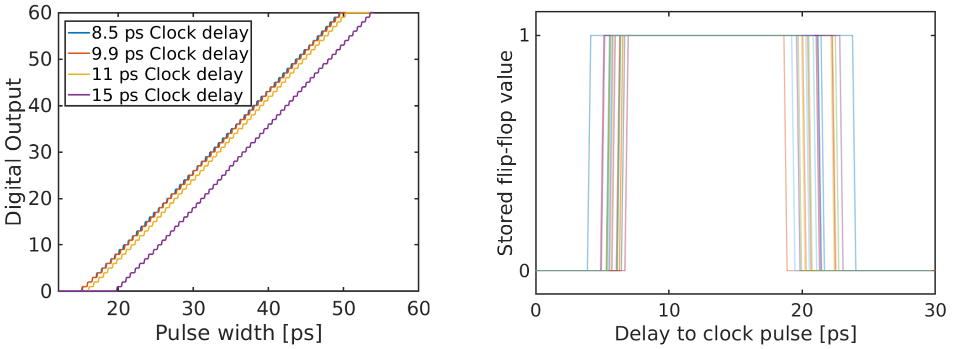
Not only must the clock pulse be delayed compared to the data pulse. It is also crucial that it propagates longer than the data pulse. This is in order to make



(a) Delay from data pulse to clock pulse.

(b) Digital output.

Figure 4.8: Monte Carlo showing the issues in figure 4.7b resolved.



(a) Digital output for different initial clock delay.

(b) The value stored in the flip-flop as a function of clock pulse delay.

Figure 4.9: Clock pulse delay investigations.

sure that atleast one zero is captured when the data pulse has "died off". While extending is not the only way to ensure this, it is a simple measure and has the benefit of also delaying it which was discussed in the previous section. The pulse must be extended more than 1 Δ, not only to ensure that the clock pulse always propagates one stage longer than the data pulse when taking process variation into account, but also to ensure that the clock pulse is strong and long enough not to violate the flip-flop hold time. Table 4.3 shows the mean of the clock pulse extending and clock pulse delay denoted μ as well as their standard deviations denoted σ for each corner.

Table 4.3: Clock pulse extending and delay.

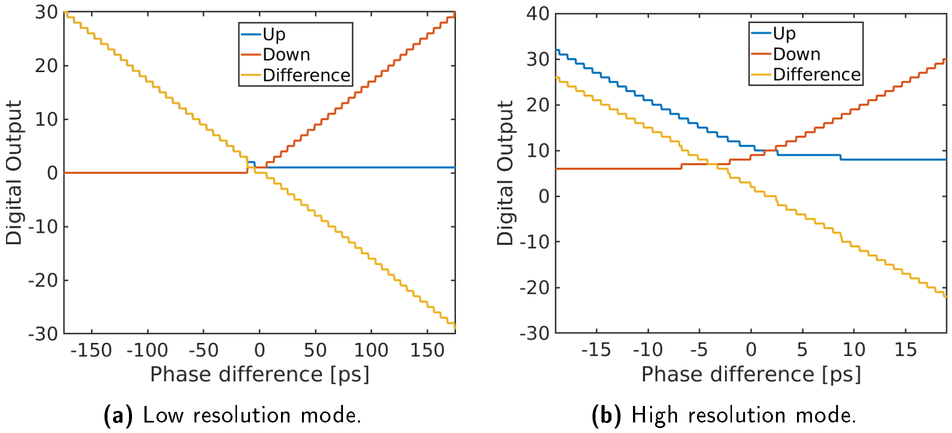
Corner	Pulse extending		Clock pulse delay	
	μ [ps]	σ [ps]	μ [ps]	σ [ps]
FF	1.95	0.52	11.7	0.40
FS	1.47	0.47	14.0	0.50
TT	2.15	0.53	13.8	0.64
SF	2.84	0.48	13.4	0.46
SS	2.61	0.58	16.3	0.59

4.3 System simulations

4.3.1 Phase error

Figure 4.10 shows the phase error when the value for the "Down"-TDC is subtracted from that of the "Up"-TDC. Just as in section 4.2 the length of the pulse is digitally represented by the stage number where the first zero is sampled in the flip-flops. The simulated reference frequency was 4 GHz where the phase difference is created by starting off with a leading reference and letting the divided frequency catch up and overtake. Both TDCs consisted of 60 stages. As well as the difference, the up- and down-pulses are quantized and included in the plot. For the low resolution mode, the difference ranged in digital output from 30 to -30 for phase differences between -180 and 180 degrees. In the high resolution mode, the digital output ranged from 26 to -22 for phase differences between -20 and 20 ps. The reason for the non-symmetry is likely due to the fact that Qa starts from all ones going down to a majority of zeros whereas Qb starts from a majority of zeros and samples ones. At some phase errors, for instance at +8 ps phase difference in the high resolution mode, both the quantized "Up" and "Down" pulse has a step. This results in the difference between them taking a step of 2 since $1-(-1)=2$.

Similar simulations were then redone but including Monte Carlo statistical variations and only the difference between the quantized "Up" and "Down". This



(a) Low resolution mode.

(b) High resolution mode.

Figure 4.10: Simulations in the typical case where the phase error is calculated as the difference from each of the TDC lines.

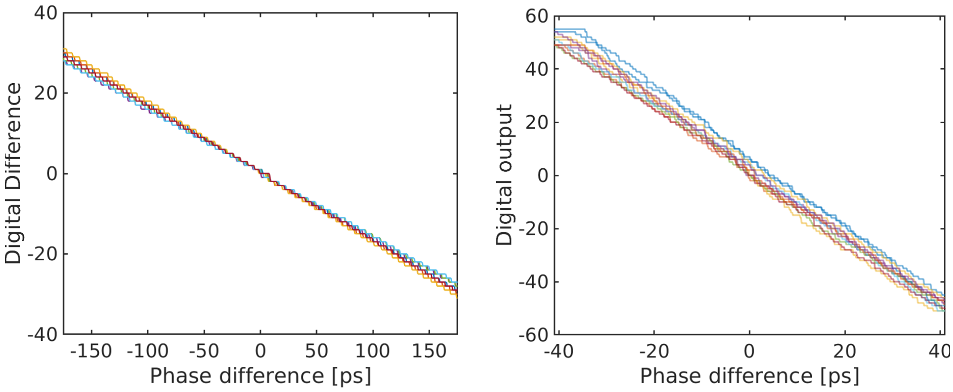
is shown in figure 4.11. The low resolution mode shows consistent behaviour whereas the high resolution mode exhibit some variation in relative step size.

4.3.2 Power consumption

Figure 4.12a shows a moving average of the power consumption when simulating with a reference frequency of 4 GHz in the high resolution mode. When the two signals are in phase, the power consumption is around 2.3 mW in total for the TDCs and the PFD combined. The PFD consumes around 200μW whereas each of the TDCs consume a bit more the 1 mW. In this simulation, the output pulse width was set such that at zero phase difference, 6 stages were active in each TDC chain. The digital output from the same simulation is shown in 4.12b.

In a second simulation regarding power consumption, the phase error was set to zero and the two input frequencies both to 4 GHz. The PFD and TDCs were simulated for 100 ns and the root-mean-square of the voltage-current product was determined. The results are summarized in table 4.4 where even lower power consumption was achieved. In the shortest pulse width setting, 6 stages were active for a total of 2.1 mW power. The reason that these numbers differ slightly from that of figure 4.12a is likely because the phase error in 4.12a is only zero during one single cycle and the power is averaged over a few thousand cycles giving a higher value.

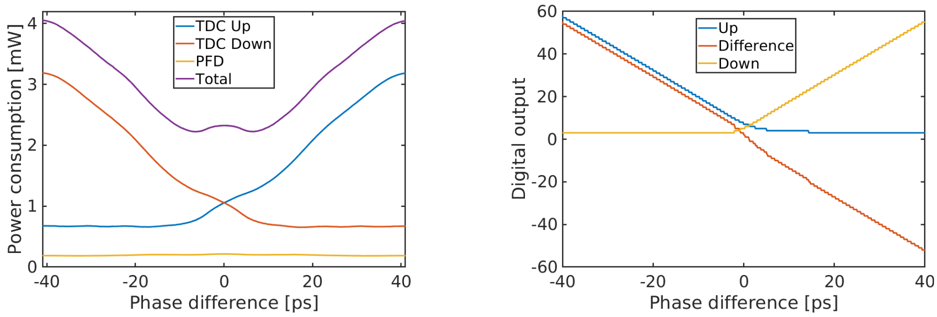
The performance of the PFD can be boosted by using body bias. The devices used in the PFD are "Super Low Threshold" type FETs. These are flip-welled meaning an N-well under NFETs and a P-well under PFETs. With flip-well devices the N-well can be forward biased up to 2 V and the P-well can be forward biased to -2 V with good modelling. This increases the current from the devices with the trade-off being higher leakage power. [7]



(a) Low resolution mode.

(b) High resolution mode.

Figure 4.11: Monte Carlo simulations of the difference of digital output between the up- and down-TDC.



(a) Power consumption divided up in PFD, TDC Up and TDC Down.

(b) Digital output showing active stages at each phase error.

Figure 4.12: Power consumption simulated at 4 GHz.

Table 4.4: Power consumption while in lock for different PFD settings.

Setting	Power [mW]			Active stages per TDC
	PFD	Single TDC	Total	
00	0.24	1.21	2.7	13
01	0.25	1.07	2.4	9
10	0.25	0.97	2.2	7
11	0.24	0.92	2.1	6

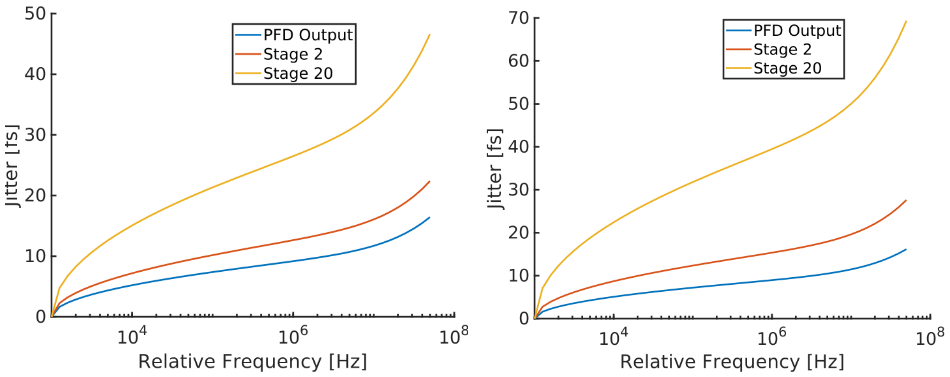
Table 4.5: Power consumption while in lock for different PFD body biases.

Body bias [V]	Power [mW]			Active stages per TDC
	PFD	Single TDC	Total	
0	0.244	0.924	2.09	6
0.2	0.252	0.867	1.99	5
0.4	0.259	0.803	1.87	4
0.6	0.268	0.750	1.77	2
0.8	0.277	0.693	1.66	1
1.0	0.287	0.639	1.57	0

Body bias ranging from 0 to 1 V was tested in the PFD. Positive for the N-devices and negative for the P-devices. For these tests, the PFD was set to the lowest pulse width setting, i.e. 11. The results are summarized in table 4.5. Using body bias, the output pulses could be made even shorter, activating fewer stages. Without body bias, 6 stages were active in each TDC giving a total power consumption of 2.1 mW. With 0.6 V body bias, the TDCs only sampled 2 ones each before the first zero. With the pulses "dying off" earlier, power consumption was reduced to 1.77 mW. At 1 V body bias, no ones were sampled in the flip-flops. The clock pulse managed to propagate at least one stage causing the first flip-flop to store a zero. The power consumption was reduced to 1.57 mW, although this might be to push the PFD too far. As body bias was increased, power consumption in the PFD increased as well. This is likely due to increased leakage power. However, the power consumption in the TDCs reduced more than the PFD increase making it an overall decrease in power consumption.

4.3.3 Noise

Figure 4.13a shows the jitter at 4 GHz at the PFD output, 2$^{\text{nd}}$ and 20$^{\text{th}}$ stage of the TDC. Using a 30 MHz loop bandwidth, jitter of about 16 fs is observed at the PFD output and 22 fs at the output of the second TDC stage. The jitter increases further into the line. At the 20$^{\text{th}}$ stage, jitter has increased to 48 fs. However, by design only about 2 to 5 stages should be active while the PLL is in lock. The jitter was also simulated for another TDC with similar resolution but using devices with half the width, shown in figure 4.13b. At the PFD output, both TDCs show the same amount of jitter. In the TDC however, the jitter is 28 and 70 fs at the 2$^{\text{nd}}$ and 20$^{\text{th}}$ stage. Therefore, a trade off between jitter and power in the TDC is recognized.



(a) Larger devices.

(b) Smaller devices.

Figure 4.13: Simulation showing jitter in the PFD and TDC.

Chapter 5
Conclusion

A digital phase-locked loop was presented that uses the phase/frequency detector from a charge pump PLL. The phase difference is quantized by sending the PFD pulses into a TDC which quantizes the length of the pulse based on how far into the chain it propagates. This is done by having a small amount of pulse shrinking each stage. The number of propagated stages is then related to the length of the pulse. When the PLL is in lock, there are short pulses from the PFD which do not propagate many stages in the TDC. Therefore, the system can be used at high reference frequency which implies a lower division ration and less magnification of reference and quantization noise.

The pulse shrinking is achieved by having an inverter with longer fall time than rise time. This is followed by a balanced inverter to restore the pulse shape. The pulse shrinking cell can be made adjustable by controlling the fall time. Several ways of doing this has been presented with the conclusion that a current starved pull-down path has lower pulse shrinking standard deviation, power consumption and propagation time. The standard deviation of pulse shrinking was 241 fs in the typical case.

The TDC is adjustable in two settings with a high resolution mode that has 0.8 ps step size and a low resolution mode with 6 ps steps. To accommodate for process variation and to reduce jitter and power consumption, the PFD output pulse width was made adjustable by having a current starved reset path. Monte Carlo simulations of the TDC showed step size errors up to 2Δ and cumulative errors up to 2Δ in the high resolution mode. Flip-flops store the information on whether the pulse has reached the stage or not. These are clocked by separate pulse shrinking inverter chains which propagate the same pulse although extended and delayed. The optimal delay for robustness was found to be around 13 to 14 ps and the extension of the pulse was chosen to around 3Δ.

Finally the PFD and TDC was tested together and the phase error was calculated as the difference in digital output for the up- and down-TDC in both the high and low resolution mode. The power consumption for the PFD and TDC was also determined. When the loop is in lock, the power consumption was reduced to 2.1 mW at 4 GHz. About 0.25 mW of this was consumed by the PFD whereas each of the TDC-chains used just above 0.9 mW of power.

The phase jitter in the TDC was determined at different locations in the chain. At the PFD output, 16 fs of phase jitter was simulated when using a 30 MHz bandwidth. At the second and 20th stage, simulations showed phase jitter of 22 and 48 fs respectively.

References

[1] Razavi Behzad. *RF microelectronics.* Vol. 2. New York: Prentice hall, 2012.

[2] Razavi, Behzad. *Design of CMOS Phase-Locked Loops: From circuit level to architecture level.* Cambridge University Press, 2020. Razavi Behzad. *RF microelectronics.* Vol. 2. New York: Prentice hall, 2012.

[3] Staszewski, Robert Bogdan, and Poras T. Balsara. *All-digital frequency synthesizer in deep-submicron CMOS.* John Wiley & Sons, 2006.

[4] William F. Egan. *Frequency Synthesis by Phase Lock*, John Wiley & Sons, 1998.

[5] Lee, Won-Hyo, Jun-Dong Cho, and Sung-Dae Lee. *A high speed and low power phase-frequency detector and charge-pump.* In Proceedings of the ASP-DAC'99 Asia and South Pacific Design Automation Conference 1999 (Cat. No. 99EX198), pp. 269-272. IEEE, 1999.

[6] Sharma, Arvind K., Meghna Madhusudan, Steven M. Burns, Parijat Mukherjee, Soner Yaldiz, Ramesh Harjani, and Sachin S. Sapatnekar *Common-centroid layouts for analog circuits: Advantages and limitations.* In 2021 Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 1224-1229. IEEE, 2021.

[7] Presenter Blackwell, Don. *"Analog Design Workshop for 22FDX 22nm FD-SOI Technology".* Webinar from GlobalFoundries. August 2017. https://www.youtube.com/watch?v=Rf2kL1lUJ10.

Acronyms

CMOS	Complimentary Metal-Oxide Semiconductor.
CP	Charge Pump.
CPPLL	Charge Pump Phase Locked Loop.
DCO	Digitally Controlled Oscillator.
DNL	Differential Non-Linearity.
DPLL	Digital Phase Locked Loop.
FET	Field Effect Transistor.
FF	Fast NMOS Fast PMOS Corner.
FS	Fast NMOS Slow PMOS Corner.
INL	Integral Non-Linearity.
LF	Loop Filter.
LSB	Least Significant Bit.
MIMO	Multiple Input Multiple Output.
NMOS	n-type Metal-Oxide Semiconductor.
PFD	Phase/Frequency Detector.
PLL	Phase Locked Loop.
PMOS	p-type Metal-Oxide Semiconductor.
PS-TDC	Pulse Shrinking Time-to-Digital Converter.
PS-DPLL	Pulse Shrinking Digital Phase Locked Loop.
SF	Slow NMOS Fast PMOS Corner.
SS	Slow NMOS Slow PMOS Corner.
TDC	Time-to-Digital Converter.
TT	Typical NMOS Typical PMOS Corner.
VCO	Voltage Controlled Oscillator.