Contents

5 Phase-Locked Loops

Chapter Contents

In this chapter we present an introduction to Phase-Locked Loop, PLL, circuits, the gold-standard for on-chip clock generation.

PLLs have many different applications with uses in radio, telecommunications, computers, and other electronic devices. Each of these applications has specific functional requirements resulting in a wide variety of circuit designs. In purely analog circuits, the signals are typically sinewaves, while in digital circuits, the signals are usually square waves. We will focus here on their application in computer chips, and specifically for on-chip clock generation for digital circuits where all the clock signals are square waves. We discuss PLL characteristics focusing on the quality of their clock output signal, circuit complexity, size, sensitivity, and flexibility.

PLL circuits are chosen because of their ability to generate very high quality stable clock signals. Furthermore, they are very flexible and can be programmed to change their output frequency over a wide range. As chip complexity and sizes have grown, PLL generated clocks have enabled reliable data transfers within the modules on the chip, between these modules, and from chip to chip. Their ability to operate at very high frequencies has also been a key factor for their continued use and development. Their stability is achieved by use of a very stable reference clock source and being intentionally slow to respond to environmental changes. Thus, they are excellent for applications where

the clock frequency is not required to change very often. Different types of chip designs often have specific clocking requirements, and there is no single PLL circuit that is optimal for all applications. Furthermore, the combination of analog and digital circuit components in conventional PLL designs has resulted in their circuit development becoming a very specialized design skill.

The development of fully digital PLL designs provides smaller and lower power replacements in many applications, but they do not yet achieve the same quality of output clock signals as conventional PLL circuits.

Some of the clock control schemes presented in later chapters of this book have non-typical needs that require fast response times when changing the output frequency. In these situations PLLs are unsuitable, and alternative on-chip clock generation techniques are required.

There are a wide range of good authoritative books and papers on PLL circuit design which cover the theory, detailed operation, and practical circuit implementations. The material presented here is a summary drawn from many of these sources and we recommend the references, [1], [2], and [3], for those readers that wish to delve deeper.

5.1 Overview

In this chapter we present two main types of PLLs, traditional analog PLLs, and more recently developed digital PLLs. Our objective is to supply enough details and operational descriptions of PLLs to understand how they work, why they work well, and to understand their limitations. We present simple circuit implementations to illustrate how they work, supported by simulations to demonstrate their operation. We start by covering analog PLL design because these are the mainstay of on-chip clock generators. A PLL consists of multiple, separable modules, each of which can have a wide variety of implementations. Better module designs result in improved clock quality, in terms of greater stability and lower jitter, but usually come at the cost of increased size and power consumption. We then describe digital PLLs, presenting example implementations of each of their modules, supported by simulations to demonstrate their operation. We conclude by summarizing the advantages of both analog and digital PLLs.

5.2 Introduction

Compute circuits on chip are commonly designed as pipelined stages of combinational logic sandwiched between clocked registers as illustrated in Figure 5.1. In most chips the clock runs at a single fixed frequency. The registers separate the current data values from the next data in much the same way that the plastic

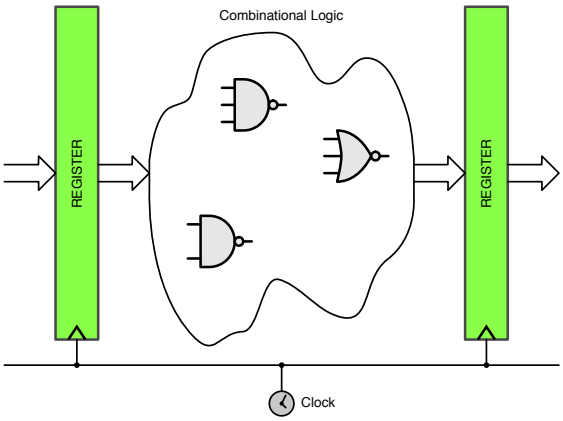

Figure 5.1
Combinational compute logic placed between clocked registers.

Figure 5.2
Grocery store checkout conveyor belt dividers.

dividers separate the grocery items of one customer from the next at the grocery store as shown in Figure 5.2. Data pipelining enables combinational logic to be utilized a greater proportion of the time. The amount of combinational logic per stage is chosen to have about the same delay per stage, with the clock frequency set to minimize the idle time of the combinational logic while also accommodating the worst-case delay.

At each clock tick all registers are updated: the results of the computation in the current stage are stored in the output register, which is the input register of the next stage; and new data values are captured in the input register of the current stage for the next computation. Operation of complex circuit modules is greatly simplified using such pipelined stages, where clocking of all the registers is carried out by a single clock signal. For correct computation, the data must propagate completely through the combinational logic before the next clock tick. Alternatively, the clock must be made slow enough to accommodate the longest delay path through the combinational logic. Higher clock rates enable more processing to be carried out per second, and so a huge amount of effort is focused on minimizing logic path delays and balancing the amount of logic per stage. Variations in the arrival time of the clock signal, known as clock jitter, between the input and output registers subtract from the amount of time available to complete the computation function of the combinational logic. Such clock jitter can be compensated for by lowering the clock frequency, which provides more time margin per stage, but this reduces performance. A similar situation arises when passing data between modules on a chip, or when transferring data between chips. When a single clock is used, variations in the arrival time of the clock signal at these interfaces must be taken into account to ensure reliable data transfers. Thus, compute performance and data transfer reliability are improved when the clock signal has less jitter.

As chip technology has advanced and the circuits have become faster, higher frequency clocks were needed. Delivering a high frequency, high quality clock for direct use on a chip became a challenge for operational frequencies above a few hundred MHz. At the time, generating a high frequency stable clock was not only difficult, but delivering this clock to chips via the traces on a printed circuit board was a major challenge. At these high frequencies, the dimensions of wire traces and their proximity to neighboring wires results in clock signal waveform degradation. In addition, signal coupling from the clock wire to adjacent data wires introduces noise onto data wires. Furthermore, the clock wires act as antennas, with radiation of the clock signal causing interference with other circuits. These problems led to a multitude of techniques for generating on-chip high frequency clocks. The characteristics of these on-chip clock generators span the range from small ring oscillator designs, that have poor stability and high sensitivity to environmental changes such as supply voltage and temperature, to very large circuits that produce high quality stable clocks.

Phase-Locked Loop, PLL, clock generator circuits, despite their large size, became the standard for clocking chips because they can generate very stable,

low jitter, on-chip clock signals. A further advantage is that their output frequency is programmable, although changing frequency can take many tens to even hundreds of clock cycles. PLLs use a high quality, stable, and low frequency reference clock source, which is usually off-chip. Their output clock signal is usually a multiple of the reference frequency, while achieving low jitter that is similar to that of the reference source.

High stability, reference clock sources usually employ a quartz crystal to govern their oscillation frequency. Crystal clock generators at 100MHz with jitter of 0.1 ps or less are readily available.

PLL circuits for digital logic come in two main design styles, traditional PLL circuits that contain multiple analog circuit components, and the more recently developed fully digital PLL implementations. Currently, analog PLL circuits generate higher quality clocks. However, digital PLL designs are becoming increasingly popular because they are considerably lower power and also smaller in size. New and improved digital PLL designs continue to be developed, although currently they do not achieve such high quality clocks as the traditional designs.

5.3 Analog PLL Circuits

The gold-standard on-chip clock generator has become the Phase-Locked Loop, PLL. PLLs are chosen for their ability to generate a very high quality on-chip clock signal that has low jitter. Furthermore, the PLL output clock frequency can be digitally controlled. These PLLs use an off-chip clock reference, typically from a very stable low frequency quartz crystal oscillator, usually in the range 10 to 100MHz. The on-chip PLL circuit generates a clock that is a multiple of this reference clock frequency, with the extremely important characteristic that the PLL clock has almost the same stability as the reference clock.

The basic PLL concept is shown in Figure 5.3. The circuit deploys an off-chip oscillator, OSC, typically a high-quality stable quartz crystal reference frequency oscillator, f_{REF}. The other components of the PLL are all on-chip. A second oscillator, a Voltage Controlled Oscillator, VCO, generates the output clock signal from the PLL, f_{PLL}. Both of these oscillators generate square wave signals. A negative feedback loop delivers the VCO output signal to the Phase Detector component where it is subtracted from the incoming reference signal. The Phase Detector produces a series of pulses whose width are proportional to the phase difference of these two input signals. These pulses are then filtered by the Loop Filter to produce the analog control voltage for the VCO. When the phase of f_{REF} leads that of f_{PLL}, a series of positive pulses will be generated,

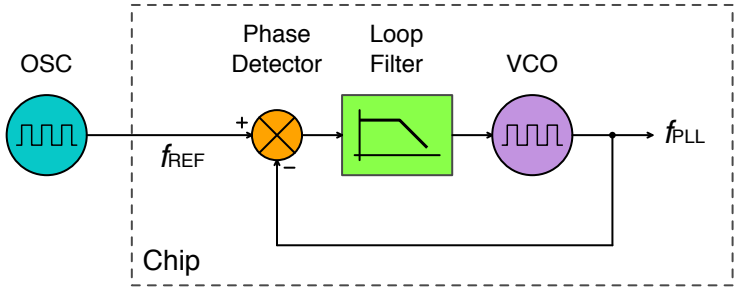

Figure 5.3
Basic PLL design that uses an off-chip stable reference frequency source, f_{REF}, for generating an on-chip clock signal f_{PLL}.

producing a rising positive VCO control voltage, increasing the frequency of the VCO. Conversely, when the phase of f_{PLL} leads that of f_{REF}, negative pulses lower the frequency of the VCO. In this manner the control loop of the PLL not only makes f_{PLL} the same frequency as f_{REF}, but it also maintains a fixed phase offset between them. The feedback loop of a PLL causes the output signal to settle to its designed frequency and to lock to the phase of the reference clock.

The fixed phase relationship property of a PLL is extremely valuable because it means that a single reference oscillator can be used to keep multiple clock domains in phase with each other, even when these clock domains are on other chips. Tight control of the clock phase used by each module can be achieved by selecting the feedback signal from the output of the clock distribution tree, as illustrated in Figure 5.4. This tight phase control enables direct data communication between modules as shown in Figure 5.5. However, when the clock skew between the two modules becomes too large, such as when there are multiple interface connections and tight enough phase control cannot be achieved, a Phase Buffer FIFO can be inserted to accommodate for the clock skew, as illustrated in Figure 5.6. A phase buffer is a form of FIFO where both the inputs and output interfaces are clocked, where the clocks have the same frequency, but different phase. A phase buffer can also accommodate a range of phase drift. A dual-port SRAM can be used as a phase buffer, while for small buffer sizes, a flow-through FIFO using flip-flops out-performs SRAM circuits. Just a few stages of a well designed phase buffer circuit can compensate for several cycles worth of phase difference. PLLs also enable reliable communication between quite different modules and between chips, as illustrated in Figure 5.7 and Figure 5.8.

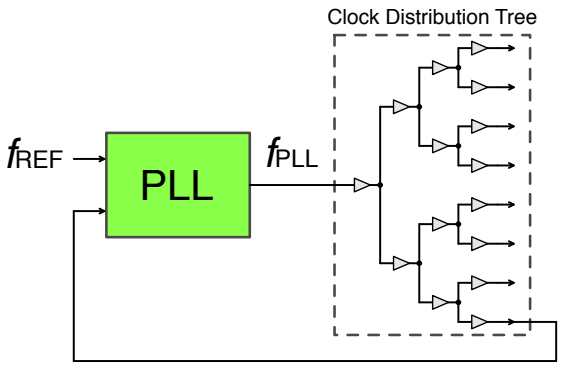

Figure 5.4
Example PLL use that enables control of the frequency and phase of the clock signals at the output of a clock distribution network.

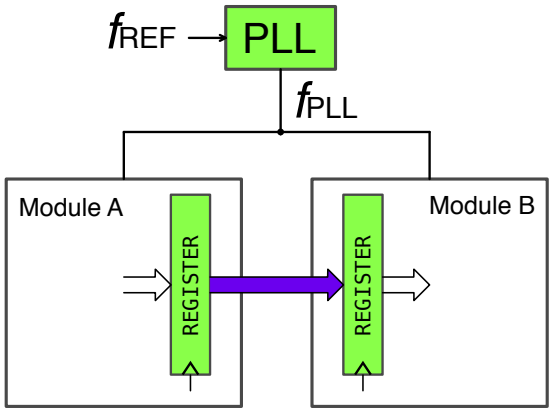

Figure 5.5
A single PLL enabling data transfer between two similar modules.

The PLL design of Figure 5.3 is limited to making f_{PLL} a copy of f_{REF}. Next comes another real advantage of a PLL: it can be used to generate frequencies that are different from the reference frequency, while maintaining almost the same stability as the reference clock. By dividing down the frequency of the f_{PLL} signal that is fed back to the phase detector, f_{PLL} is made to be an integer multiple of the frequency of the reference clock f_{REF}. Figure 5.9 illustrates

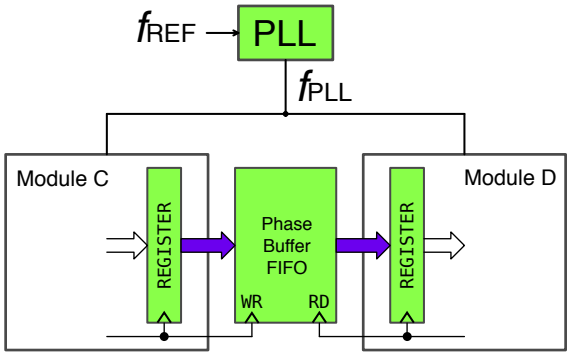

Figure 5.6
A single PLL clocking to two different modules using a Phase Buffer FIFO to accommodate clock skew between the modules.

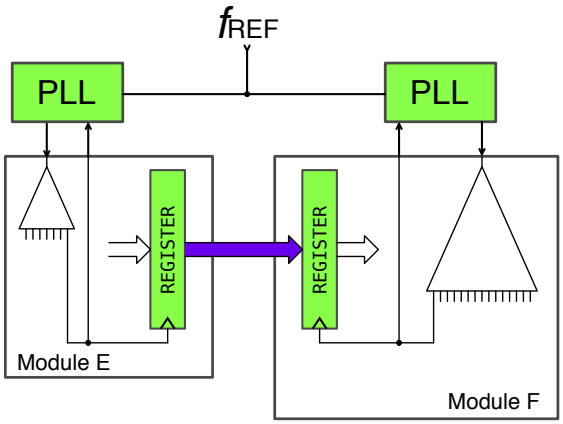

Figure 5.7
Two PLLs that use the same clock reference source to enable reliable communication between quite different modules.

a PLL where a divide-by-N counter has been introduced into the negative feedback loop, to generate an output clock signal $f_{\mathrm{PLL}} = \mathrm{N} \times f_{\mathrm{REF}}$.

PLLs that use the same reference clock, but with different integer multipliers, generate different frequency clocks that are phase-related as illustrated in Figure 5.10. Often modules on a chip cannot run at the same high frequency as other modules, and there is a big advantage to run them at a fixed integer ratio

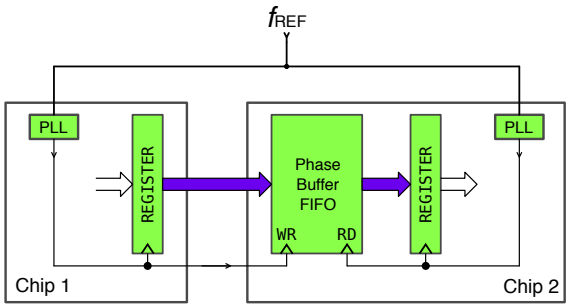

Figure 5.8
Two PLLs that use the same clock reference source to enable reliable communication between chips.

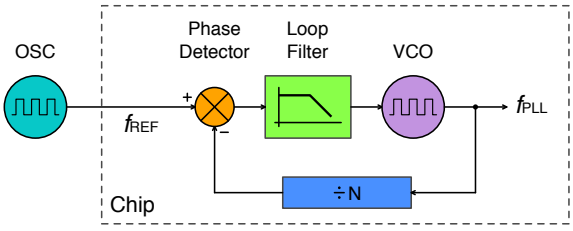

Figure 5.9
Basic PLL design with a divide-by-N counter in the feedback loop for generating an on-chip clock signal with frequency $f_{PLL} = N \times f_{REF}$.

clock frequency to neighboring modules, e.g., large caches often are clocked at half the frequency of cpu cores. Maintaining a fixed phase relationship between these clocks facilitates data transfers without the need for synchronizer circuits.

5.4 Analog PLL Components

Before progressing on to more elaborate PLL circuits, let us first look at the components in these first two designs to learn about their features, constraints, and limitations.

A common circuit implementation of the Phase Detector, PD, component is shown in Figure 5.11. This circuit, a Phase Frequency Detector, PFD, is used because it has greater functionality than a pure phase detector. It responds to both the phase difference and the frequency of the inputs, and operates well over a wide frequency range. This circuit contains a pair of flip-flops clocked by f_{REF} and f_{PLL} respectively. These flip-flops are configured to produce a

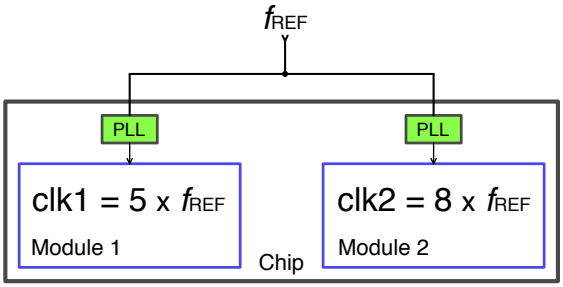

Figure 5.10
Two PLLs generating phase-related, different frequency, on-chip clocks.

logic-1 output after every rising clock edge. The AND gate followed by the delay element clears both flip-flips after both of their outputs have become logic-1, resulting in pulses on their Q output signals q_P and q_N respectively. The duration of these pulses corresponds to the phase difference between f_{REF} and f_{PLL}, with q_P having a longer duration than q_N when f_{REF} arrives before f_{PLL} and vice versa. The signal q_P is used to pull the output PFD_{OUT} high, while the signal q_N is used to pull the output PFD_{OUT} low. Example timing waveforms for the PFD circuit are illustrated in Figure 5.12 where both f_{REF} and f_{PLL} have about the same frequency, but different phase offsets. In the upper part of the figure the phase of f_{REF} leads that of f_{PLL}, resulting in short high-going pulses on PFD_{OUT} that will increase the frequency of f_{PLL} slightly, thus reducing the phase difference between these two signals. While in the lower part of the figure the phase of f_{REF} lags that of f_{PLL}, resulting in short low-going pulses on PFD_{OUT} that will decrease the frequency of f_{PLL} slightly, again reducing the phase difference between these two signals.

The delay element in the PFD circuit is important to avoid a dead zone in operation when the phases of f_{REF} and f_{PLL} are closely aligned. The dead zone is illustrated in the diagram shown in Figure 5.13. This is a plot of the PFD output voltage, PFD_{OUT}, vs input phase difference, $\Delta\Phi = \Phi f_{REF} - \Phi f_{PLL}$, where the slope of this response corresponds to the gain, $K = \delta PFD_{OUT} / \delta\Delta\Phi$, of the PFD circuit. The dead zone is the region in the center of the plot, when the input phase difference is small, and the gain has become zero. This dead zone arises from the non-ideal behavior of transistors, and adding the delay before resetting the flip-flops removes this dead zone.

The issue of avoiding such a dead zone will come up again when discussing practical circuit implementations. Another benefit of this PFD circuit is that it is insensitive to the duty cycle of the incoming frequency waveforms, because

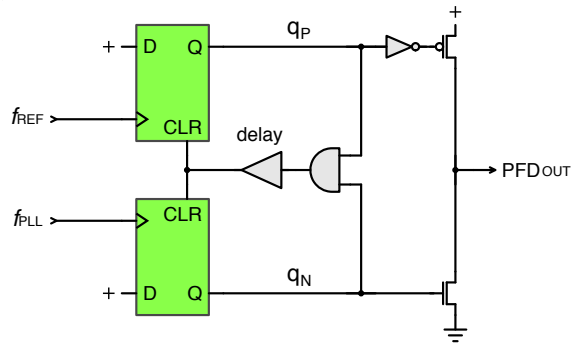

Figure 5.11
PLL Phase Frequency Detector, PFD, circuit.

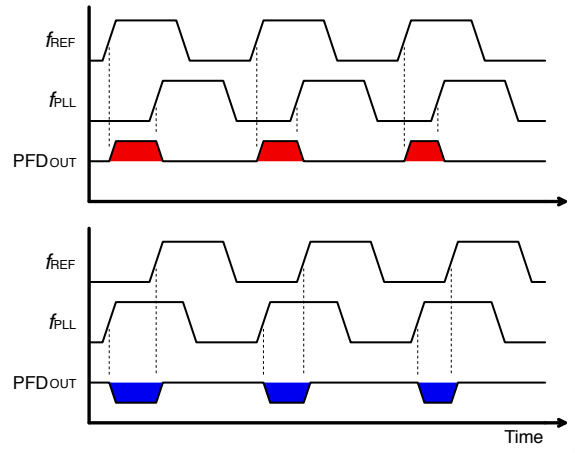

Figure 5.12
PLL Phase Frequency Detector, PFD, circuit timing waveforms.

it operates on rising input signal edges. This is important because in circuit designs it is hard to achieve square wave signals that have equal logic high and low durations except by dividing a signal by 2. It is also hard to achieve equal rise and fall times of a signal, because different transistor types are used to pull a signal up vs down, further unbalancing the duty cycle of a signal.

Waveforms from a simulation of a Phase Frequency Detector are presented in Figure 5.14. In this simulation the frequency of f_{REF} is kept constant while the phase and frequency of f_{PLL} is swept over a small range from leading to lagging

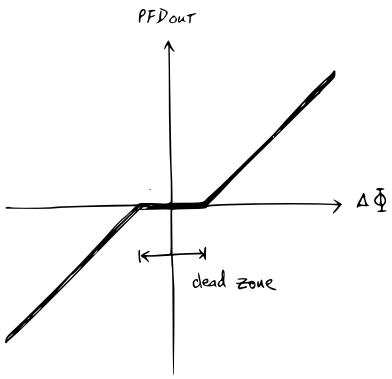

Figure 5.13
PFD output voltage, PFD_{OUT}, vs input phase difference, $\Delta\Phi$, illustrating the dead zone region when $\Delta\Phi$ is small.

f_{REF} in phase. As these simulation results confirm, each flip-flop produces a rising output when it is clocked by its input frequency signal, and after both flip-flop outputs have transitioned high, they are both cleared, returning their outputs low. As illustrated in this figure, the flip-flops produce pulses on their outputs q_{P} and q_{N}, where the width of the pulses are proportional to the phase difference between f_{REF} and f_{PLL}. The delay element ensures that these pulses always exist and have a minimum width. The pulse signals on q_{P} and q_{N} are combined by the output circuit into the single signal PFD_{OUT}, such that positive-going pulses are generated when the phase of f_{PLL} leads that of f_{REF}, while negative-going pulses result when the f_{PLL} lags that of f_{REF}.

The waveforms of a circuit simulation of the Phase Frequency Detector of Figure 5.11 are presented in Figure 5.15. Asymmetries in these waveforms arise from non-ideal behaviour of the transistors. Improved circuits that control current rather than voltage overcome some of this non-ideal operation, but at the cost of greater circuit complexity and power.

We now move on to the Loop Filter, whose role is make the PLL control loop stable. Commonly this filter is a Low Pass Filter, LPF, which integrates out the pulses generated by the Phase Detector to produce a smoothed control voltage for the Voltage Controlled Oscillator. A simple low pass filter circuit, composed of an R-C combination, is shown in Figure 5.16.

The capacitor required for this filter, if implemented as a parallel plate capacitor using metal layers of the chip, would take up an impractically large amount of chip area. Instead, CMOS transistor gate capacitors are used because these provide roughly 10x greater capacitance per unit area. However,

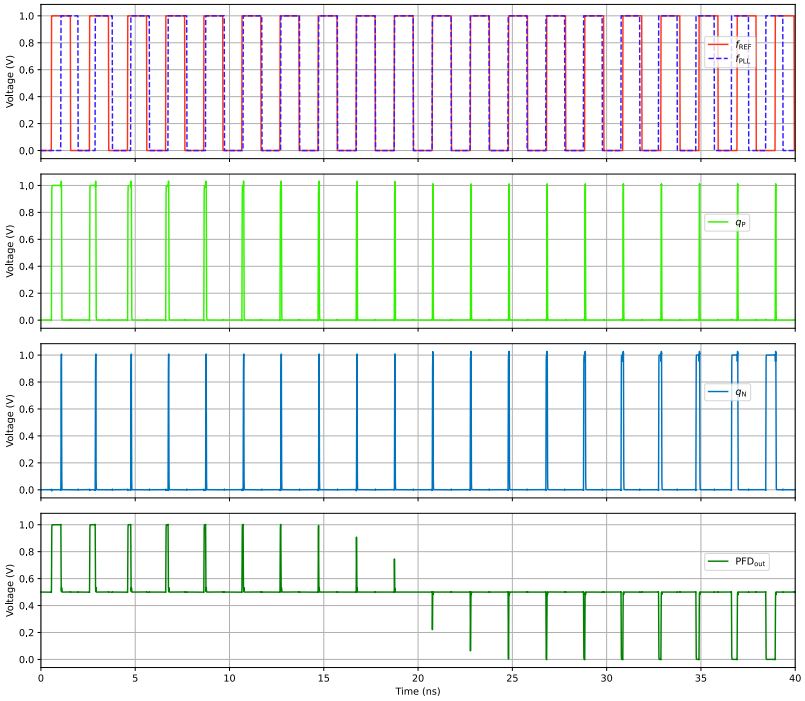

Figure 5.14
Phase Frequency Detector ideal circuit waveforms.

the capacitance of the gate capacitors is voltage dependent, and a P-transistor and N-transistor pair is used to provide a less voltage sensitive circuit as is illustrated in Figure 5.17.

A PLL can produce an output frequency f_{PLL} that is a fractional multiplication of f_{REF} as shown in Figure 5.18. The design present in this figure produces an output frequency of $f_{\text{PLL}} = f_{\text{REF}} \times (\text{N} \div (\text{R} \times \text{D}))$.

There are a variety of circuits that operate well as Voltage Controlled Oscillators, VCOs. We present a simple VCO circuit implementation, shown in Figure 5.19, which uses current-starved inverters in a ring oscillator. This circuit uses analog control voltages, V_{CNTL} and $\text{V}_{\text{DD}} - \text{V}_{\text{CNTL}}$ to adjust the delay of the inverter stages in the ring.

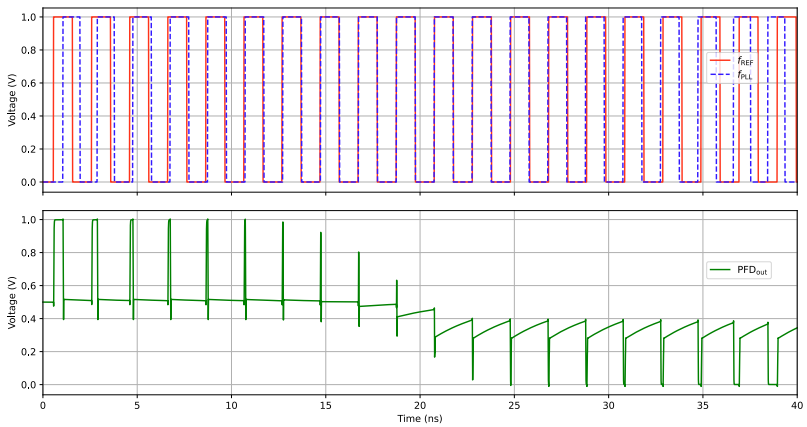

Figure 5.15
Phase Frequency Detector real circuit waveforms.

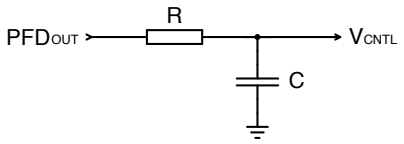

Figure 5.16
R-C Low Pass Filter circuit.

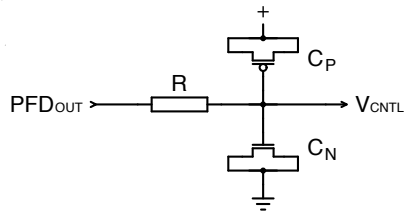

Figure 5.17
R-C Low Pass Filter circuit where the capacitor is implemented by a pair of CMOS transistor gate capacitors, C_P and C_N.

The range of output frequencies produced by sweeping the VCO control voltage, V_{CNTL}, of the 17-stage current-starved inverter ring oscillator circuit

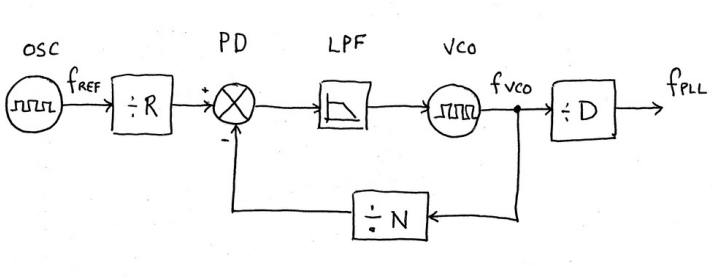

Figure 5.18
PLL with fractional ratio frequency multiplication. Counters R, N, and D, enable f_{PLL} to cover a wide frequency range in small step increments. The PLL frequency is $f_{PLL} = f_{REF} \times (N \div (R \times D))$

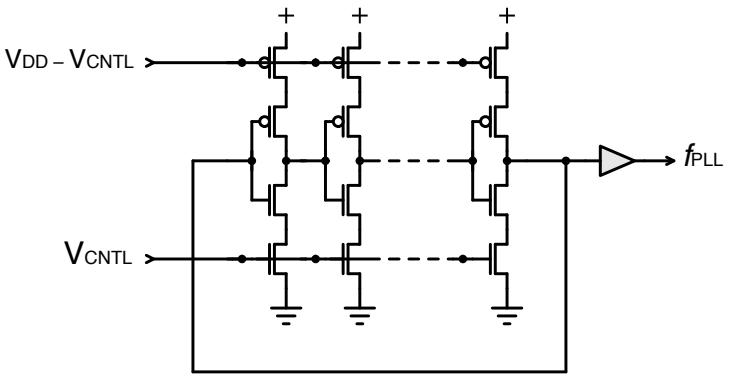

Figure 5.19
Example VCO circuit using a current-starved inverter ring oscillator.

implementation is shown in Figure 5.20. In this case, the output frequency ranges from 100 MHz to almost 3 GHz. The response of this VCO circuit is fairly linear in the middle of its range, but as expected, the frequency changes saturate when the input control voltage approaches the supply rails. In fact, this circuit completely stops oscillating when the control voltage goes below 0.3V. Fortunately, this does not matter, because this type of oscillator is self-starting as soon as the control voltage rises above 0.3V. The important characteristic of the VCO is that the frequency changes monotonically over the whole range of the input control voltage because a non-monotonic response can cause the PLL to be unstable or even lock to the wrong frequency.

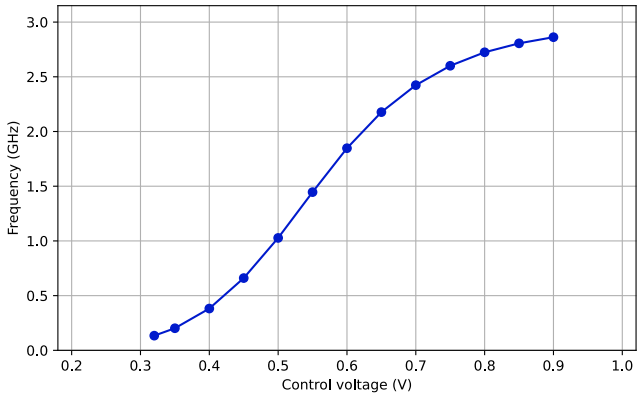

Figure 5.20
Frequency vs Control voltage of a 17-stage current-starved inverter ring oscillator.

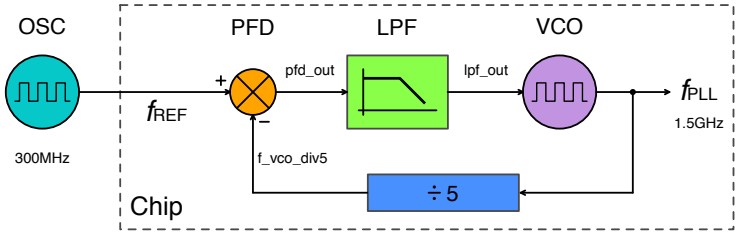

Figure 5.21
Simple x5 PLL circuit.

We can now build a simple PLL by assembling together these components. We have chosen to generate a 1.5 GHz output frequency for f_{PLL}, thus operating the VCO in the center of its range. By selecting a reference oscillator frequency of 300 MHz, we require a divide-by-5 counter in the feedback loop, as shown in Figure 5.21. The divide-by-5 counter implementation chosen is a 3-bit state machine that cycles through the binary count sequence, wrapping around after the fifth value, shown in Figure 5.22. An extra flip flop is used to produce an equal duty cycle output.

Simulation results of this circuit, using a Spice circuit simulator, are shown in Figure 5.23. For this simulation, the node voltages were initialized such that the circuit starts with the VCO running very slowly. The upper pair of signals plotted are f_{REF} and the output of the divide-by-5 counter, $f_{\text{VCO\_div5}}$. The

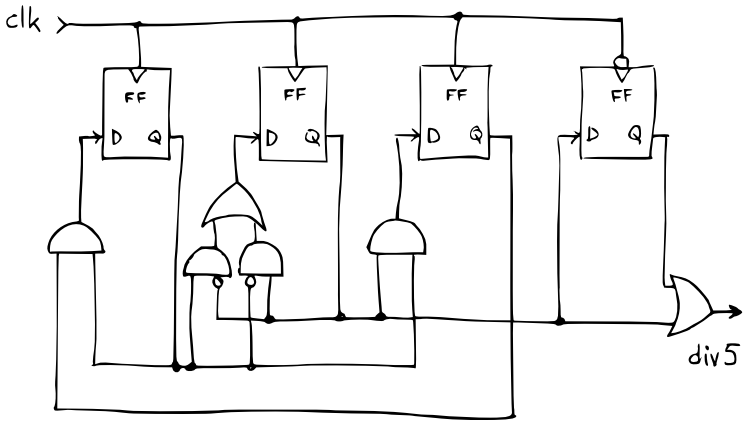

Figure 5.22
Divide-by-5 counter circuit that produces a uniform duty cycle output, div5.

second pair of signals plotted are the output pulses from the Phase Frequency Detector, and the output of the Low Pass Filter, that is the control voltage input of the VCO. The third pair of signals are f_{REF} and f_{PLL}, where I have zoomed-in towards the end of the simulation time to show that the f_{PLL} signal is running at a 5 times higher frequency than f_{REF}. Over the course of the first part of the simulation, the VCO frequency speeds up in response to the rising control voltage from the filter output, until $f_{\mathrm{VCO\_div5}}$ reaches about the same frequency as f_{REF}. During the remainder of the simulation this VCO control voltage stays almost constant, although some small variations are visible. Similarly, the results of a second simulation in which the VCO starts off running at almost 3 GHz are illustrated in Figure 5.24. The PLL responds slightly quicker in this case and the small variations in the VCO control voltage are more obvious in this simulation. The consequence of the varying VCO control voltage is that the frequency of f_{PLL} varies slightly even though it is locked to f_{REF}. These simulations show that this particular PLL design is robust and can accommodate starting at either extreme of the VCO frequency range.

However, this simple PLL circuit also illustrates many of the implementation shortcomings that have caused engineers to devise a whole range of circuit improvements to produce a higher quality f_{REF} output to reduce variations in frequency.

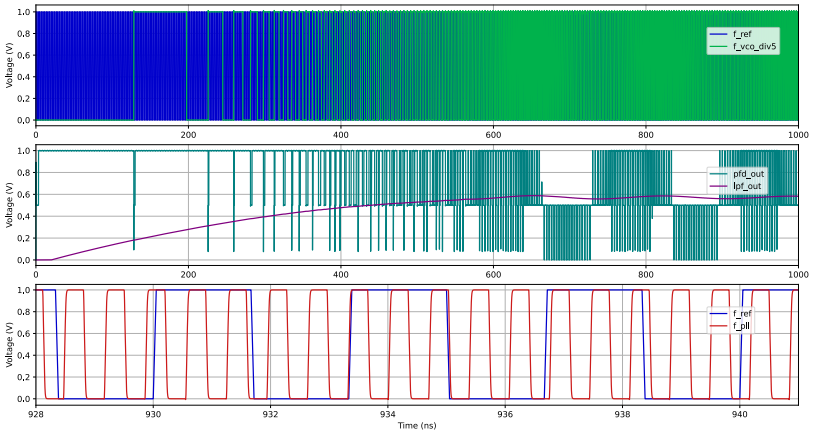

Figure 5.23
Simple x5 PLL simulation with the VCO starting at a low frequency.

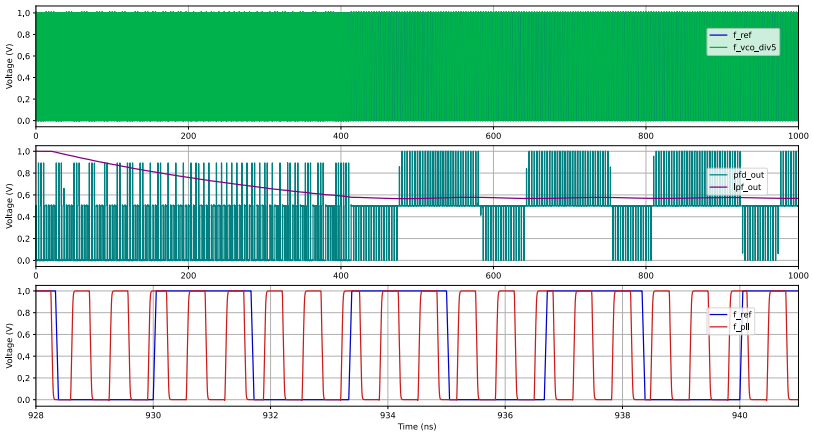

Figure 5.24
Simple x5 PLL simulation with the VCO starting at a high frequency.

5.5 Analog PLL Circuit Drawbacks

Until now, we have mostly focused on the advantages of PLLs for on-chip clock generation, but have commented briefly on some of their drawbacks. Real circuit implementations suffer from several issues that are progressively

becoming more of a problem. The main issues are their size and power consumption. Analog circuits tend to have large transistors, large capacitors, and are always drawing current. In comparison, digital circuits tend to draw significantly less current. As chip frequencies increase, greater demands are placed on the stability and skew of the PLLs. These demands can be met by using higher quality low-pass filters, but this requires large capacitors, which contributes significantly to their physical size – chip real estate that we would prefer to devote to compute logic.

As on-chip clock frequencies have risen, radiation from the clock has become an issue because it generates undesirable Electro-Magnetic Interference, EMI, that can disrupt the operation of adjacent circuits and nearby devices. For this reason, manufactured systems must meet strict EMI regulations. It is much harder to contain the EMI within the system with higher frequencies because it can leak out of small holes and sneak onto cables going in and out of the system. Thus, reducing the amount of signal radiation at the source is required, high frequency clock signals being the worst offenders. The clock output stability from a PLL concentrates most of the radiated energy at the clock frequency, which makes the EMI very much worse. To combat this issue, in some chip designs, small amounts of frequency variation are intentionally introduced to spread out the radiation spectrum, and thus reduce the EMI. Of course, this partially negates the value of using a PLL to achieve a low jitter clock and requires greater attention to track and design for the timing relationships at clock domain boundaries.

5.6 Digital PLLs

Digital PLL circuits have been developed to address many of the issues with the conventional analog PLL designs. Although Digital PLLs do not yet achieve such high quality output frequency signals, they are catching up rapidly. Further attractions are that they typically are less than half the physical size and consume significantly less power than equivalent analog PLLs. These attributes make Digital PLLs a good choice in applications that place lower demands on the clock's quality, especially in chips that require a large number of phase-related clocks.

The components of a digital PLL are shown in Figure 5.25, where essentially each analog component of a standard PLL has been replaced by a digital counterpart. In this figure, the phase detector is replaced with a Digital Phase Detector, DPD. The low pass filter is replaced by a Digital Low Pass Filter, DLPF, and the voltage controlled oscillator is replaced by a Digital Controlled

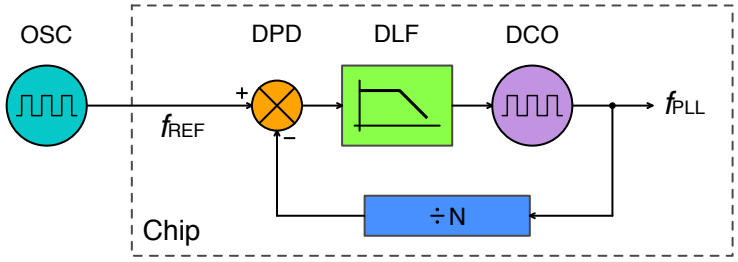

Figure 5.25
Digital PLL components.

Oscillator, DCO. The digital filter component contributes the greatest area and power savings, avoiding the large capacitors of standard PLLs.

Even in a digital PLL, to achieve greater performance, some of the components still have partial analog behavior, such as the DPD and the DCO. Furthermore, some components have analog behavior that might not be obvious to digital circuit designers. An example of this is when a Time to Digital Converter, TDC, is employed in the phase detector component. As described earlier in chapter [reference TDC chapter], TDCs have internal analog behavior because they sample input signals independent of the sampling clock. This will inevitably result in occasionally sampling an input signal just at the moment that it is changing, potentially causing the sampling latch to linger in a metastable state, introducing a random additional delay in generating its output. However, propagation of a metastable value from the TDC must first pass through the digital low pass filter, which has multiple clock cycles of delay, and provides plenty of time for any metastable voltage to resolve to a valid logic level. Furthermore, some DCO circuit designs can tolerate both digital and analog control input signals and are well behaved when the voltage of one or more of its digital control inputs is metastable and lies between logic 0 and logic 1 values. The DCO circuit design presented shortly in Figure 5.36, has such tolerance to metastable control inputs.

For demonstration purposes we present a digital PLL circuit using straightforward implementations of each digital component and show with simulation results how the digital PLL can generate the output clock, f_{PLL}.

A Digital Phase Detector, DPD, can be implemented using a Time to Digital Converter, TDC. There are multiple ways of implementing a TDC, such as with a tapped delay line or with an up/down counter. For our example circuit, we use a TDC using a tapped delay line, as illustrated in Figure 5.26.

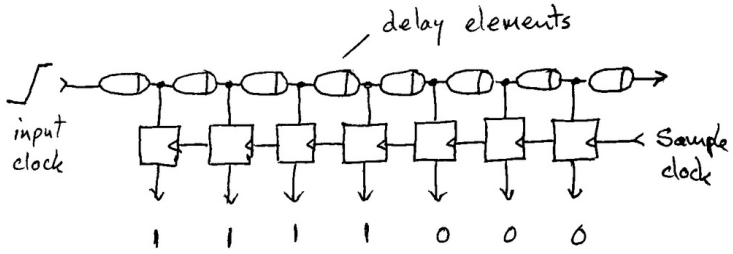

Figure 5.26
Time to Digital Converter, TDC, sampled delay line.

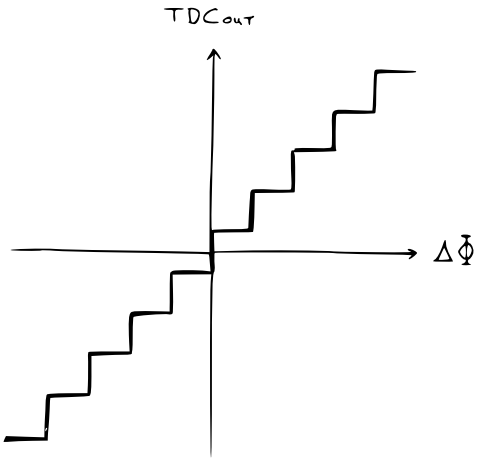

Figure 5.27
TDC thermometer code output, TDC_{OUT}, vs input phase difference, $\Delta\Phi$.

The signal phase difference is measured by sampling how far along the delay line the input clock signal has traveled when it is sampled by the reference sample clock. The TDC output, TDC_{OUT}, is an n-bit binary thermometer code word whose value is proportional to the input phase difference, $\Delta\Phi$, as illustrated in Figure 5.27. The stepped response is the result of the digital quantization of sampling the analog phase difference. Smaller step sizes along the $\Delta\Phi$ axis correspond to finer resolution of the input phase difference, requiring smaller time differences per stage of the TDC.

A very basic delay line consists of a chain of inverters, tapped off after each inverter pair, as shown in Figure 5.28. This circuit design has two main

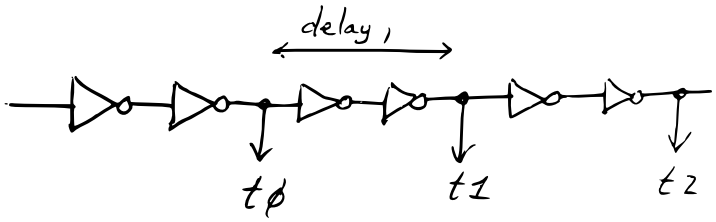

Figure 5.28
Basic tapped delay line using pairs of inverters.

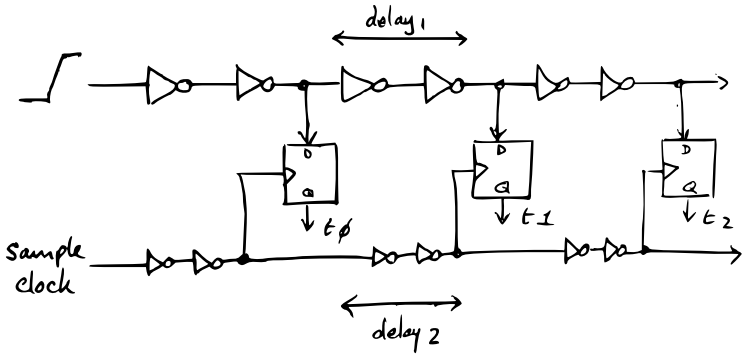

Figure 5.29
Vernier tapped delay line.

issues: the time resolution is an interval of 2 inverter delays per tap, which is quite coarse, and the variation in the delay interval per tap is very sensitive to variations in the transistor sizes. A Vernier tapped delay line reduces the delay interval per stage by adding a second tapped delay line for the sampling clock, as illustrated in Figure 5.29. This delay line's time resolution is greatly improved and corresponds to the difference $delay_1 - delay_2$, where the sample clock delay line is made to have less delay per stage than the upper delay line. However, this circuit still suffers from sensitivity to variations in the transistor sizes.

A third style of tapped delay line that achieves both improved time resolution and less sensitivity to transistor size variations, uses a feed-forward lattice arrangement of tightly cross-coupled inverters, as illustrated in Figure 5.30. Simulation waveforms for a 16-stage feed-forward tapped delay line are shown in Figure 5.31, where the rising input clock transition has traveled roughly halfway along the delay line when sampled. The sampled values are captured by

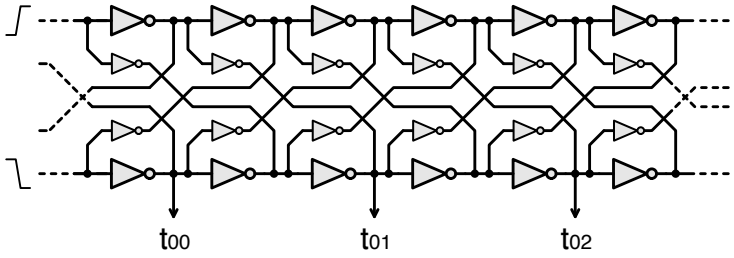

Figure 5.30
Feed-forward tapped delay line.

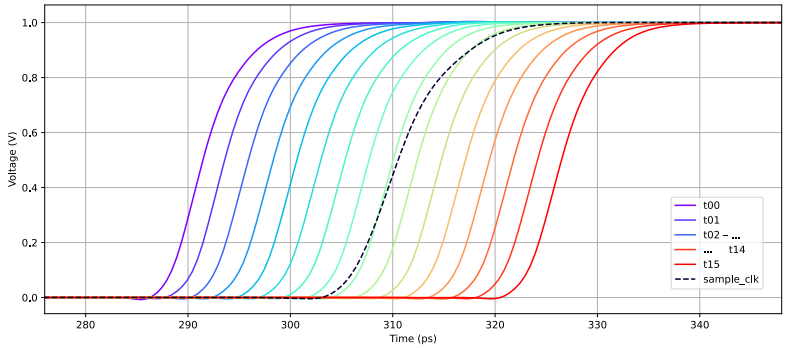

Figure 5.31
Waveforms of a 16-stage tapped delay line Time to Digital Converter, TDC.

a register of flip-flops producing a digital thermometer code value representing the phase difference between the two clocks. As described in earlier chapters, the sampling process of an asynchronous signal can lead to metastable values when the input signal is transitioning at the instant that it is being sampled. Care needs to be taken that any such metastable values do not adversely affect the digital circuits' operation. Fortunately, the low-pass filter that follows the sampling circuit takes multiple clock cycles, which significantly increases the probability that any metastable values have been resolved to valid logic values before they are used.

For reasons explained shortly when discussing the digital low pass filter, there should be only a single rising transition, and no falling transition, in the tapped delay line per sample clock period. One way to ensure this is to restrict significantly the range of frequency output of the DCO, which is common practice, [3]. Alternatively, an input metering circuit can be added to achieve

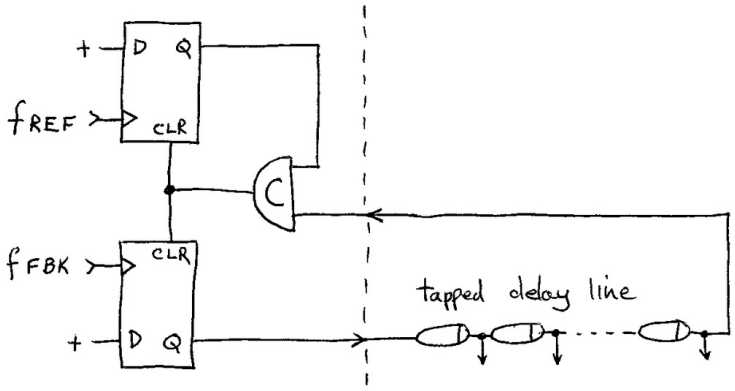

Figure 5.32
Digital PFD circuit that ensures at most one transition in the delay line per sample.

a	b	x_n
0	0	0
0	1	x_{n-1}
1	0	x_{n-1}
1	1	1

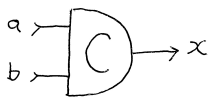

Figure 5.33
Truth table and symbol of a Muller-C element.

this objective of ensuring that at most a single transition is present in the delay line, as shown in Figure 5.32. This metering circuit uses a pair of flip-flops, similar to those used in the analog PFD circuit, to deliver the transition signals into the delay line. The circuit waits for transitions to have reached the output of the delay line before permitting a new transition to enter the delay line. This metering circuit waits for both rising and falling transitions and uses a Muller-C element for this function. The Muller-C element symbol is like that of an AND gate, but with the addition of a letter "C", because it performs the AND function of transition inputs: it waits until both inputs have transitioned, either both HI or both LO, before producing a corresponding output transition, as described by its Truth Table in Figure 5.33. A Muller-C element is also known as a "wait-on" or "rendezvous" gate, because it waits for the last input to arrive before generating its output signal. The implementation of a Muller-C element that responds to transition inputs contains one bit of internal storage to maintain its current output value until both of its inputs have changed value.

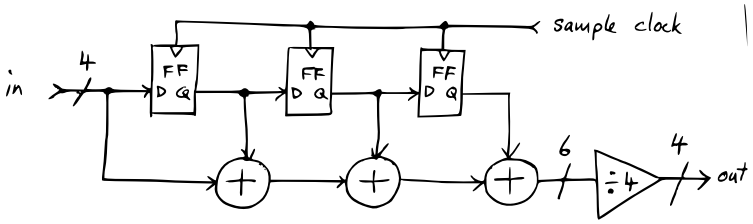

Figure 5.34
4-stage digital FIR Low Pass Filter.

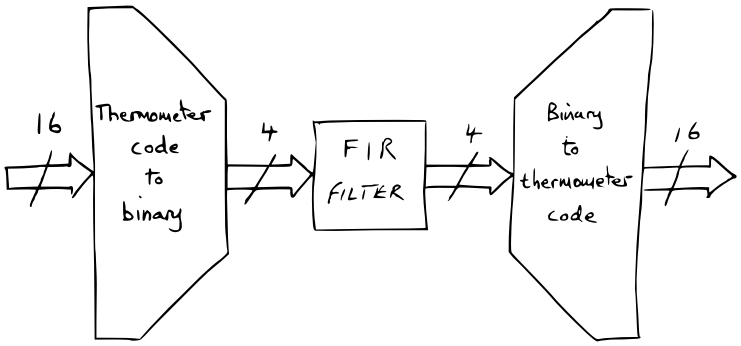

Figure 5.35
Convert from 16-bit thermometer code to 4-bit binary, use a 4-bit FIR Low Pass Filter, convert back from 4-binary to 16-bit thermometer code.

The digital low pass filter component is often less than half the chip area of its analog counterpart. A simple filter circuit that performs integration of consecutive samples is a digital finite impulse response (FIR) filter. A 4-stage FIR filter is illustrated in Figure 5.34.

A binary FIR filter is a very compact design, and so there is an advantage to convert the 16-bit thermometer code output of the TDC into 4-bit binary. Following the FIR filter, the 4-bit binary is converted back to a 16-bit thermometer code because this provides suitable control signals for the digital controlled oscillator, as shown in Figure 5.35. These converters are quite small and shallow combinational logic circuits.

An implementation of a Digital Controlled Oscillator, DCO, very similar to the current-starved inverter ring oscillator, is shown in Figure 5.36. With digital control signals, rather than analog ones, additional small transistors, that are always conducting, are required per stage to prevent the oscillator from

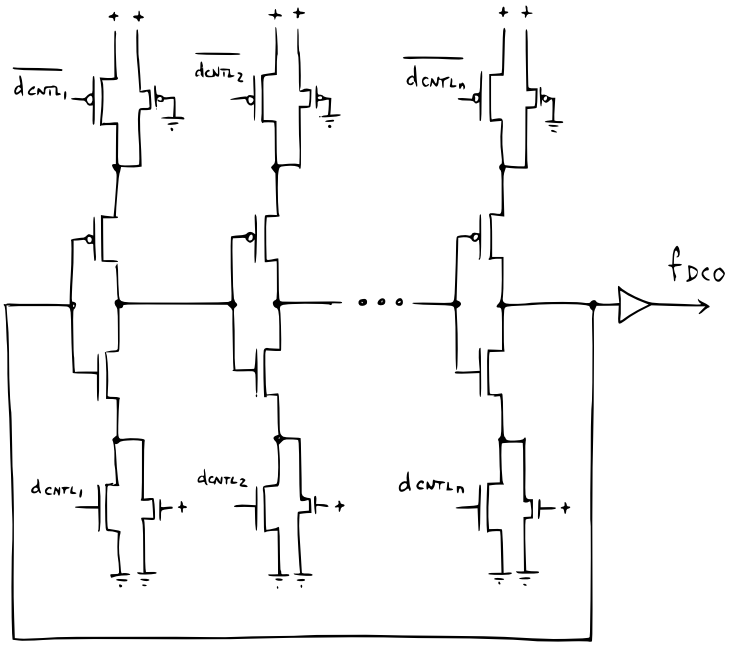

Figure 5.36
Current-starved inverter Digital Controlled Oscillator, DCO, circuit.

stopping. Each inverter stage is controlled by one bit of the thermometer code. The range of operation of a 16-bit current-starved inverter DCO is presented in Figure 5.37. As these simulation results show, the oscillation period varies very linearly with changes in the thermometer code value.

The divide-by-5 circuit in the feedback loop remains identical to the circuit used in the analog PLL, shown in Figure 5.22.

Spice simulation results of a complete digital PLL circuit using all these digital components is presented in Figure 5.38.

5.7 Conclusion

PLLs have become the standard way of generating a high quality on-chip clock. They use a stable, low frequency, low jitter, external clock reference, which is usually a quartz crystal oscillator. PLLs provide high frequency very stable clocks with a jitter that is similar to that of their external reference clock. They are programmable by selecting different internal divider values. An example situation, where it is essential to switch between different frequencies, occurs

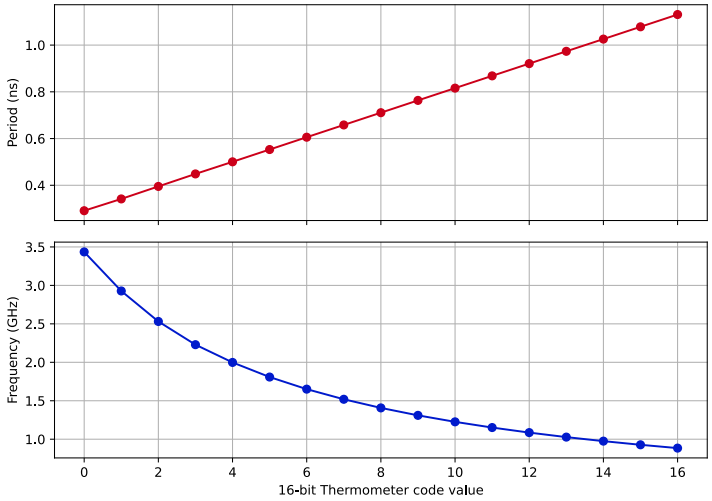

Figure 5.37
DCO range with thermometer code value.

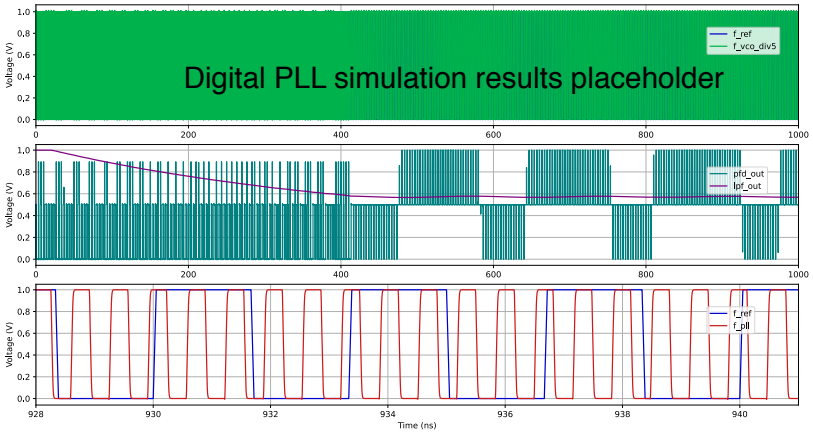

Figure 5.38
Digital PLL simulation waveforms.

when transitioning to operate circuits at a different voltage, such as operating in low power mode. Switching a PLL from one frequency to another usually

takes many tens to even hundreds of clock cycles. Applications that require fast switching between different frequencies often deploy a pair of PLLs: while the first PLL output is selected, the second PLL is re-programmed many clock cycles in advance of when a frequency transition must occur.

Analog PLLs provide the highest quality clocks, but are very large and draw significant current. The introduction of Digital PLLs offers smaller, lower power clock generation solutions, but have higher jitter. Digital PLLs are becoming more common for chips with many clock domains that require multiple PLLs, because of their smaller size and lower power.

Bibliography

[1] Banerjee, Dean. 2017. *Pll performance, simulation, and design*, 5th edn. Texas Instruments. https://www.ti.com.cn/cn/lit/ml/snaa106c/snaa106c.pdf.

[2] Best, Roland E. 2003. *Phase-locked loops: Design, simulation, and applications*, 5th edn. New York, USA: McGraw-Hill.

[3] Perrott, Michael H. 2009. Tutorial on Digital Phase-Locked Loops. IEEE Custom Integrated Circuits Conference, CICC 2009.