A Cascaded Dual-Fractional-N Digital Phase-Locked Loop

Warrick Lo

01 May 2026

Contents

List of Figures

List of Tables

1. Introduction

Phase-locked loops (PLLs) are vital components for digital and analog electronics. Their uses include clock generation for digital circuits, frequency synthesis for wireless communications and RFICs, clock recovery for communications, and demodulation of frequency and phase modulated signals.

PLLs are essential for wireless communications. Due to various reasons, however, spurious emissions in the output remain a large challenge for high-frequency clock synthesis. In this report, we will explore a proposed architecture that cascades two fractional-*N* PLLs together to improve spur performance.

2. Basic PLL Design

The basic idea behind the PLL is quite simple. It first compares the reference clock and the feedback clock signals. Depending on which arrives first, it will pass an output to a filter, which will then control an oscillator. The oscillator will then respond by changing the output frequency, which is then fed back to the phase detector. By adding a divider in the feedback path, the phase detector will see a lower frequency. Thus, the oscillator will generate a higher frequency to compensate. This is how high frequency clocks are generated from a slow reference clock.

In the classic PLL, all of the components are analog and work in continuous-time. This is the analog PLL (APLL) and uses an analog phase detector, an RC network as the loop filter, and a voltage-controlled oscillator (VCO). Generally these are low noise and good for RF applications. However, they are hard to integrate into modern CMOS processes and are also sensitive to process variations.

The digital PLL (DPLL) replaces some of the analog blocks with digital equivalents. Usually, the phase detector will be a digitally implemented phase/frequency detector, along with a digital loop filter. The VCO may still remain in a DPLL design. These changes make it easier to integrate into modern CMOS ICs and also makes the DPLL more tolerant to process variations. However,

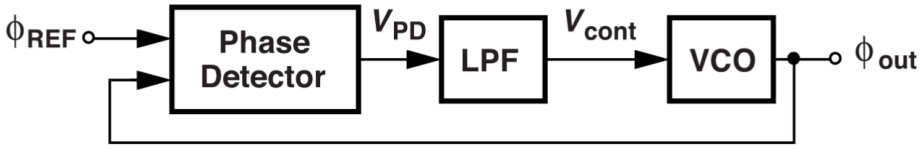


Figure 2.1. System-level overview of a PLL.

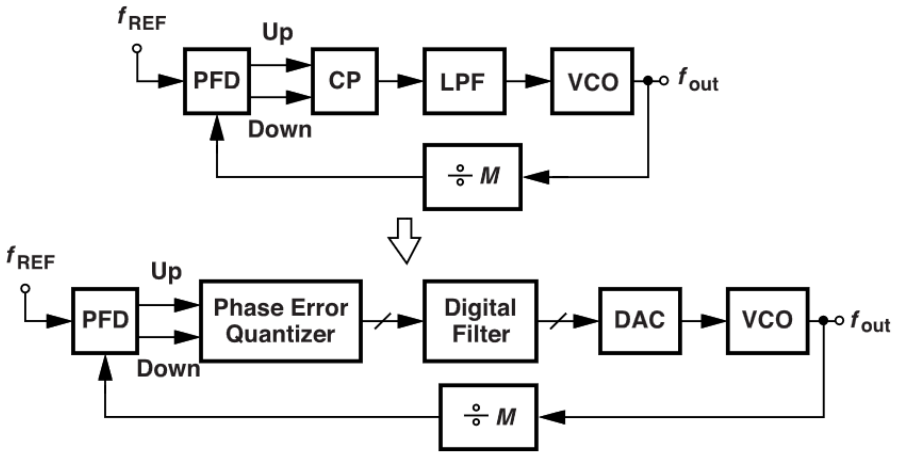


Figure 2.2. Transformation of an analog PLL to a digital PLL.

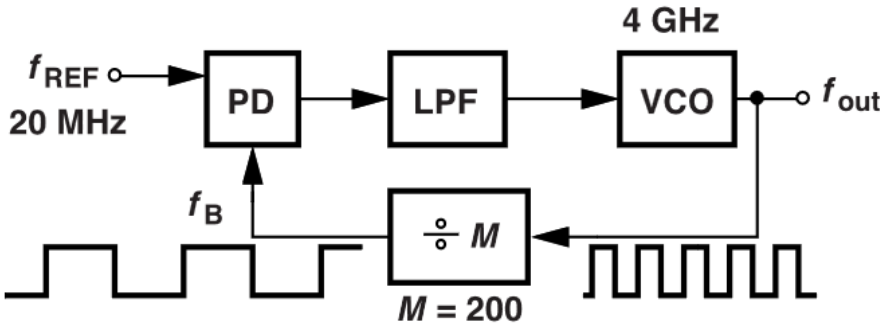


Figure 2.3. A frequency-multiplying PLL.

because of the discrete-time nature, quantisation noise appears in the output.

The all-digital PLL (ADPLL) replaces all of the analog blocks with digital equivalents. In most cases, the phase detector is replaced with a time-to-digital converter (TDC). The VCO is also replaced with a digitally-controlled oscillator (DCO). The ADPLL is very common for system-on-chip (SoC) designs and are very scalable with CMOS processes. However, like the DPLL, there is quantisation noise. Furthermore, they have limits on the resolution.

2.1. Integer- and Fractional-N PLLs

In the integer-*N* PLL, the output frequency is an integer multiple of the reference. As seen in figure 2.3, this is accomplished by dividing feedback frequency by *M*. This, compared to the fractional-*N* PLL, leads to a simpler architecture. The spurious tones are also lower in the output. However, due to its nature, it will also have very coarse output frequency control. This makes it not suitable for high-frequency applications.

Fractional-*N* PLLs, on the other hand, allows the output frequency to be a non-integer multiple. As an easy example, *N* may be 2.5 or 2.36 in the fractional-*N* PLL, whereas *N* will limited to 2 or 3 for the integer-*N* PLL. The use of non-integer values of *N* allow for finer control in the output frequency, making it highly useful for RF applications.

3. Digital PLL Architecture

DPLLs are able to cancel noise from the ΔΣ modulator (DSM) more effectively, compared to analog designs, according to [1]. ΔΣ modulation is used in fractional-N PLL designs to obtain a fractional division ratio in the feedback loop. We will cover DSMs later on. However, the quantisation also leads to an increase in phase noise and jitter.

3.1. Time-to-Digital Converters

The TDC is the digital equivalent of the phase detector. It's role is the same: to convert the phase difference between the reference and feedback signals to a digital value.

To generate a digital output code, we want to run the reference signal through a delay line. These delays should be all equally spaced. The rising edge of each delayed signal can be used as a reference to compare the feedback signal with. The concept is similar to using equally-spaced reference voltages in an analog-to-digital converter.

Using a flip-flop to compare the delayed references and feedback signals, we can obtain a thermometer code output.

Because the analog phase error is now represented digitally, the TDC introduces quantisation noise. The power of the noise is roughly proportional to the square of the TDC resolution (the unit delay per stage in seconds), according to [1]. More specifically, the noise spectral density

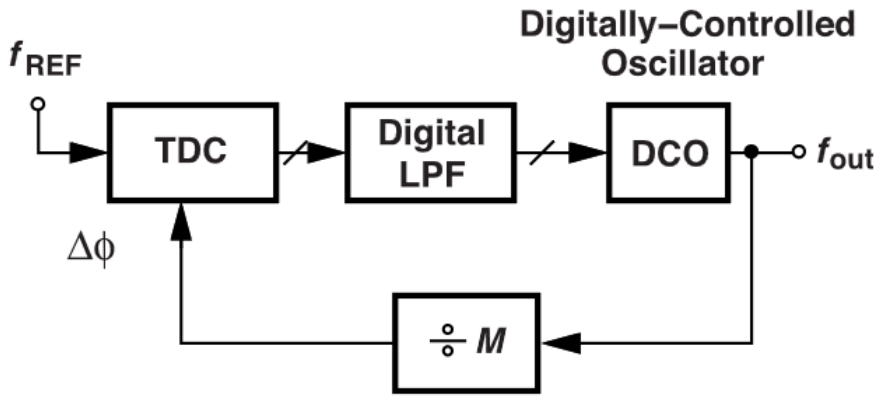


Figure 3.1. System-level overview of a digital PLL.

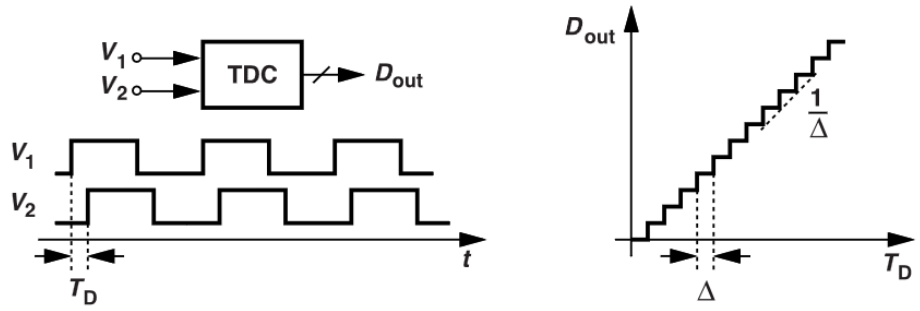


Figure 3.2. Input waveforms and the input-output characteristic of a TDC.

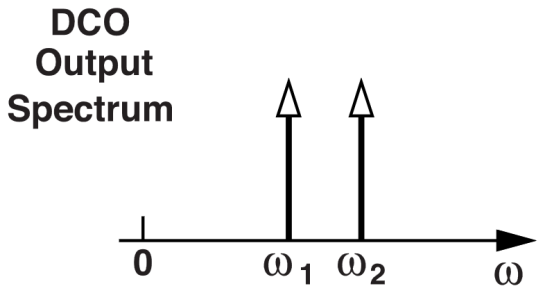


Figure 3.3. Output frequency spectrum of a DCO.

S_{ϕ_e} can be given by

$$S_{\phi_e} = \frac{4\pi^2\Delta^2}{12\ T_{\text{ref}}}, \tag{3.1}$$

where Δ is the LSB size (TDC resolution).

The quantisation noise will increase the phase jitter of the output clock. In many cases, the error is periodic and not completely random. This will also cause spurious tones to appear in the output spectrum. We will cover more about spurs in a later section.

3.2. Digitally-Controlled Oscillators

A DCO behaves like a DAC with its output connected to a VCO. Therefore, a DCO will only generate discrete frequency values, where the frequency resolution is $\Delta\omega_{\text{LSB}}$.

For fractional-N PLLs, the DCO will need to generate two different frequencies for the two different division ratios, N and $N + 1$. The average of the frequencies will then be the reference frequency multiplied by the fractional-N ratio. However, since this means the oscillator will only be operating at two tones, the output frequency spectrum will consist of two strong spikes and there will be large amounts of phase jitter in the output.

Two different types of DCOs exist: ring DCOs and LC DCOs. Ring DCOs consist of delay elements in a chain. These elements can be inverters or differential stages. By adjusting the current

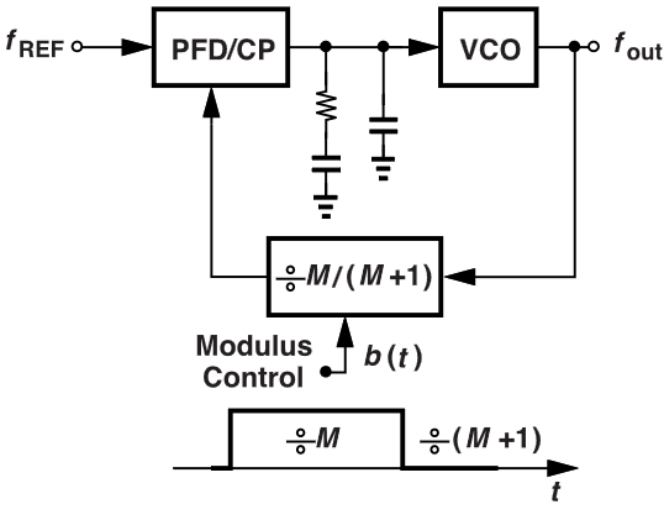


Figure 3.4. Controlling the modulus to obtain a fractional-*N* value.

or capacitance, the delay per stage can be controlled. LC DCOs utilise an LC resonant tank. The oscillation frequency is determined by the capacitance, which can be controlled digitally.

Ring DCOs are much easier for CMOS processes to manufacture. However, they have poor phase noise and is extremely sensitive to noise from the supply or substrate. On the other hand, LC DCOs have excellent phase noise and have lower jitter, while being more costly and complex because of the need for inductors and coarse and fine capacitor banks.

3.3. Delta-Sigma Modulators

Frequencies for RF communication, WiFi, GNSS, et cetera, are all in the gigahertz range. Frequency synthesis from a low-frequency reference crystal such as 50 MHz will almost always require a fractional-*N* modulus (division ratio). As the target output frequency is so much larger than the reference, an integer-*N* modulus will in general not give enough accuracy and precision for RF applications.

By using a non-constant integer for the division ratio, the average value of the modulus can be made to be the fractional-*N* number. If we have a fractional-*N* value of $M + \alpha$, where M is the integer part and α is between 0 and 1, then the division ratio will be M for $1 - \alpha$ time units, and $M + 1$ for α time units. Then, the average value of the modulus will be

$$(1 - \alpha) \cdot M + \alpha \cdot (M + 1) = M + \alpha,$$

giving us an average output frequency of $f_{\text{out}} = (M + \alpha)\, f_{\text{ref}}$.

However, this periodic toggling of the division ratio creates significant spurious tones (spurs), as seen in figure 3.5. Since the divider switches between two different integers, the feedback

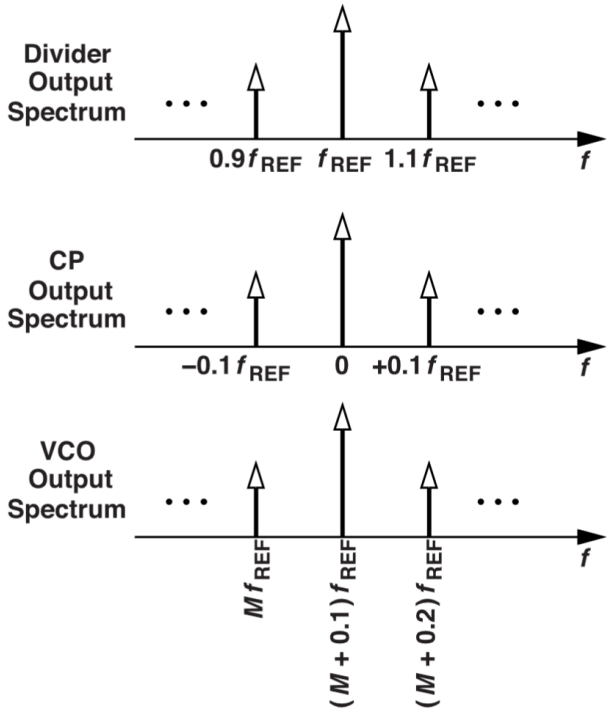


Figure 3.5. Spurious tones at the divider output, phase/frequency detector and charge pump output, and VCO output. Here, the modulus is equal to M for nine reference cycles, and $M+1$ for one reference cycle, giving us $\alpha = 0.1$.

(divider output) is not perfectly periodic, resulting in tones at $(1 \pm k\alpha)f_{\text{ref}}$. These spurs appear at the charge pump output as $\pm k\alpha f_{\text{ref}}$, which then get modulated by the VCO, creating peaks at frequency offsets of $\pm k\alpha$ around the output frequency in the final output.

The fundamental spur frequency is given by

$$f_{\text{spur}} = \begin{cases} \alpha f_{\text{ref}}, & \alpha \leq 0.5 \\ (1-\alpha) f_{\text{ref}}, & \alpha > 0.5, \end{cases} \tag{3.2}$$

and the phase detector and charge pump output spurs can thus be given by $\pm k \cdot f_{\text{spur}}$. The VCO output spurs can similarily be given by $f_{\text{out}} \pm k \cdot f_{\text{spur}}$.

When N is very close to an integer, that is when α is small like 0.01 or close to 1 like 0.99, the spuriours tones will appear at very small offset frequencies. At these frequencies, it is very hard for the loop filter to attenuate properly, which can lead to large spurious emissions. We can see this mathematically with equation 3.2. A small α will lead to a low frequency, which is hard for the loop filter to eliminate. The same applies to α close to 1 because of the piecewise nature of f_{spur}.

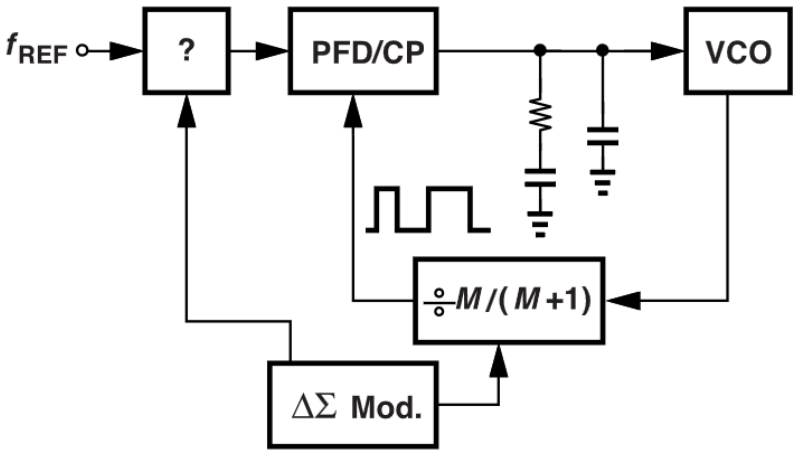


Figure 3.6. Inserting a block in the feedforward path to reduce DSM noise. We can implement the block marked with a question mark with a digital-to-time converter to add or subtract phase in the reference path.

3.4. Digital-to-Time Converters

Digital-to-time converters (DTCs) can be used to cancel the ΔΣ noise. By having the DSM also control the delay of the reference signal, the phase/frequency detector and charge pump output will have reduced noise from the DSM quantisation.

The DTC is essentially a variable-delay line. We input a digital code and it adds phase delay to the output wave.

To visualise the effects of the DTC correction, I used MATLAB to plot waveforms with and without the DTC correction, and also comparing it with an ideal square wave, as seen in figure 3.7.

The frequency spectrum of the waves are also plotted in figure 3.8. Without the DTC correction, we see strong spurs caused by the divider sequence (2, 2, 2, 3). These spurs arise from deterministic timing error introduced by fractional division. The DTC suppresses periodic timing error by applying cycle-by-cycle delay compensation. As a result, spur components are significantly reduced, and the spectrum approaches that of an ideal fractional waveform.

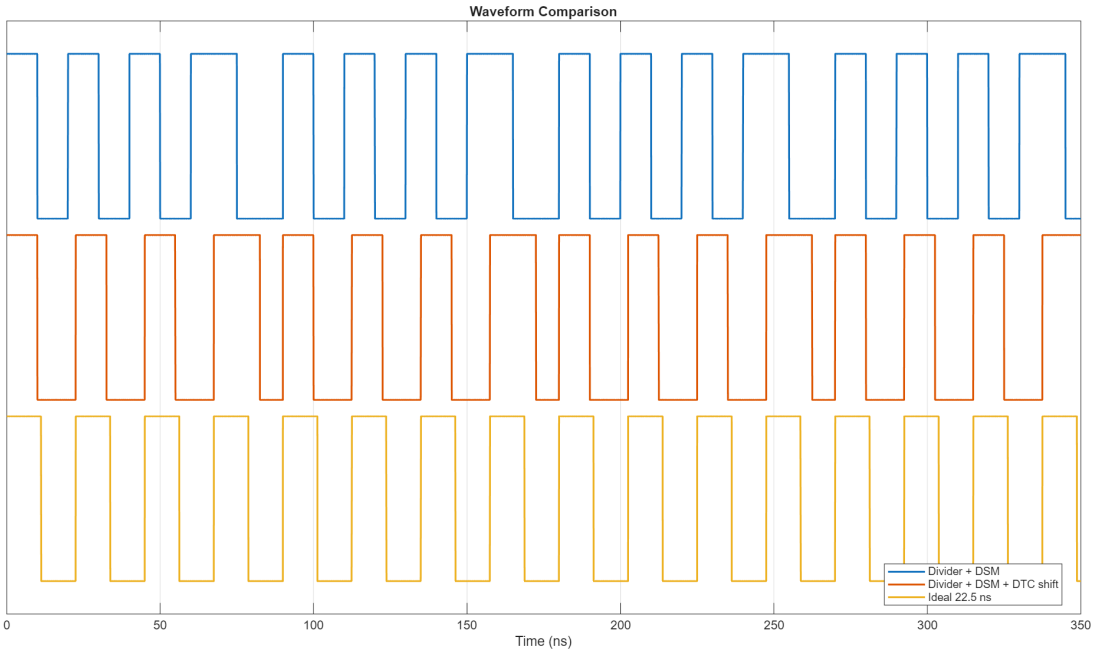
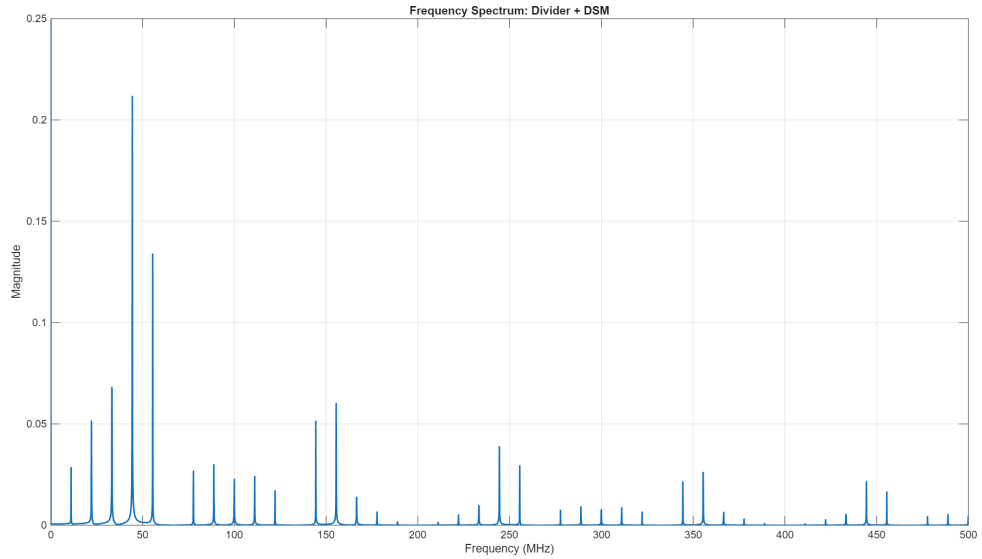
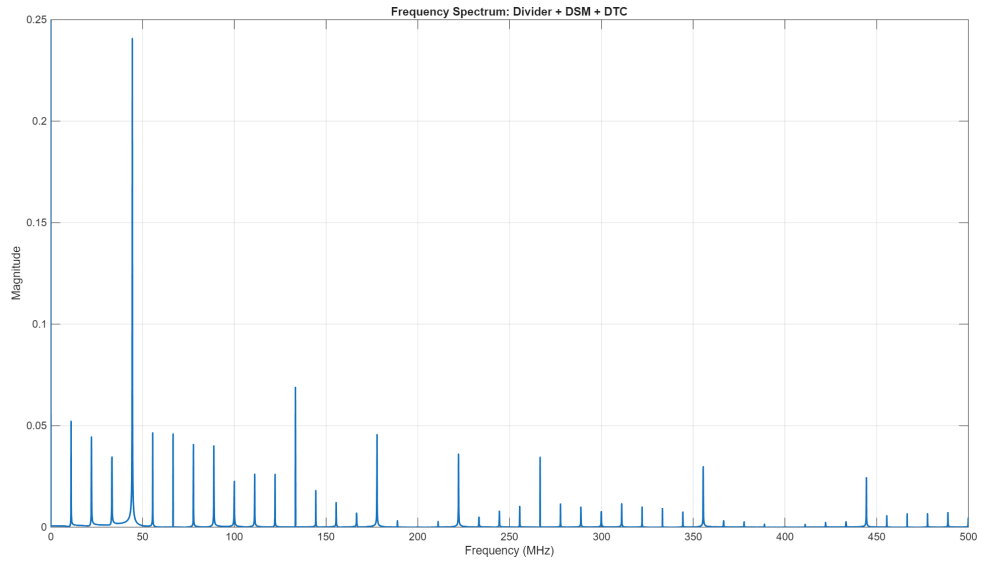


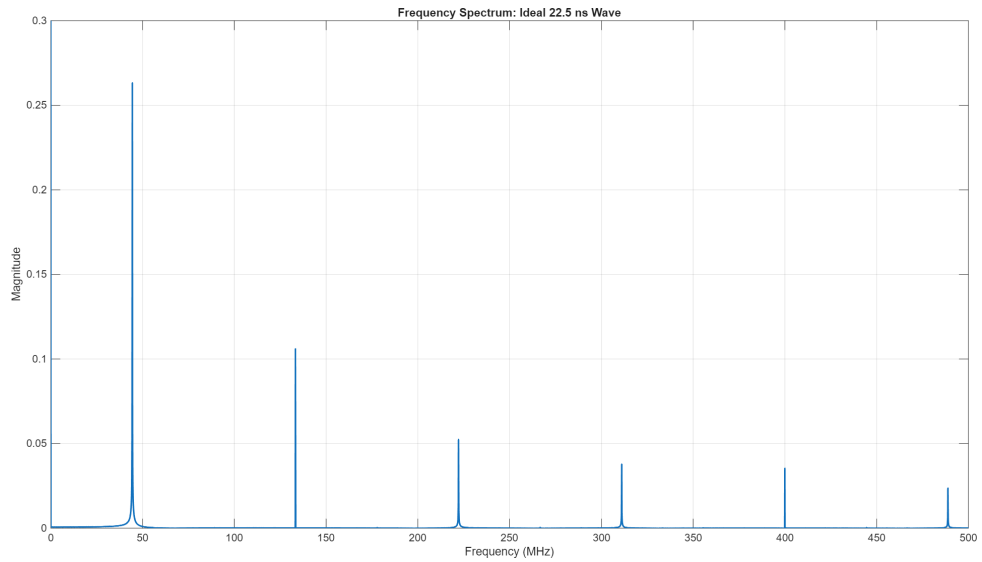
Figure 3.7. Waveform comparison of (blue) divider and DSM output only, (red) divider and DSM output with DTC correction, and (yellow) ideal waveform with a constant 22.5 ns period.



(a)



(b)



(c)

Figure 3.8. Frequency spectrum of (a) divider and DSM output only, (b) divider and DSM output with DTC correction, and (c) ideal waveform with a constant 22.5 ns period.

4. Spurious Emissions in Fractional-N PLLs

As stated earlier, fractional-*N* PLLs are very often used for high-frequency frequency synthesis in RF communication applications. A requirement for communication systems is that the carrier clock should be a single pure tone. However, fractional spurs caused by the DSM results in a degredation of frequency accuracy and increases the error vector magnitude (EVM).

4.1. Spurious Tones

When synthesising frequencies that are near integer multiples of the reference, spurious tones will appear with low offset frequencies. This means that the spurs are very close to the target output frequency, close enough that the loop filter is unable to filter the tones out [1].

Xu et al. proposes in [2] to cascade two fractional-*N* PLLs together to obtain near-integer values of *N* without large spurious content in the output.

For example, if we want to generate a 7.50005 GHz output from a 50 MHz reference, like in figure 4.2, we will need an overall division ratio of $N = 150.001$. Using one single fractional-*N* PLL to obtain this will give us a fractional modulus $N_{\text{frac}} = 0.001$, which will give us an offset frequency f_{spur} of 50 kHz. By using a fractional-*N* stage cascaded with an integer-*N* stage, the fractional part of N_1 is $N_{1,\,\text{frac}} = 0.0001$, giving us a 5 kHz spur in the first stage.

If we follow the proposed dual-fractional-*N* architecture in [2], we can set $N_1 = 15.2$ and $N_2 = 9.868487$, for example. Thus, $N_{1,\,\text{frac}} = 0.2$ and $N_{2,\,\text{frac}} = 0.868487$, giving us fundamental spur frequencies of $N_{1,\,\text{spur}} = 10$ MHz and $N_{2,\,\text{frac}} = 660$ MHz according to equation 3.2 (the intermediate frequency f_{IF} is $15.2 \cdot 50$ MHz). These offsets are large enough to be in the second-stage loop filter's stop-band, which will greatly attenuate these spurious tones.

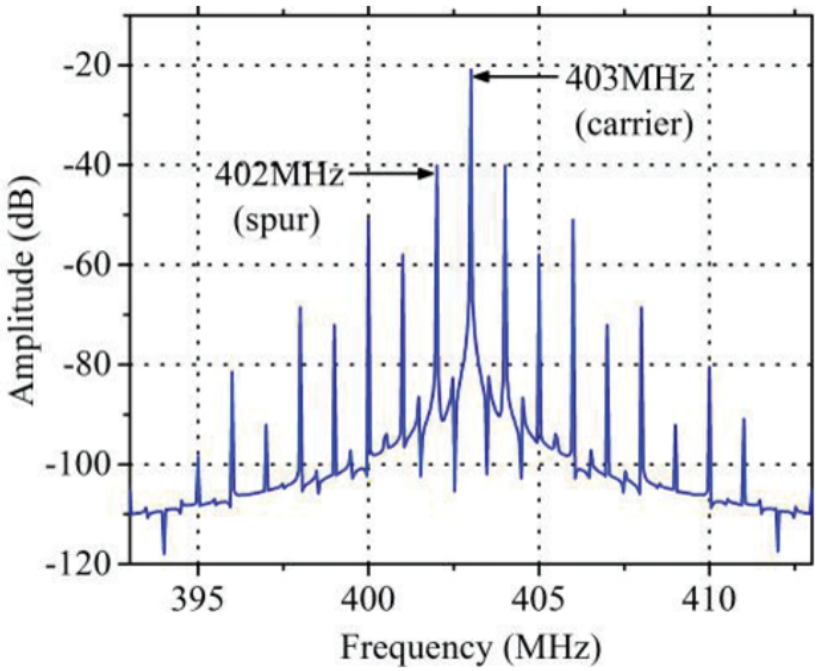


Figure 4.1. An example of spurious tones on the frequency spectrum. Here, the output frequency is 403 MHz and the offset frequency for the spurs is 1 MHz. We see spurs appear at $\pm k$ MHz from the output, at 401, 402, 404, 405 MHz, et cetera.

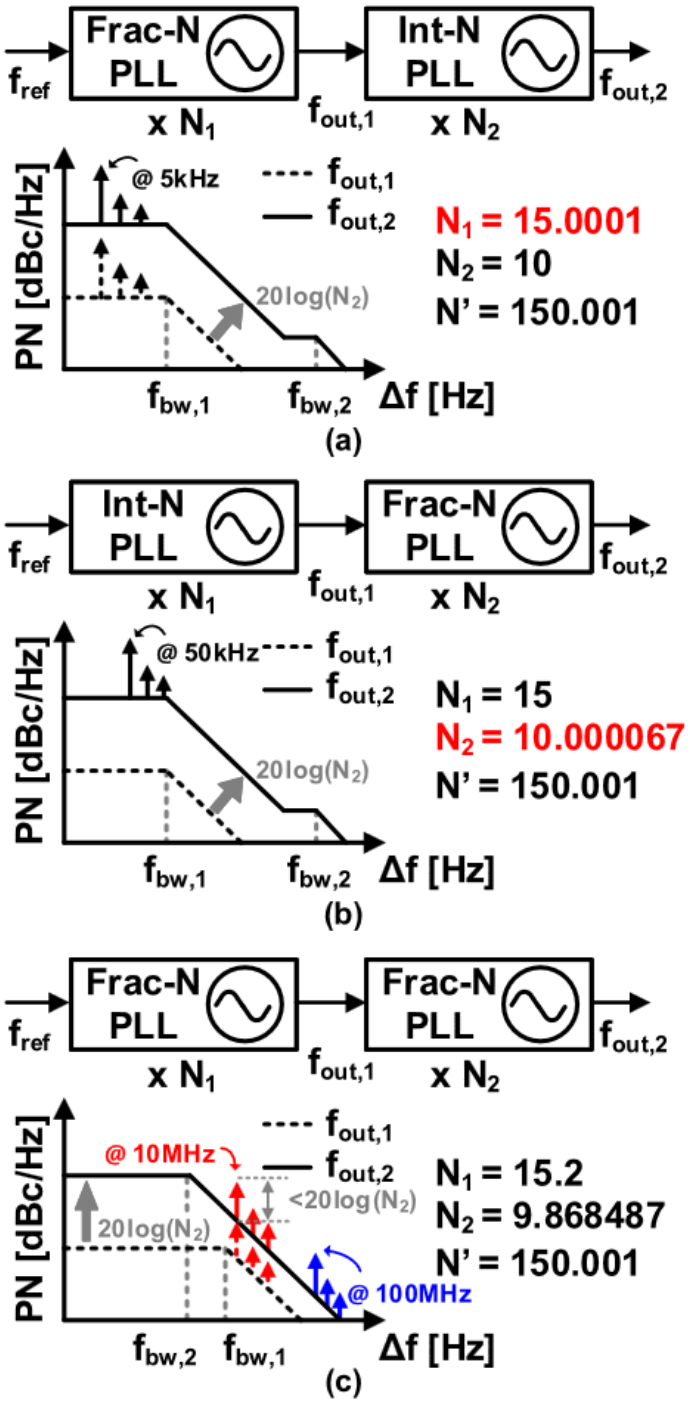


Figure 4.2. Conceptual comparison of the phase noise spectra of (a) cascaded fractional-*N*-integer-*N* PLL, (b) cascaded integer-*N*-fractional-*N* PLL, and (c) cascaded dual-fractional-*N* PLL proposed by [2]. We see that by properly choosing N_1 and N_2 in (c), we can push the spurious emissions to higher frequencies and eliminate them more effectively in the final output.

5. Implementation of the First-Stage Fractional-N Multiplying Delay-Locked Loop

The first stage is a fractional-*N* multiplying delay-locked loop (MDLL). This stage takes in a 50 MHz reference signal and generates a sub-10 GHz intermediate frequency (IF) f_{IF}. The frequency is usually chosen to be in the 1.2-to-1.5 GHz range, which is then fed into the second stage to generate the final desired frequency. The IF range is specifically chosen to avoid the long DTC delay ranges required at lower frequencies, which would otherwise degrade the noise floor, while allowing enough frequency separation for the second stage to properly filter out this stage's quantisation noise and spurs [2].

5.1. MDLL Architecture

The first stage employs an MDLL specifically to provide a higher loop bandwidth f_{bw} over a traditional PLL. This is needed to aggressively supress the phase noise of the ring oscillator [2].

An MDLL uses a multiplexer to periodically inject a clean edge from the reference signal directly into the ring of the oscillator. This is unlike a standard PLL which would gradually adjust the oscillator's phase usign a slow-reacting control voltage. This process is called "edge injection". This effectively resets the accumulated jitter of the RO at every injection of the reference cycle. Because the phase is reset every cycle, the loop acts as if it has a very wide bandwidth for supressing the RO's phase noise.

A wider loop bandwidth will allow the loop to more aggressively supress the phase noise of the internal RO. In standard PLLs, this is around one-tenth of the reference frequency to maintain stability, according to [1].

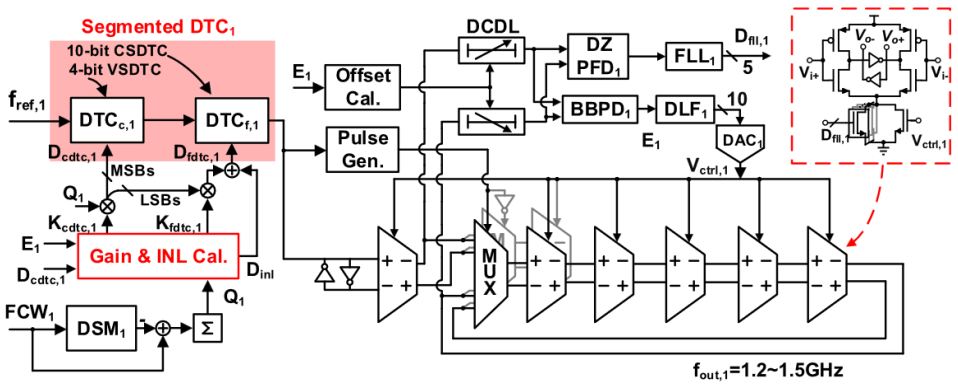


Figure 5.1. Detailed schematic of the first-stage multiplying delay-locked loop.

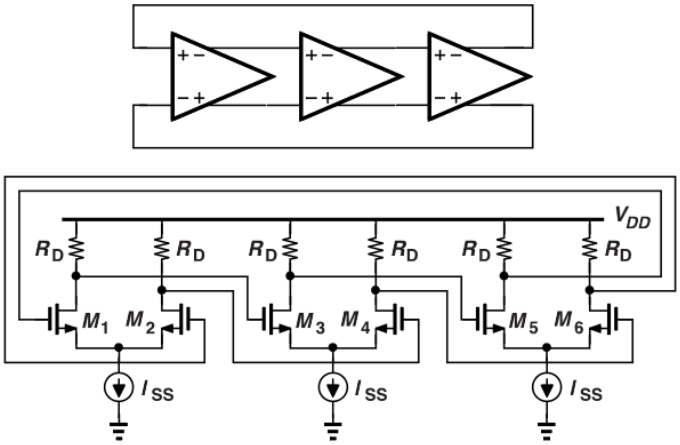


Figure 5.2. A three-stage differential ring oscillator.

5.2. The Pseudo-Differential Ring Oscillator

At 1.2-1.5 GHz, an LC oscillator will be too large, even when it offers far superior phase noise. By utilising an MDLL architecture along with a ring DCO, we can achieve the needed performance for high-frequency clock synthesis while staying compact, instead of using a large LC oscillator.

The pseudo-differential RO consists of differential delay cells, as seen in figures 5.2 and 5.3. This topology greatly reduces the supply noise [1] [2] by rejecting the common-mode noise. Standard single-ended ROs, on the other hand, are extremely sensitive to fluctuations in the supply voltage, as discussed earlier in section 3.2.

The pseudo-differential ring DCO is managed by two control loops in parallel: a frequency-locked loop (FLL) and a PLL. The FLL controls a 5-bit current-starver bank to maintain frequency stability. This loop acts as the coarse tuning mechanism for the RO, ensuring that the oscillator is within the correct frequency range. The PLL tracks the precise phase of the RO to align it with the reference, acting as the fine tuning mechanism. The PLL uses a 10-bit DAC to generate a control voltage used to control the delay cells within the RO.

5.3. The Delta-Sigma Modulator

The first-stage DSM controls both the modulus (division ratio) and the stage 1 DTC. A frequency control word chosen and given to the DSM, which determines both the modulus N_1 and DTC delay. A 1st-order DSM is utilised here, chosen specifically to supress the phase noise that would have been induced by a higher-order DSM [2]. A higher-order DSM will typically require a longer DTC delay range to operate, which introduces additional phase noise.

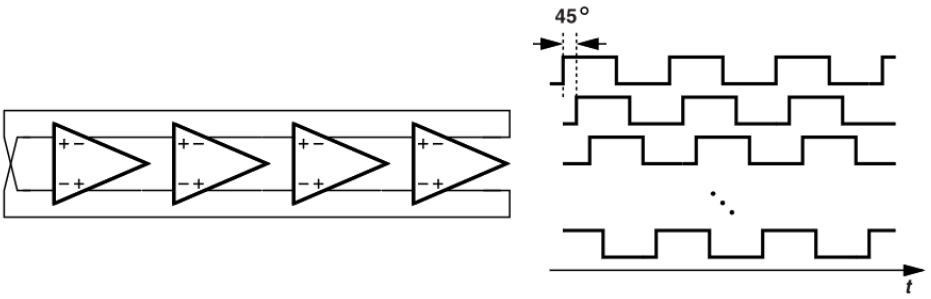


Figure 5.3. A four-stage differential ring oscillator and its waveforms.

5.4. The First-Stage Segmented Digital-to-Time Converter

There are two DTC designs. Constant-slope DTCs (CSDTCs) give high linearity but suffer from noise due to long delay offsets. Meanwhile, variable-slope DTCs (VSDTCs) have low noise but have poor linearity characteristics. Xu et al. proposes using a segmented DTC architecture. The coarse stage is a 4-bit VSDTC with 80 ps resolution and a 1.2 ns range. This ensures that the system can cover the RO period of up to 833 ps. The fine stage is a 10-bit CSDTC with 0.2 ps resolution and a 200 ps range. The range was designed specifically to cover more than 2 LSBs of the coarse VSDTC (160 ps), providing a safety margin for the piecewise linear (PWL) calibration to settle without range overflow. This range also ensures that the delay in the fine stage is below 100 ps to prevent it from dominating the MDLL's noise floor [2].

To correct for the VSDTC's nonlinearities, a PWL integral nonlinearity calibration is done, employing a 15-accumulator lookup table and converging in roughly 80,000 reference cycles, or 1.6 ms [2].

5.5. The First-Stage Frequency-Locked Loop

The FLL acts as the coarse tuning mechanism for the stage 1 RO. It tracks the RO's operating frequency and maintains stability to ensure that the frequency stays within range for the phase-tracking loop to lock.

A dead-zone phase/frequency detector (DZ PFD) is used to sense frequency errors. The output of this is fed to the digital logic, which generates a 5-bit digital bus to control the current-starver bank. The current-starver bank directly controls the delay cells in the ring DCO.

The DZ PFD will give an output proportional to the error. A small phase difference will result in small pulse widths, and vice versa for a large phase difference. In the ideal scenario, this will exhibit a completely lineary behaviour. However, in practice, there is a region of near zero phase difference where the PFD will not produce any output, resulting in small phase errors not being

corrected at all. The accumulation of the small errors also increases jitter.

5.6. The First-Stage Phase-Locked Loop

The PLL provides the fine tuning mechanism to align the oscillator output with the reference signal.

A subsampling bang-bang phase detector (BBPD) is utilised in this loop. The detector gives a 1-bit output to identify timing errors between the reference and feedback signals. The output is then passed to the digital loop filter, which in turn drives the 10-bit DAC to control the bias voltage of the RO.

The BBPD, unlike the DZ PFD, only indicates whether the feedback signal is early or late compared to the reference. Any infinitesimal phase error will trigger a full UP or DOWN output, regardless of the amount of error.

6. Implementation of the Second-Stage Fractional-N Type-II Digital Phase-Locked Loop

The second stage is a fractional-*N* Type-II DPLL. The proportional path controls a 10-bit DAC directly, while the integral path controls a 6-bit capacitor bank. This stage is designed to take in the IF from the first stage and output a frequency in the 6.5-to-8 GHz range.

The proportional and integral control paths are both separated in the analog domain to reduce power consumption. Typically, digital loop filters at high clocking frequencies will consume large amounts of power [2]. Instead of utilising a single large DAC, each path controls the oscillator independently, and the effects are then summed by the LC DCO.

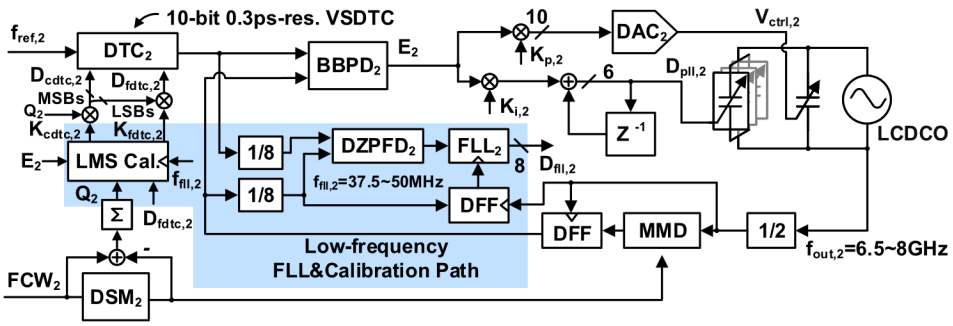


Figure 6.1. Detailed schematic of the second-stage type-II PLL.

6.1. The Proportional and Integral Paths

Both paths take the output of the second-stage BBPD. The proportional path directly controls a 10-bit DAC, which is then used to control the bias voltage of the varactors within the LC tank for fine frequency control. The integral path controls a 6-bit capacitor bank, which handles the coarse frequency control. The effects of the two paths are summed by the capacitors in the LC DCO, rather than a discrete digital adder.

6.2. The Second-Stage Frequency-Locked Loop

Xu et al. proposes to divide the reference and feedback signals by 8 first, before passing it to the FLL. This allows the FLL to operate between 37.5 and 50 MHz. Because of the lower clocking frequency, the FLL will operate with a better power efficiency.

The FLL is used to track the frequency and maintain stability, similar to the first stage. The FLL controls a separate 8-bit capacitor bank, as seen in figure 6.2.

6.3. The Second-Stage Variable-Slope Digital-to-Time Converter

The second-stage implements a VSDTC architecture to cancel the quantisation noise generated by the DSM. Again, the VSDTC has low noise but poor linearity, compared to the constant-slope design. The variable-slope design is chosen for the second-stage due to the high-frequency settlement requirements, giving the needed speed while maintaining a delay range of 300 ps, which is sufficient for covering two DCO periods [2]. This DTC design has a 10-bit resolution, which is further divided into 5-bit coarse and 5-bit fine capacitor banks.

The DTC is calibrated with a least mean square based gain calibration to maintain accuracy. Xu et al. found this allowed the system to reach stability within 40,000 cycles at 37.5 MHz (1.07 ms) [2].

6.4. The Class-B Digitally-Controlled Oscillator

A Class-B DCO is chosen to achieve high spectral purity. These are typically designed to be a low-phase-noise frequency source. The Class-B topology avoids the jitter degradation that usually occurs when an oscillator operates under a narrow loop bandwidth [2]. Since the second-stage has a narrow bandwidth, the Class-B DCO is crucial for maintaining low jitter.

As seen in figure 6.2, the tail inductor is designed as a figure-8 shape and placed inside the main inductor to minimise chip area. The primary inductor is designed without metal overlaps to

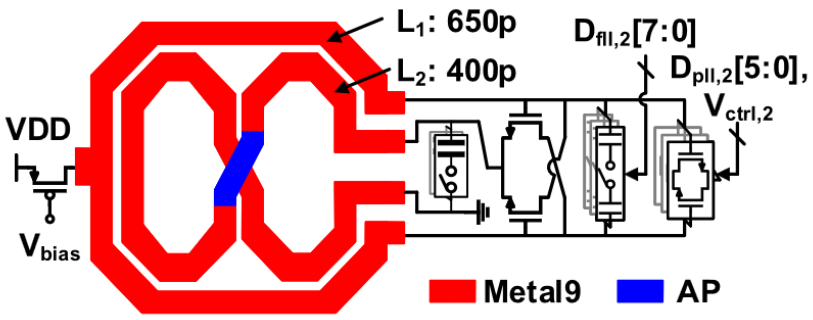


Figure 6.2. Schematic of the DCO with an figure-8-shaped tail inductor, as implemented by Xu et al.

achieve a high quality factor, further improving the phase noise performance.

7. Implementation of the PLL in Verilog

I attempted to model the dual-fractional-*N* PLL described in [2] in SystemVerilog and Verilog-A. The behavioural model for the first-stage FLL was finished, but because of time constraints I wasn't able to finish the rest of the PLL.

For the Verilog source code, go to `https://github.com/warricklo/cascaded-pll`.

7.1. Simulations

The stage 1 FLL was simulated with AMD Vivado 2025.2. The waveforms can be seen in figures 7.1 and 7.2, where $N_1 = 25.368$.

The intermediate frequency eventually stabalises to around 1.26838 GHz, which is very close to the target IF. The stage 1 PLL might need further tuning because of this.

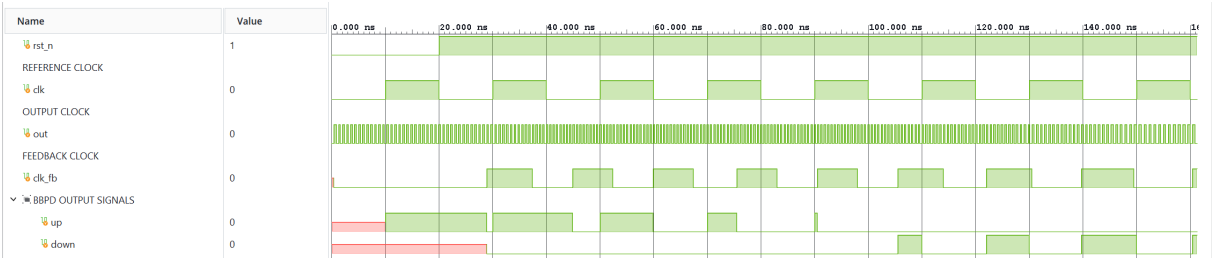


Figure 7.1. Simulation waveform of the stage 1 FLL.

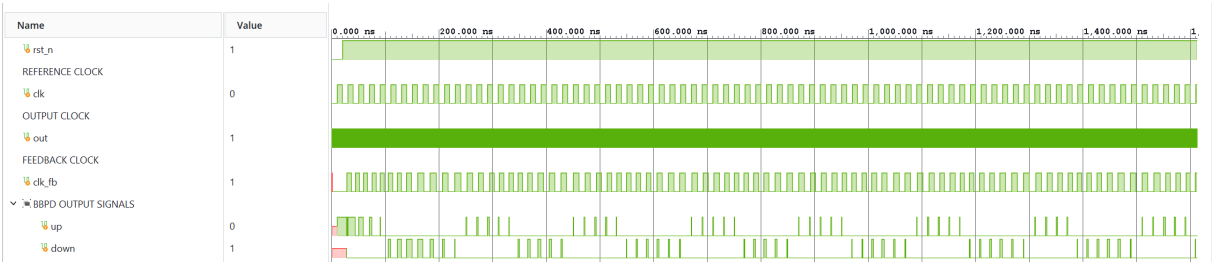


Figure 7.2. Simulation waveform of the stage 1 FLL, zoomed out.

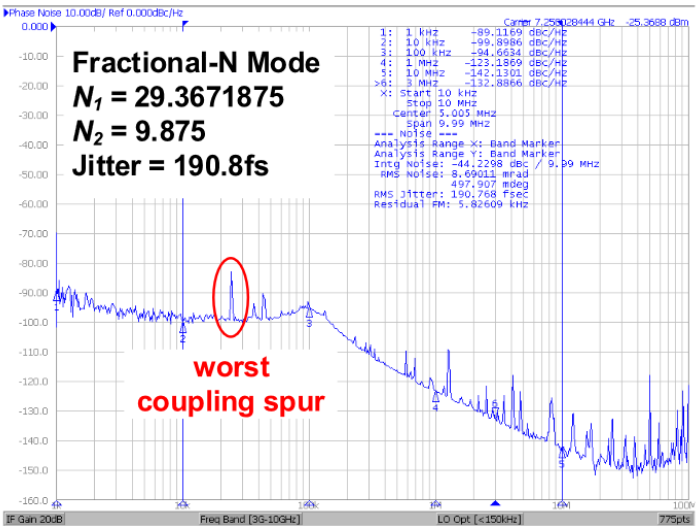
7.2. Next Steps

In the near future (next 1–2 months), I'd like to complete the full behavioural model of the cascaded PLL. Afterwards, I'd like to try model the analog parts in Verilog-A, as well as introduce noise into the SystemVerilog behavioural model.

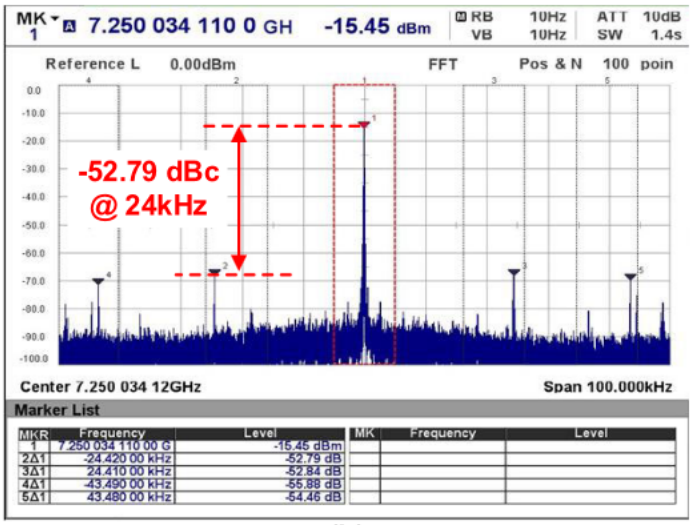
There are some parts of the SystemVerilog model that still need improvement, for example the DSM module and potentially reimplementing the DTC as a segmented DTC.

8. Performance Results

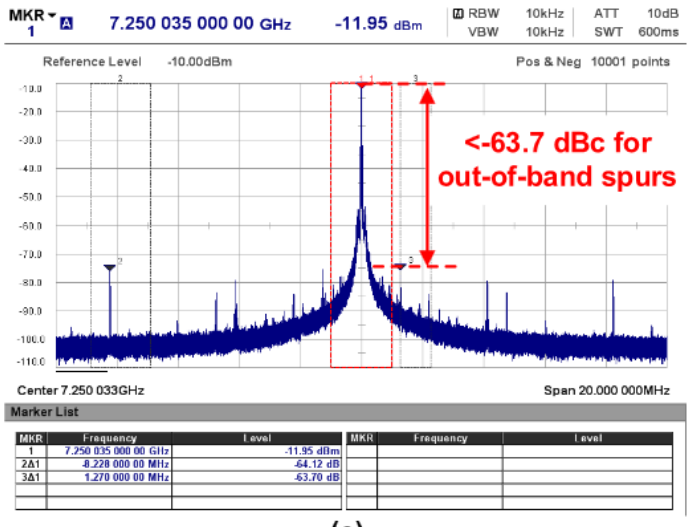
The 65 nm CMOS implementation was verified with a 50 MHz reference. Silicon measurements confirm the robustness of the dual-fractional scheme. The measurements shows the design achieves a worst-case fractional spur of -52.79 dBc, an integrated jitter is 190.8 fs, and power consumption of 14.2 mW [2]. The design achieved a figure of merit of -242.9 dB, which is highly competitive and outperforms recent designs by Murphy et al. and Zhang et al.



(a)



(b)



(c)

Figure 8.1. Measured (a) phase noise and (b) and (c) spur performance at different offset frequencies of the cascaded PLL in a near-integer channel nearby 7.25 GHz.

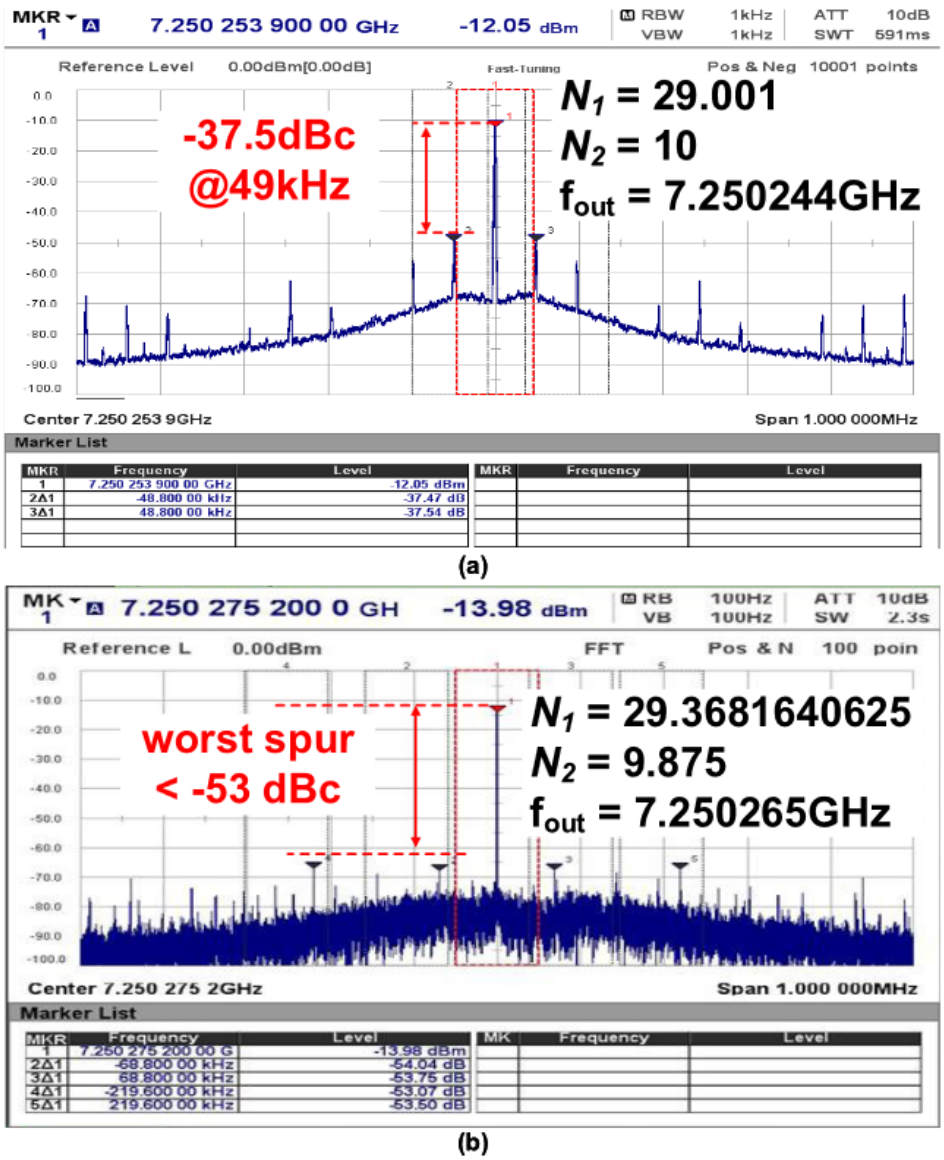


Figure 8.2. Fractional spur of the cascaded PLL when (a) $N_1 = 29.001$ and $N_2 = 10$ and (b) $N_1 = 29.3681640625$ and $N_2 = 9.875$.

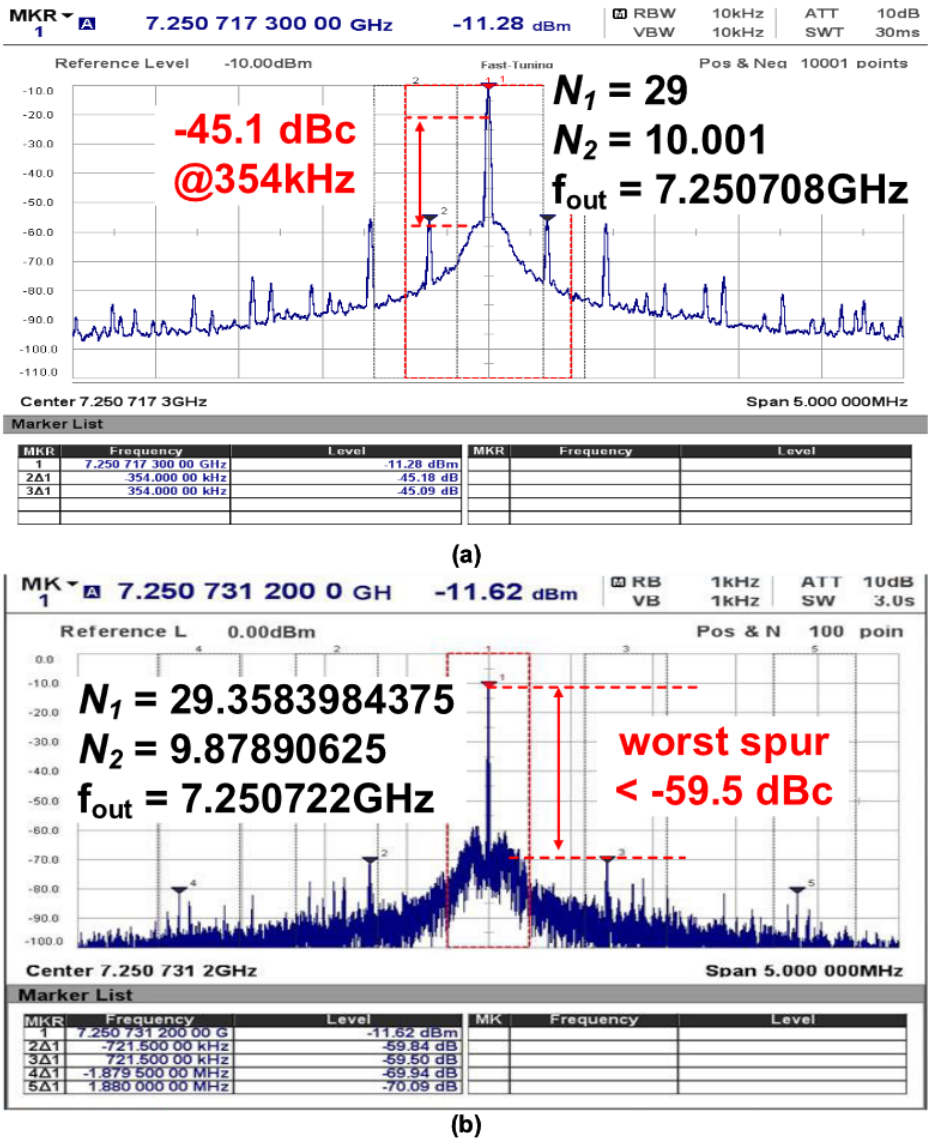


Figure 8.3. Fractional spur of the cascaded PLL when (a) $N_1 = 29$ and $N_2 = 10.001$ and (b) $N_1 = 29.3583984375$ and $N_2 = 9.87890625$.

	This Work	JSSC'23 Y. Jo [31]	JSSC'12 D. Park [17]	JSSC'17 L. Kong [18]	JSSC'22 D. Yang [3]	JSSC'23 M. Osada [32]	JSSC'21 W. Wu [33]	JSSC'23 D. Murphy [2]	JSSC'23 Z. Gao [11]	JSSC'23 S. Dartizio [34]	JSSC'24 Q. Zhang [13]
Structure	Cascaded Frac+Frac	Cascaded Frac+Sub-Int	Cascaded Int+Frac	Cascaded Int+Frac	Dual-Loop HM-PLL	Dual-Loop Dual-FB	SPLL	APLL	DPLL	DPLL	MDLL
Process	65nm	65nm	130nm	45nm	7nm	65nm	14nm	7nm	40nm	28nm	65nm
f_{out} [GHz]	6.5~8	0.6~7.7	2.9~4	2.3~2.6	25~28	2.9	6.2	4.66~5.22	2.56~4.1	9.25~10.5	1.5
f_{ref} [MHz]	50	150	50	22.6	74	40	76.8	79.96	40	250	50
Integrated Jitter [fs] Integration Range [Hz]	191 [10k~10M]	135 [1k~100M]	255 [100~40M]	1680 [10K~50M]	88 [10k~40M]	869 [1k~100M]	93.2 [10k~40M]	154 [10k~100M]	182 [10k~40M]	76.7 [10k~100M]	800 [10k~10M]
Ref. Spur [dBc]	-72.4	-77	-87	-70	-83	-65	-66	-70	-73.5	-70.5	-58
Frac. Spur [dBc]	-52.7	-51	-53.9	-52.5	-70**	-49	-66.4***	-60	-59	-71.9	-67
Frac. Spur Normalized to 1GHz [dBc]	-69.9	-68.1	-62.4	-60.1	-98.9**	-58.25	-82.2	-73.7	-67.5	-91.2	-70.52
Power [mW]	14.2	17.9	14.2	6.4	12.9	15.38	14.2	1.11	3.48	17.2	13.56
FoM* [dB]	-242.9	-244.9	-240.3	-227.4	-250	-229.4	-249.1	-255.8	-249.4	-250.6	-230.6
Core Area [mm^2]	0.48	0.64	0.26	0.03	0.24	0.112	0.31	0.21	0.31	0.33	0.23

*FoM = $20\log_{10}$(RMS Jitter/1s) + $10\log_{10}$(Power/1mW) **Ignoring coupling spurs ***Normalized to PLL frequency

Table 8.1. Performance comparison with other PLL structures.

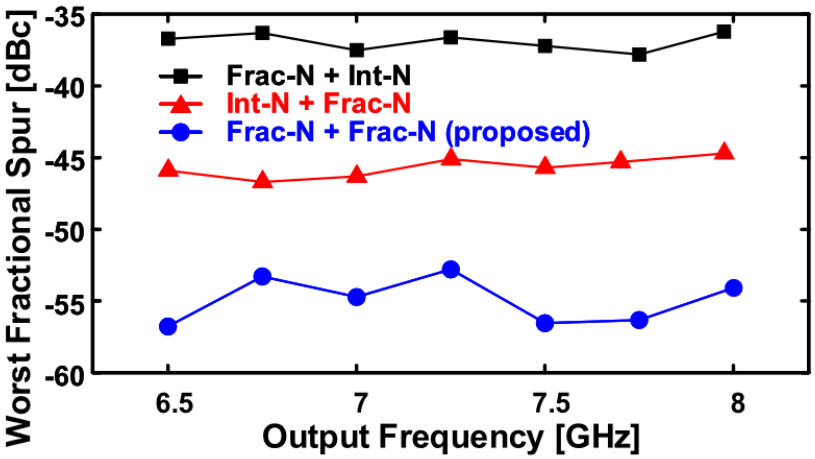


Figure 8.4. Measured worst-case fractional spur levels when the cascaded PLL is set to different cascading schemes at different near-integer channels.

9. Conclusion

The cascaded dual-fractional-*N* architecture successfully decouples frequency resolution from spur performance, providing a robust solution for near-integer channel synthesis. By steering quantisation noise out of the loop bandwidth via dual fractional stages, the design achieves high spectral purity comparable to PLLs using significantly higher reference frequencies.

See Also

Behavioural model of the PLL in SystemVerilog and Verilog-A

`https://github.com/warricklo/cascaded-pll`

References

[1] B. Razavi, *Design of CMOS phase-locked loops: From circuit level to architecture level*. Cambridge University Press, 2020.

[2] D. Xu, Y. Zhang, H. Huang, *et al.*, "A 6.5-to-8-GHz cascaded dual-fractional-N digital PLL achieving −52.79-dBc fractional spur with 50-MHz reference," *IEEE Journal of Solid-State Circuits*, vol. 60, no. 3, pp. 1043–1055, Mar. 2025. DOI: 10.1109/jssc.2024.3447021.