



UNIVERSITY OF TWENTE.

Faculty of Electrical Engineering,
Mathematics & Computer Science

Design of a frequency dividing subsampling phase-locked loop

J.B. Geertsema

MSc. Thesis

August, 2017

Supervisors

dr. ir. R.A.R. van der Zee

J.B. Lechevallier, MSc

prof. dr. ir. B. Nauta

prof. dr. ir. G. Krijnen

Report number: 067.3750

Chair of Integrated Circuit Design

Faculty of Electrical Engineering,

Mathematics and Computer Science

University of Twente

P.O. Box 217

7500 AE Enschede

The Netherlands

Abstract

In Ultra Low Power wireless systems it is crucial to generate clock signals with as low power consumption and phase noise as possible. The solution is generating the clock with an LC-oscillator that uses much less than 1mW of power. It can be beneficial to increase the frequency of this LC-oscillator from 2.5GHz to 5GHz, for a better power and noise trade-off. To arrive at the desired lower output frequency, a frequency divider is required that adds as little power consumption and noise as possible. This way a system is implemented with a better power and noise trade-off.

This thesis describes the analysis, design and comparison of a low power frequency divider. For a reliable comparison, an analysis is done to find the existing divider with the lowest power consumption using estimations. This divider turned out to be a counter implemented with true single phase clocked flip-flops. The counter is simulated to find the power consumption, noise and bandwidth. Because the counter's power increases with division ratio, a new topology is implemented that does not have this shortcoming; the dividing subsampling phase-locked loop. This subsampling phase-locked loop is designed aiming at low power consumption.

Unfortunately, the phase noise and the power of the implemented subsampling phase-locked loop is worse than that of the benchmark counter. With an input frequency of $5GHz$ and an output frequency of $2.5GHz$, the subsampling phase-locked loop has a phase noise of $-116dBc/Hz$, which is $27dB$ more than that of the counter divider. The subsampling phase-locked loop has a power consumption of $8.8\mu W$, that is double the power consumption of the counter. However, it is expected when a division ratio higher than 2 is used, the sub-sampling phase-locked loop has a lower power consumption than the benchmark counter.

Acknowledgements

Firstly I like to thank my parents for the support they gave my the past years during my study. I also would like to thank my supervisors, Joeri Lechevallier and Ronan van der Zee, for their supervision. Lastly I like to thank the other students of ICD for the cosiness around the workplace.

Contents

Abstract	iii
Acknowledgements	v
Contents	vi
1 Introduction	1
1.1 Motivation	1
1.2 Goal	2
1.3 Thesis outline	3
2 Frequency divider topologies	5
2.1 Miller divider	5
2.2 Injection locked oscillator	6
2.3 Counter	7
2.4 Conclusion	8
3 Benchmark divider	11
3.1 Miller and Injection locked oscillator power estimation	11
3.2 Counter in more detail	12
3.3 Counter power estimation	14
3.4 True single phase clocked counter design	16
3.5 True single phase clocked counter results	18
3.6 Power consumption estimation versus simulation	20
3.7 Conclusion	21
4 Frequency dividing subsampling phase-locked loop	23
4.1 Phase-locked loop	23
4.2 Subsampling phase-locked loop	28
4.3 Subsampling phase-locked loop divider design	30

<i>CONTENTS</i>	vii
4.4 Subsampling phase-locked loop divider results	44
4.5 Conclusion	48
5 Conclusion	51
5.1 Recommendations	51
A Flip flop logic types	55
B Power dissipation estimation	63
C Oscillator power estimation	73
Bibliography	77

Acronyms

AOI And-Or-Inverter Gate.

Balun Balanced to Unbalanced.

BLE Bluetooth Low Energy.

C²MOS Clocked CMOS.

CML Current Mode Logic.

CMOS Complementary Metal-Oxide Semiconductor.

CP Charge Pump.

FD Frequency Detector.

FF Flip-Flop.

FLL Frequency Locked-Loop.

FOM Figure of Merit.

HVT High Voltage Threshold.

ILO Injection Locked Oscillator.

IOT Internet Of Things.

LPF Low Pass filter.

LVT Low Voltage Threshold.

OAI Or-And-Inverter Gate.

PD Phase Detector.

PFD Phase Frequency Detector.

PLL Phase Locked-Loop.

PN Precharged N TSPC stage.

PP Precharged P TSPC stage.

PSS Periodic Steady State.

Q Quality factor.

RVT Regular Voltage Threshold.

SN Static N TSPC stage.

SP Static P TSPC stage.

SSB Single Side Band.

SSPD Subsampling Phase Detector.

SSPLL Subsampling Phase-Locked Loop.

TSPC True Single Phase Clocked logic.

VCO Voltage Controlled Oscillator.

Chapter 1

Introduction

1.1 Motivation

The wireless data-rates of mobile devices improves, while battery life should not degenerate, especially for **Internet Of Things (IOT)** applications. In these wireless connections, co-existence of interfering channels cause problems. One of these problems is reciprocal mixing, where large signals from interfering channels (f_{INT}) mix down with the phase noise of the local oscillator (f_{LO}) on top of the wanted signal (f_{IN}), as shown in **Figure 1.1**. The phase noise in the local oscillator is caused by up-converted $1/f$ and current noise originating from the transistors in the oscillator.

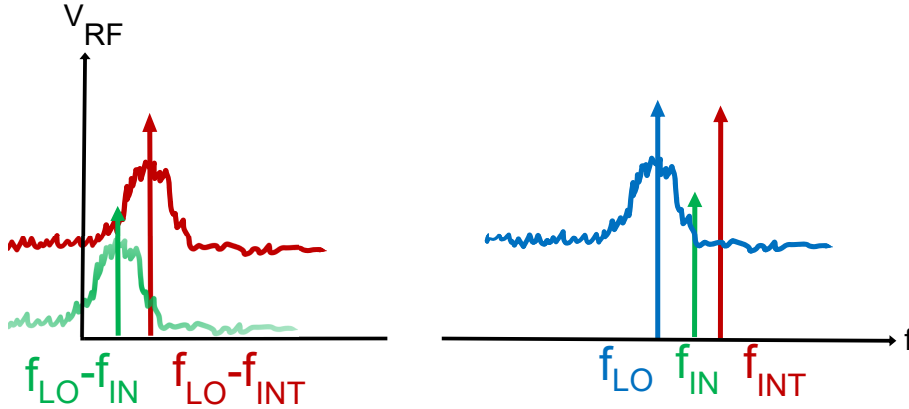


Figure 1.1: Reciprocal mixing of the large signals from neighboring channels, f_{INT} , on top of the wanted signal, f_{IN} [1]

The phase noise of an oscillator, under the assumption that it only contains current noise, is approximated as[2]:

$$\mathcal{L}(f_m) = \frac{kT}{2P} \frac{1}{Q^2} \left(\frac{f_0}{f_m}\right)^2 \quad (1.1)$$

This phase noise scales with the DC power consumption of the oscillator core (P), the unloaded **Quality factor** (Q) of the oscillator (Q), Boltzmann constant

(k), absolute temperature in Kelvin (T), the oscillation frequency (f_0), and the offset frequency (f_m) from f_0 at which the phase noise is measured in a 1Hz bandwidth. To compare oscillators a quantity of oscillator quality was devised[2], in which P , f_m and f_0 play no role. This is the **Figure of Merit (FOM)**, where a lower FOM means a better power and noise trade-off:

$$FOM = 10 \log(\mathcal{L}(f_m)) + 10 \log\left(\left(\frac{f_m}{f_0}\right)^2 \frac{P}{1mW}\right) \quad (1.2)$$

The FOM is logarithmic inversely proportional to the oscillator Q. When an LC oscillator is implemented that runs at 2.5GHz and uses less than 1mW of power, the Q of the inductance (Q_L) dominates the Q of the oscillator. Previous research done at the ICD group [3] shows that Q_L of a small inductor increases when the oscillation frequency is increased from 2.5GHz to 5GHz and 10GHz. This is also shown by the graph in Figure 1.2, where Q_L is simulated over the inductance for different frequencies. Therefore a higher FOM can be attained by increasing the oscillation frequency of the LC-oscillator.

To obtain a system with a higher FOM that has an output frequency of 2.5GHz, the LC-oscillator with a higher FOM and oscillation frequency in combination with a frequency divider can be used. This frequency divider should convert the oscillation frequency down, while adding as little phase noise and power consumption as possible.

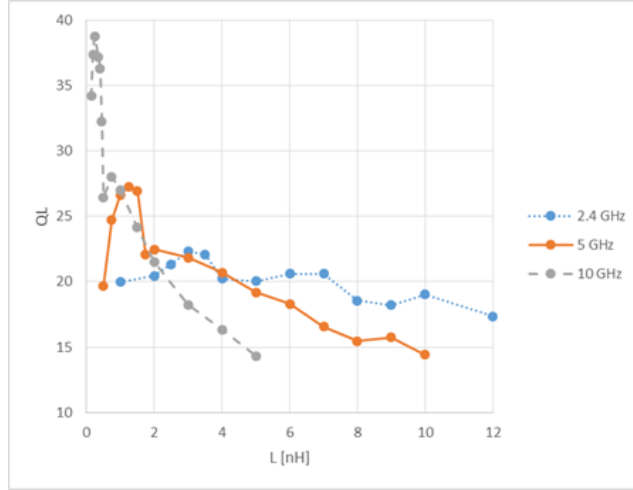


Figure 1.2: The Q versus the inductance for $f_0 = 2.4, 5$ and 10GHz[3].

1.2 Goal

The goal of this thesis is the analysis, design and comparison of a low power frequency divider. Therefore the research question is; “*How can a low power frequency divider be implemented?*” These are the goals for the design:

- Power: *as low as possible, by any means less than 1mW*

- Noise: *compliant to the Bluetooth Low Energy (BLE) specification: -105dBc/Hz at $f_m = 3MHz$*
- Frequencies: *Constant 2.5GHz output frequency, for either a 5GHz, 10GHz or 20GHz input frequency.*
- Output phases: *Preferably differential quadrature out*

1.3 Thesis outline

To explore what types of frequency dividers exist, these are studied in [chapter 4](#). Then in [chapter 3](#), the existing divider with the lowest power consumption is found using estimations and its specifications are found using simulation as a benchmark. After this a new divider, the [Subsampling Phase-Locked Loop \(SSPLL\)](#), is analyzed and implemented in [chapter 4](#). Finally the benchmark is compared with the developed divider, conclusions are given and recommendations done in [chapter 5](#).

Chapter 2

Frequency divider topologies

To examine what solutions exist for low power dividers, this chapter contains an examination of existing frequency dividers. These are the Miller, Injection Locked and Counter divider. The Miller divider is covered in [section 2.1](#) and the Injection Locked dividers and counter divider are mentioned in [section 2.2](#) and [section 2.3](#), respectively.

2.1 Miller divider

With a Miller divider, the input (*IN*) and output (*OUT*) are fed into a mixer, as shown in [Figure 2.1](#)¹. If the loop functions properly, then $f_{out} = \frac{1}{2}f_{in}$ [4, p. 699]. This produces a $\frac{1}{2}f_{in}$ and $\frac{3}{2}f_{in}$ term at the output of the mixer (*X*). By filtering out the $\frac{3}{2}f_{in}$ term, only the $\frac{1}{2}f_{in}$ term remains at *OUT*. Advantages

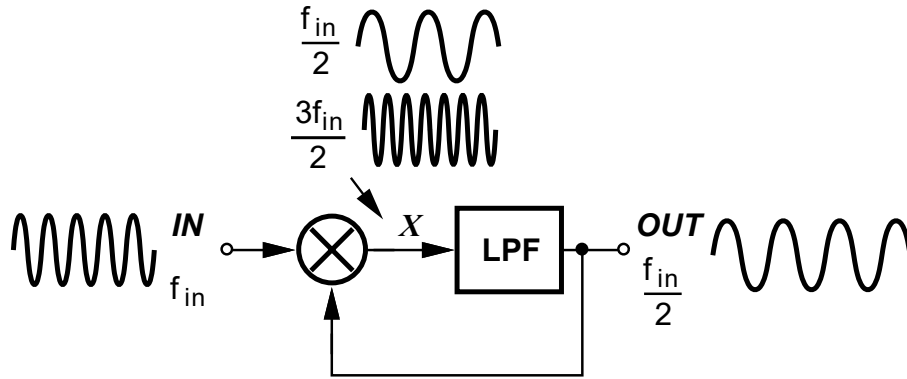


Figure 2.1: Miller divider topology. [4, p.699-707]

of the Miller divider:

- The topology is useful for very high frequencies. Input frequencies above $50GHz$ are no exception[5].

¹Throughout this report the following font styles are used for figures; *signals names* in bold italic, **component names** in bold, frequencies and properties in normal and *degrees* in italic font.

Disadvantages of the Miller divider:

- Require a filter of an order higher than one to filter out the $\frac{3}{2}f_{in}$ component.
- A maximal division ratio of 2 can be achieved per Miller divider [4, p. 705].
- There is loss from input to output with a passive mixer, gain depending on the conversion gain for an active mixer.

2.2 Injection locked oscillator

An injection locked oscillator consists of an input with frequency f_{in} , that is injected in an oscillator. Such an oscillator is shown in Figure 2.4. The connection between X and the oscillator denotes injection rather than frequency control.

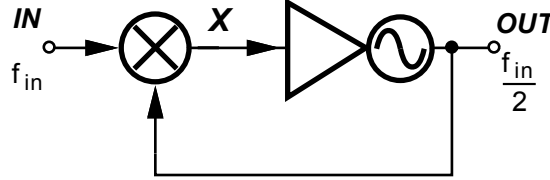


Figure 2.2: An Injection-locked divider. [4, p. 707]

Without injection, this oscillator produces a periodic signal with frequency f_0 and jitter. This translates to a single harmonic f_0 with surrounding phase noise in the output spectrum, also called the main lobe. When the injected signal is outside the lock range of the oscillator, two beats appear alongside the main lobe in the output spectrum. These beats with mixing frequencies $f_0 - f_{in}$ and $f_0 + f_{in}$, decrease in magnitude, and the main lobe is pulled toward f_{in} , when f_{in} approaches f_0 . The beats vanish and the main lobe follows f_{in} , if f_{in} is within the lock range [6]. A plot of the magnitude of the beats versus the input frequency is shown in Figure 2.3a. Injection locking also happens when f_{in} is close to an integer multiple (harmonic or superharmonic) or division (subharmonic) of the resonant frequency. This allows the use of an injection locked oscillator as a frequency divider or multiplier.

Because the locked oscillator follows the input signal within the lock range, it also follows the input signal's phase noise in this range. Thereby any phase noise generated by the oscillator in this range is suppressed, as shown in Figure 2.3b. An expression for the lock range of an injection locked oscillator [7] is:

$$f_L \approx \frac{f_0}{2Q} \cdot \frac{2}{\pi} \cdot \frac{I_{inj}}{I_{osc}} \quad (2.1)$$

This means that the locking range is inverse proportional to the Q of the oscillator, and therefore proportional to the phase noise produced by the oscillator without injection (1.1). It is also proportional to the ratio of tail current through the oscillator (I_{osc}) and the injected current in the oscillator (I_{inj}).

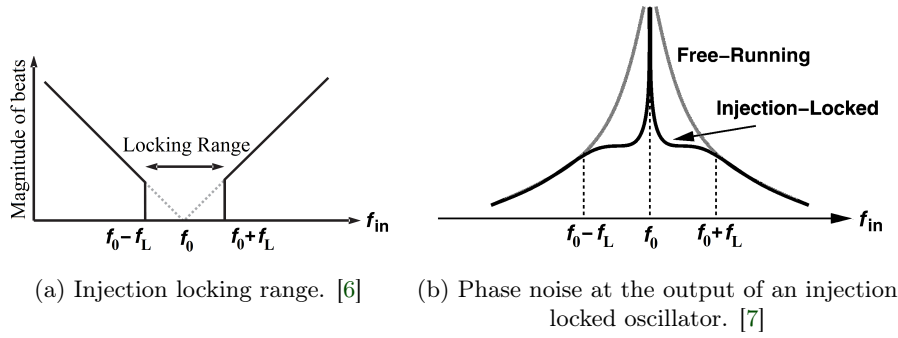


Figure 2.3

Advantage of the injection locked oscillator:

- It is usable for very high input frequencies, frequencies above 50GHz are not uncommon [8–10].
- The output noise follows the input noise within the locking range, possibly producing a low noise output.

Disadvantages of the injection locked oscillator:

- The oscillator can lock to the wrong integer of the input frequency.
- There is a limited locking range, so the noise of the input is tracked in a limited range.

2.3 Counter

A Flip-Flop (FF) is a device that passes its input to its output on one falling clock edge or rising clock edge. If the input is always the inverted output, the output is flipped on one of the clock edges, as shown in Figure 2.4. This means that the clock flips twice and the output flips once every clock period, and a frequency division by 2 is realized.

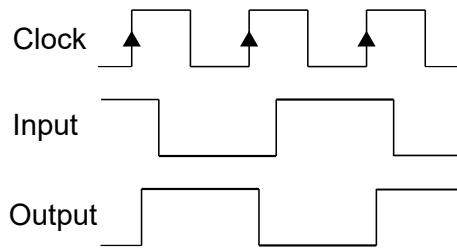
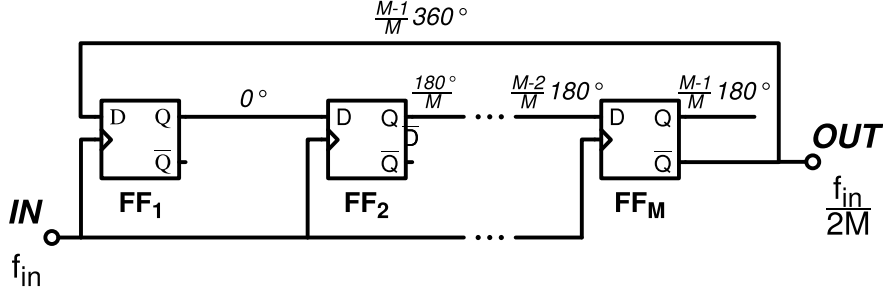


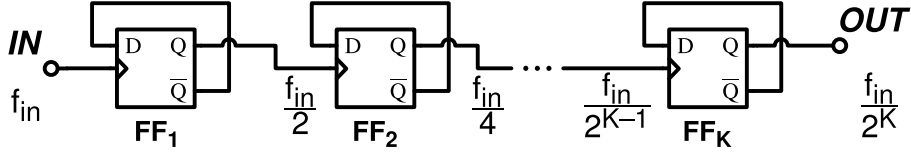
Figure 2.4: The clock, input and output waveforms of a positive edge triggered FF, when the input is always the complement of the output.

Cascading FFs is conventionally done in two ways; with a feedback from inverse output to input per FF, and with this feedback per cascade of FFs. An illustration where this is done with D-FFs is shown in Figure 2.5. With

feedback per FF, an asynchronous counter is realized, the input is the inverted output per FF and the frequency is halved every FF. The number of equal FFs in this figure is K , and the division ratio (N) is 2^K . With feedback per cascade of FFs a synchronous counter is realized, and the input is the inverted output per cascade of FFs. It takes the number of FFs in the cascade (M) for the inverted input to reach the output, giving a phase shift of $\frac{180^\circ}{M}$ per FF, and a frequency division of $2M$. Synchronous and asynchronous counters combined with optional logic can be used to realize various division ratio's.



(a) A synchronous counter topology.



(b) An asynchronous counter topology.

Figure 2.5

Advantages of asynchronous compared to synchronous counter are:

- It consumes less power, each FF along the cascade switches at a lower frequency.
- There is less loading of the clock source.
- Higher division ratio's can be realized while using less FFs.

Disadvantages of asynchronous compared to synchronous counters:

- The jitter accumulates along the cascade of FFs.
- There are less additional phases on one frequency present in the cascade of FFs.
- Only a division ratio of 2^K instead of $2M$ can be achieved.

2.4 Conclusion

There are three fundamental types of frequency dividers; the Miller, injection-locked and counter divider. These three fundamental ones are covered in this

section. A Miller divider consists of a mixer, and can therefore only give a division ratio of two per Miller divider. This divider is mostly used for higher frequencies than 10GHz, and could therefore be unsuitable for this thesis. The injection locked oscillator has the advantage of following the phase noise of the input within the locking range, and can lock to any integer multiple or division of the input frequency. However, the phase noise and power consumption performance could not be sufficient due to the limited locking range. A counter is the most commonly used divider, and consist of cascaded FFs with feedback from the complement output to the input per FF or per cascade of FFs. This could be a promising topology because there are many low power flip-flops. A more in depth comparison of these three divider types in terms of power consumption will be done in the next chapter.

Chapter 3

Benchmark divider

Because low power consumption is the main goal of this thesis, an assessment of the power consumption is done for all three divider types explained in [chapter 2](#).

First an estimation for the Miller and of the injection locked oscillator is done in [section 3.1](#). This gives an unacceptable high power consumption, therefore the counter is examined in more detail in [section 3.2](#). An estimation of the power consumption of the counter implemented in various logic types is done in [section 3.3](#). With this assessment, the counter with the lowest power consumption is determined in this section. This counter is designed and simulated in [section 3.4](#) and [section 3.5](#) respectively. Comparing the estimated and simulated power consumptions of the divider with the lowest power consumption is done in [section 3.6](#). Some conclusions are drawn in [section 3.7](#). Finding a divider with the lowest power consumption is done to provide a set a set of specifications, a benchmark, to compare the developed subsampling phase-locked loop divider with.

3.1 Miller and Injection locked oscillator power estimation

To find the divider topology with the lowest power consumption, recently published Miller and Injection locked dividers are summarized. A selection of these low power miller and Injection Locked Oscillator (ILO) are shown in [Table 3.1](#). Almost all the dividers shown in this table operate at high frequencies, except for the ILOs described in [\[11\]](#) and [\[12\]](#). This is logical because all these high frequency dividers use a specialized topology for high frequencies with a differential pair with inductors as load. This is done because this gives additional output impedance and output voltage at the resonance frequency of the circuit, without using extra power. However, this differential pair is not a low power solution, because the tail current through this circuit has to be scaled up for biasing the transistors in the correct region, and to obtain sufficient output voltage swing. This translates into a larger power consumption in excess of $400\mu W$.

Using a ring based architecture is also possible, because this could save power consumption, as this does not rely on a tail current source. This power consumption of a ring oscillator is dominated by the CV^2f switching and

crowbar current of the circuit. This CV^2f switching power consumption is due to the charging and discharging of capacitance. Each period, $\frac{1}{2}CV^2f$ is dissipated in the PMOS while charging and the other $\frac{1}{2}CV^2f$ in NMOS while discharging. Crowbar current is current that does not contribute to charging capacitance, but goes directly from V_{dd} to GND through all transistors. Using an ring oscillator in an ILO is the most power efficient solution, as also shown in Table 3.1. The power consumption however is still more than $100\mu W$ for these dividers.

Table 3.1: A selection of low power Miller and ILO dividers.

Reference	Type	Power [μW]	Frequency [GHz]	$\mathcal{L}(1MHz)$ [dBc/Hz]	Technology [nm]
[13]	Miller	1600	35.7	-126 ¹	65
[5]	Miller	810	59.5	-117	130
[11]	ILO	200	2	-108	180
[8]	ILO	1900	53.7	NS	65
[9]	ILO	800	55	-118	90
[10]	ILO	440	60	NS	65
[12]	ILO	740	1.2	-130	180

3.2 Counter in more detail

To investigate how a low power counter can be made, first the FF is examined, then the latch types contained in this FF and the logic types implementing these latches are explored.

Flipflop

A Clk edge-triggered FF can consist of 2 gated latches cascaded with one latch having \overline{Clk} , as shown in Figure 3.1. This is also called a master-slave configuration, where the first latch is called the master and the second the slave. Sometimes the two latches are combined to form a optimized transistor level topology. Latches are also called non edge-triggered FFs, and have the same symbols as edge-triggered FFs. In the context of this thesis a non edge-triggered memory device is called a latch, while an edge-triggered device is called a FF.

Latch types

Gated latches are usually implemented as 4 different types; SR, JK, D and T. The name, truth table and schematic of these latches are given in Table 3.2. A gated latch consist of a clock gate part and a latch part. The implementation of the clock gate part depends on the type of latch, and how the inputs effect the output. However the outputs are always latched if $Clk = 0$. The latch part of all latch types is the same in Table 3.2, two cross coupled NAND gates

¹Extrapolated from -106dBc/Hz at $f_m=100kHz$

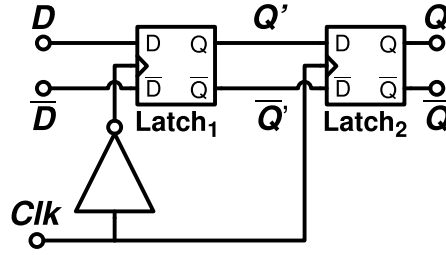


Figure 3.1: A positive edge-triggered FF build up from two latches.

with a Q and inverted \overline{Q} output in CMOS, but could also be implemented differently dependent on the logic type. The function of the circuits, if the input is evaluated ($Clk = 1$), is as follows:

- SR: the latch is set and reset with 1 on the S and R inputs respectively. The previous output is stored if both S and R are 0. Both inputs 1 is forbidden as this gives $Q = \overline{Q} = 0$, which is not complementary.
- D: The circuit passes the D input.
- JK: same as the SR latch, but with inputs J and K instead of S and R and this latch passes the previous outputs inverted when both inputs are 1.
- T: Passes the previous outputs non-inverted and inverted if $T = 0$ and $T = 1$ respectively.

Logic types

The latch types can be implemented in various sorts of logic. These logic types are described in detail in [Appendix A](#). There is always one optimal latch type for one type of logic. [Complementary Metal-Oxide Semiconductor \(CMOS\)](#), [Transmission gate](#), [Clocked CMOS \(C²MOS\)](#), [True Single Phase Clocked logic \(TSPC\)](#) and [Current Mode Logic \(CML\)](#) logic are described in the appendix. Other, less power efficient logic types, such as pseudo-NMOS and PMOS logic are not discussed in this thesis.

[CMOS](#) is the most basic type of logic and contains most transistors. Transmission gate logic consists of transmission gates, which pass the input, depending on the clock. One transmission gate combined with an inverter and a capacitance forms a latch. With [C²MOS](#), the transmission gates are integrated in the inverters to solve the overlapping-clock problem of transmission gate logic. [TSPC](#) is a derivate of [C²MOS](#), where transistors are removed to prevent the need for complementary clock.

Table 3.2: Table with the 4 commonly used latch types.

Name	Truth table if $Clk = 1$	logic gate schematic																				
SR	<table><tr><th>S</th><th>R</th><th>Q</th><th>state</th></tr><tr><td>0</td><td>0</td><td>$Q(\text{prev})$</td><td>latch</td></tr><tr><td>0</td><td>1</td><td>0</td><td>reset</td></tr><tr><td>1</td><td>0</td><td>1</td><td>set</td></tr><tr><td>1</td><td>1</td><td>?</td><td>undef</td></tr></table>	S	R	Q	state	0	0	$Q(\text{prev})$	latch	0	1	0	reset	1	0	1	set	1	1	?	undef	
S	R	Q	state																			
0	0	$Q(\text{prev})$	latch																			
0	1	0	reset																			
1	0	1	set																			
1	1	?	undef																			
D	<table><tr><th>D</th><th>Q</th><th>state</th></tr><tr><td>0</td><td>0</td><td>reset</td></tr><tr><td>1</td><td>1</td><td>set</td></tr></table>	D	Q	state	0	0	reset	1	1	set												
D	Q	state																				
0	0	reset																				
1	1	set																				
JK	<table><tr><th>J</th><th>K</th><th>Q</th><th>state</th></tr><tr><td>0</td><td>0</td><td>$Q(\text{prev})$</td><td>latch</td></tr><tr><td>0</td><td>1</td><td>0</td><td>reset</td></tr><tr><td>1</td><td>0</td><td>1</td><td>set</td></tr><tr><td>1</td><td>1</td><td>$\overline{Q}(\text{prev})$</td><td>invert & latch</td></tr></table>	J	K	Q	state	0	0	$Q(\text{prev})$	latch	0	1	0	reset	1	0	1	set	1	1	$\overline{Q}(\text{prev})$	invert & latch	
J	K	Q	state																			
0	0	$Q(\text{prev})$	latch																			
0	1	0	reset																			
1	0	1	set																			
1	1	$\overline{Q}(\text{prev})$	invert & latch																			
T	<table><tr><th>T</th><th>Q</th><th>state</th></tr><tr><td>0</td><td>$Q(\text{prev})$</td><td>latch</td></tr><tr><td>1</td><td>$\overline{Q}(\text{prev})$</td><td>invert & latch</td></tr></table>	T	Q	state	0	$Q(\text{prev})$	latch	1	$\overline{Q}(\text{prev})$	invert & latch												
T	Q	state																				
0	$Q(\text{prev})$	latch																				
1	$\overline{Q}(\text{prev})$	invert & latch																				

3.3 Counter power estimation

To find the counter and FF logic type that has the lowest power consumption, an estimation is done in Appendix B. First, expressions for the power consumption of a synchronous and asynchronous counters are given in terms of the number of switched data (d) and clocked transistors (c) in a FF. These are:

$$P_{\text{synchronous counter}} = \frac{1}{2}(Nc + d)CV_{dd}^2f_{in} \quad (3.1)$$

$$P_{\text{asynchronous counter}} = (2c + d)CV_{dd}^2f_{in}(1 - \frac{1}{N}) \quad (3.2)$$

Then the capacitance of one minimum sized TSMC65 transistor is estimated by simulation. Finally, the number of switched transistors is obtained for the various types of logic. Combined with the same supply voltage used for the implemented SSPLL, namely 0.7V, this information gives an estimation for the power consumption of the two counter types, implemented in various types of logic (minimum sized) in TSMC65 technology. CML logic is not taken into account in this comparison, as the tail current of the circuit is required to be large to achieve the 2.5GHz bandwidth. This causes a large (DC) power

consumption. It is important to note that the following causes for additional power consumption are not included in this estimation:

- Power consumption other than dynamic CV^2f_{in} power consumption. This includes leakage, short circuit, DC standby currents and glitch power. These powers are almost non computational, and therefore not included in the estimation.
- Supply voltage scaling; all logic types are compared with the same supply voltage. Lower supply voltages could be used for logic types that have less transistors stacked. However when using counters in a system, most of the time counters are not the limiting factor on the supply voltage.
- Clock circuitry other than inverters for complementary clock. This could cause an additional power consumption for transmission gate FFs, as these require non-overlapping clock.
- Additional capacitances. It is assumed that the parasitic transistor capacitances are sufficiently large to hold the logic levels at the frequency that the circuit operates. This means no capacitance (C in Figure A.1a and Figure A.2) is added to the outputs of non-regenerative latches. Because the frequency becomes lower with higher division ratios, this could cause problems with high division ratios.
- Input, Output and clock buffers. Adding these buffers would add power consumption to all the logic types, and would not significantly influence the comparison.

Plotting this power over the division ratio (N) gives Figure 3.2. Note that for the asynchronous counter power consumption this plot only contains points for $N = 2, 4, 6, 8, 10, \dots$, and $N = 2, 4, 8, 16, \dots$ for the synchronous counters. This is displayed by placing \circ and \times markers for the contained points of the synchronous and asynchronous counters respectively. The power consumption of synchronous and asynchronous counters implemented in the same logic type, is the same for $N = 2$. This is true because in this case the counter only contains 1 FF with feedback from \overline{Q} to D . The synchronous counter power consumption increases linearly, because the number of clocked FFs increases linearly with N . The asynchronous counter power consumption approaches a horizontal asymptote with N , because the power consumption is dominated by the first FF in the cascade. All other FFs use less than the power consumption of the first FF. These two different functions for power consumption cause the power consumption of asynchronous counters to be lower than synchronous counters above a certain N . This N is dependent on the ratio of clocked transistors to data transistors in a FF.

The TSPC divider is a clear winner in terms of power consumption. This is due to the absence of extra inverters for complementary clock and output, less data transistors, but a little more clocked transistors compared to the second lowest power consuming FF type, the C^2MOS FF. C^2MOS And dynamic transmission gate counters are more power efficient than synchronous TSPC counters when $N > 4$. This is true because synchronous counters power consumption scales with the number of clocked transistors, and TSPC FFs have more clocked transistors than C^2MOS and dynamic transmission gate FFs.

Counter power consumption, with $C=210\text{aF}$, $V_{dd}=0.7\text{V}$, $f_{in}=5\text{GHz}$, $f_{out}=f_{in}/N$

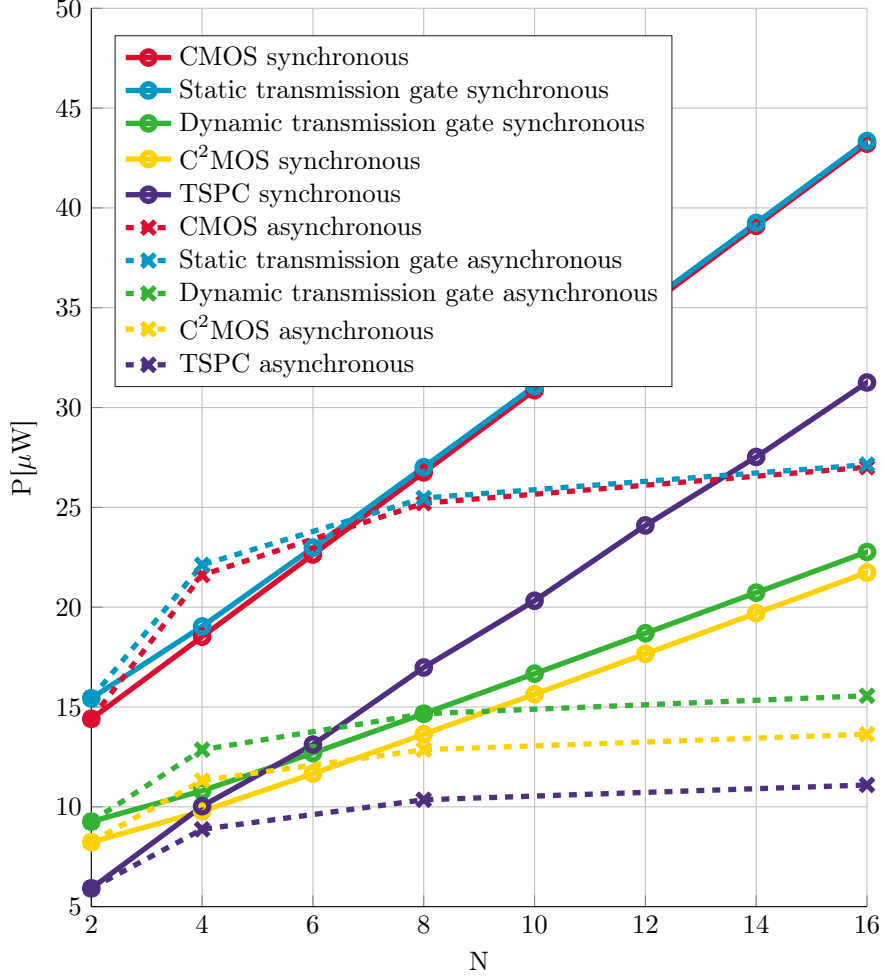


Figure 3.2: The estimation of counter power over division ratio for different types of FF logic types.

Therefore a asynchronous counter, build with TSPC FFs is used as a benchmark for the designed frequency divider.

3.4 True single phase clocked counter design

The used TSPC FF in the asynchronous counter is shown in Figure 3.3. The colored arrows in this figure show which part of the circuit passes a value, when the input is a logic 1 (blue) or 0 (purple). Passing these values also depend on the value of Clk , therefore the value of Clk is annotated along these arrows. This is how the 1st, 2nt, and 3th stage operate:

- The first "static P" stage blocks a 1 to OUT_1 if $Clk = 1$ or $D = 1$, but always passes a 0 to OUT_1 if D is 1.

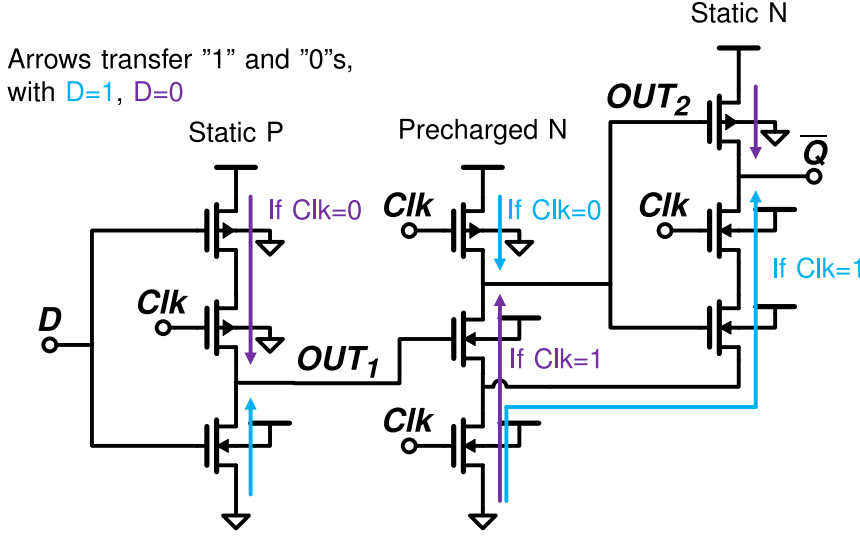


Figure 3.3: The TSPC FF implemented with body biasing. The color of the arrows indicate which part of the circuit conducts if the input is 1 (blue) or 0 (purple), and Clk is 1 or 0.

- The second "Precharged N" stage needs this blocked 1 on OUT_1 for passing a 0 to OUT_2 when $Clk = 1$, and always blocks a 1 to OUT_2 when $Clk = 1$.
- The third stage needs this blocked 1 on Out_2 for passing a 0 to \bar{Q} when $Clk = 1$, but always passes a 1 to \bar{Q} if OUT_2 is 0.

The first two stages therefore form a non-inverting positive edge-triggered FF for $D = 0$. The last two stages form a non-inverting positive edge-triggered FF for $D = 1$. The three stages combined form an inverting positive edge-triggered FF for both values of D .

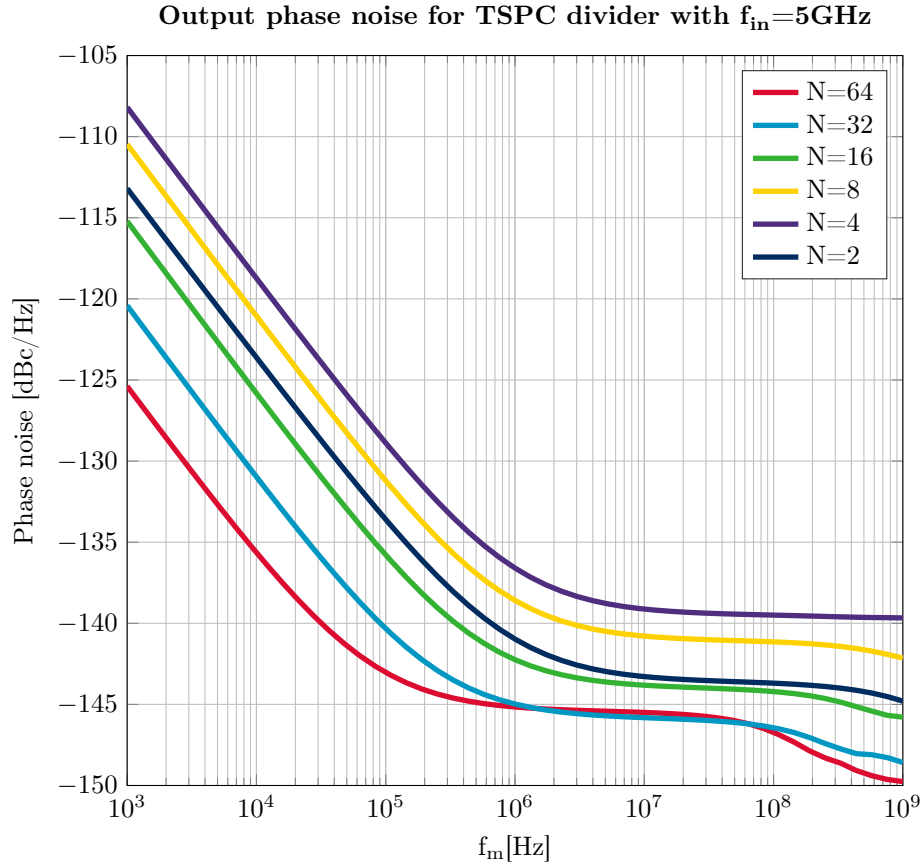
This counter is implemented in TSMC65 technology with all NMOS minimum sized, and the PMOS wider to balance the NMOS and PMOS driving strength: $\mu_N \frac{W_N}{L_N} = \mu_P \frac{W_P}{L_P}$. With $\frac{W_N}{L_N}$ and $\frac{W_P}{L_P}$ the width to length ratio of the NMOS and PMOS transistor respectively. This is done in Cadence Virtuoso, and this design is simulated using Spectre(RF). Low Voltage Threshold (LVT) transistors are required to operate the FF with the same supply voltage used for the implemented SSPLL, namely 0.7V. While simulating with these ascending N and f_{in} , it is observed that the bandwidth of this divider is 5GHz. This means that with a fixed output frequency of 2.5GHz, only a divider of 2 can be implemented with TSPC FFs. To provide an adequate benchmark for the SSPLL divider, a wider range of division ratios is required. Therefore $N = 2, 4, 8, 16, 32$ and 64, is used with a fixed input frequency of 5GHz. This means that the desired 5GHz to 2.5GHz division is still simulated, but there is a deviation from the goal by also dividing to 1.25GHz, 625MHz, 312.5MHz, 165MHz and 78MHz. Because the main goal of this thesis is low power con-

sumption, the TSPC divider is used as benchmark and not a divider with a higher bandwidth, like a Miller or Injection locked divider.

3.5 True single phase clocked counter results

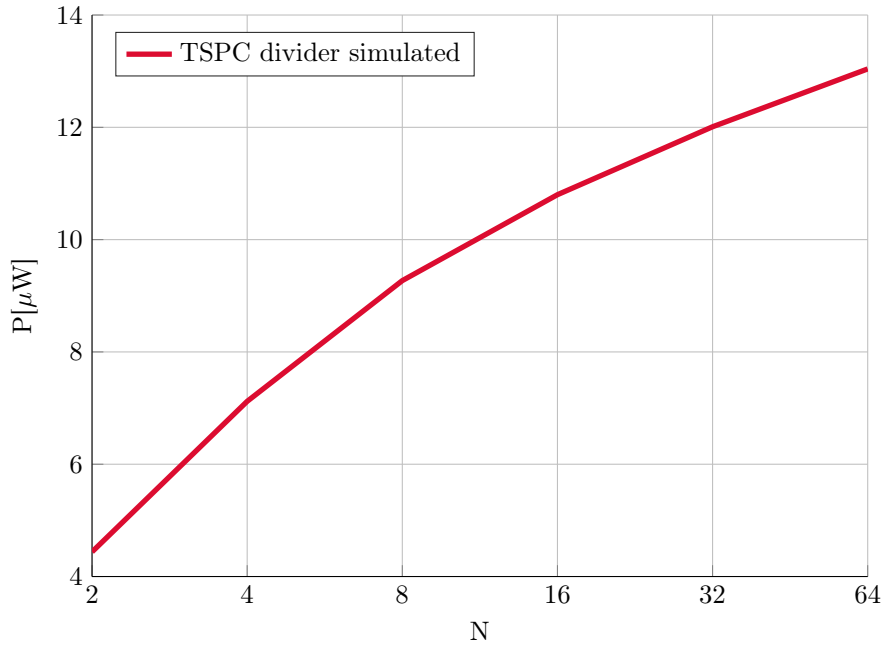
Simulating the TSPC FF divider with various division ratios gives a set of phase noise and power consumption. The phase noise over f_m with several division ratios, and the power consumption over division ratio, are shown in figures 3.4a and 3.4b respectively. The phase noise decreases if N is increased, except between $N = 2$ and $N = 4$. This phase noise should decrease with N , because phase noise decreases with output frequency (f_{out})[14], which in turn decreases with N . The phase noise increases when going from a $N = 2$ to $N = 4$ divider. This is presumably caused by a greater noise addition by the first 2 FFs to the noiseless clock source, than the noise reduction due to a lower output frequency.

Figure 3.4a also shows that the frequency divider operates within the stated specifications in section 1.2. The simulated phase noise of $-143dBc/Hz$ at $f_m = 3MHz$ for $N = 2$ is well below the $-105dBc/Hz$ BLE specification. The power consumption of $4.4\mu W$ is also far less than $1mW$ for all division ratios.



(a) The output phase noise over offset frequency (f_m) for various division ratio's (N).

Counter power consumption, with $C=210\text{aF}$, $V_{dd}=0.7\text{V}$, $f_{in}=5\text{GHz}$, $f_{out}=f_{in}/N$



(b) The power consumption for various division ratio's.

Figure 3.4

3.6 Power consumption estimation versus simulation

To test the power consumption estimation method, the estimated power consumption is compared to the simulated one. This is done by plotting these two in Figure 3.5. When $N < 16$, the estimated power consumption is higher than the calculated one. A cause could be the approximated that all the transistors in a cascode add the same capacitance to the total output capacitance, which could be an overestimation. If $N > 16$, the simulated power consumption is higher than the estimated one. With a very high division ratio, an added FF would be running at a very low frequency. In the estimation it would cause almost no additional power consumption, but in simulation it still adds power consumption due to leakage currents. Therefore the power consumption of the asynchronous counter keeps rising with increasing division ratio in simulation. The power consumption of the second most power efficient divider, the asynchronous C²MOS divider, is also plotted in Figure 3.5. This shows that if the CMOS power consumption is overestimated, its power would still be in the same order of magnitude as the power consumption of the TSPC counter.

Counter power consumption, with $C=210\text{aF}$, $V_{\text{dd}}=0.7\text{V}$, $f_{\text{in}}=5\text{GHz}$, $f_{\text{out}}=f_{\text{in}}/N$

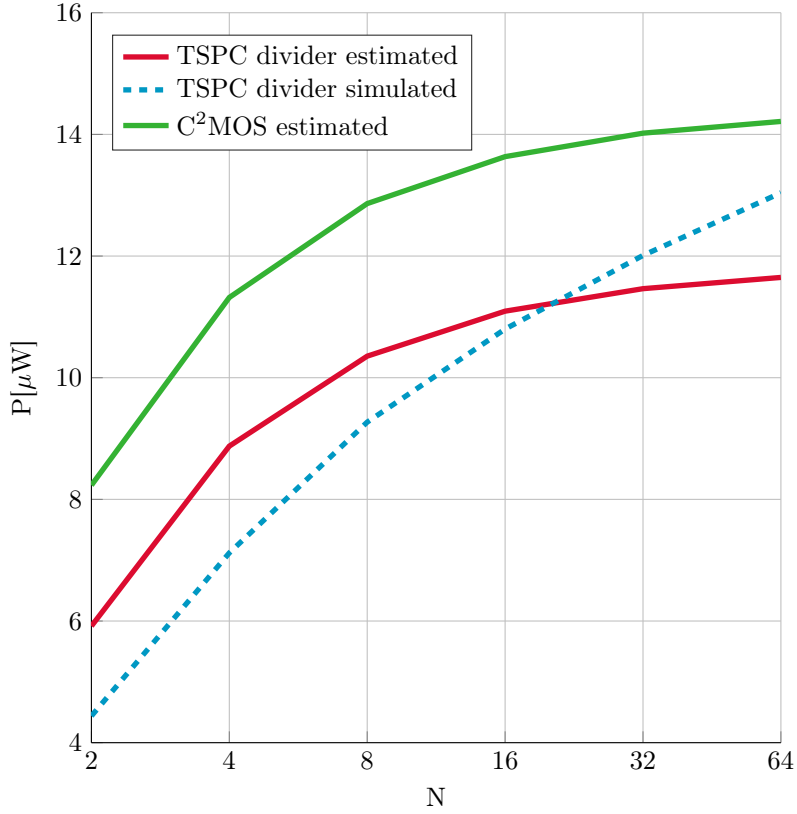


Figure 3.5: The estimated and simulated power consumption of the TSPC divider.

3.7 Conclusion

First an assessment of the power consumption of the ILO and Miller dividers is done. The power of these dividers is in excess of $100\mu W$. Then an estimation is done for counters implemented in various sorts of logic. The counter with the lowest power consumption turned out to be the asynchronous counter divider implemented in TSPC logic. The estimated power consumption of this counter is $6\mu W$, and thus much lower than the power consumption of a ILO or Miller divider

This TSPC counter divider is implemented in Cadence and simulated in section 3.5 to obtain all the necessary specifications. The result of simulating this counter with $N = 2$ is a noise of $-143dBc/Hz$ at $f_m = 3MHz$, and a power consumption of $4.4\mu W$, which is sufficient for the stated goal of this thesis. If the power consumption of the second most power efficient is overestimated, its power would still be in the same order of magnitude as the power consumption of the TSPC counter. Therefore this TSPC counter can be used as a benchmark.

Unfortunately, higher input frequencies than 5GHz are not supported by TSPC FFs, even if body biased LVT transistors are used in the FF. Because a counter implemented in TSPC FFs is the most power efficient solution, and low power consumption is the main goal of this thesis, it is decided to still use these TSPC counters. Because of the limited bandwidth of these TSPC FFs, there is a deviation from the goal by using a fixed input frequency of 5GHz, and dividing this down to 2,5GHz, 1,25GHz, 625MHz, 312,5MHz, 165MHz and 78MHz. This way the desired division from 5GHz to 2,5GHz is implemented, and the rest of the range of the range of division ratios is present to compare the developed SSPLL frequency divider in the next chapter to.

Chapter 4

Frequency dividing subsampling phase-locked loop

A SSPLL that is typically used for multiplying frequencies is adopted for dividing them. To understand this topology first the fundamental Phase Locked-Loop (PLL) is explained in section 4.1. Then the derivative of the PLL, the SSPLL, is explained in section 4.2. Both the PLL and SSPLL are normally used as frequency multipliers. However with some simple modifications in the Charge Pump (CP)/Phase Frequency Detector (PFD), a frequency divider is obtained from the SSPLL topology. This topology is chosen because a dividing SSPLL has the advantage that the power consumption does not expand that much with increasing the division ratio compared to a counter.

Implementing this topology involves a system and circuit design, which are discussed in section 4.3. Results of this implemented SSPLL divider are in section 4.4. Finally some conclusions are given in section 4.5.

4.1 Phase-locked loop

A PLL is a control system that consist of a PFD/CP, Low Pass filter (LPF), Voltage Controlled Oscillator (VCO) and frequency divider. Such a system is shown in Figure 4.1. The PFD compares the phase of the input signal (IN) with that of the divided output signal, and outputs an error signal. This error signal is converted to a current (i_{cp}) with the CP. The current is integrated by the LPF, resulting in a voltage that is fed to the input of the VCO. This voltage steers the frequency of the VCO. If the PLL is in lock, the phase difference between the input and divided output signal are constant, but not necessarily zero. This way a frequency multiplier is realized where $f_{out} = Nf_{in}$.

First the phase domain model, and the effect that changing parameters in this model has on the output phase noise of the PLL is explained, to provide a means for analyzing the behavior of the developed SSPLL. Finally an explanation about the Phase Detector (PD) in the PLL is given, because this subsystem in the PLL is adopted to form a SSPLL. The PD is again changed to obtain a dividing SSPLL from a frequency multiplying SSPLL.

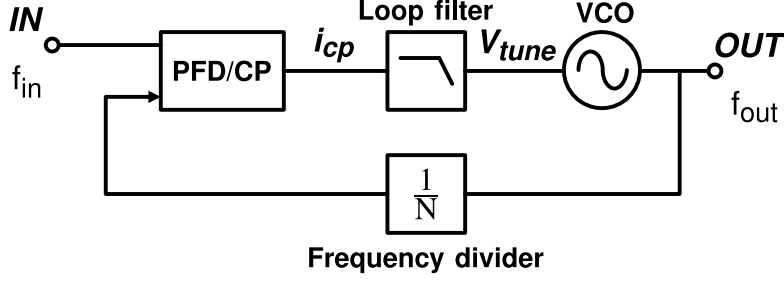


Figure 4.1: An integer-N frequency multiplying PLL.

Phase domain model

To analyze the stability and phase noise of the PLL, it is useful to construct a model where all the signals are defined in phase as quantity; the phase domain model. The noise contributions originating from the PLL can be modeled by adding noise sources in this phase domain model, as shown in Figure 4.2. The PFD is modeled by a subtractor and a PFD gain. It is modeled as a subtraction point because it compares the phase of IN and OUT , and outputs the phase difference ($\Delta\varphi = \varphi_{in} - \varphi_{div}$). This phase difference is converted to a average output current ($\overline{i_{cp}}$) with a certain gain $K_{PFD} = \frac{\Delta\overline{i_{cp}}}{\Delta\varphi}$. The LPF integrates this current, resulting in a tuning voltage (V_{tune}). This integration can be modeled by an impedance Z_{LP} . The VCO transfers V_{tune} , which is dependent on integrated phase difference, to an output frequency with a gain of K_{VCO} . The VCO can be seen as a phase integrator, because phase is integrated frequency ($\varphi = \int f \cdot dt$). The frequency divider also functions as a phase divider in the phase domain model.

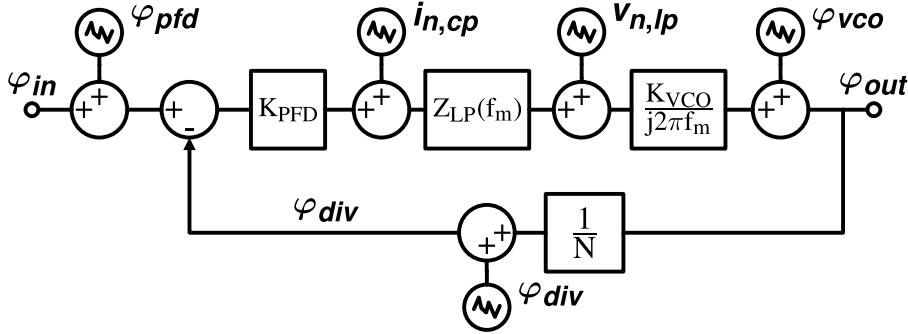


Figure 4.2: The phase domain model with noise sources of a PLL [15, p. 273].

With a type-I PLL there is one integrator with one pole at the origin in the circuit, and with a type-II there are 2. Only type-II PLLs are discussed in this thesis.

For a type-II PLL, the LPF and VCO function as second integrator in the

PLL. Given that the input of the LPF is a current, this LPF consist of a capacitance. This means that there are two integrators, and two poles at the origin in the PLL, rendering the system instable. Therefore an additional zero is placed in the PLL as shown in Figure 4.3. This can be done by adding a resistor in series to the capacitor in the LPF. This zero is located at $f_{zero} = \frac{1}{2\pi R_1 C_1}$

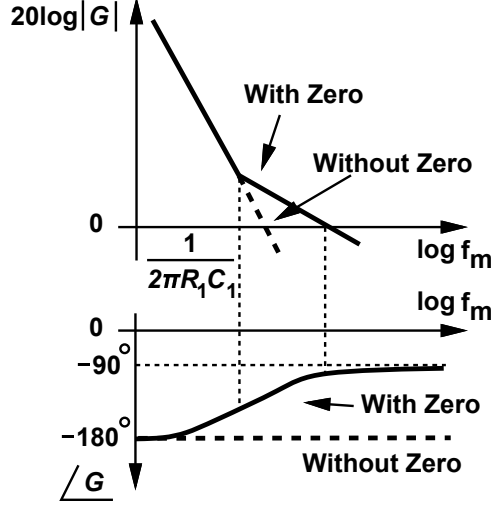


Figure 4.3: Bode plots of the open loop gain (G) of the PLL with and without a zero [4, p. 620].

Noise shaping

All individual blocks in the PLL contribute noise to the output at the place they are located in the PLL. Placing the noise of the PFD ($\varphi_{pfd}(f_m)$) in front of the positive terminal, as done in the model, or behind the output of the subtractor does not matter for the output noise. The CP adds current noise ($i_{n,cp}(f_m)$) at the output of the PFD/CP. The LPF adds voltage noise ($v_{n,lp}(f_m)$) at its output. The VCO produces phase noise ($\varphi_{vco}(f_m)$) at the output of the VCO, while the frequency divider also adds phase noise ($\varphi_{div}(f_m)$) to its output. In Leenaerts[15, p. 273] there is also a noise source of an input (or reference) divider, but an input divider is disregarded in this thesis.

The output phase noise ($\varphi_{out}(f_m)$) consist of a high ($\varphi_{out,hp}(f_m)$) and low ($\varphi_{out,lp}(f_m)$) pass filtered part. These noise contributions can be added:

$$\varphi_{out}^2(f_m) = \varphi_{out,lp}^2(f_m) + \varphi_{out,hp}^2(f_m) \quad (4.1)$$

The scaled CP current noise, the phase noise of the frequency divider and of the phase of the PFD combined are called loop noise. This loop noise together with the input ($\varphi_{in}(f_m)$) noise are low pass filtered with the closed loop PLL transfer function ($H(f_m)$)[15, p. 274]:

$$\varphi_{out,lp}^2(f_m) = N^2 |H(f_m)|^2 (\varphi_{div}^2(f_m) + \varphi_{pfd}^2(f_m) + \frac{i_{n,cp}^2(f_m)}{K_{PFD}^2} + \varphi_{in}^2(f_m)) \quad (4.2)$$

The phase noise of the VCO and the scaled LPF voltage noise are high pass filtered[15, p. 277]:

$$\varphi_{out, hp}^2(f_m) = |1 - H(f_m)|^2 (\varphi_{vco}^2(f_m) + v_{n, lp}^2(f_m) \frac{K_{VCO}^2}{f_m^2}) \quad (4.3)$$

The closed loop PLL transfer function is[15, p. 263]:

$$H(f_m) = \frac{G(f_m)}{1 + G(f_m)} = \frac{K_{PFD} Z_{LP} K_{VCO} / (j f_m N)}{1 + K_{PFD} Z_{LP} K_{VCO} / (j f_m N)} = \frac{K_{PFD} Z_{LP} K_{VCO}}{j f_m N + K_{PFD} Z_{LP} K_{VCO}} \quad (4.4)$$

The open loop transfer function of the PLL is $G(s) = \frac{K_{PFD} Z_{LP} K_{VCO}}{j f_m N}$.

For attenuating more VCO and LPF noise, the PLL closed loop bandwidth $f_c \approx \frac{K_{PFD} K_{VCO} R_1}{2\pi N}$ should be increased. To attenuate more loop noise, the loop bandwidth should be decreased, as shown in Figure 4.4. The contributions of the shaped noise sources are shown in this figure. The optimal loop bandwidth is where the spectrum of the VCO and loop noise intersect[16].

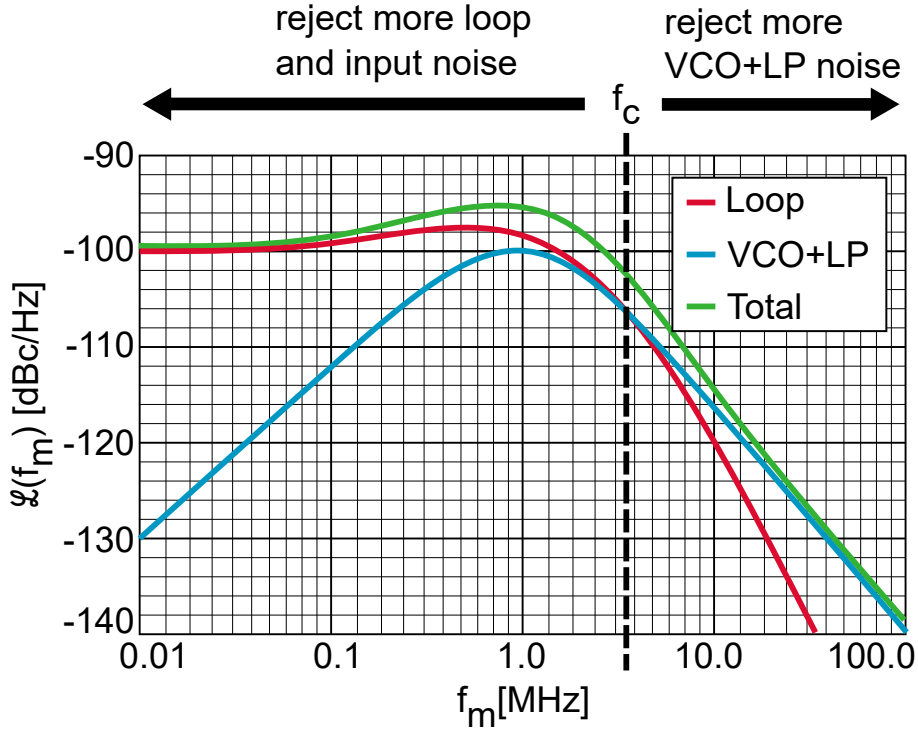


Figure 4.4: Single Side Band (SSB) phase noise power density illustration depicting the contribution from the different shaped phase noise sources[15, p. 278].

Phase frequency detector

When looking at a PLL and $\overline{i_{cp}}$ is positive, V_{tune} increases, the frequency increases, and the phase difference ($\Delta\varphi$) decreases. If $\overline{i_{cp}}$ is negative, V_{tune}

decreases, the frequency decreases, and the phase difference increases. So when there is a phase difference present, this difference becomes larger if the resulting $\overline{i_{cp}}$ is negative and smaller if it is positive. When plotting $\overline{i_{cp}}$ versus $\Delta\varphi$, the phase always travels left when $\overline{i_{cp}}$ is positive and right if it is negative. To always arrive at $\Delta\varphi = 0$, $\overline{i_{cp}}$ has to have the same sign as $\Delta\varphi$ in the characteristic. When this is true, a PFD is realized. A characteristic of such a detector is shown in the top of Figure 4.5. The locking point in this figure is shown with a dot, and the path towards this locking point with arrows.

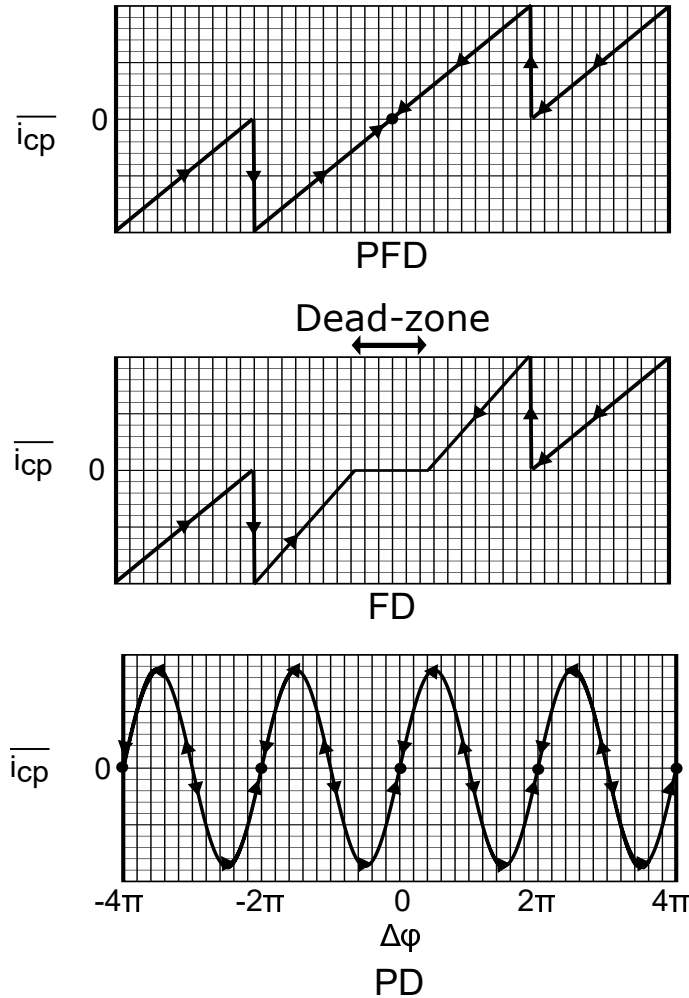


Figure 4.5: The characteristic of a PFD, Frequency Detector (FD), and PD.

Sometimes around $\Delta\varphi = 0$ there is a zone where no $\overline{i_{cp}}$ can be produced, called a dead-zone. $\Delta\varphi$ can change in this zone without affecting $\overline{i_{cp}}$, so the phase of the in and output are not locked within this zone, and the detector has no locking point. If f_{out} and f_{in} are unequal, the phase difference between the in and output increases or decreases every period. This means that the phase difference becomes very large after an amount of periods, and the detector with

the dead-zone can detect this large difference. So it can tune the loop to the correct frequency. Therefore this detector is called a **FD**, and a loop using this detector is called a **Frequency Locked-Loop (FLL)**. A characteristic of such a **FD** is shown in the middle of Figure 4.5.

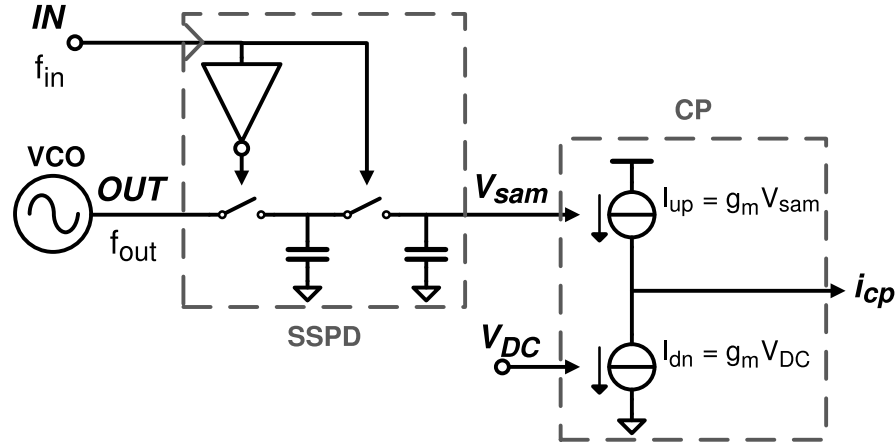
If the characteristic is periodic, there is a locking point with $\overline{i_{cp}} = 0$ every positive or negative integer 2π distance from the origin. This means the loop will not change the frequency if $\Delta\varphi$ is a positive or negative integer of 2π away from the origin. When f_{out} is an integer multiple of f_{in} , or f_{in} is an integer multiple of f_{out} , an integer periods are passed on one signal while 1 is passed on the other, and $\Delta\varphi$ increases a negative or positive integer times 2π every period of one of the signals. This means that $\overline{i_{cp}}$ can be zero, while f_{out} is an integer multiple of f_{in} or f_{in} is an integer multiple of f_{out} . The loop can still compensate for phase with the periodic transfer, because phase disturbance will results in phase that differs from a negative or positive integer 2π distance from the origin. This means the loop can lock phase, but not frequency. The detector in such a loop is called a **PD**. The characteristic of such a detector is shown in the bottom of Figure 4.5. If a loop tracks phase with either a **PFD** or a **PD** it is called a **PLL**.

4.2 Subsampling phase-locked loop

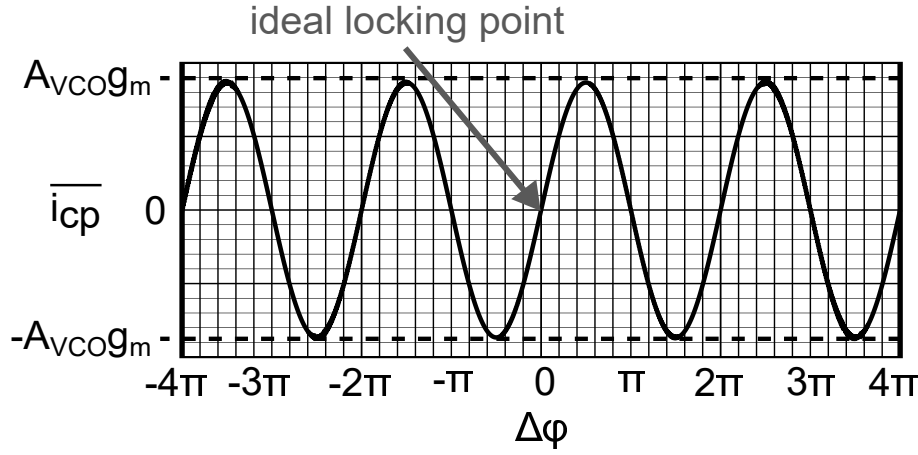
The low pass filtered noise (eq. 4.2) is multiplied by the squared division ratio (N^2), and the frequency divider also adds noise and power consumption to the **PLL**. Therefore it is useful to remove the need for a frequency divider. This could be done by using a **PLL** incorporating a **PD** instead of **PFD**, because with this **PD** the **PLL** can lock to an integer division or multiple of the input signal. A **Subsampling Phase Detector (SSPD)** is used because this is a relatively simple circuit, hence the name **SSPLL**. Such a **SSPD** with **CP** is shown in Figure 4.6a, with V_{DC} the DC voltage of *OUT*. With this detector, the **PLL** can lock to any integer multiple of f_{in} . This is because the characteristic of the **SSPD** is periodic, with zero $\overline{i_{cp}}$ every multiple of $\pi \Delta\varphi$. The characteristic of a **SSPD** is shown in Figure 4.6b.

The control signal of the **VCO** has to be steered to the right range in order for the outputs frequency to be within an integer multiple or division distance from the desired frequency. This can be done using a **FLL** with a dead-zone around $-\pi \leq \Delta\varphi \leq \pi$, the range that the **SSPLL** is able to lock phase. When the phase difference is such that the **FLL** operates in the dead-zone, it can be turned off to save power. A **SSPLL** with a **FLL** is shown in Figure 4.7.

For analyzing the noise of the **SSPLL**, it is assumed that the **FLL** does not contribute to the output noise, as the **FLL** is only initially used to steer the loop to the right frequency and it is not used to lock phase. For examining the phase noise, the phase domain model is determined and shown in Figure 4.8. This model is the same as the model of the regular **PLL**, however there is no divider in the feedback path and there is a 'virtual' frequency multiplier in front of the subtracter. This 'virtual' frequency multiplier originates from the sub-sampling process. The frequency of the alias that appears at the input of the **VCO** is $f_{alias} = f_{VCO} - N \cdot f_{ref}$, because that is the only alias that is not filtered away by the **LPF**. Therefore the sub-sampling process works as if the **VCO** is sampled by a signal with frequency N times higher than the input[17].



(a) The SSPD and CP diagram[17].



(b) The characteristic of the SSPD and CP[17].

Figure 4.6

This means the low pass filtered noise becomes:

$$\varphi_{out,lp}^2(f_m) = |H(f_m)|^2(\varphi_{sspd}^2(f_m) + \frac{i_{n,cp}^2(f_m)}{K_{PFD}^2} + N^2\varphi_{in}^2(f_m)) \quad (4.5)$$

With a closed loop gain of:

$$H(f_m) = \frac{G(f_m)}{1 + G(f_m)} = \frac{K_{SSPD}Z_{LP}K_{VCO}/(jf_m)}{1 + K_{SSPD}Z_{LP}K_{VCO}/(jf_m)} = \frac{K_{SSPD}Z_{LP}K_{VCO}}{jf_m + K_{SSPD}Z_{LP}K_{VCO}} \quad (4.6)$$

The high pass filtered phase noise ($\varphi_{out,hp}^2(f_m)$) is still the same as with a regular PLL.

It is also good to notice that the gain of the SSPD and CP (K_{SSPD}) is the slope of the characteristic (Figure 4.6b) at $\Delta\varphi = 0$. This characteristic is the

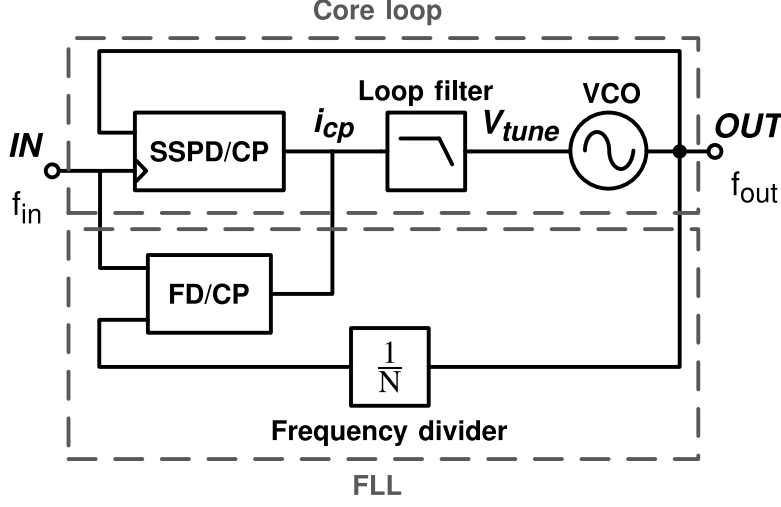


Figure 4.7: A diagram of a SSPLL frequency multiplier.

output voltage, with amplitude A_{OUT} , times the transconductance of the CP (g_m)[17]:

$$K_{SSPD} = \frac{\Delta \bar{i}_{cp}}{\Delta \varphi} = \frac{A_{OUT} \sin(\Delta \varphi) \cdot g_m}{\Delta \varphi} \approx A_{OUT} g_m \quad (4.7)$$

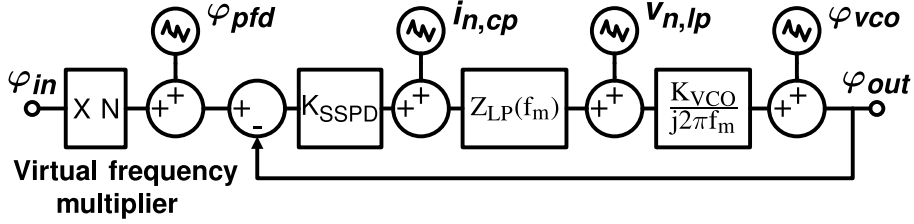


Figure 4.8: The phase domain model of the SSPLL[17].

4.3 Subsampling phase-locked loop divider design

To attain a frequency dividing SSPLL, first a system level design is done, in which the basic concept of the frequency dividing SSPLL is explained. In this system level design a SSPD with CP and a VCO are present. These two blocks are designed in the circuit level design subsection for minimum power. When the specifications of these blocks are determined, the SSPLL is tuned for minimal output noise by designing the LPF.

System level design

As explained in the previous section the output of the SSPLL is sub-sampled at the rate of the input frequency. This creates a periodic characteristic, which

has the effect that the input frequency is locked to an integer multiple of the output frequency.

If OUT and IN are exchanged in the $SSPD$, an integer periods on IN can pass in one sampling period of OUT , without a change of $\overline{i_{cp}}$, due to the periodic characteristic. This has the effect that f_{out} can lock to an integer division of f_{in} , and a frequency divider can be constructed without the use of a counter. An illustration of the frequency dividing $SSPLL$ is shown in Figure 4.9. This figure shows a $SSPLL$ without a FLL . This lack of a FLL could be solved by using an initial voltage on V_{tune} while simulating.

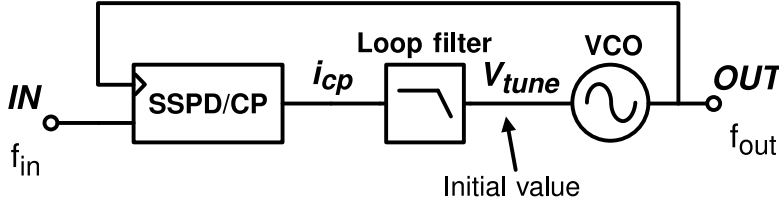


Figure 4.9: The diagram of the $SSPLL$ divider.

The phase domain model of the $SSPLL$ frequency divider is the same as the regular $SSPLL$, with the exception of a virtual frequency divider instead of a virtual frequency multiplier. This phase domain model is shown in Figure 4.10. This divider is present due to the fact that if noise at a higher frequency is transferred to a lower frequency, the phase noise becomes lower [14]. This means the low pass filtered noise becomes:

$$\varphi_{out,lp}^2(f_m) = |H(f_m)|^2(\varphi_{sspd}^2(f_m) + \frac{i_{n,cp}^2(f_m)}{K_{PFD}^2} + \frac{1}{N^2}\varphi_{in}^2(f_m)) \quad (4.8)$$

With a closed loop gain of:

$$H(f_m) = \frac{G(f_m)}{1 + G(f_m)} = \frac{K_{SSPD}Z_{LP}K_{VCO}/(jf_m)}{1 + K_{SSPD}Z_{LP}K_{VCO}/(jf_m)} = \frac{K_{SSPD}Z_{LP}K_{VCO}}{jf_m + K_{SSPD}Z_{LP}K_{VCO}} \quad (4.9)$$

The high pass filtered phase noise ($\varphi_{out,hp}^2(f_m)$) is still the same as with a regular PLL . The gain of the $SSPD$ and CP is [17]:

$$K_{SSPD} = \frac{\Delta \overline{i_{cp}}}{\Delta \varphi} = \frac{A_{IN} \sin(\Delta \varphi) \cdot g_m}{\Delta \varphi} \approx A_{IN} g_m \quad (4.10)$$

This is again the slope of the characteristic at $\Delta \varphi = 0$. This characteristic is the input voltage, with amplitude A_{IN} , times the transconductance of the CP (g_m).

Circuit level design

In the $SSPLL$ divider two subsystems are present, the VCO and the $SSPD/CP$. First an explanation for the choice for the VCO and $SSPD/CP$ type is given. Then some expressions for the relevant specifications of both subsystems are

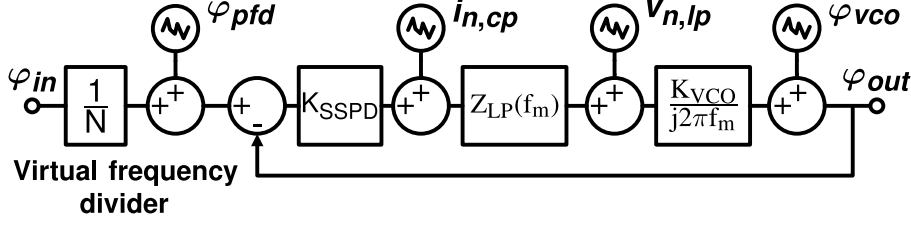


Figure 4.10: The phase domain model of the SSPLL divider.

presented. These expressions then are used to design the subsystems for minimal power. Finally the specifications are simulated using Cadence Virtuoso and Spectre(RF).

Voltage controlled oscillator

A LC, single ended ring or differential Ring based VCO can be used. An estimation of the power of a LC, single ended ring, and differential ring oscillator can be found in Appendix C. A single ended ring oscillator consumes much less ($\frac{1}{80}$) minimal power than a LC-based oscillator. A differential ring oscillator would also consume more ($\frac{4}{3}$) power than a single ended ring oscillator, even $\frac{16}{3}$ more when quadrature output generation is required. Because the counter implemented with TSPC FFs lacks quadrature outputs, it is also omitted in the SSPLL for a fair comparison. More than 3 stages should not increase the dynamic power consumption, however it does increase the leakage current. Therefore it is chosen to use a single ended ring oscillator of 3 stages.

A simple power efficient implementation of a single ended current starved ring oscillator is chosen, and is shown in Figure 4.11. With V_{tune} , a current through N_1 and N_2 (I_{bias}) is controlled. I_{bias} generates a voltage V_P with P_1 , this voltage controls P_2 such that the current through P_2 (I_D) is also I_{bias} . This way the maximal current that can flow through P_3 and P_2 is I_D . This current controls the speed at which the (parasitic) capacitance at the in and outputs of P_3 and N_3 are charged and discharged, and thus controls the oscillation frequency. The capacitance C is added representing the input capacitance of the SSPD/CP, and is estimated to be 400aF. For all the transistors to stay in strong inversion and saturation, V_{tune} and V_{dd} have the following voltage ranges:

$$V_{th} < V_{tune} < \frac{V_{dd} + V_{th}}{2} \quad (4.11)$$

The output frequency of such a VCO is approximated as[18]:

$$f_{out} \approx \frac{\mu_n W_{N2} C_{ox} (V_{tune} - V_{th})^2}{12\eta L_{N2} q_{max}} \quad (4.12)$$

The tuning constant ($K_{VCO}[\frac{Hz}{V}]$) is:

$$K_{VCO} = \frac{\partial f_{out}}{\partial V_{tune}} \approx \frac{\mu_n W_{N2} C_{ox} (V_{tune} - V_{th})}{24\eta L_{N2} q_{max}} \quad (4.13)$$

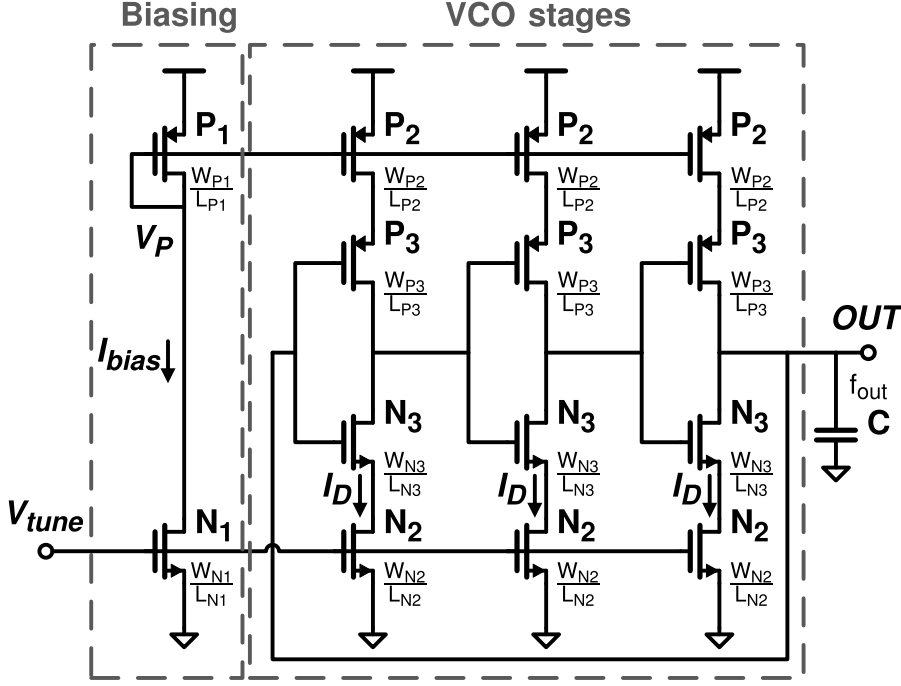


Figure 4.11: The circuit diagram of the VCO.

This shows that K_{VCO} is not constant, but depends on V_{tune} . K_{VCO} is approximated as constant around the bias voltage of V_{tune} , for a simpler analysis. The bias circuit has a power consumption of $I_{bias}V_{DD}$, and the VCO stages have $2\eta CV_{DD}^2 f_{out}$ [18] power consumption. This is not only power consumption due to charging and discharging output capacitance, but also power consumption due to crowbar currents. This contribution is approximately equal to the charging power [18]. The total power consumption, with neglectable bias current, is then:

$$P = 2K\eta q_{max} f_{out} + I_{bias}V_{DD} \approx 6\eta q_{max} V_{dd} f_{out} \quad (4.14)$$

The output phase noise of a ring oscillator is [18]:

$$\mathcal{L}(f_m) = \frac{8}{3\eta} \left(\frac{f_{out}}{f_m} \right)^2 \frac{kT}{P} \frac{\gamma V_{dd}}{(V_{tune} - V_{th})} \quad (4.15)$$

This phase noise does assume dominant noise of the VCO stages over the bias circuit, and it assumes long channel transistors. K is the number of VCO stages, and η is the proportionality factor and is close to 1. This factor is further explained in Appendix C. The maximum charge is the output capacitance times the voltage swing (V_{swing}) on this capacitance; $q_{max} = CV_{swing}$.

As a starting point all NMOS are minimum sized, and the PMOS are sized wider to balance the driving strength of the PMOS and NMOS: ($\frac{W_{N1}}{L_{N1}} = \frac{W_{N2}}{L_{N2}} = \frac{W_{N3}}{L_{N3}} = \frac{120nm}{60nm}$, $\frac{W_{P1}}{L_{P1}} = \frac{W_{P2}}{L_{P2}} = \frac{W_{P3}}{L_{P3}} = \frac{500nm}{60nm}$).

To allow lower supply voltage, and thus lower power, LVT transistors are used in the VCO.

I_{bias} can be scaled down for lower power, while I_D (hence f_{out}) stays the same by increasing L_{N1} and L_{P1} . However this is not done due to the increase in capacitance on V_{tune} . Increasing this capacitance introduces stability problems later on when tuning the SSPLL.

V_{dd} is scaled down to a minimum with some margin (700mV) for lower power. V_{tune} also has to scale down with V_{dd} for all the transistors to stay in the correct operating region. For maximum bandwidth, V_{tune} is biased at almost at the maximum of its range (4.11), with some room for voltage swing on top of the bias voltage. This is beneficial for the bandwidth, because with higher V_{tune} , W_{P2} and W_{N2} can be smaller for the same I_D and f_{out} , and there is less parasitic capacitance. With $V_{th} = 0.25V$, (4.11) then becomes $0.25V < V_{tune} < 0.475V$, so V_{tune} is biased around 400mV.

To obtain the proper output frequency of 2.5GHz, I_D is scaled up by scaling W_{P2} and W_{N2} up, while keeping L_{P2} and L_{N2} the same. This also gives (approximately) the same parasitic input capacitance. Even if scaling up W_{P2} and W_{N2} would give significantly more output capacitance, this results in scaling the power scale linearly up. Scaling V_{dd} down however results in scaling the power quadratically down, so it is still profitable to scale V_{dd} down. A lower power however does give a higher phase noise. W_{P2} is scaled up to $5.5\mu m$ and W_{N2} to $1.8\mu m$. This scaling is determined with simulation.

P_3 and N_3 are kept minimum size to give less input and output capacitance of the VCO stages, and thus less power consumption. A circuit diagram with all the $\frac{W}{L}$ ratios and voltages notated is shown in Figure 4.12.

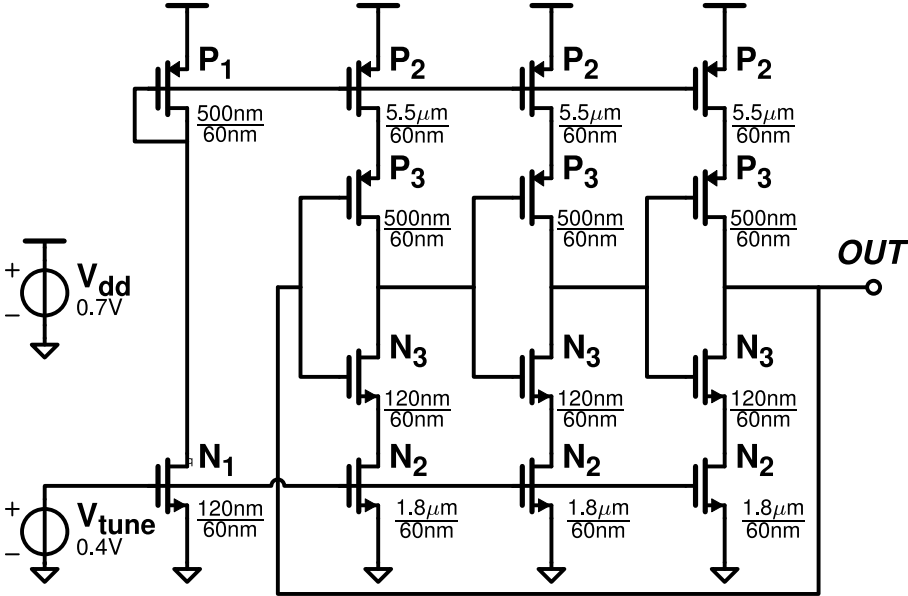


Figure 4.12: The circuit diagram of the VCO with $\frac{W}{L}$ ratios and voltages.

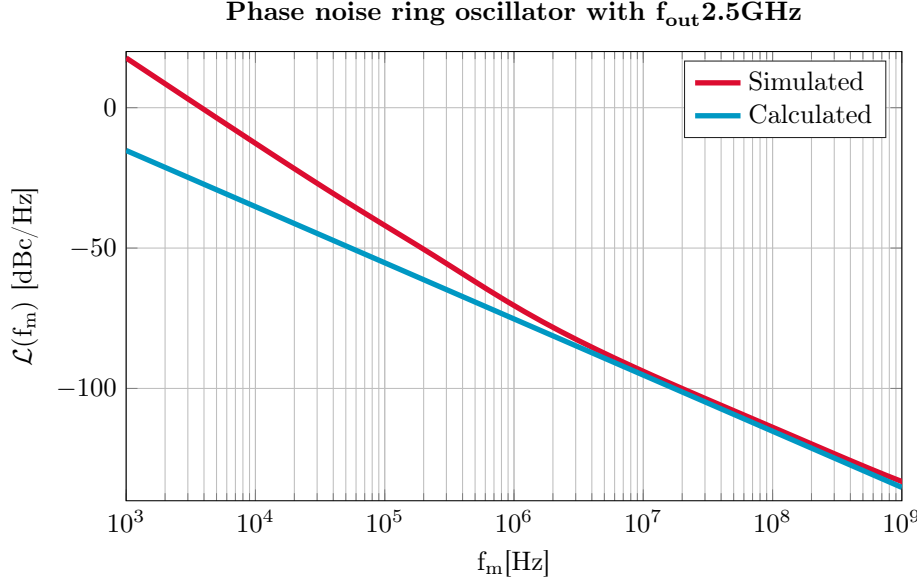


Figure 4.13: The estimated using (4.15) and simulated power consumption of the VCO.

Simulating this VCO gives a power consumption of $4.5\mu W$, from which $1.7\mu W$ originates from the biasing circuit. This means that the power consumption of the VCO stages dominate the power consumption. Calculating the power gives $3\mu W$, with $\eta = 1$, $q_{max} = CV \approx 400aF * V_{dd}$ (capacitance from section B.2), $V_{dd} = 700mV$, and $f_{out} = 2.5GHz$, so this is very close to the simulated value.

The estimated phase noise with (4.15) and the simulation are both plotted in Figure 4.13. This phase noise is estimated with $kT = 4 \cdot 10^{-21} \frac{m^2 kg}{s^2}$, $P = 4.5\mu W$, $\gamma = 1$, $V_{tune} = 0.4V$ and $V_{th} = 250mV$ (obtained from Cadence). In this figure the phase noise goes below $-105dBc/Hz$ around $40MHz$. This means that f_c of the SSPLL has to be higher than $40MHz$ to achieve the BLE specification stated in section 1.2. This also limits the sizing of $L_{N1} = L_{N2}$, W_{N2} and W_{P2} as the bandwidth from V_{tune} to I_D should not dominate the bandwidth of the SSPLL. The simulated phase noise below $f_m = 1MHz$ is larger than the estimated one as (4.15) does not consider $\frac{1}{f^3}$ flicker noise, only $\frac{1}{f^2}$ current noise. The simulated phase noise clearly shows $\frac{1}{f^3}$ noise with a $\frac{-30dB}{decade}$ roll-off. The flicker noise corner frequency f_C is around $5MHz$, where the phase noise continues as $\frac{1}{f^2}$ phase noise. Below $40KHz$ the phase noise is positive, that means there is more noise than signal on the output. This has to do with the phase noise model used. In this model the $\frac{1}{f^3}$ flicker noise continues to increase as it approaches the carrier, even if it becomes more than the carrier itself.

The tuning constant is obtained by simulating f_{out} over a range of $380mV - 420mV$ of V_{tune} , as shown in Figure 4.14. At the biasing point ($400mV$), $f_{out} = 2.5GHz$ and $K_{VCO} = 16 \frac{GHz}{V}$. The calculated values are $f_{out} = 25GHz$

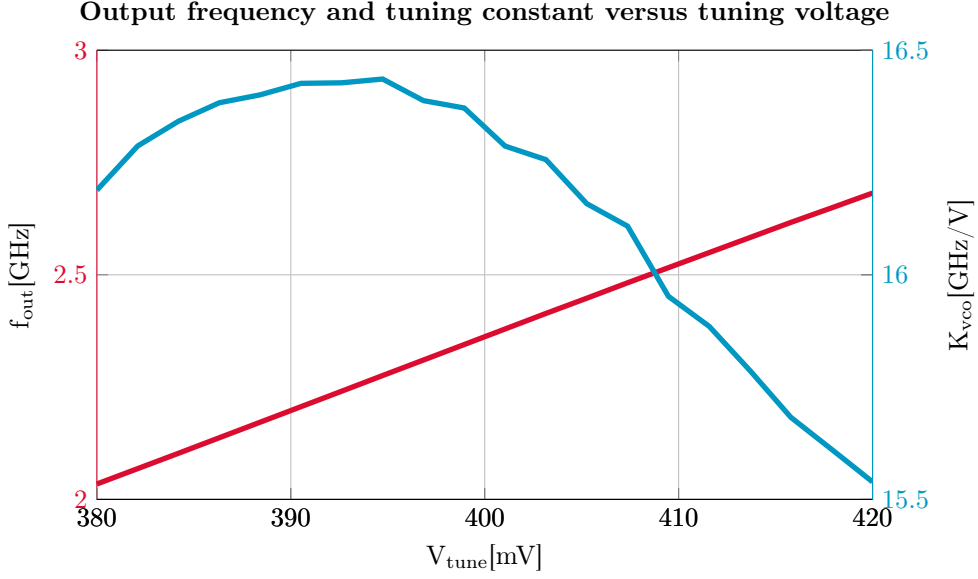


Figure 4.14: The tuning constant (K_{VCO}) and the output frequency (f_{out}) over the tuning voltage

(4.12) and $K_{VCO} = 2.7GHz/V$ (4.13) at this bias point. The tuning constant should increase with V_{tune} according to (4.13), and f_{out} should quadratically increase with V_{tune} according to (4.12). However f_{out} increases linearly and K_{VCO} goes quadratically with V_{tune} . This is probably due to the fact that P_2 and N_2 , with the sizing used, does not limit I_D , but the P_3 and N_3 do. Therefore (4.13) and (4.12) are not accurate anymore. Nonetheless (4.12) and (4.13) do give a useful indication for scaling.

Charge pump

Due to time constraint, the CP/SSPD used in Gao et al.[17] is adopted for this thesis, this CP is shown in Figure 4.15. The adjusted CP/SSPD is shown in Figure 4.16. Note that Gao et al.[17] uses Ref instead of IN , and VCO instead of OUT . To give a SSPLL dividing behavior, the differential VCO signals $VCOP$ and $VCOP$ are replaced by differential input signals IN_P and IN_N , and Ref is replaced by OUT . All the cascode stages, attached to nodes voltages V_{BN} and V_{BP} , in the CP are removed, because of the used low supply voltage. This will also reduce power due the absence of a bias circuits to determine V_{BN} and V_{BP} . Removing the cascodes will however result in a CP with lower output impedance, and thus inferior gain and linearity.

The CP also uses a pulser to duty cycle the CP. This duty cycling means that only a fraction of the time a current is outputted. This means K_{SSPD} (4.7) can be lowered without changing A_{OUT} and g_m . The pulser is also present to form a sample and hold, which would otherwise require a second track-and-hold circuit. The pulser circuit is omitted due to its power consumption. This means K_{SSPD} can only be lowered with A_{IN} and g_m of the CP, and only a

track and hold circuit is present. This should not be a problem, because the tracked input signal should be attenuated by the loop filter. This does require that f_c is sufficiently lower than f_{IN} .

There are no buffers present at IN , but there is one inverter present at OUT . Removing the input buffers is done to give a fair comparison with the TSPC counter divider, that also contains no buffers. To achieve a bandwidth that is great enough to prevent the CP from limiting the bandwidth of the SSPLL, transistor N_1 becomes large. This presents a large capacitive load to the VCO, which may prevent the VCO from running at frequency f_{out} . Therefore a single inverter is placed at the In , which presents a significantly smaller load to the VCO.

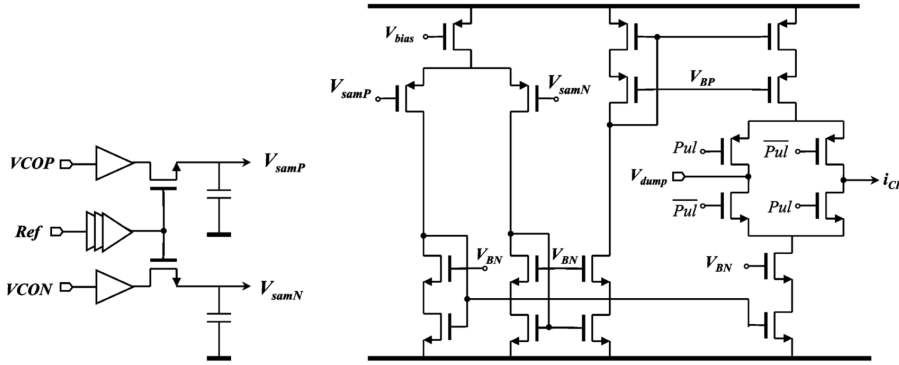


Figure 4.15: The CP/SSPD used in [17].

The CP is shown in Figure 4.16. The differential input (IN_P and IN_N) is tracked & hold by the SSPD, producing a differential tracked output $V_{sam} = V_{samP} = V_{samN}$. This voltage is transformed to current with P_2 , and back to a differential voltage $V_L = V_{LP} - V_{LN}$ with N_2 in the Differential amplifier. This differential voltage is then transformed to a single ended current in the Balanced to Unbalanced (Balun) stage.

For the sampling capacitance (C_{sam}), a reasonable value of 10fF is chosen. This is done to keep N_1 small while achieving the loop bandwidth, which means a smaller load for the inverter. With this value of C_{sam} , the $\frac{kT}{C_{sam}}$ noise of the SSPD does not dominate $\varphi_{out,lp}(f_m)$ (4.5). Transistor N_1 is sized $\frac{W_{N1}}{L_{N1}} = \frac{600nm}{60nm}$. The inverter is one minimum sized inverter implemented with LVT transistors, for a smaller load of the VCO, and less power consumption of the inverter itself. This single inverter is adequate for driving the sampling switches N_1 .

To reject the effect of common mode excitement of IN_P and IN_N , the transconductance of the current through P_4 (i_P) divided by IN_P , and of the current through N_4 (i_N) divided by IN_N should be equal. Otherwise common mode excitement of IN_P and IN_N would cause a resulting current i_{cp} . This means $\overline{i_{cp}} \neq 0$ while $\Delta\varphi = 0$, and the loop does not operate in the ideal locking point (Figure 4.6b) of $\overline{i_{cp}} = 0$ and $\Delta\varphi = 0$. This would mean that K_{SSPD} is not maximized at $g_m A_{IN}$.

Therefore both sampling switches (N_1), the Differential amplifier's differential pair transistors (P_2), and the Differential amplifier's load transistors

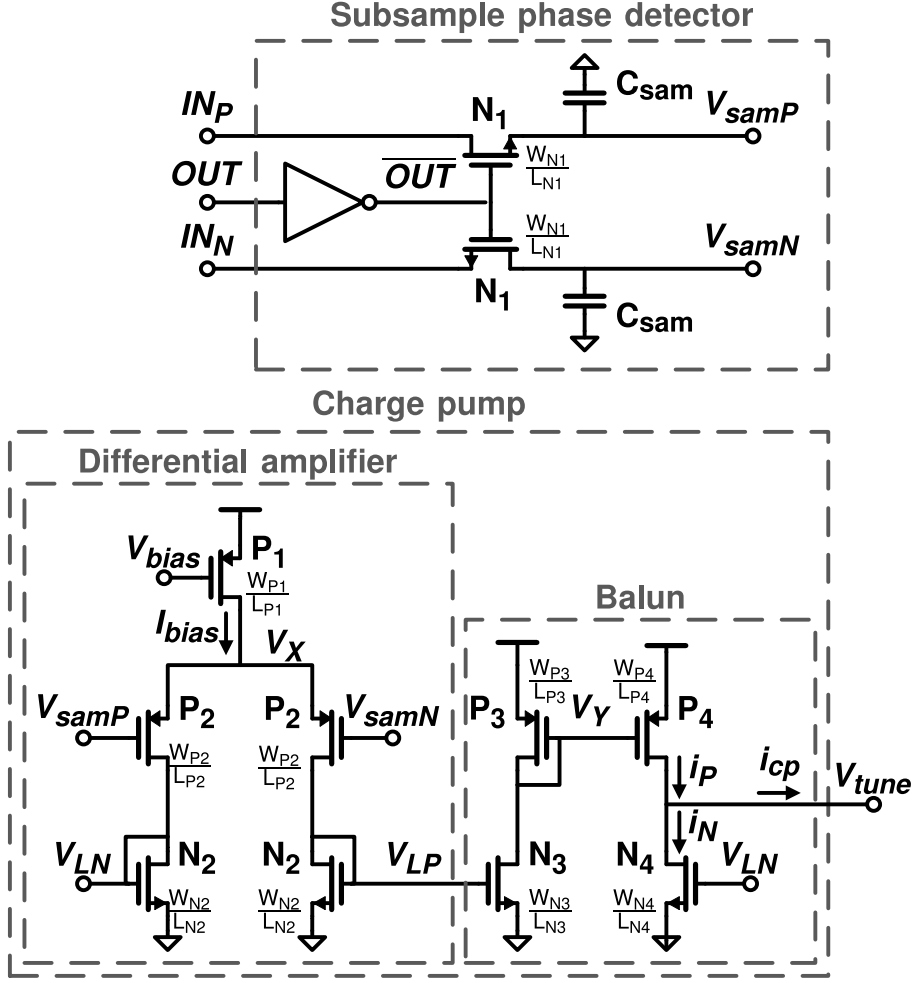


Figure 4.16: The circuit diagram of the CP/SSPD used in this thesis.

N_2 are the same size. The trans-conductances of the Balun $g_{mP} = \frac{i_P}{V_{LP}}$ and $g_{mN} = \frac{i_N}{V_{LN}}$ should also be equal, giving:

$$g_{m,N4} = g_{m,P4} \frac{g_{m,N3}}{g_{m,P3}}$$

$$\frac{\mu_N \frac{W_{N4}}{L_{N4}}}{\mu_P \frac{W_{P4}}{L_{P4}}} = \sqrt{\frac{\mu_N \frac{W_{N3}}{L_{N3}}}{\mu_P \frac{W_{P3}}{L_{P3}}}}$$

This means that if the output stage's P_4 and N_4 are not sized for equal driving strength to bias V_{tune} on the correct bias voltage for the VCO, this can be compensated by sizing of P_3 and N_3 . Doing so drives voltage V_Y up or down. This scaling is therefore limited as voltage V_Y should stay lower than $V_{dd} - V_{th}$ and higher than V_{th} , otherwise transistors in the Balun leave the correct operating region.

The total transconductance of the CP is:

$$g_m = g_{m,N4} \frac{g_{m,P2}}{g_{m,N2}}$$

The bandwidth of the circuit is dominated by the parasitic capacitance on node V_{LP} and V_{LN} caused by transistors N_2 and P_2 and the parasitic capacitances on node V_Y due to transistors P_3 , P_4 and N_3 .

Again as a starting point all transistors are minimum ($\frac{120nm}{60nm}$) sized. First all node voltage in the differential amplifier are determined. Voltage V_{bias} is biased at GND, to prevent the need of a bias circuit. For transistor P_1 a **High Voltage Threshold (HVT)** transistor with the bulk connected to V_{dd} is used. This is done to gain a higher threshold of $600mV$, and to get a lower overdrive voltage ($V_{gs} - V_{th}$) of $100mV$. This means voltage V_X only has to be less than $600mV$, leaving more voltage room to tune the rest of the circuit. Using $\frac{W_{P1}}{L_{P1}} = \frac{120nm}{100nm}$ the tail current is set to a tolerable value of around $1\mu A$.

The voltage $V_{sam,CM}$ is set to $0V$, otherwise the resistance of sampling transistors N_1 becomes too large to achieve the bandwidth of the loop. Voltage V_X is biased approximately $\frac{V_{sam}}{2}$ above the threshold voltage of P_2 by sizing $\frac{W_{P2}}{L_{P2}}$ to $\frac{120nm}{200nm}$. The sizing of P_2 is kept as small as possible, for C_{sam} to dominate the sampling capacitance, not the parasitic capacitance of P_2 , and for the transconductance of the CP (B.1) to be maximal. A differential input voltage of $100mV$ is used, created using an ideal **Balun**. This voltage is $100mV$ in order for V_X to become lower for the gain of the differential amplifier to become higher, and because it is a regular output swing for an oscillator to produce. A **Regular Voltage Threshold (RVT)** transistor is used for P_2 , and its bulk is connected to GND to increase V_{th} to $480mV$. This is done in order for V_X to be higher.

When sizing L_{N2} up for more gain, the size of L_{N3} and L_{N4} also increases due to matching, causing a quadratic increase in parasitic capacitance, but a linear increase in transconductance (B.1). Therefore the maximum size of L_{N2} is determined with simulation, in order for the bandwidth of the CP to fall around $1GHz$, so the CP bandwidth does not dominate the bandwidth of the loop. This ratio $\frac{W_{N2}}{L_{N2}}$ is set to $\frac{120nm}{500nm}$. The bulk of N_2 is connected to GND for higher V_{th} and less transconductance, so more gain at a smaller sizing. **LVT** transistors are used for the rest of the transistors, because of the limited voltage headroom.

The length of transistors P_3 and P_4 are scaled to $100nm$ and not the minimum length for V_{th} to go down. The width of P_3 is scaled minimum size ($120nm$), for minimum capacitance on node V_Y . The width of transistor N_3 is the same as that of transistor N_4 for a symmetrical parasitic capacitance on voltage V_{LP} and V_{LN} . This is scaled $\frac{300nm}{500nm}$, for keeping V_Y on a correct voltage, such that all transistors in the **Balun** stay in the correct operating region. Transistor P_4 is scaled $\frac{180nm}{100nm}$ for voltage V_{tune} to be approximate $400mV$. With this sizing, $g_{m,P4} \frac{g_{m,N3}}{g_{m,P3}} = g_{m,N4}$ is approximately true. A circuit diagram with all the final component values and biasing voltages is shown in Figure 4.17.

First a DC simulation is done to show that all the transistors in the CP are in the correct operating region. This gives the node voltages $V_X = 530mV$, $V_{L,CM} = 350mV$, $V_Y = 275mV$ and $V_{tune} = 420mV$. The calculated **Balun**



SSPLL tuning

The SSPLL is tuned to be stable, and to produce as little output noise as possible. To do this, the phase domain model is used as already mentioned in

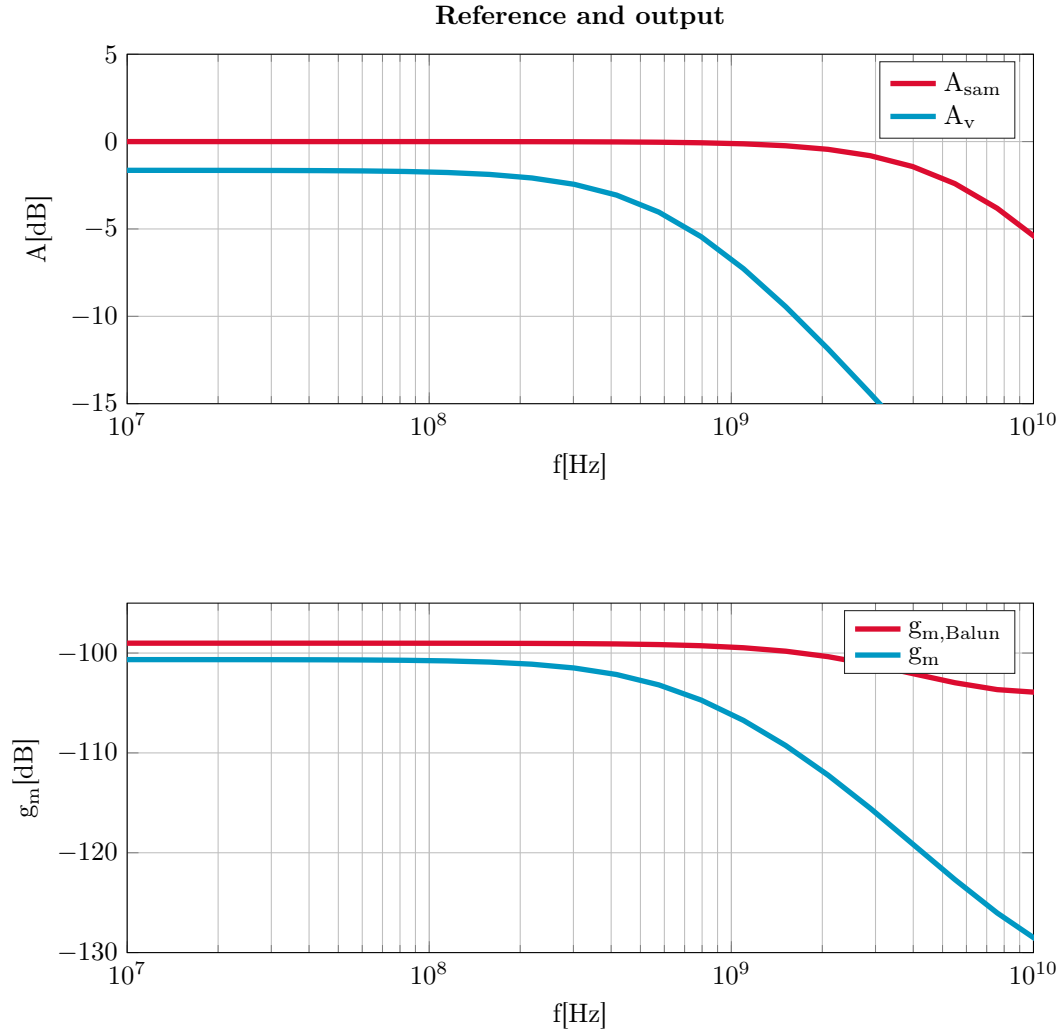


Figure 4.18: The voltage gain of the SSPD (A_{sam}) and differential amplifier (top), and the transconductance of the Balun and CP (bottom).

section 4.1. The transfer functions described using this model, the open loop $G(f_m)$, low pass closed loop $H(f_m)$ and high pass closed loop $(1 - H(f_m))$, are simulated using Matlab. The transconductance of the CP is reconstructed with a transfer function in Matlab. A bode plot of this transfer function is shown in Figure 4.19. This is done to include the bandwidth of this CP in the simulations. This transfer function is used as K_{SSPD} in model. The tuning constant of the VCO (K_{VCO}) that is obtained previously is also used.

To ensure SSPLL stability, the phase of open loop transfer function ($\angle G(f_m)$) has to be less than 180° on the frequency that the magnitude of the open loop function crosses unity ($|G(f_m)| = 0\text{dB}$). This frequency is also the bandwidth of the SSPLL, f_c . Because there are 2 poles in the system, the margin from 180° is achieved by placing a zero before f_c , which shifts $\angle G(f_m)$ back to 90° .

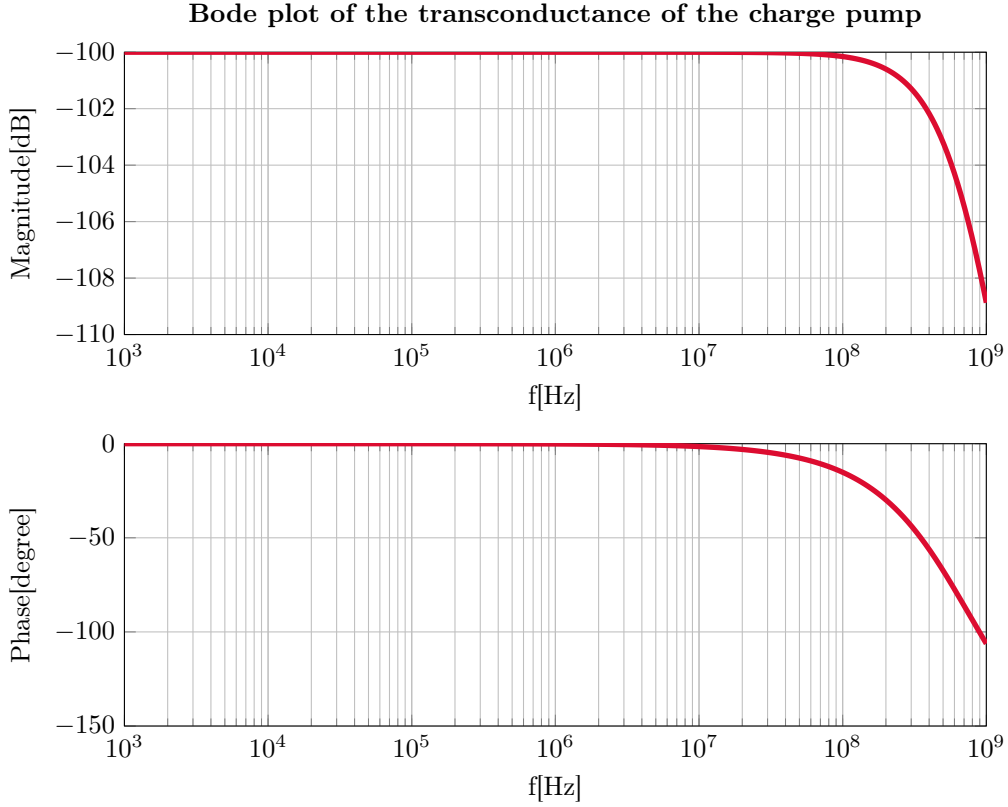


Figure 4.19: A bode plot of the reconstructed transconductance of the CP.

The 2 poles in the transconductance of the CP also give 180° phase shift, so f_{zero} has to be lower than the bandwidth of the CP, while it also has to be lower than f_c .

Increasing f_c can be done by increasing R_1 , while increasing f_{zero} can be done by decreasing the size of C_1 . Because a very low power VCO is used, the VCO is very noisy. Therefore the bandwidth of the SSPLL has to be higher than 40MHz, as already determined while examining the VCO phase noise on page 35. Because an ideal noiseless source is used for IN in the simulation, the loop noise is dominated by the noise of the SSPD/CP. This noise is much lower than the noise of the VCO, and therefore the frequency where the spectra of the VCO and loop noise intersect is high, and it is preferred to have a high f_c . With a non ideal source, it could be beneficial to place f_c lower to suppress more noise from the input, as previously shown in Figure 4.4. The two poles of the CP are situated around 500MHz, so f_c is moved at 120MHz, while the zero is placed at 3MHz to ensure stability. This results in a bode plot of $G(f_m)$ that is shown in Figure 4.20a. This pole and zero placement is achieved with $R_1 = 50K\Omega$ and $C_1 = 1pF$. The bode diagram of $G(f_m)$ shows that there is a phase margin of about 80° . A bode plot of the closed loop low pass transfer function, by which the loop and input noise are filtered, is shown in Figure 4.20b. The VCO noise is filtered by the high pass transfer function,

from which a bode plot shown in Figure 4.20c. An important note to make is that K_{SSPD} changes if the locking point shifts and K_{VCO} varies if V_{tune} changes, therefore these bode plots can change with a different locking point and tuning voltage.

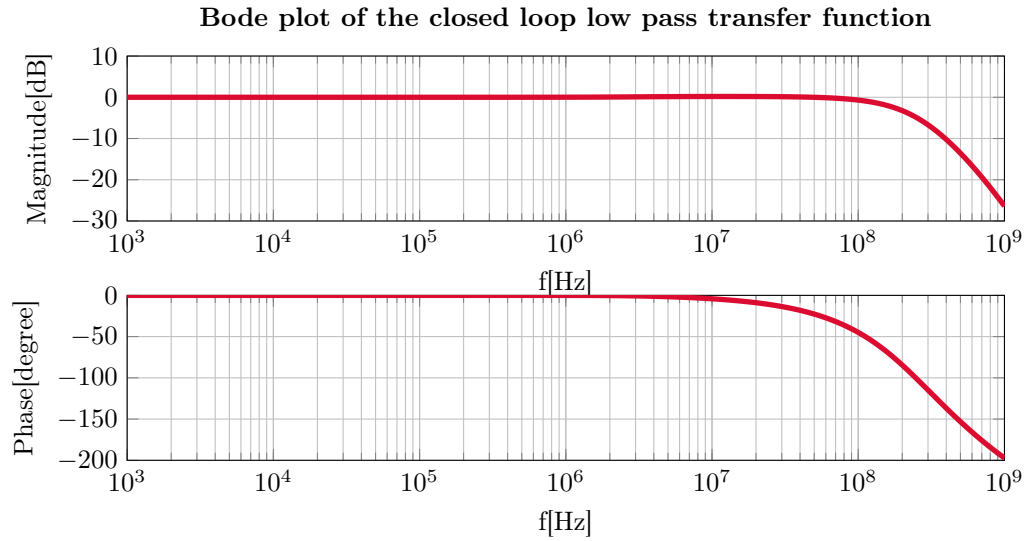
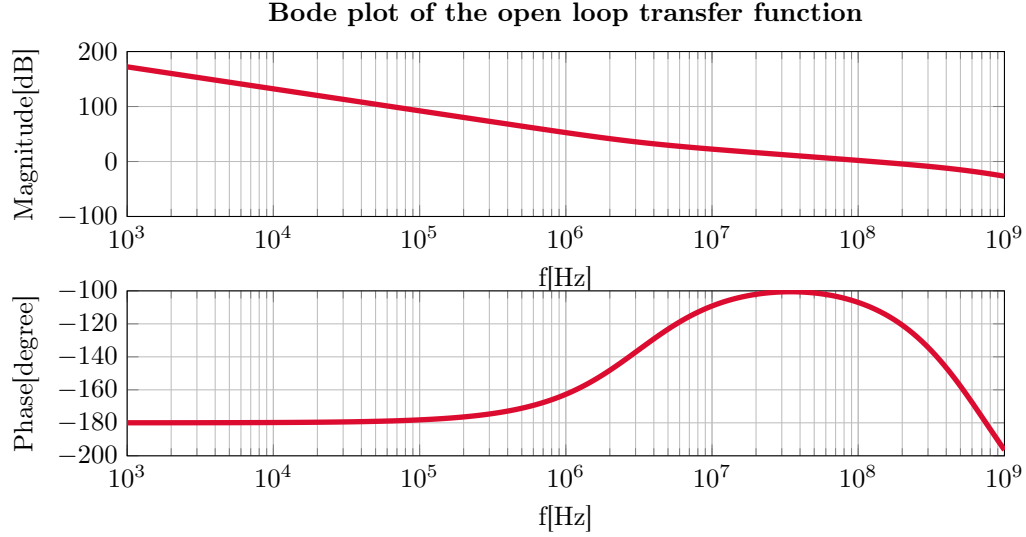
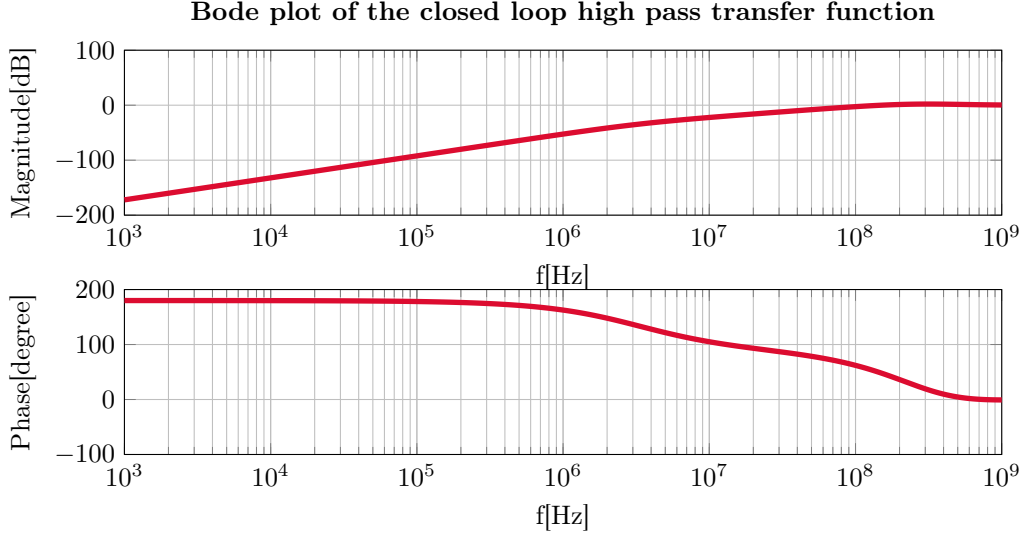


Figure 4.20



(c) A bode plot of the high pass closed loop transfer function.

Figure 4.20

4.4 Subsampling phase-locked loop divider results

Locking test and harmonics

For testing if the SSPLL can lock to the input, the SSPLL is tested by applying a phase jump on the input of half a period of IN . Doing this gives the result of Figure 4.21. After $t = 0$ there is some start up behaviour where V_{tune} settles to a constant value around 410mV. Then around $t = 50ns$ there is a disturbance in the tuning voltage due to a phase shift of half a period in the input signal. The loop settles in 30ns to continue at its previous tuning voltage.

The input and output at the time the loop is settled after startup at $t = 40ns$, and after it is settled after the first phase shift at $t = 150ns$ is shown in Figure 4.22. It demonstrates that the zero crossing of output are at the same time of the zero crossings of the input before and after the phase shift, the phase of the input is however shifted 180° after the phase shift. It also shows that the frequency is correct, as the output signal has a period of 400ps.

The hold moment of the SSPD is around 400mV, the threshold voltage of the sampling switches. The track and hold signals are shown in Figure 4.23. So the input signal is held around the peak of the sine wave, which is a locking point with far less gain. This means the SSPD/CP gain calculated previously does not hold. This could also explain the ringing on V_{tune} while there should be a phase margin of 80° . That it does not lock to the zero crossing of the input is probably due to the non-symmetrical Balun transconductance, where $g_{m,N4} \neq g_{m,P4} \frac{g_{m,N3}}{g_{m,P3}}$. This could be solved by further scaling the CP or choosing a different CP topology. The locking test together with the time domain signal demonstrate that the loop does correct for phase changes, and therefore is in lock.

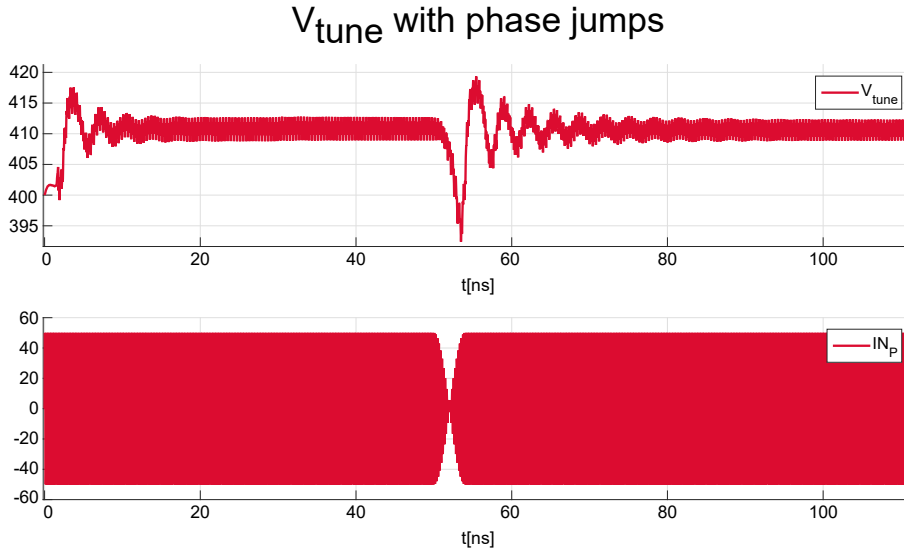


Figure 4.21: The positive differential voltage (IN), the output (OUT), and the tuning voltage with phase shift applied.

Power consumption

Simulating the loop with the previously obtained values for components is done using a **Periodic Steady State (PSS)** simulation. With this simulation the loop is simulated until it is stable, with a tuning voltage around $410mV$ and a output frequency of $2.5GHz$. Then the energy dissipation during 1 period of the slowest harmonic, in this case the output signal, is obtained during simulation and is multiplied with the frequency of the output signal to obtain the power consumption of the circuit when it is in lock; $P = V \int_0^{\frac{1}{f_{OUT}}} I \cdot dt \cdot f_{OUT}$. During this period the phase noise is also measured in simulation. The power consumption per **SSPLL** block is shown in Table 4.1 together with the total power consumption. These **SSPLL** blocks are the **SSPD/CP** without the inverter, the **VCO** and the inverter in the **SSPD/CP**. The power consumption of the **SSPD/CP** also includes the power drawn from the input source. Table 4.1 shows that the power consumption of the circuit is dominated by the **VCO**. The total power consumption of the **SSPLL** is $8.8\mu W$, which is well below the $1mW$ specification stated in section 1.2.

Table 4.1: Power consumption of the **SSPLL** per block.

SSPLL block	P [μW]
SSPD/CP	1.8
VCO	5
Inverter	2
Total	8.8

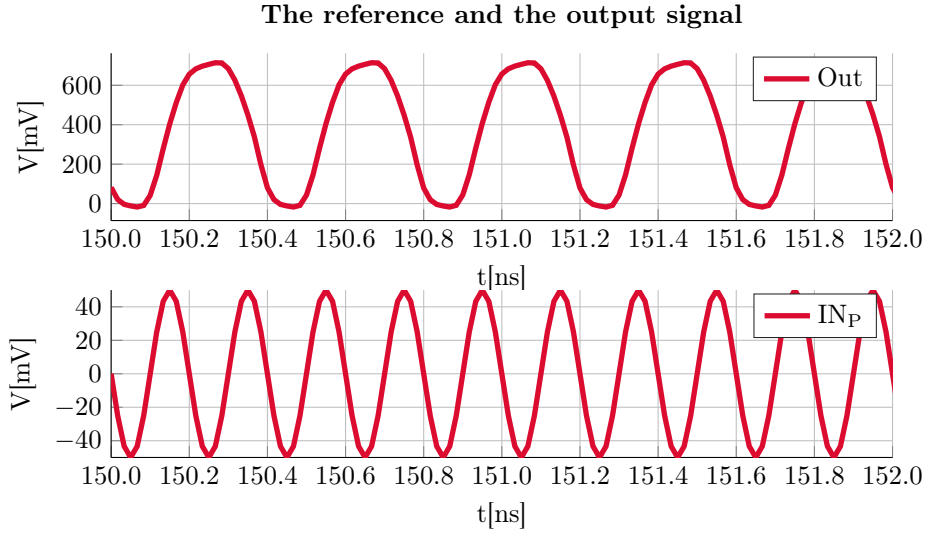
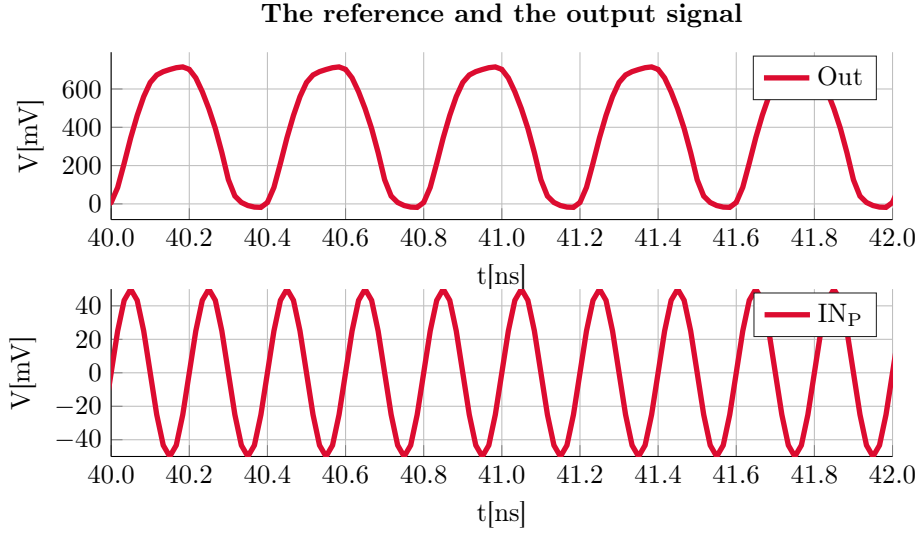


Figure 4.22

Phase noise

Obtaining the noise in simulation gives the the output phase noise shown in Figure 4.24. The free running oscillator noise is also shown in this figure to illustrate that the SSPLL does suppress the noise of the VCO up to the bandwidth of the CP, which is around 500MHz. That it does not suppress it up to the bandwidth of the loop is due to the fact that the gain of the SSPD is different than previously simulated due to the locking point at the peak of the sinusoidal input, instead of the zero crossing. At higher frequencies than the

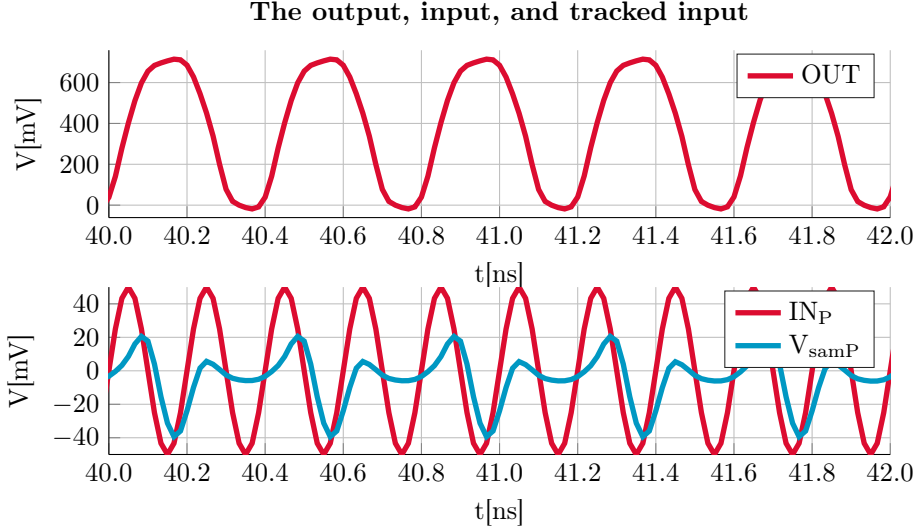


Figure 4.23: The output (OUT), positive differential input (IN_P) and the sampled positive differential input (V_{samP}) signal.

bandwidth of the CP, output noise follows the noise of the VCO, and the loop noise is attenuated. There is a ripple in the output phase noise around 500MHz. This is the point where both the shaped loop and the VCO noise contribute to a noise that is significantly higher than one of the individual contributions. At lower frequencies the loop noise dominates the output noise. According to Figure 4.4, the output phase noise below the bandwidth of the SSPLL should be flat, but in Figure 4.24 there is a $\frac{1}{f}$ or $\frac{-10dB}{dec}$ roll-off below 1MHz. This is due to the $\frac{1}{f}$ flicker noise of the CP, specifically that of transistors P_3 and P_4 in the Balun (Figure 4.16). Sizing these transistors up to decrease this $\frac{1}{f}$ noise is not possible due to the limited bandwidth of this CP. This $\frac{1}{f}$ noise does not show up in Figure 4.4, because the model that is used to generate this figure only includes current noise. The flat part of the noise between 1MHz and 100MHz is due to the 1st order part of the transfer where the phase bends back to 90° .

The noise according to the BLE specification should not exceed $-105dBc/Hz$ at $f_m = 3MHz$. In Figure 4.24 the phase noise is around $-116dBc/Hz$. This means that the BLE specification is achieved with this SSPLL. With a more strict noise specification the increasing phase noise in the pass band of the SSPLL could also be a problem, as well as the ripple on the phase noise around 500MHz.

This shaped output noise is also evidence that the SSPLL is in lock with the input. If the loop would not function at all, it would follow the output noise of the free running oscillator.

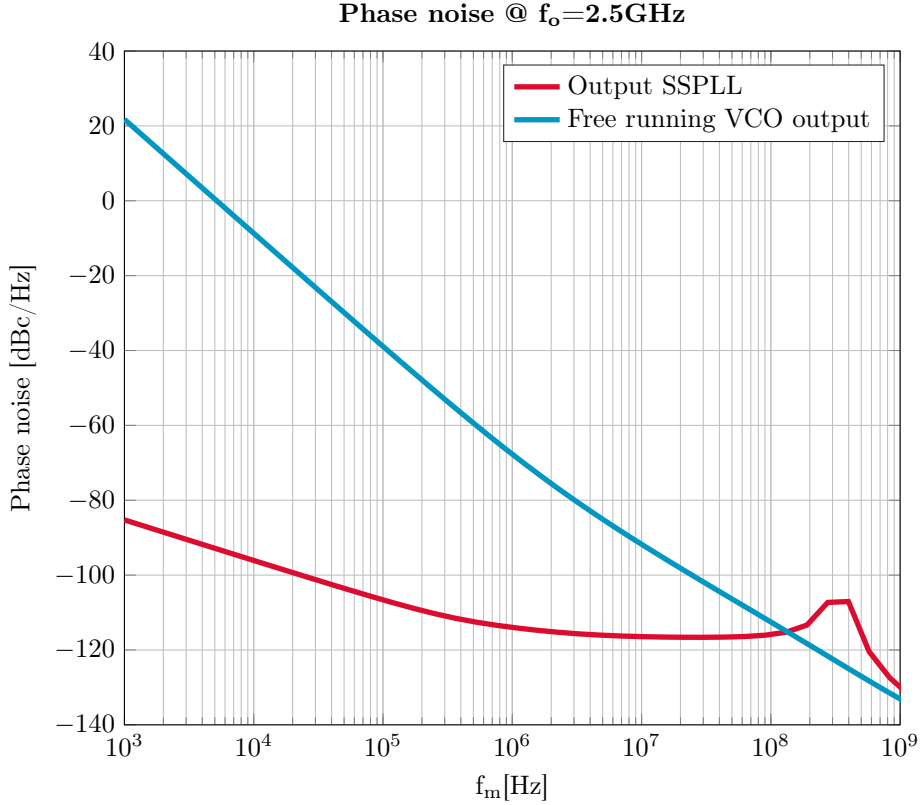


Figure 4.24: The output phase noise of the SSPLL divider, and the free running VCO.

4.5 Conclusion

A SSPLL is implemented that achieves a frequency division using sub sampling of the input at the rate of the output frequency. This is done by first examining the regular PLL and SSPLL. From this SSPLL, a system design of a dividing SSPLL is derived. The SSPLL is implemented using a division ratio of $N = 2$, without a FLL due to time constrain. The subsystems of the dividing SSPLL, the SSPD/CP and VCO, are implemented for minimum power dissipation. The VCO is implemented as a current starved ring oscillator, while the SSPD/CP is implemented by adopting the implementation used in the paper of Gao et al.[17]. No IQ generation is implemented, because this is also omitted in the TSPC counter divider, and it would cause a significant increase in power consumption in the VCO. With specifications of these individual subsystems, the SSPLL is tuned using the low pass filter. Because the VCO is very noisy due to its low power consumption, the bandwidth of the SSPLL is maximized. However this bandwidth is limited by the bandwidth of the CP. Eventually the loop is tuned to a phase margin of 80° and a bandwidth of 120MHz . However these values are not very reliable because of the shifted locking point of the SSPD, and therefore the different gain of the SSPD. This shifted locking point

is probably due to the non-symmetrical transconductance of the CP.

With this implementation, a noise of -116dBc/Hz at a offset frequency 3MHz is achieved with a power consumption of $8.8\mu\text{W}$. This specification is sufficient to achieve the BLE specification set in section 1.2. However the SSPLL does track the phase of the input signal. This is demonstrated with the locking test, and can also be concluded from the fact that the phase noise does not follow the noise of the VCO. The locking point of the SSPD is however different than expected, giving a different detector gain and different loop behavior.

An improvement in the implementation could be to use a different SSPD/CP implementation. With the implementation used, the differential amplifier in the CP only attenuates the sampled voltage, while offering limited bandwidth. The Balun is difficult to bias to a symmetrical transconductance, while it also has a limited bandwidth and generates significant flicker noise. If the bandwidth, flicker noise and symmetry of the CP is improved, a lower output phase noise could be achieved.

Chapter 5

Conclusion

For a correct comparison of the developed frequency divider, the lowest power frequency divider is found using estimation. From the Miller, injection locked and counter divider, it turned out that this is the counter implemented with TSPC FFs. The estimated power consumption of this divider is $4.4\mu W$, so this divider is evaluated further using simulation. With simulation it is determined that the bandwidth of this divider is not high enough to support input frequencies higher than $5GHz$. However low power has a higher priority than bandwidth, so this divider is still used. To provide an adequate benchmark for the developed SSPLL divider, a fixed $5GHz$ input frequency is used while dividing this frequency down to $2.5GHz$, $1.25GHz$, $625MHz$, $312.5MHz$, $165MHz$ and $78MHz$. The power and phase noise of this TSPC divider is found using simulation. Then the SSPLL divider's VCO and CP are implemented for minimum power consumption, and its loop filter is designed for stability and minimal output noise. The frequency dividing SSPLL's power consumption and phase noise is also found using simulation.

It is possible to use a SSPLL as a frequency divider, however the power and phase noise performance of the existing TSPC divider is better for a division ratio of 2. The simulated power consumption of the SSPLL divider is two times higher than that of the TSPC divider for $N = 2$, namely $4.4\mu W$ for the TSPC divider compared to $8.8\mu W$ of the SSPLL divider. The phase noise of the SSPLL is $-116dBc/Hz$ at $f_m = 3MHz$. This means it is $27dB$ higher than the $-143dBc/Hz$ phase noise of the TSPC divider at the same offset frequency.

5.1 Recommendations

If the division ratio of the SSPLL divider increases, the VCO and inverter in the SSPD/CP run at a lower frequency and thus use less power, as opposed to the TSPC divider, where the power consumption increases with division ratio. When the division ratio of the TSPC divider is increased to obtain a lower output frequency, the power consumption increases, as simulated in section 1.2. For estimating this power consumption over division ratio for the SSPLL, the power consumption of the VCO and the inverter in the SSPD/CP can be divided by the division ratio. The power consumption of the SSPD itself would also decrease when f_{out} decreases, for simplicity this decrement

is not taken in to account in the estimation. Doing this estimation gives the power consumption of the TSPC and SSPLL divider over N in Figure 5.1. This figure shows that the power consumption for division ratios higher than 2 is lower with the SSPLL divider than with the TSPC divider. This presents that using this SSPLL with higher division ratios could give benefit in term of power consumptions. To find the actual power consumption at higher division ratios, more simulations need to be done.

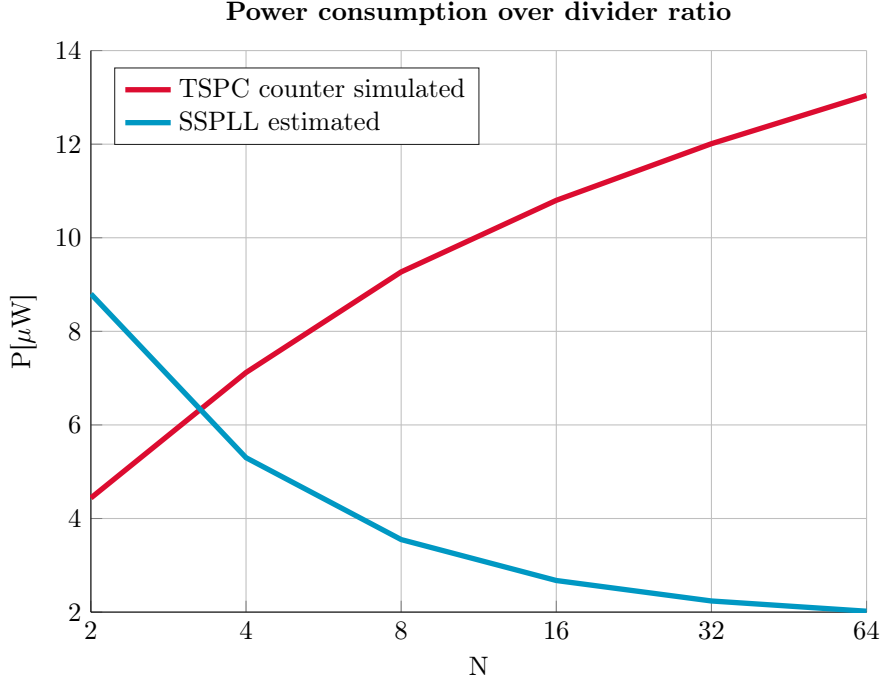


Figure 5.1: The power consumption of the TSPC counter divider, and the estimated power consumption of the SSPLL.

The bandwidth of the SSPD could be scaled up for higher input frequencies by scaling the sampling switches up, this however means that the inverter(s) driving these switches also have to be scaled up, giving an increase in power consumption. The bandwidth of the TSPC circuit however is almost at its limit. Increasing V_{dd} could help, but this gives a quadratic increase in power consumption. So it would be more sensible to choose a different divider as a benchmark to compare the SSPLL to.

To obtain a better noise/power performance for the SSPLL, a more wide band low-power, low-noise and symmetrical CP topology should be chosen. The topology chosen in this thesis is the same as that used in Gao. et al.[17], and is not a wide band low-power, low-noise symmetrical CP topology.

To gain better noise performance, a reclocking scheme could be used at the output of the SSPLL. This scheme selects an input clock edge if one output clock edge of the divider has passed. This way the output only contains edges from the input, and it follows the phase noise of the input. This scheme could be used for every divider and is not restricted to the SSPLL. It is useful in case

of the *SSPLL*, because the *SSPLL* has insufficient noise performance. There however is a limitation of the noise the divider is allowed to have, as otherwise the wrong edge may be selected. The reclocking scheme would also add power consumption to the system.

Appendix A

Flip flop logic types

The latch types mentioned in [section 3.2](#) can be implemented in various types of logic, however there is always one optimal latch type for one type of logic. Therefore a detailed description of CMOS, Transmission gate, C²MOS, TSPC and CML logic included in this chapter.

A.1 CMOS

The most commonly used logic type is CMOS, this is a logic type which uses both NMOS and PMOS to propagate the ground and supply voltage to the output, depending on the input voltage.

SR-latches are the types of latches which consume the least amount of power in CMOS, as these contain the least amount of transistors. The schematic for such a latch is shown in [Table 3.2](#). For a clock inversion and combining logic ports, several optimizations on transistor level can be used to reduce the power consumption.

The type of latch used in this structure is regenerative. If $Clk = 0$, the outputs of the clock gates are both one, and the two cross coupled NAND gates act as two cross coupled inverters. If $Q \neq \overline{Q}$, the feedback from Q to \overline{Q} causes this difference to be amplified until both signals clip to the supply voltages. In case of a perturbation on Q or \overline{Q} , the feedback regenerates both signals back to the supply voltages. With non-regenerative latches, the output logic level is stored on a capacitance. If $Clk = 1$, the inputs are evaluated; the clock gate acts as two inverters, breaking the feedback in the latch and causing one of the latch NAND's to pull Q or \overline{Q} to 1, while the other latch NAND inverts this 1 and pulls Q or \overline{Q} to 0.

Advantages of the CMOS latch:

- Regenerative latch structure, no minimum clock speed.
- Full (supply voltage) swing of the output.

Disadvantages of the CMOS latch:

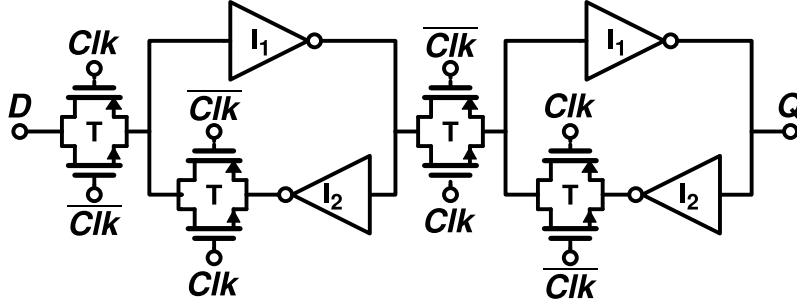
- Many transistors causing a high dynamic power dissipation.
- When combining logic gates to reduce the number of logic gates and less transistors, more transistors are stacked, requiring a higher supply voltage.

A.2 Transmission gate

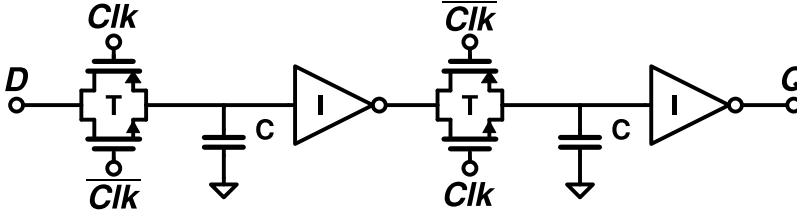
The clock gate consists of one transmission gate (T) which propagates the input to the output on a high Clk . This transmission contains a NMOS and PMOS to propagate 0 and 1 logic levels respectively, without a threshold voltage drop over the transmission gate. A D-FF therefore is the most power efficient and simple implementable FF in this logic type, although this topology still needs an inverter to generate a \bar{Q} output. The value of the output can be stored both regeneratively and non-regeneratively, as shown in Figure A.1. The two latches in this FF need a non-overlapping clock, otherwise Q is directly connected to D , causing problems in a counter made in this logic type.

With regenerative latching, the feedback gain (I_2) is broken during the evaluation of the input. It is also possible to implement a regenerative transmission gate without breaking the feedback, this is done with inverters that are rationed. This means that the feedback inverter (I_2) is weaker than the forward inverter (I_1).

With non-regenerative latching, the output is stored on the capacitance (C). This requires periodic recharging of the capacitance, as the capacitance drains over time by leakage current. The period over which the draining occurs should not be too long, and therefore the circuit has a minimal input frequency. Because the required recharging, this logic is called dynamic logic. The regenerative latch is static, as this does not require periodic recharging. These non-regenerative latches also require some inverters (I) as buffers, to prevent charge sharing between the two store capacitances (C).



(a) A regenerative transmission gate FF.



(b) A non-regenerative transmission gate FF.

Figure A.1

Advantages of the transmission gate latch:

- Less transistors and power consumption than CMOS.
- Full (supply voltage) swing of the output.
- Low supply voltages possible due to no stacking of transistors.

Disadvantages of the transmission gate latch:

- Requires a complementary clock.
- Needs a non-overlapping clock.
- Due to being non-regenerative it has a minimum clock speed.

A.3 C^2MOS

The overlapping clock problem of transmission gate FFs can be solved by integrating the transmission gate into the inverters to form a clocked inverter, as shown in Figure A.2, hence the name C^2MOS . In case of an overlapping clock ($Clk = \overline{Clk}$), as shown in Figure A.3, only one logic level (0 or 1) can be propagated through the first inverter. The second inverter needs the inverse of this logic level to conduct a logic level to the output, and thus the FF is not transparent with an overlapping clock.

Placing the clocked transistors inwards (Figure A.2a) or outwards (Figure A.2b), mainly has effect on the moment charge injection occurs in the capacitor (C). For realizing a \overline{Q} output, an extra inverter is needed.

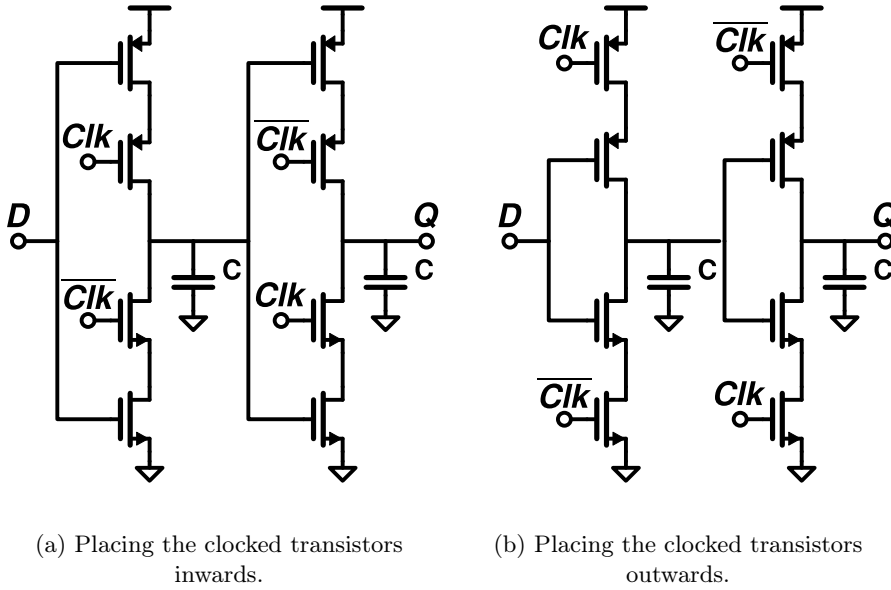


Figure A.2: C^2MOS FFs.

Advantages of the C^2MOS latch:

- Has less transistors and power consumption than CMOS.

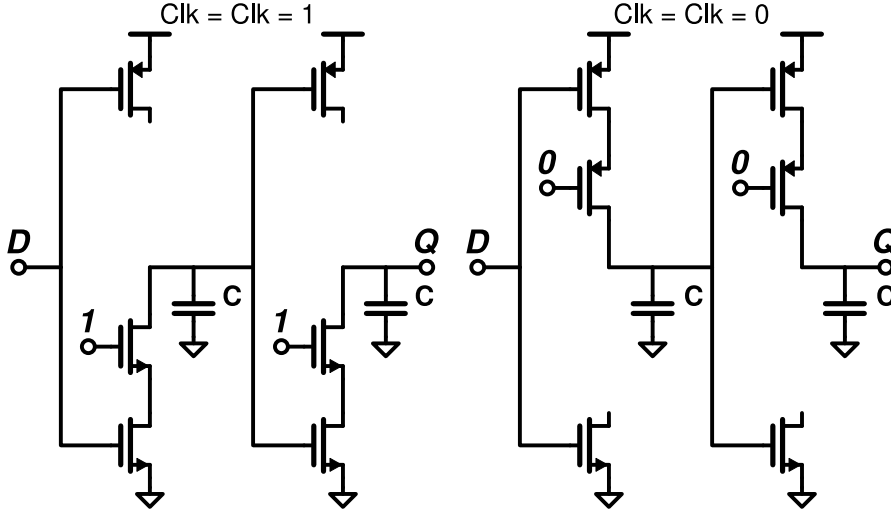
(a) $Clk = \overline{Clk} = 1$ overlap.(b) $Clk = \overline{Clk} = 0$ overlap.

Figure A.3: Clock overlap with C2MOS FFs [19].

- Provides full (supply voltage) swing of the output.

Disadvantages of the C²MOS latch:

- Needs a complementary clock.
- Just as with CMOS FF's, higher supply voltages are required for the 4 stacked transistors.
- There is a clock speed when using non-regenerative transmission gate FFs

A.4 True single phase clock

With C²MOS FFs, both logic levels are blocked by one latch while passed through the other on one clock phase. If 2 cascaded clocked inverters block the same logic level on the same clock phase, latching functionality is also present. This means that one logic level has to be blocked per clocked inverter with Clk , and one of the two clocked transistors in a clocked C²MOS inverter can be removed, thus taking out the need for a complementary clock. Removing these transistors gives the Static P TSPC stage (SP) and Static N TSPC stage (SN) in Figure A.4. This also explains the name TSPC, as only one clock phase is necessary. Note that parasitic capacitance is used to store the latch value, so this topology is non-regenerative. Cascading two SN and two SP stages Out to In forms a FF.

In a SP and SN stage, one logic level (L) is blocked to the output (Out) with Clk . There is a second output (Out') in a static stage where the inverse logic level (\overline{L}) of Out is blocked. By connecting Out and Out' to two transistor that need these two blocked logic levels to conduct, a latch can be made. These two transistors form a split output stage in Figure A.4.

In a Precharged N TSPC stage (PN) or a Precharged P TSPC stage (PP), there is a precharge and an evaluation clock phase. The output is precharged to a logic level (LL) on the precharge clock phase. The output is pulled up or down to the inverse logic level (\overline{LL}), if the input is the precharged logic level (LL) in the evaluation clock phase. If the previous stage only supplies the input with the required logic level (LL) on the precharge phase, or the succeeding stage only uses the precharged logic level (LL) on the evaluation stage, a latch is realized. If it does both, a FF shaped. This means that with SP-PN-SN or SN-PP-SP stages cascaded from *Out* to *in*, a FF can be made. When a TSPC FF is made from 3 stages, an inversion is given to the output.

It is also possible to incorporate additional logic in the FFs, by placing logic parts between the two clocked transistors in PN and PP stages and complementary logic parts at both ends of SN and SP stages. It is also possible to incorporate non-clocked stages between TSPC stages. This way additional logic functions can be implemented within a FF.

A PP or PN contains two precharged nodes, of which one is the output. The other precharged node can be used as a Vdd or Ground for the succeeding SP or SN stage, as shown for cascaded PN and SN stage in Figure A.5. This gives lower power consumption at the cost of speed. It is also possible to incorporate stages of other logic types in TSPC FFs, such as C²MOS[20] and ratioed logic[21] stages.

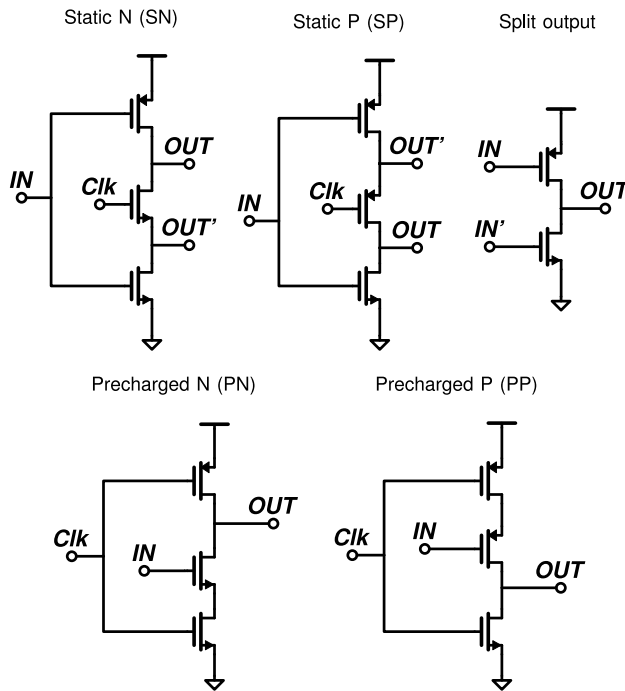


Figure A.4: The five TSPC stages which can be combined to form a FFs [20],[22].

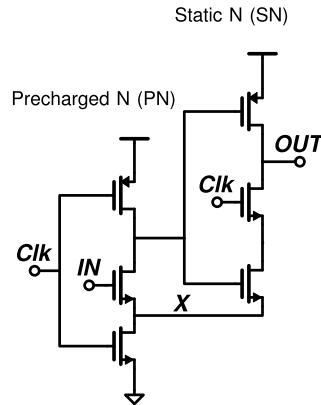


Figure A.5: Using the conditionally pulled up node of a PN stage as a ground for succeeding SN stage.

Advantages of the TSPC latch:

- Less transistors and power consumption than CMOS.
- Full (supply voltage) swing of the output.
- Does not need a complementary clock.
- Inversion of the output if a 3 stage TSPC FF is made.

Disadvantages of the TSPC latch:

- Just as with CMOS FF's, higher supply voltages are required for the 3 stacked transistors.
- Due to being non-regenerative it has a minimum clock speed.

A.5 Current mode logic

With CML the current through the tail current source, the transistor with *Bias* in Figure A.6 is switched between a differential pair and a latch. When $Clk = 1$, the logic levels of D and \overline{D} are transferred by the differential pair to the parasitic capacitances of the nodes Q and \overline{Q} . When $Clk = 0$, these logic levels are regeneratively stored by the latch transistors. To form a FF, two of these CML latches are required.

The speed of this circuit depends on the tail current used, and the parasitic output capacitance. The power consumption also depends on this tail current, and so there is a clear power speed tradeoff.

Advantages of the CML latch:

- Due to being regenerative, it has no minimum clock speed.
- Possible high speed circuit.

Disadvantages of the CML latch:

- High static power consumption, due to bias current source.

- Higher supply voltages are required for the 3 stacked transistors and load resistor.
- No full (supply voltage) swing of the output.
- A complementary clock is necessary.

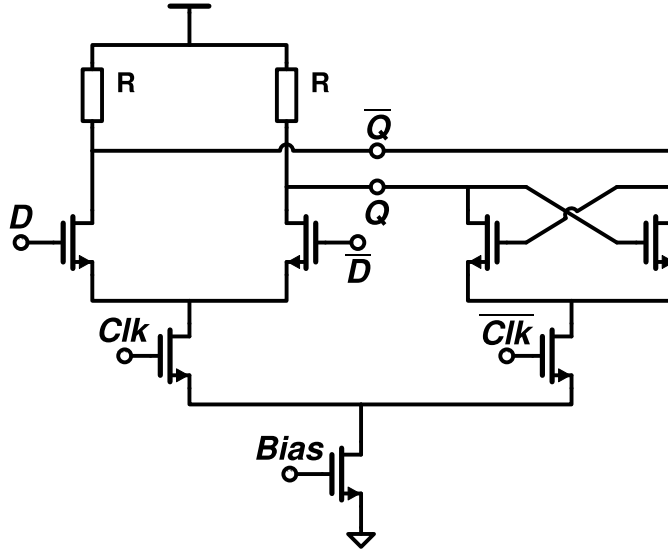


Figure A.6: CML Latch.

A.6 Summary

The types of logics discussed in this thesis are; CMOS, static and dynamic transmission gate, C²MOS and TSPC. CMOS is the most basic type of logic and consist of the most transistors. Transmission gate logic consists of transmission gates. With C²MOS, the transmission gates are integrated in the inverters to solve the overlapping-clock problem. TSPC Is a derivate of C²MOS, where transistors are removed to take out the need for a complementary clock. An overview of the advantages and disadvantages of all the logic types is shown in Table A.1. In the first column the logic types are present, in the second column it is indicated if there is a complementary clock required, the third column indicates if a non overlapping clock is required, in the fourth column it is indicated if a minimum clock speed is mandatory, the fifth column indicates if there is a full swing voltage on the output, and the last column indicates how many transistors are minimally stacked from between the supplies. The logic type "Optimized CMOS", is the CMOS logic used in section B.3. In this logic the NAND and OR gates are combined to reduce the number of transistors, and to omit the inverter for complementary clock.

Table A.1: Advantages and disadvantages of the logic types

Logic type	Complementary clock	non-overlapping clock	Minimum clock frequency	Delivers full output swing	stacked transistors
Optimized CMOS	no	no	no	yes	3
Regenerative transmission gate	yes	yes	no	yes	2
Non-regenerative transmission gate	yes	yes	yes	yes	2
C ² MOS	yes	no	yes	yes	4
True single phase clock	no	no	yes	yes	3
Current mode logic	yes	no	no	no	3

Appendix B

Power dissipation estimation

In [section B.1](#), an expression for the power of a synchronous and asynchronous counter is given in terms of the number of switched data and clock transistors. The capacitance of one TSMC65 transistor is estimated using simulation in [section B.2](#). Then the amount of switched transistors is obtained for the various types of logic in [section B.3](#)-[section B.8](#). Combining this information gives an estimation for the power consumption of the two counter types implemented in various types of logic in TSMC65.

The following causes for additional power consumption are not taken into account:

- Power consumption other than dynamic CV^2f_{in} power consumption. This includes leakage, short circuit, DC standby currents and glitch power. These powers are almost non computational, and therefore not included in the estimation.
- Supply voltage scaling; all logic types are compared with the same supply voltage. Lower supply voltages could be used for logic types that have less transistors stacked. However when using counters in a system, most of the time counters are not the limiting factor on the supply voltage.
- Clock circuitry other than inverters for complementary clock. This could cause an additional power consumption for transmission gate FFs, as these require non-overlapping clock.
- Additional capacitances. It is assumed that the parasitic transistor capacitances are sufficiently large to hold the logic levels at the frequency that the circuit operates. This means no capacitance (C in [Figure A.1a](#) and [Figure A.2](#)) is added to the outputs of non-regenerative latches. Because the frequency becomes lower with higher division ratios, this could cause problems with high division ratios.
- Input, Output and clock buffers. Adding these buffers would add power consumption to all the logic types, and would not significantly influence the comparison.

B.1 Asynchronous and Synchronous divider

The total power consumption of counters is approximated by adding the power consumption of the clocked transistors (c) and the transistors in the data path (d), with switching activities α_c and α_d respectively:

$$P_{FF} = cf_{in}CV_{dd}^2 + \alpha_d df_{in}CV_{dd}^2 = (c + \alpha_d d)f_{in}CV_{dd}^2 \quad (B.1)$$

In this context, the switching activities determine how many times a zero to one and one to zero transitions per clock period is present in the clock or data in a FF, therefore $\alpha_c = 1$. The output and input parasitic capacity of one transistor is estimated to be equal, to simplify the calculation.

Synchronous divider

The synchronous counter is shown in Figure B.1, the number of equal FFs in this figure is M and the division factor (N) is $2M$. There is one (zero to one, or one to zero) data change in the whole cascade of FFs per clock period, therefore the data switching activity per FF is $\frac{1}{2M}$. However, the clock frequency is the same for the whole cascade. This gives the following total power:

$$P_{\text{synchronous counter}} = P_{FF} \cdot M = (c + \frac{d}{2M})CV_{dd}^2 f_{in} \cdot M = (Mc + \frac{1}{2}d)CV_{dd}^2 f_{in} = \frac{1}{2}(Nc + d)CV_{dd}^2 f_{in} \quad (B.2)$$

This is the same result as taking a frequency of $\frac{f_{in}}{2M}$ for the data, f_{in} for the clock, and a $\alpha_c = \alpha_d = 1$.

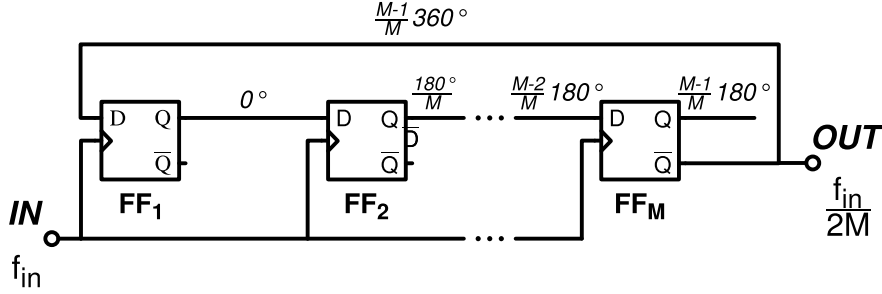


Figure B.1: A synchronous counter.

Asynchronous divider

The asynchronous counter is shown in Figure B.2, the number of equal FFs in this figure is K , and the division ratio is 2^K . There is one data change in every FF per clock period, therefore the switching activity for the data path (α_d) is $\frac{1}{2}$. However, the clock frequency, and thus the power, is decreased per FF in the cascade, meaning that the first FF dominates the power consumption.

This gives the following total power:

$$\begin{aligned}
 P_{\text{asynchronous counter}} &= \frac{P_{FF}}{f_{in}} (f_{in} + \frac{f_{in}}{2} + \dots + \frac{f_{in}}{2^{K-1}}) = (c + \frac{d}{2}) CV_{dd}^2 f_{in} \sum_{k=1}^K \frac{1}{2^{k-1}} = (2c + d) CV_{dd}^2 f_{in} (1 - \frac{1}{2^K}) \\
 &= (2c + d) CV_{dd}^2 f_{in} (1 - \frac{1}{N})
 \end{aligned} \tag{B.3}$$

This is the same result as reducing the switching activity per FF in the cascade and taking f_{in} as the clock speed for all FFs.

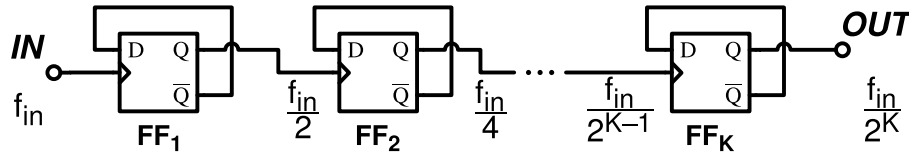


Figure B.2: An asynchronous counter.

B.2 In and output capacitance

The in and output capacity of one transistor is calculated by measuring the input and output current of a minimum sized inverter with a load in simulation. This load is present to decrease the slope of the output voltage, and to decrease the crowbar currents in the inverter. Minimum sized in the TSMC65 process means that the NMOS and PMOS are sized $\frac{W}{L} = \frac{120nm}{60nm}$ and $\frac{W}{L} = \frac{240nm}{60nm}$ respectively for equal driving strength. The capacitance is calculated by integrating the current flowing in and out the intrinsic transistors capacitance over the rise and fall time of a square-wave input signal, and dividing this by the voltage change. This gives an input, output and average capacity of $160aF$, $260aF$ and $210aF$ respectively.

B.3 CMOS

To estimate the amount of switched transistors in a FF, the gate equivalents method described in "Low Power Design Methodologies" [23, p. 47-51] is used. To get a lower power consumption for a CMOS FF, the circuit is optimized for power using boolean algebra resulting in the circuit in Figure B.3. In this FF, the And-Or-Inverter and Or-And-Inverter combinations can be combined in one And-Or-Inverter Gate (AOI) and Or-And-Inverter Gate (OAI), as shown in Figure B.4. Every AOI/OAI gate input capacitance consist of 2 transistor input capacitances. The gate output capacitance consist of 6 transistor output capacitances due to the Miller effect, as shown in Figure B.4. As shown in Figure B.3, there are 4 gate inputs driven by the clock, consisting of 8 input capacitances (red dots). There are 8 inputs (green dots) and 4 outputs (blue dots) driven by the data path, consisting of 16 input and 24 transistor output

capacitances. This gives a total of $d = 40$ and $c = 8$. The total power consumption, with colored numbers representing the transistor capacitances indicated as colored dots in Figure B.3, is:

$$P_{\text{synchronous counter}} = \frac{1}{2}(\mathbf{8}N + \mathbf{16} + \mathbf{24})NCV_{dd}^2f_{in} = (4N + 20)CV_{dd}^2f_{in} \quad (\text{B.4})$$

$$P_{\text{asynchronous counter}} = (2 \cdot \mathbf{8} + \mathbf{16} + \mathbf{24})CV_{dd}^2f_{in}(1 - \frac{1}{N}) = 56(1 - \frac{1}{N})CV_{dd}^2f_{in} \quad (\text{B.5})$$

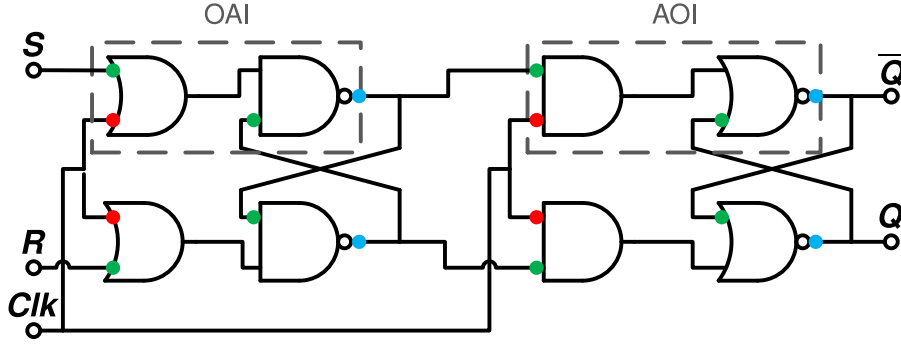


Figure B.3: A CMOS SR FF, simplified with boolean algebra.

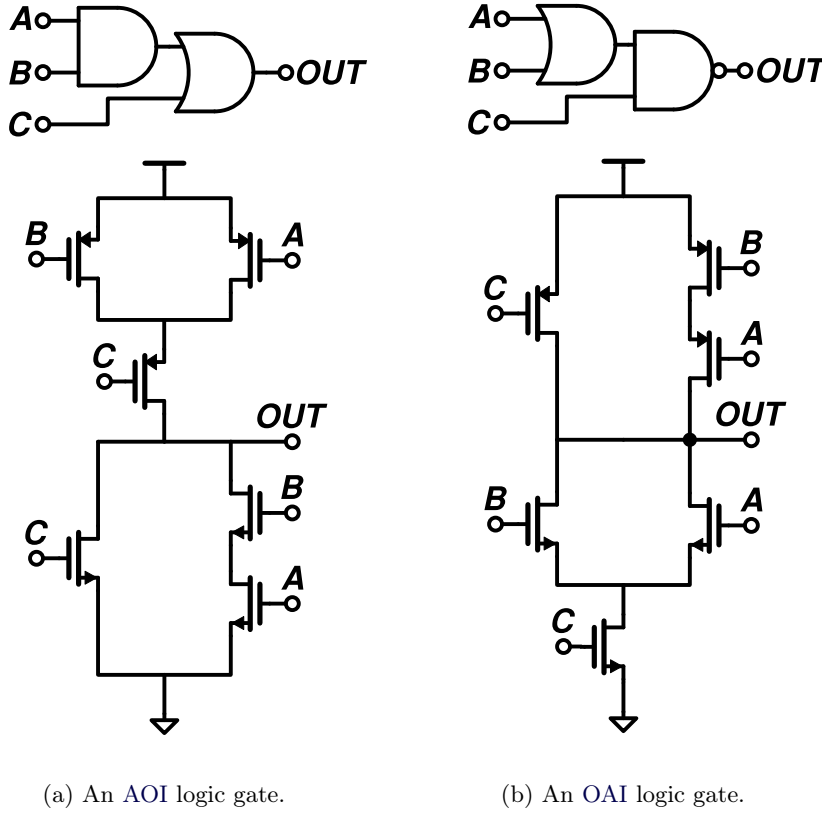


Figure B.4

B.4 Regenerative transmission gate

A regenerative FF with a transmission gate after the feedback inverter is chosen. Using regenerative feedback with stronger inverters in the forward path than in the feedback path costs a lot of power. The power consumption of a circuit used to ensure a non overlapping clock is not taken in account in this calculation.

As shown in Figure B.5, there are 8 transistor input capacitances driven by the clock, shown as red dots. There are 4 inverter inputs (green dots) and outputs plus 8 transmission gates outputs (blue dots) driven by the data path, consisting of 8 input and 24 transistor output capacitances. This could be an exaggeration of the capacitance in a transmission gate [23, p. 48], because not all transistors in a cascode add capacitance to the output.

For an asynchronous counter \overline{Q} is needed every FF, requiring one additional inverter per FF. For a synchronous divider, \overline{Q} is required once per counter. This means that one of the M FFs in a synchronous counter has an additional inverter, and has $d = 28$ instead of $d = 24$. This one inverter runs at the output ($\frac{f_{in}}{N}$) frequency, which means there is an additional $\frac{2+2}{N} CV_{dd}^2 f_{in}$ term in eq. (B.2).

There is also a complementary clock needed once per counter, both asyn-

chronous and synchronous, that requires an additional inverter running at the input frequency f_{in} , giving an additional $4CV_{dd}^2f_{in}$ term in both eq. (B.2) and eq. (B.3). These additional terms are necessarily, as eq. (B.2) and eq. (B.3) assume equal FFs, without additional logic. This gives a total power, with colored numbers representing the transistor capacitances indicated as dots in Figure B.5, of:

$$\begin{aligned} P_{\text{synchronous counter}} &= \frac{1}{2}(8N + 8 + 24)CV_{dd}^2f_{in} + \frac{2+2}{N}CV_{dd}^2f_{in} + 4CV_{dd}^2f_{in} \\ &= (4N + 20 + \frac{4}{N})CV_{dd}^2f_{in} \end{aligned} \quad (\text{B.6})$$

$$\begin{aligned} P_{\text{asynchronous counter}} &= (2 \cdot 8 + 8 + 24 + (2 + 2))CV_{dd}^2f_{in}(1 - \frac{1}{N}) + 4CV_{dd}^2f_{in} \\ &= (56 - \frac{52}{N})CV_{dd}^2f_{in} \end{aligned} \quad (\text{B.7})$$

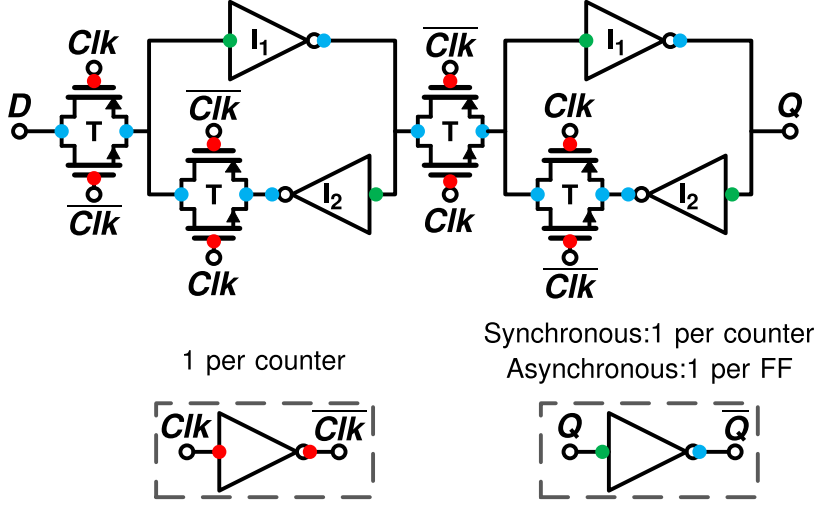


Figure B.5: A regenerative transmission gate D FF.

B.5 Non-regenerative transmission gate

For the estimation for the power consumption of non-generative FFs, it is assumed that the parasitic transistor capacitances are sufficiently large to hold the values for the 2.5GHz clock speed. Therefore no additional capacitances (C in Figure B.6 and Figure B.7) are added to the outputs of non-regenerative latches.

As shown in Figure B.6, there are 4 transistor input capacitances driven by the clock, shown as red dots. There are 2 inverter inputs (green dots) and outputs plus 4 transmission gates outputs (blue dots) driven by the data path, consisting of 4 input and 12 output capacitances.

For an asynchronous counter, \bar{Q} is needed in every FF, requiring one additional inverter per FF, and thus $c = 4, d = 20$.

Adding the same terms as with the regenerative transmission gate FF for the additional inverters for the \overline{Clk} and \overline{Q} gives a total power of:

$$\begin{aligned} P_{\text{synchronous counter}} &= \frac{1}{2}(4N + 4 + 12)CV_{dd}^2f_{in} + \frac{2+2}{N}CV_{dd}^2f_{in} + 4CV_{dd}^2f_{in} \\ &= (2N + 12 + \frac{4}{N})CV_{dd}^2f_{in} \end{aligned} \quad (\text{B.8})$$

$$\begin{aligned} P_{\text{asynchronous counter}} &= (2 \cdot 4 + 4 + 12 + (2 + 2))CV_{dd}^2f_{in}(1 - \frac{1}{N}) + 4CV_{dd}^2f_{in} \\ &= (32 - \frac{28}{N})CV_{dd}^2f_{in} \end{aligned} \quad (\text{B.9})$$

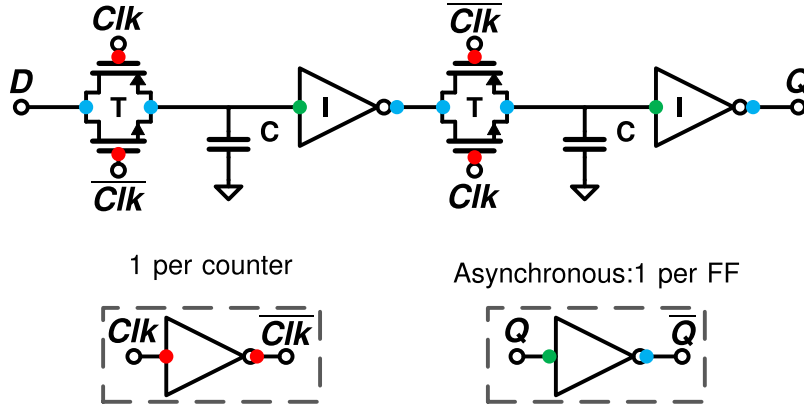


Figure B.6: A non-regenerative transmission gate D FF.

B.6 C²MOS

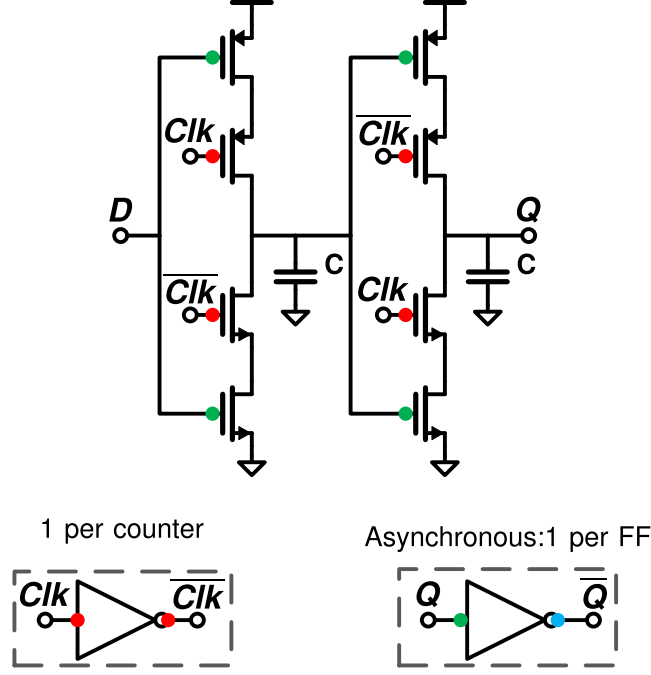
As shown in Figure B.6, there are 4 transistor input capacitances driven by the clock, shown as red dots. There are 4 transistor input (green dots) and outputs (blue dots) capacitances driven by the data path, consisting of 4 input and 8 output capacitances.

For an asynchronous counter \overline{Q} is needed every FF, requiring one additional inverter per FF, and thus $c = 4, d = 16$.

Adding the same terms as with the regenerative and non-regenerative transmission gate FF for the additional inverters for the \overline{Clk} and \overline{Q} gives a total power of:

$$\begin{aligned} P_{\text{synchronous counter}} &= \frac{1}{2}(4N + 4 + 8)CV_{dd}^2f_{in} + \frac{2+2}{N}CV_{dd}^2f_{in} + 4CV_{dd}^2f_{in} \\ &= (2N + 10 + \frac{4}{N})CV_{dd}^2f_{in} \end{aligned} \quad (\text{B.10})$$

$$\begin{aligned} P_{\text{asynchronous counter}} &= (2 \cdot 4 + 4 + 8 + (2 + 2))CV_{dd}^2f_{in}(1 - \frac{1}{N}) + 4CV_{dd}^2f_{in} \\ &= (28 - \frac{24}{N})CV_{dd}^2f_{in} \end{aligned} \quad (\text{B.11})$$

Figure B.7: A C²MOS D FF.

B.7 True single phase clock

As shown in Figure B.8, there are 4 transistor input capacitances driven by the clock, shown as red dots. There are 3 normal input (green dots) and 6 outputs (blue dots) transistor capacitances driven by the data path. 2 Transistor gates are connected to precharged nodes (purple dots), and these transistors have 3 output capacitances. An additional precharge node is present as ground for the last Static N stage. These precharged nodes have an activity factor of $\frac{1}{2}$, because these nodes (approximately) function half the time as a \overline{Clk} , when $Q = 0$ with a half duty-cycle Q .

For this 3 stage TSPC FF, no additional inverter for \overline{Q} and \overline{Clk} is needed as this topology already has an inversion on the output and needs one clock phase. However, for synchronous counters with even number of FFs an extra inverter is needed. This gives a power of:

$$P_{\text{synchronous counter}} = \begin{cases} (3.5N + 4.5 + \frac{4}{N})CV_{dd}^2f_{in}, & \text{if } \frac{N}{2} \text{ is even} \\ (3.5N + 4.5)CV_{dd}^2f_{in}, & \text{otherwise} \end{cases} \quad (\text{B.12})$$

$$\begin{aligned} P_{\text{asynchronous counter}} &= (2(\textcolor{red}{4} + \frac{1}{2} \cdot \textcolor{violet}{2} + \frac{1}{2} \cdot \textcolor{orange}{4}) + \textcolor{green}{3} + \textcolor{blue}{6})CV_{dd}^2f_{in}(1 - \frac{1}{N}) \\ &= (23 - \frac{23}{N})CV_{dd}^2f_{in} \end{aligned} \quad (\text{B.13})$$

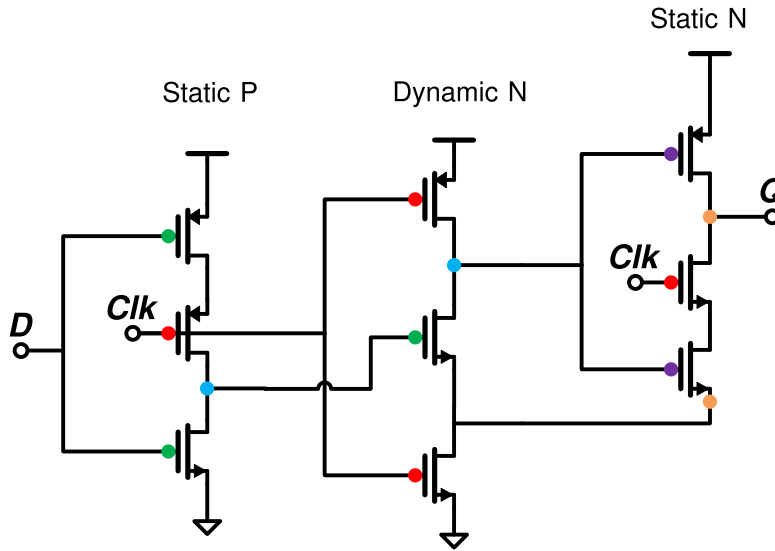


Figure B.8: A TSPC D-FF.

B.8 Summery

First the power consumption of the asynchronous and synchronous counters are calculated. Than the output and input capacitance of one transistor is estimated. Finally an estimation for the number of data (d) and clocked (c) is done. An overview of this estimation is given in Table B.1. In this table the logic types, number of data and clocked transistors, and it is indicated if the logic needs an additional inverter for \overline{Clk} and \overline{Q} .

Table B.1: Power consumption of the SSPLL per block.

logic type	d	c	inverter for \overline{Clk}	inverter for \overline{Q}
CMOS	40	8	no	no
Regenerative transmission gate	32	8	yes	yes
Non-regenerative transmission gate	16	4	yes	yes
C ² MOS	12	4	yes	yes
True single phase clock	9	7.5	no	no

Appendix C

Oscillator power estimation

C.1 LC oscillator

An LC and a Ring based VCO can be used. The LC based oscillator needs to meet the oscillation condition with some margin by adding the more energy with a transconductance (g_m), than it is dissipated in the equivalent parallel resistance of the LC tank (R_p); $g_m R_p \geq 4$ [4, p. 517]. This gives a power consumption for an oscillator with the transistors operating in weak inversion (for minimal power consumption) of:

$$P = V_{dd} I_{ss} = g_m V_{dd} n V_t \geq \frac{4}{R_p} V_{dd} n V_t \quad (\text{C.1})$$

With some realistic values of; $R_p = 1k\Omega$ for on chip inductors, slope factor (n) of 1.6, $V_{dd} = 0.7V$ and $V_t = \frac{kT}{q} = 26mV$ at 300K, there is a minimal power consumption $120\mu W$.

C.2 Ring oscillator

A ring oscillator also can be used. The output frequency of such an oscillator is[18]:

$$f_{out} \approx \frac{\mu_{eff} W_{eff} C_{ox} \Delta V^2}{8\eta K L q_{max}}$$

With $\mu_{eff} = \frac{\mu_n W_N + \mu_p W_P}{W_N + W_P}$ the effective channel mobility, $W_{eff} = W_n + W_p$ the effective width, C_{ox} the oxide capacitance, $\Delta V = (V_{gs} - V_{th})$ the overdrive voltage, η the proportionality constant, K the number of stages in the oscillator, L the length of the NMOS and PMOS, W_N and W_P the width of the NMOS and PMOS in the oscillator and $q_{max} = CV^2$ the maximal charge on the output node capacitance.

The rise and fall time of the output of one oscillator stage is approximated as equal; $t_F = t_R = \frac{1}{f'_{max}}$, with f'_{max} the maximal slope of the output as shown in figure C.1. The normalized stage delay (t_D) is the time it takes, if one output of a VCO stage is at half supply, for the succeeding stage output to reach half supply. This time is $t_D = \frac{\eta}{f'_{max}}$, so the proportionality constant

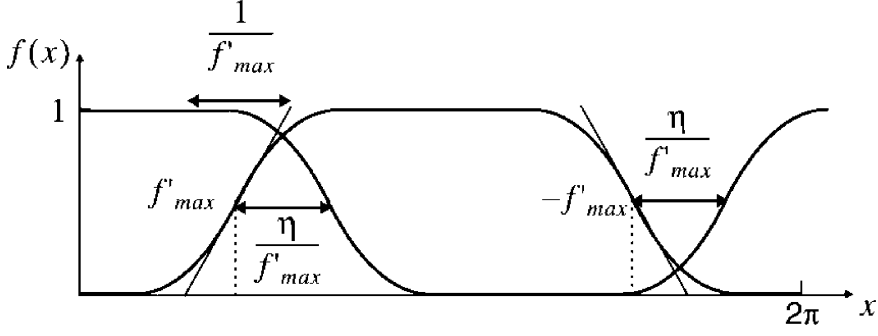


Figure C.1: Relationship between rise and fall times and delay[18].

(η) is a measure of how quickly cascaded stages respond to their inputs. In practice this constant is close to 1[18].

The power consumption is then is:

$$P = 2\eta K q_{max} f_{out} V_{dd} \approx \frac{\mu_{eff} W_{eff} C_{ox} \Delta V^2}{4L} V_{dd} \quad (C.2)$$

This is not just the dynamic $CV_{dd}^2 f$ power, but it also contains a contribution for crowbar currents, current that does not contribute to charging but goes directly from V_{dd} to GND through all transistors. This contribution is approximated to be equal to the charging power[18]. With the values obtained from cadence; $\mu = \frac{\mu_n + \mu_p}{2} = \frac{110 + 190}{2} \frac{cm^2}{V \cdot s} = 150 \frac{cm^2}{V \cdot s}$, $\frac{W_{eff}}{L} = \frac{W_n + W_p}{2L} = \frac{120nm + 240nm}{60nm} = 3$, $C_{ox} = 8.5 \frac{fF}{\mu m^2}$, $V_{dd} = 0.7V$ and $\Delta V = (V_{gs} - V_{th}) = 0.15V$, there is a power consumption of $1.5\mu W$.

C.3 Differential or single ended ring oscillator

A ring oscillator can be used differentially and single ended. With a differential ring oscillator quadrature outputs can be generated. This differential oscillator has a output frequency of[18]: $f_{out} = \frac{I_{tail}}{2\eta K q_{max}}$, with I_{tail} the tail current of the differential pair, η again the proportionality constant, K the number of stages and q_{max} the charge on the output capacitance. This gives a power consumption of:

$$P = K I_{tail} V_{dd} = 2\eta K^2 q_{max} f_{out} V_{dd} \quad (C.3)$$

Increasing the number of stages while keeping a constant output frequency requires an increasing number of increasingly fast stages, meaning that the power goes quadratically up with the number of stages.

To compare the differential and single ended ring oscillator's power consumption, the ratio of the power consumption of a differential and single ended ring oscillator can be found, by approximating the output capacitance and proportionality constant of the differential and single ended equal:

$$\frac{P_{diff}}{P_{single}} = \frac{2\eta K_{diff}^2 q_{max} f_{out} V_{dd}}{2\eta K_{single} q_{max} f_{out} V_{dd}} = \frac{K_{diff}^2}{K_{single}} \quad (C.4)$$

The differential oscillator requires an even or odd number of stages (with a minimum of 2), while the single ended needs an odd (with a minimum of 3) number of stages. The differential oscillator uses therefore at least $\frac{4}{3}$ times more power. When IQ generation is required, 4 stages of differential oscillator are needed, giving $\frac{16}{3}$ more power consumption.

Bibliography

- [1] E. Klumperink, “Oscillators & PAs,” 2016.
- [2] J. van der Tang and D. Kasperkovitz, “Oscillator design efficiency: a new figure of merit for oscillator benchmarking,” in *2000 IEEE International Symposium on Circuits and Systems. Emerging Technologies for the 21st Century. Proceedings (IEEE Cat No.00CH36353)*, vol. 2, 2000, pp. 533–536 vol.2.
- [3] J. Lechevallier, “(ultra) low power oscillators,” Mar. 2017.
- [4] B. Razavi, *RF Microelectronics*. Prentice Hall, 2012.
- [5] W. S. Chang, K. W. Tan, and S. S. H. Hsu, “A 56.5 - 72.2 ghz transformer-injection miller frequency divider in 0.13 μm cmos,” *IEEE Microwave and Wireless Components Letters*, vol. 20, no. 7, pp. 393–395, July 2010.
- [6] L. Zhang, A. Carpenter, B. Ciftcioglu, A. Garg, M. Huang, and H. Wu, “Injection-locked clocking: A low-power clock distribution scheme for high-performance microprocessors,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 16, no. 9, pp. 1251–1256, Sep. 2008.
- [7] B. Razavi, “A study of injection locking and pulling in oscillators,” *IEEE Journal of Solid-State Circuits*, vol. 39, no. 9, pp. 1415–1424, Sep. 2004.
- [8] J. Yin and H. C. Luong, “A 0.8v 1.9mw 53.7-to-72.0ghz self-frequency-tracking injection-locked frequency divider,” in *2012 IEEE Radio Frequency Integrated Circuits Symposium*, June 2012, pp. 305–308.
- [9] T. N. Luo and Y. J. E. Chen, “A 0.8-mw 55-ghz dual-injection-locked cmos frequency divider,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 56, no. 3, pp. 620–625, March 2008.
- [10] Y. Chao and H. C. Luong, “A 440- μw 60-ghz injection-locked frequency divider in 65nm,” in *2013 IEEE Radio Frequency Integrated Circuits Symposium (RFIC)*, June 2013, pp. 111–114.
- [11] K. Takano, M. Motoyoshi, and M. Fujishima, “4.8ghz CMOS frequency multiplier with subharmonic pulse-injection locking,” in *2007 IEEE Asian Solid-State Circuits Conference*, Nov 2007, pp. 336–339.
- [12] Y. C. Lo, H. P. Chen, J. Silva-Martinez, and S. Hoyos, “A 1.8v, sub-mW, over 100% locking range, divide-by-3 and 7 complementary-injection-locked 4 ghz frequency divider,” in *2009 IEEE Custom Integrated Circuits Conference*, Sept 2009, pp. 259–262.

- [13] Y. H. Lin and H. Wang, "A 35.7 -64.2 ghz low power miller divider with weak inversion mixer in 65 nm cmos," *IEEE Microwave and Wireless Components Letters*, vol. 26, no. 11, pp. 948–950, Nov 2016.
- [14] V. F. Kroupa, "Jitter and phase noise in frequency dividers," *IEEE Transactions on Instrumentation and Measurement*, vol. 50, no. 5, pp. 1241–1243, Oct. 2001.
- [15] D. Leenaerts, C. S. Vaucher, and J. van der Tang, *Circuit Design for RF Transceivers*. Springer, 2001.
- [16] X. Gao, E. A. M. Klumperink, P. F. J. Geraedts, and B. Nauta, "Jitter analysis and a benchmarking figure-of-merit for phase-locked loops," *IEEE Transactions on Circuits and Systems*, vol. 56, no. 2, pp. 117–121, Feb. 2009.
- [17] X. Gao, E. A. M. Klumperink, M. Bohsali, and B. Nauta, "A low noise sub-sampling PLL in which divider noise is eliminated and PD/CP noise is not multiplied by N^2 ," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 12, pp. 3253–3263, Dec. 2009.
- [18] A. Hajimiri, S. Limotyrakis, and T. H. Lee, "Jitter and phase noise in ring oscillators," *IEEE Journal of Solid-State Circuits*, vol. 34, no. 6, pp. 790–804, Jun. 1999.
- [19] J. Plusquellic, "CMOS logic structures," Lecture slides "Principles of VLSI Design/VLSI Systems", 2000. [Online]. Available: http://ece-research.unm.edu/jimp/vlsi/slides/chap5_2.html
- [20] J. Yuan and C. Svensson, "New single-clock CMOS latches and flipflops with improved speed and power savings," *IEEE Journal of Solid-State Circuits*, vol. 32, no. 1, pp. 62–69, Jan. 1997.
- [21] B. Razavi, "TSPC logic," *IEEE Solid-State Circuits Magazine*, vol. 8, pp. 10–13, Fall 2016.
- [22] J. Yuan and C. Svensson, "High-speed CMOS circuit technique," *IEEE Journal of Solid-State Circuits*, vol. 24, no. 1, pp. 62–70, Feb. 1989.
- [23] J. M. Rabaey and M. Pedram, *Low Power Design Methodologies*. Springer Us, 2012.