Spread spectrum for clock generators
# Contents

1 Introduction ......................................................... 2

2 Clock properties .................................................. 3

2.1 Harmonic analysis .................................................. 3

3 Spread spectrum modulation ........................................ 6

3.1 Profiles comparison ............................................... 7

3.1.1 Sinusoidal profile .............................................. 7

3.1.2 Square profile .................................................. 8

3.1.3 Sawtooth profile ............................................... 9

3.1.4 Triangular profile ............................................. 9

3.1.5 Hershey-kiss profile .......................................... 10

3.1.6 Conclusions .................................................. 12

4 PLL-based SSCG .................................................... 13

4.1 PLL internals .................................................... 13

4.1.1 Reference input ............................................... 14

4.1.2 Phase-frequency detector ................................... 14

4.1.3 Voltage Controlled Oscillator ............................... 16

4.1.4 Loop filter ................................................... 16

4.1.5 Charge pump .................................................. 17

4.1.6 Feedback divider ............................................. 18

4.2 Direct VCO modulation ......................................... 18

4.3 Fractional-N divider using Σ∆ modulation .................... 20

5 All-digital SSCG .................................................... 23

5.1 Direct DCO modulation .......................................... 23

5.2 A DCDL approach ................................................ 24

5.2.1 Top-level block diagram ................................... 26

5.2.2 Waveform generator ........................................ 27

5.2.3 Digital period synthesizer .................................. 28

6 Conclusion .......................................................... 30
Introduction

As the consumer market demands more computational power and smaller devices, the electronic equipment has to adapt and reach higher signal switching speed than ever. This can have bad consequences on the radiated electromagnetic interferences (EMI), creating troubles for the device itself and for the ones in the nearbies. Every nation apply to some EMI regulations, dictated by international or government organizations. The main organization in the United States is the FCC, while the European Union has adopted the EMC Directive. Both of them limit the amount of radiated EMI, distinguishing among various classes of devices, based on the device’s target user and deployment environment.

Historically, many solutions have been adopted to lower a device’s EMI. For communications systems, the use of coaxial cables and more generally shielded cables has always been common practice and brought good results. The same can be said for desktop PCs, where the big chassis can provide an effective shielding, in case the metal cage is connected to the power supply ground line.

Unfortunately, this solutions are not adoptable on pocket equipment, where strong size constraints are posed. Furthermore, the casing is usually made of plastic, which sometimes requires the use of toxic metal sprays or paints to reduce EMI.

This attracted research interest in finding in-circuit solutions suitable for a wider variety of devices.

One of the main source of EMI in a digital circuit is the clock line, being a constantly oscillating signal with strong requirements on the sharpness of its edges. Furthermore, a clock signal has to reach the entire circuit, and carry a lot of current to allow transistor gates to (dis)charge. For this reason, altering a clock signal properties is a delicate process and needs careful design and testing.

In this thesis I will first discuss the main properties of the clock signal. With that in mind, I will explore the most common clock modulation techniques, along with some implementations proposed in literature.
Clock properties

The clock is one of the main signals in a synchronous digital circuit. As the name suggests, in a synchronous circuit the inputs of the digital functions are evaluated only on discrete time steps, and the circuit state advances like a discrete-time dynamic system, using the clock as a global synchronization signal. This simplifies the design phase compared to an asynchronous circuit, where inputs are continuously evaluated and dangerous race conditions can arise.

A typical synchronous digital circuit is described in figure 2.1. The state transitions happen either on the rising or the falling edge of the clock. For this reason, pushing forward the clock frequency we can achieve faster state transitions, which translates to better computation times. However, flip flops and combinatorial circuits also present some delay time, which impose an upper bound on clock frequency. In fact, switching the clock too soon can lead to spurious state transitions, from which the system can hardly recover.

Harmonic analysis

We can use the Fourier theory to analyze a clock signal spectrum components. The ideal clock signal is a periodic square wave, with 50% duty cycle. For this reason we have frequency components only at multiples of the main frequency, known as harmonic frequencies. Moreover, given the discontinuous nature of the waveform, the harmonics’

Figure 2.1: Sequential digital circuit
amplitude fall like $\frac{1}{n}$ as $f \to \infty$.

A period of a clock signal can be described as

$$f(t) = \begin{cases} -A & -\frac{T}{2} \leq t \leq 0 \\ A & 0 < t \leq \frac{T}{2} \end{cases}$$

(2.1)

The overall clock signal, per the Fourier theory, can be written as:

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left[ a_n \cos \left( \frac{2\pi n}{T} t \right) + b_n \sin \left( \frac{2\pi n}{T} t \right) \right]$$

(2.2)

with

$$a_n = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(t) \cos \left( \frac{2\pi n}{T} t \right) dt$$

(2.3)

$$b_n = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(t) \sin \left( \frac{2\pi n}{T} t \right) dt$$

(2.4)

The $a_0$ term is null as the signal has no DC component.

The $a_n$ terms are also null, as the function $f(t) \cos \left( \frac{2\pi n}{T} t \right)$ is odd around the zero.

We can now calculate the $b_n$ sequence.

$$b_n = 4 \int_{0}^{\frac{T}{2}} f(t) \sin \left( \frac{2\pi n}{T} t \right) dt = 4A \int_{0}^{\frac{T}{2}} \sin \left( \frac{2\pi n}{T} t \right) dt = 4A \left[ -\cos \left( \frac{2\pi n}{T} t \right) \right]_{0}^{\frac{T}{2}} =$$

$$= \frac{2A}{\pi n} \left[ 1 - \cos (\pi n) \right] = \begin{cases} 0 & n \text{ even} \\ \frac{4A}{\pi n} & n \text{ odd} \end{cases}$$

From our calculations, substituting $b_n$ into (2.2) we can write $f(t)$ as:

$$f(t) = \frac{4A}{\pi} \sum_{n=1, \text{odd}}^{\infty} \frac{1}{n} \sin \left( \frac{2\pi n}{T} t \right)$$

(2.5)

It can be noted that only odd harmonics contribute to the signal spectrum. We can confirm our results by plotting the resulting waveform in figure 2.2a, where the number of harmonics is swept, and comparing the waveforms to the ideal square wave. The 5th harmonic truncated square wave will be useful to us, as it will be considered as the clock signal in the next chapters’ simulations.

In figure 2.2b it is also reported a plot of the harmonic’s amplitudes, each normalized to the first one. The dashed horizontal lines represent the 10% and 5% threshold. It should
Figure 2.2: Fourier analysis results

be noted that just past the 21th harmonic we get values below the 5% threshold. This is an expected result, given the sharp waveform of the signal.
Spread spectrum modulation

We adopt a frequency modulation to vary the instantaneous frequency of the clock signal, with the goal of flattening the spikes presented at the fundamental frequency and its multiplies. The resulting signal has the same spectrum energy, but spread on a wider bandwidth.

We now introduce some mathematical notation to better describe this scheme. Let’s call the clock signal \( c(t) \), which is a square wave with an nominal period \( T_0 \) and frequency \( f_0 = 1/T_0. \) Then we have a baseband signal, say \( h(t) \), which we suppose normalized in the \([-1, 1]\) range.

The clock frequency is not fixed, as it changes with time. The instantaneous frequency can be written as:

\[
f_i(t) = f_0 + f_\delta h(t)
\]  

(3.1)

where \( f_\delta \) is called the frequency deviation, and its value determines the maximum distance from the clock initial frequency to the instantaneous frequency. The spread ratio is then defined as \( \delta = f_\delta / f_0 \), and is usually expressed as a percentage.

The function \( h(t) \), called modulation profiles, thus determines how the clock frequency evolves over time. Being the clock signal critical for a synchronous system, we have to choose a profile waveform that doesn’t affect overall system functionalities, in terms of correctness and performances. Eventually the choice can be also be lead by implementation requirements, which we examine later.

In the classification of the various profiles, we must first distinguish between center-spread, up-spread and down-spread ones. As the names suggest, with our modulation profile we can control the direction of the flattened spectrum. Although it might seem obvious, the choice among the threes depends highly on the implementation scenario. If we know that the clock frequency has already been brought to its upper bound, the only solution is to apply a down-spread modulation, avoiding problems related to the clock timing constraints. However, in other scenarios it may be plausible that the clock frequency has been chosen with a tolerance margin, so we can push it above its nominal value, using a center-spread or up-spread modulation.
Those three spread directions directly translate into contraints onto $h(t)$:

- center-spread: $h(t) \in [-1, 1]$
- down-spread: $h(t) \in [-1, 0]$
- up-spread: $h(t) \in [0, 1]$

It also has to be considered that the modulation profile is usually periodic. Thus, the modulation frequency is another degree of freedom. Actually, in literature it is common practice to choose this frequency in the $[30, 100]$ KHz range.

Profiles comparison

Now we describe the most common profiles adopted in literature, and the resulting spectrum obtained through the simulation process in the Matlab environment, which gives a first qualitative appreciation of the benefits.

All the cases covered consider a clock signal with nominal frequency set to $f_0 = 50$ MHz. The clock signal is a square wave like the one presented in the previous chapters, approximated to the 5th harmonic component, to prevent aliasing problems during the simulation. The modulation profiles are periodic, with frequency set to $F_m = 30$ KHz. Moreover, we implement a center-spread modulation, thus the profile amplitude will be in the full $[-1, 1]$ range. The sampling frequency is set as 11 times the clock frequency. The simulation captures one entire modulation period, and then computes the Fast Fourier Transform over the gathered samples. The resulting spectrum is plotted, centered around the first harmonic, with amplitudes normalized to the maximum value.

Sinusoidal profile

The first profile under test is the sinusoidal waveform.

$$h(t) = \sin(2\pi F_m t)$$

As the figure 3.1 suggests, a spread ratio of 1% gives a 17 dB attenuation. Anyhow, some non desirable spurious components are present at the bandwidth edges, with a resulting attenuation of 12 dB. The resulting bandwidth is about 2 MHz.
The spurious components are a direct consequence of the sine waveform. Given that the sine is slower at its peaks, the clock instantaneous frequency spends more time around its higher and lower bounds, as dictated by the modulation waveform.

Better results are obtained with $\delta = 5\%$, where the spurious components are attenuated, and the attenuation is kept at 16 dB on the worst case. The bandwidth is about 8 MHz.

### Square profile

A 50% duty cycle alternating square wave has the following expression.

$$ h(t) = \begin{cases} 
-1 & -\frac{T}{2} \leq t \leq 0 \\
1 & 0 < t \leq \frac{T}{2}
\end{cases} $$

As expected from the previous observation on the sine profile, the square wave presents
Figure 3.3: Sawtooth profile

even higher spikes, as depicted in figure 3.2. The attenuation is about 7 dB with both spreading ratio.

Sawtooth profile

A sawtooth waveform has the following expression.

\[ h(t) = \frac{2}{T} t - \frac{T}{2}, \quad -\frac{T}{2} < t < \frac{T}{2} \]

A sawtooth presents a much better spectrum, reported in figure 3.3, whose amplitude is almost flat everywhere but on its edges. The attenuation is respectively 15 dB and 22 dB, the bandwidths 2 MHz and 8 MHz.

Anyway it can present some implementation difficulties. Its discontinuous nature requires a very precise generation circuit, and the fast clock jump can put under pressure some circuit elements like PLLs, which have to follow the clock frequency and so present very tight settling time. As it will be seen in the next chapters, this is not so cheap to achieve.

Triangular profile

A triangular waveform has the following expression.

\[ h(t) = \begin{cases} 
-\frac{4}{T} t - 1 & -\frac{T}{2} < t < 0 \\
\frac{4}{T} t - 1 & 0 \leq t < \frac{T}{2} 
\end{cases} \]
The resulting spectrum, as reported in figure 3.4, is pretty flat, with no spurs and an attenuation of 12 dB and 20 dB. The bandwidths are 2 MHz and 8 MHz, like the sawtooth profile.

This profile brings significant improvements over the others, and it also doesn’t require complex generation circuitry.

### Hershey-kiss profile

A very common nonlinear profile is the so called Hershey-kiss profile, which is a third order polynomial function. A possible realization, deducted from [10], is given by the following equation

\[
h(t) = \begin{cases} 
0.45\left(\frac{2}{\pi} t - 1\right)^3 + 0.55\left(\frac{2}{\pi} t - 1\right) & -\frac{T}{2} < t < 0 \\
-0.45\left(\frac{2}{\pi} t - 1\right)^3 - 0.55\left(\frac{2}{\pi} t - 1\right) & 0 \leq t < \frac{T}{2}
\end{cases}
\]

which describes the waveform reported in figure 3.5.

The resulting spectra can be found in figure 3.6.

This profile reveals to be as good as the triangular one, providing 11 dB and 18 dB of attenuation. The spectrum is very flat and homogeneous, spread among respectively 2 MHz and 12 MHz.
Figure 3.5: Hershey-kiss waveform

Figure 3.6: Hershey-kiss profile

Simulation results at $\delta = 5\%$

<table>
<thead>
<tr>
<th></th>
<th>Sinusoidal</th>
<th>Square</th>
<th>Sawtooth</th>
<th>Triangular</th>
<th>Nonlinear</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attenuation (dB)</td>
<td>16</td>
<td>7</td>
<td>22</td>
<td>20</td>
<td>18</td>
</tr>
<tr>
<td>Bandwidth (MHz)</td>
<td>8</td>
<td>10</td>
<td>8</td>
<td>8</td>
<td>12</td>
</tr>
</tbody>
</table>
The triangular, sawtooth, and nonlinear profiles have proven to give good attenuation values. At this point, the choice among the three should be lead by other considerations. Among the linear profiles, the triangular one delivers the best results. The nonlinear one, although presenting very similar attenuation results, has the interesting characteristic of spreading the spectrum over a larger bandwidth. This can reveal useful in case of certification problems: in fact the EMI regulations impose, among the other things, the resolution bandwidth of the measuring equipment. For this reason, obtaining a large attenuation over a small bandwidth can lead to a significant underrating of the results. An electronic designer should choose, when possible, the nonlinear profile if he wants to avoid certification issues.
PLL-based SSCG

The most basic spread spectrum clock generators can be built starting from a well-known circuit called phase locked loop, which is used in a wide range of applications, for example:

- FM modulators and demodulators
- clock recovery
- clock synthesis

An overview of PLL’s mechanics is now presented, and then we discuss how it can be modified to achieve the desired frequency modulation.

PLL internals

A phase-locked loop is an electronic feedback system, used to generate a periodic signal which follows the phase of a reference waveform.

Conceptually, the PLL circuit can be represented by the block diagram in figure 4.1. The output of this system is generated by a Voltage Controlled Oscillator, which as the name suggests, generates a waveform whose frequency is related to the voltage applied to its input. The goal is to align the VCO output to the reference signal, eventually

![Figure 4.1: Phase-locked loop](image)
multiplied by a scaling factor of N. To achieve this, the VCO output is compared to 
the reference signal using a Phase Detector, which generates a signal proportional to the 
phase difference. This signal is then low-pass filtered to ensure system stability and the 
desired steady state/transient requirements.

Reference input

The reference input must present very stable oscillations and low jitter. 
For this reason, if an oscillator circuit is going to be used (like LC oscillators), it must 
be tuned around a low frequency, usually around the 30 KHz range, to ensure oscillation 
stability. 
Quartz crystals are used when higher frequencies are required. They exploit the piezo-
electric effect to create a very stable and self-sustaining oscillation. Their resonating 
frequency also depends very loosely on temperature, making them ideal candidates for a 
solid reference signal.

Phase-frequency detector

The output of this stage should be in some way related to the phase difference between 
the two input signals. If digital signals are used, this stage has a very straight-forward 
implementation.

XOR gate

In the most basic case, a XOR gate can serve the purpose. In fact, per the truth table, 
a XOR gate has its output high when its input signals differ from each other. 
Despite its simple implementation, it presents some drawbacks, the biggest one being 
losing the error sign. In fact, when the output is high it cannot be determined which 
signal leads the other. 
It should also be noted that the phase error is not encoded in the amplitude of the output, 
but in the pulse width. 
An example scenario is provided in figure 4.2a, with simulation outputs in 4.2b.
Flip flop configuration

If we want to keep track of which signal is ahead of the other, we need some sort of storage element to preserve state informations. A suitable configuration is reported in figure 4.3a.

As the D flip flops are always loaded with a positive input on the D line, the first signal to reach a rising edge will trigger the output high, and it will be kept high until the other signal also reaches a rising edge, resetting the configuration. The example is depicted in figure 4.3b.

A few properties should be noted: the output is again a pulse, whose width encodes phase error information. Moreover, the output signals cannot be both high, as it should be expected.

An exception is presented when both signals are perfectly aligned, as we hope they will be after a transient phase. While the Q signals propagate through the AND gate and
trigger the reset, the output can present an high spike. So, the steady state output for this configuration is a pulse train, whose frequency is equal to the frequency of the input signals.

**Voltage Controlled Oscillator**

Multiple techniques can be employed to build a VCO. A common approach is to use an LC oscillator circuit. The input voltage can be used to control the capacitance of a variable capacitor, called *varactor*, which is a purposely-built PN-junction exposing very fine control over the junction capacitance when reverse biased. Whatever the technique, in the end we hope to achieve an almost linear output characteristic, at least in the zone of interest.

To have an idea of the kind of filter we need to design, let’s obtain a linear representation of the VCO characteristic.

In a first approximation, we can say that

\[ f_{out}(t) = K_{VCO} V_{ctrl}(t) \]

where \( K_{VCO} \) is the slope of the input-output transfer function, and \( V_{ctrl} \) is the control voltage. We also know that

\[ \theta_{out}(t) = 2\pi \int_0^t f_{out}(\tau) d\tau \]

which gives us

\[ \theta_{out}(t) = 2\pi K_{VCO} \int_0^t V_{ctrl}(\tau) d\tau \]

or in the Laplace-domain

\[ \Theta_{out}(s) = \frac{2\pi K_{VCO}}{s} V_{ctrl}(s) \]

**Loop filter**

The loop filter determines the performances of our PLL.
First of all, as the system will follow a reference phase, which has a linear growth, this is a so-called velocity problem. For this reason, to achieve steady-state null error, the open loop transfer function needs two integrators. As the VCO characteristic already provides one, the filter must provide the other one.

But this is not enough, as the resulting system is unstable. The usual implementation also adds a zero and a pole to the open loop transfer function.

**Charge pump**

A charge pump is basically a voltage-controlled current source. It is typically used as the first stage in the loop filter, to convert phase detector voltage pulses in current pulses. A simple implementation is depicted in figure 4.4a.

The control signals, which corresponds to the outputs of the phase detector stage, decide whether the charge pump should generate a current, and its direction.

The resulting PLL design will thus need at least two capacitors and a resistor, and the most simple implementation is shown in figure 4.4b.

If we ignore the $R_z - C_z$ path, the current will inject charges onto the capacitor $C_p$. If the current is kept constant, the amount of charges, thus the voltage across the capacitor, will be the integral of the PFD error signal. When both inputs are low, the charge can be considered constant.
The transfer function between charge pump current output and VCO input voltage is

\[
\frac{V_c(s)}{I_{CP}(s)} = \frac{R_z C_p s + 1}{s(R_z C_p C_z s + C_p + C_z)}
\]

which gives us three degrees of freedom on the pole-zero placement.

Feedback divider

The feedback divider is needed in frequency synthesis circuits, like an SSCG. As the steady state for the PLL is reached when the phase comparator inputs are in phase, if a divider is present on the feedback path the resulting steady state is

\[
f_{out} = N f_{ref}
\]

which allows to achieve an arbitrary multiple of the reference frequency. The divider can be built using a counter, like in a flip flop cascade configuration (figure 4.5). Anyway, this configuration allows to achieve only integer multiplies of the reference frequency. This means that the frequency resolution of our synthesizer is dependant on the input frequency.

Direct VCO modulation

Direct VCO techniques obtain the desired modulation by modifying the VCO input signal appropriately. One possible implementation of a triangular modulation is proposed in [4] and depicted in figure 4.6. In the new circuitry, a slow square wave is obtained from the reference clock, through a
divider. This square wave is fed into a charge pump, which converts it into current pulses, as shown in figure 4.7a. The authors have shown that the transfer function between the new charge pump and the VCO input is

\[ F(s) \approx \frac{1}{s(C_1 + C_2)} \]

It is an integrator, which is fed with a square wave, producing a triangular waveform. By adding it to the VCO input, we can easily achieve a triangular modulation. The modulation frequency corresponds to the square wave frequency, thus varying the divider modulus we can alter it.
The spread ratio is determined by the charge pump current gain, and as the figure suggests, it can be digitally manipulated. Figure 4.7b shows a programmable current source, made of a set of current mirror stages, each enabled by a control signal. If all stages generate the same current, the output current will be proportional to the number of active stages.

**Fractional-N divider using ΣΔ modulation**

Another proposed technique is the modulation of the feedback divider modulus. However, as already mentioned, an integer-N divider allows the generation of integer multiplies of the input signal, thus providing very poor frequency resolution, not suitable for the generation of continuous waveforms like the linear ones.

For this reason, a fractional-N divider is needed. In its most simple form, it can be implemented using a dual modulus counter, which offers a control signal to choose among two modulus, usually N and N + 1.

If we call the probability of dividing by N \( P_N \), and \( P_{N+1} = 1 - P_N \), on average the dividing factor will be

\[
N.F = NP_N + (N + 1)P_{N+1}
\]

However, periodicity should not be present in the output pattern, otherwise frequency spurs are generated and injected into the PLL filter.

To achieve an apparently-random behaviour, a ΣΔ modulator can be employed. ΣΔ modulation consists in mapping a continuous input signal, say \( x(t) \in [-A, A] \), to a digital oscillating representation, whose average value matches the value of \( x(t) \).

An example is provided in figure 4.8.

This way we can modulate any signal, provided we arrange for a digital representation of it.

A ΣΔ modulator can easily be implemented in a configuration called Digital Phase Accumulator, reported in figure 4.9a.

The input signal is continuously accumulated, and the carry bit is used as the output.

The carry bit thus can drive the control signal of a dual modulus divider, like shown in figure 4.9b, as proposed by [14].
Figure 4.8: $\Sigma\Delta$ modulation example

(a) Digital phase accumulator

(b) $\Sigma\Delta$ divider modulation

Figure 4.9
The up/down counter is useful to generate a triangular waveform. However, using an up counter, we could feed its output as an input for a look-up table, which could allow the generation of an arbitrary waveform.

A drawback of this configuration is the high frequency noise (quantization noise) introduced by the ΣΔ modulator. For this reason, the bandwidth of the PLL loop filter should be tight enough to suppress most of this noise. However, by reducing the bandwidth we are making settling time worse. An acceptable trade-off should be obtained.
All-digital SSCG

The PLL-based approaches, although quiet simple, require the use of analog signals throughout the circuit, mainly to control VCO’s output.

Under the all-digital approach we group all the implementations that make exclusive use of digital signals.

Direct DCO modulation

A possible first solution would be to replace the VCO with a digital equivalent, called Digitally Controlled Oscillator.

A simple digital oscillator circuit can be built from a ring network, where an odd number of inverters are put in series, like in figure 5.1. It’s obvious that the circuit presents no stable states.

The oscillation period will be given by the single inverter propagation delay times the number of stages. So, by dynamically modifying the number of active stages, we could modulate the output period. This concept is easily realizable, using a configuration like the one proposed in [7] and reported in figures 5.2a, 5.2b.

We can distinguish two stages: the coarse-tuning and the fine-tuning stage. The coarse-tuning inputs allow to select the length of the propagation path. Let’s suppose for example that all inputs but the RESET are low. The signal propagates through the first NAND, which is effectively an inverter; than it flows through the inverting multiplexer which is another inverting stage; at the end of the path another NAND gate used as an inverter is encountered. This path corresponds to the minimum delay period, or maximum oscillation frequency.

This signal is then fed into the fine-tuning stage, along with the output of the delay cell

![Figure 5.1: Ring oscillator](image)

23
preceeding the last one. The fine-tuning stage is made of two parallel buffer array, driven by complementary signals. The frequency output will be closer to one input or the other, depending on which array has the largest number of buffers activated.

A DCO-based architecture is presented in figure 5.3.

We can notice that the analog filter has been converted to a digital one.

The SSC controller implements the modulation profile, adding the desired waveform on top of the filter output.

The generated control code is fed to a $\Sigma\Delta$ modulator to improve the DCO resolution.

### A DCDL approach

A *Digitally Controlled Delay Line* is basically a digital circuit whose output is a delayed replica of its input. The delay amount is controllable through some digital input signals.
The implementation relies on a concept similar of that of the DCO explained before, but this time the configuration must not present oscillations, so the inverting stages are not closed in a loop, and the number of them is even. See figure 5.4a, taken from [2], for an implementation.

Through the control pins, we can select the length of the propagation chain, and so the total input-output delay.

The transfer function is plotted in figure 5.4b, where is should be noted that this structure presents a small asymmetry between even and odd control codes. This is caused by process variations like asymmetries in the $tp_{LH}$ and $tp_{HL}$ of the CMOS inverters.

Better results are obtained using the architecture proposed in [1] and reported in figure 5.5. The structure relies only on NAND gates, and the gates marked as D are added for load balancing purposes. There are two control codes: S and T. S inputs are driven by a thermometric code ($S_i = 0$ for $i < c$ and $S_i = 1$ for $i \geq c$); T input are driven by a one-cold code ($T_i = 1$ for $i \neq c + 1$ and $T_i = 0$ for $i = c + 1$).

This forms the basis for the SSCG proposed in [2], which achieves an arbitrary-waveform.
modulation of the clock signal.

In particular, it can perform period modulation, which can be expressed as

\[ T_{\text{out}}(t) = T_0 + \Delta T h(f_m t) \]

where \( h(t) \) is the modulation profile. However, for small spreading ratio, the period modulation can be considered equivalent to frequency modulation.

**Top-level block diagram**

The *Waveform Generator* generates an arbitrary waveform, based also on the spread ratio and modulation frequency passed as inputs. Then the *Digital Period Synthesizer* converts the input waveform into control signals for the DCDL. The DCDL outputs are fed in a XOR gate, which will generate the output signal.

By using a XOR gate (or eventually a XNOR), the output presents a transition when one of the inputs changes value. For this reason, by appropriately feeding the DCDLs, we have full control on the output clock timing.

The measurement unit accounts for PVT variations, and its implementation is omitted here for the sake of simplicity.
The two DCDLs work on the opposite edges of the clock, while all the modulation logic is executed on the rising edge.

**Waveform generator**

The waveform generator is built of three stages: an *Overflowing Accumulator*, an *Interpolator* and a last *Scaling* block.

The overflowing accumulator is basically like a digital phase accumulator, generating a periodic sawtooth waveform, with period equal to the modulation waveform period.

This sawtooth is used as a phase reference in the interpolator, which is presented in two different implementations, respectively the upper and lower path.

The upper path allows to generate a piece-wise linear approximation of an arbitrary waveform. The approximation will follow the equation

\[ h(x - x_k) \approx A_k + B_k (x - x_k) \]

for the k-th segment, where the \( A_k \) and \( B_k \) coefficients are stored inside the SRAM block.

The lower path implements a simple triangular waveform, and requires a much easier circuit.

The output is then scaled, and the result represents the desired output period normalized to the reference period, obeying the equation
Digital period synthesizer

The digital period synthesizer can be conceived as a finite state machine, where the state
variable is the interval between the next reference clock rising edge and the next output
clock rising edge to be generated; let’s call it $\tau$. A block diagram is provided in 5.8a.

The FSM can now work in four different modes:

- $\tau > 1$: the next rising edge to be generated is farther than the next reference clock rising edge. We can be idle for this cycle, and decrement $\tau$ by a unit.

- $0.5 \leq \tau \leq 1$: the next rising edge must be generated in the second half-cycle of the reference clock. So the FSM switches the $S_{FE}$ signal and sets $W_F = \tau - 0.5$. The next state is $\tau(k+1) = \tau(k) - 1 + T_{out}/(2T_{ck})$. See figure 5.8b.

- $0 < \tau \leq 0.5$: the next rising edge must be generated in the first half-cycle of the

\[
\frac{T_{out}}{T_{ck}} = \frac{T_0}{T_{ck}} + \frac{\Delta T}{T_{ck}} h(f_m t)
\]
reference clock. Two situations can arise now, depending on the value of $T_{out}/T_{ck}$:

- Only the rising edge must be generated in the first half-cycle of the reference clock. The FSM switches $S_{RE}$ and sets $W_R = \tau$. The next state is $\tau(k + 1) = \tau(k) - 1 + T_{out}/(2T_{ck})$. See figure 5.8c.

- Both the rising and the falling edge must be generated into this reference clock period. Thus the FSM switches both $S_{RE}$ and $S_{FE}$, and updates $W_R = \tau$ and $W_F = \tau - 0.5 + T_{out}/(2T_{ck})$. The next state will be $\tau(k + 1) = \tau(k) - 1 + T_{out}/(T_{ck})$. See figure 5.8d.
Conclusion

Spread-spectrum clock generation has become a standard solution to reduce EMI, because of its cheapness and effectiveness.

Various modulation profiles have been simulated and compared, showing how the triangular and the nonlinear profiles give the best results.

Then some spread-spectrum clock generators architectures have been presented.

On one side, we have the PLL-based approaches, which are the oldest one. They require the generation of analog signals to achieve the desired modulation; however their implementation is pretty well understood and solid.

When using ΣΔ modulation, particular care should be taken when designing the loop filter. A trade-off must be reached between noise suppression and loop performances.

On the other side, all-digital approaches are generally more flexible.

With the modulated DCO architecture, we find again much of the problems and solutions typical of a PLL-based implementation.

With the DCDL architecture, we can fully exploit the flexibility of digital design, obtaining a much more complex circuit, but also more powerfull and easily integrable. Arbitrary modulation waveforms can be used, and most of the problems derived from PVT variations can be reduced using a measurement logic. However, it just does clock modulation. Thus an high frequency clock source must be available, with exactly 50% duty cycle, which can require once again the use of an external PLL for clock synthesis.
Bibliography


