

## Chapter 2

# Architectures for Frequency Synthesizers

### 2.1 Overview

This chapter starts with an overview of the conventional frequency synthesis techniques as well as the hybrid architectures that can be used to implement a fast settle frequency synthesizer. This is followed by a proposed architecture for an inductorless fourteen-band CMOS frequency synthesizer for MB-OFDM UWB applications.

### 2.2 Frequency Synthesis Techniques

The majority of frequency synthesis techniques fall into two categories: either direct frequency synthesis or indirect frequency synthesis. The direct frequency synthesis technique is based on using digital techniques to achieve fine frequency steps. While the achievable switching speed in this technique is fast, this technique is limited to lower frequencies due to the speed limitation of the digital circuits. Indirect frequency synthesis is based on using a phase-locked loop (PLL) to generate multiples (integer or non-integer) of a reference frequency. PLL-based techniques can synthesize higher frequency carriers; however, they obtain a slower settling and switching time. A combination of direct and indirect frequency synthesis techniques along with some other additional circuits, such as frequency mixers, can be used to achieve the required frequency of operation and switching speed in a frequency synthesizer. This section presents a detailed study of different architectures for a frequency synthesizer that can be used to implement a fast-settling fourteen-band frequency synthesizer that meets the specification stated in Sect. 1.4.

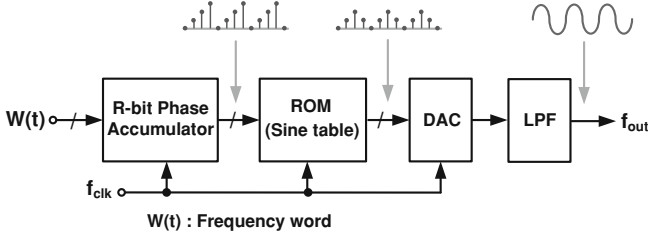


**Fig. 2.1** Block diagram of a typical direct digital frequency synthesizer (DDFS)

### 2.2.1 Direct Digital Frequency Synthesizer

Direct digital frequency synthesis (DDFS) is a technique to synthesize frequencies and achieve a very fast settling time. A DDFS is composed of an accumulator, a ROM-based lookup table, and a digital-to-analog converter (DAC), and is usually, followed by a low-pass filter. A block diagram of a typical direct digital frequency synthesizer (DDFS) is shown in Fig. 2.1. The output of the accumulator is a discrete-time ramp representing phase information. As can be seen in Fig. 2.1, the accumulator is followed by a sine lookup table that converts the discrete-time ramp phase output of the accumulator into a discrete-time sine wave. Finally, the DAC and low-pass filter generate a continuous time waveform that is suitable for frequency translation applications. The oscillation frequency of the ramp at the output of the accumulator is [1]

$$f_{\text{out}}(t) = \frac{W(t) \cdot f_{\text{clk}}}{2^R} \quad (2.1)$$

where  $f_{\text{clk}}$  is the clock to the accumulator,  $W(t)$  is the input to the accumulator, and  $R$  is the number of accumulator bits. As a result, based on the input frequency word ( $W(t)$ ), the output of the accumulator is at a frequency that is a fraction of the master clock.

A block diagram of a (DDFS)-based frequency synthesizer for MB-OFDM UWB is shown in Fig. 2.2. It consists of a DDFS and a single-sideband (SSB) mixer for frequency translation. As discussed earlier, a DDFS can achieve a very fast frequency switching time. However, implementing a DDFS for UWB requires high-speed digital circuits and implementation of such high-speed logic is not very straightforward in current CMOS technology and also requires a very high power consumption. As a result, a hybrid architecture like the one shown in Fig. 2.2 is required to meet the frequency agility requirement of UWB and also to relax the design of the DDFS. Depending on the coverage of the UWB frequency synthesizer of Fig. 2.2, the DDFS needs to generate some integer multiples of the channel spacing (528 MHz), and the SSB mixer upconverts the output of the DDFS to the actual band center frequencies. Using an LO frequency that lies between two-band center frequencies slightly relaxes the DDFS coverage since it only needs to generate odd integers of 264 MHz (half the channel spacing) to achieve the same coverage. Although the architecture of Fig. 2.2

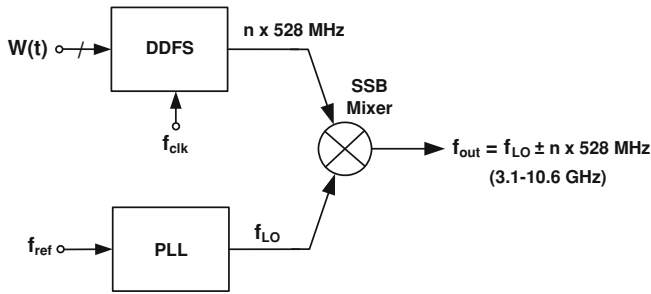


**Fig. 2.2** A simplified block diagram of a DDFS-based frequency synthesizer for UWB

relaxes the speed requirements of DDFS, the accumulator, the ROM, and the DAC need to run at several giga hertz in order to generate three times the channel spacing at the output of the DDFS, which is challenging.

Another challenge is to mitigate the spurious tones at the output of the DDFS [1, 2]. Moreover, the architecture of Fig. 2.2 requires a linear SSB mixer to suppress the mixing spurious response at the output. Implementing a linear SSB mixer in a low-voltage CMOS process poses several challenging design constraints, and generally increases the power consumption. In addition, a SSB mixer requires quadrature phases of both mixing signals with sufficiently good amplitude and phase accuracy to achieve adequate side-band cancellation. So the DDFS needs to provide quadrature output phases. SSB mixers are integral parts of many different fast hopping frequency synthesizers, as will be discussed later in this chapter.

In summary, high-speed logic requirement makes the architecture of Fig. 2.2 less attractive, and also makes it very difficult for this architecture to cover a wide span of the UWB spectrum.

In special cases the DDFS of Fig. 2.1 can be simplified to a cascade of the ROM-based lookup table, DAC, and low-pass filter, which reduces the need to implement the accumulator. This can become more important at high frequencies where the implementation of high-speed logic is challenging. An example of this technique is presented in [3], where a simplified version of a DDFS is implemented by storing sine and cosine waveforms at frequencies of  $\pm 264$  and  $-792$  MHz in two ROM-based lookup tables and selecting the appropriate frequency while using a fixed sampling rate of 4.224 GHz. The block diagram of this frequency synthesizer is shown in Fig. 2.3 [3]. As can be seen from Fig. 2.3, the outputs of the lookup tables go to the input of two four-bit current-steering DACs and low-pass filters. This frequency synthesizer uses a fractional-N PLL and a divide-by-two to generate the quadrature phases of 4.224 GHz. These signals are used in two SSB mixers to upconvert the outputs of the low-pass filters to the desired LO frequency for Band Group One. The frequency synthesizer of Fig. 2.3 is implemented in a 0.13  $\mu\text{m}$  CMOS technology and performs frequency hopping within the center frequencies for Band Group one in 2 nS.

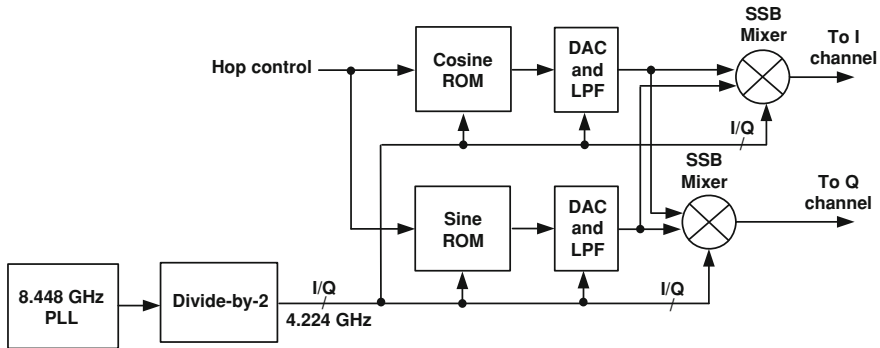


Fig. 2.3 Block diagram of the UWB frequency synthesizer of [3]

The  $\pm 264$  MHz signal for the operation of the SSB mixer of the frequency synthesizer of Fig. 2.3 could be directly derived from the PLL output (8.448 GHz) using frequency dividers. However, generating the  $-792$  MHz signal from the PLL output is not straightforward and requires another SSB mixer, or a frequency divider with a fractional division ratio. Using the DDFS to generate  $\pm 264$  and  $-792$  MHz has the advantage of suppressing the second and the third harmonics of these signals, while the remaining dominant fourth harmonic is suppressed by the low-pass filter at the output of the DAC in the DDFS [3]. In addition, the inductive tuned load in the SSB mixer and in the LO buffer after the SSB mixer provide further suppression of the remaining spurious tones.

### 2.2.2 Phase-Locked Loop-Based Approaches

A PLL is one of the most common ways to synthesize a frequency using a reference signal. Figure 2.4 shows a simplified block diagram of a PLL for frequency synthesis applications. The PLL of Fig. 2.4 consists of a phase-frequency detector (PFD), a charge-pump, a loop filter, and a voltage-controlled oscillator (VCO) in the forward path, and a frequency divider in the feedback path. The PLL aims to lock the phase of the feedback signal to the phase of the reference signal. Therefore, a phase detector is required to compare the phases of these two signals. However, a PFD can also be used to enable frequency detection as well [4]. A tri-state PFD is used in the block diagram of Fig. 2.4 [4, 5]. The output of the PFD determines the control voltage of the VCO through a charge-pump and loop filter. The PLL of Fig. 2.4 uses a second-order loop filter. Additional poles and zeros can be added to the loop filter to control the dynamic properties of the PLL as well as its stability. The frequency divider that is used in the feedback path indirectly leads to frequency multiplication at the output of the PLL, hence the output frequency of the PLL of Fig. 2.4 when phase locked is

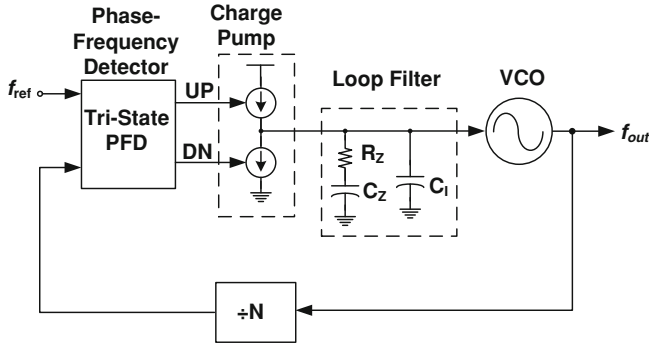
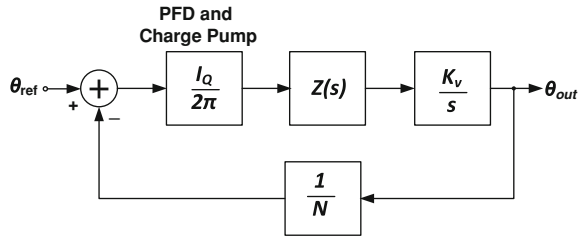


**Fig. 2.4** Block diagram of a PLL-based frequency synthesizer with charge-pump

**Fig. 2.5** Linear phase domain model for a charge-pump based PLL



$$f_{out} = N f_{ref} \quad (2.2)$$

where  $f_{ref}$  is the frequency of the reference frequency and  $N$  is the total division ratio used in the PLL.

To analyze the PLL of Fig. 2.4 we use a linear model for the phase of the input and output signals of the PLL. A linear model for the phase of the input and output signals of the PLL of Fig. 2.4 is shown in Fig. 2.5 [6]. The PLL of Fig. 2.4 tracks a phase step input if its loop gain has at least one integrator, and tracks a frequency step input if its loop gain has at least two integrators [7]. Therefore, a type-II PLL (a PLL with two poles at DC in its loop gain) is required to track a frequency with no steady-state error.

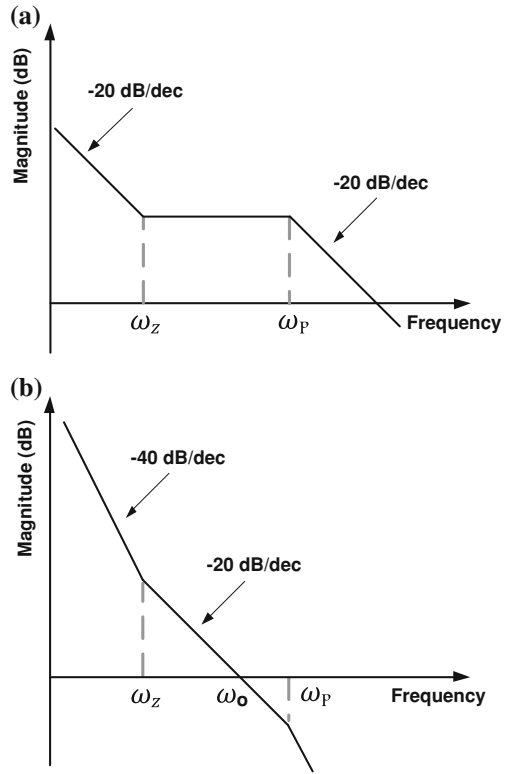
Based on this model, the loop gain of the PLL of Fig. 2.4 can be expressed by

$$G(s) = \frac{I_Q}{2\pi} Z(s) \frac{K_v}{s} \frac{1}{N} \quad (2.3)$$

where  $I_Q$  is the charge-pump current,  $Z(s)$  is the impedance of the loop filter,  $K_v$  is the VCO gain, and  $N$  is the total division ratio in the loop.

For a type-II PLL the loop filter in its simplest form requires a pole at DC and a LHP zero for stability. However, for practical reasons it also needs a higher frequency pole to suppress the reference signal. This loop filter is shown in Fig. 2.4. The zero and the pole frequencies of the loop filter are chosen to achieve the desired loop

**Fig. 2.6** **a** Magnitude of the transfer function of the loop filter used in the PLL of Fig. 2.4 and **b** magnitude of the loop gain of a charge-pump-based type-II PLL



bandwidth and phase margin. The impedance of the loop filter can be expressed by

$$Z(s) = \frac{1}{(C_Z + C_I)s} \cdot \frac{1 + R_Z C_Z s}{1 + R_Z (C_I || C_Z)s}. \quad (2.4)$$

In (2.4)  $C_I$  and  $C_Z$  are the shunt and the series capacitors in the loop filter and  $R_Z$  is the series resistor with  $C_Z$  to implement the zero of the impedance. As mentioned earlier, the impedance zero at the VCO input voltage improves the stability of the PLL, and is mainly determined by the required phase margin, while the location of the pole is determined by the required attenuation of the reference spurious tone. Therefore, the zero is chosen at a frequency that is a fraction of the unity gain frequency of the loop gain ( $\omega_o$ ) while the pole is chosen at a frequency higher than the unity gain frequency of the loop gain to minimize the effect of this pole on the phase margin of the PLL. The magnitude of the impedance of the loop filter and also the magnitude of the linear loop gain of the PLL of Fig. 2.4 are shown in Fig. 2.6a and b, respectively, where  $\omega_Z$  is the zero of the impedance and is given by  $1/R_Z C_Z$ ,  $\omega_P$  is the pole of the impedance and is given by  $1/R_Z (C_I || C_Z)$ , and  $\omega_o$  is the unity gain frequency of the loop.

It can be concluded from (2.4) that if  $(\omega_Z/\omega_o)^2 \ll 1$  and  $(\omega_o/\omega_P)^2 \ll 1$ , the magnitude of the impedance of the loop filter in the vicinity of  $\omega_o$  can be expressed by

$$|Z(j\omega_o)| \approx R_Z. \quad (2.5)$$

Consequently, from (2.3) and (2.5) the unity gain frequency of the transfer function of the loop gain of the PLL of Fig. 2.4 can be found as follows.

$$\omega_u \approx \frac{I R_Z K_v}{2\pi N} \quad (2.6)$$

The 3 dB bandwidth of the PLL of Fig. 2.4 depends on the unity gain frequency of its loop gain as well as its phase margin. For a phase margin of  $60^\circ$  the 3 dB bandwidth of the PLL of Fig. 2.4 is the same as the unity gain frequency of its loop gain and can be obtained from (2.6). In the design of Fig. 2.4, the loop filter parameters are chosen to obtain the desired phase margin and 3 dB bandwidth. The design and optimization of higher order loop filters are discussed in [8].

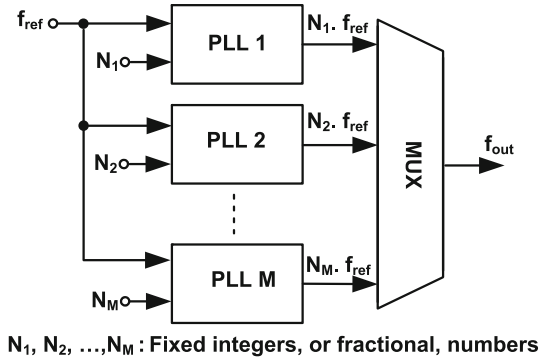
A wideband PLL is needed to meet the fast settling time requirement. But the bandwidth of a PLL also needs to be a fraction of the frequency of the reference signal due to stability reasons [6], and in practice this ratio is chosen to be no greater than 0.1. This can cause several practical issues. The first is settling time, and following approximation can be used to find the settling time of a PLL with a bandwidth of  $BW$  and a frequency step of  $f_{\text{step}}$  [9].

$$T_{\text{settling}} = \frac{1}{BW\zeta} \ln \left( \frac{f_{\text{step}}}{f_{\text{error}}} \right) \quad (2.7)$$

In (2.7)  $\zeta$  is the damping factor and is function of the phase margin, and  $f_{\text{error}}$  is the acceptable frequency error when PLL is settled [10]. For a phase margin of  $50^\circ$ , which chosen for fast settling [9],  $\zeta$  is equal to five. Nonetheless, substituting a step frequency of 528 MHz and an frequency error of 1 kHz along with the optimal value of  $\zeta$  in (2.7) leads to a reference frequency of greater than 2.8 GHz to obtain a settling time of 9.5 nS. However, in practice a slightly higher reference frequency is needed, since (2.7) is a linear approximation. To sum up, such a high reference frequency makes the implementation of the PFD or the phase detector very difficult. In addition, a reference frequency greater than the PLL step frequency mandates the use of a fractional-N PLL, which brings other challenges such as implementation of a delta-sigma modulator that operates at this reference frequency. These limitations make any single PLL frequency synthesizer approach unfeasible for these applications.

Several approaches have been suggested to overcome the stringent settling requirements of a PLL for UWB frequency synthesizer applications. One technique is based on using a dedicated PLL for each band, and using a multiplexer to select one of the PLL outputs, as shown in Fig. 2.7. In this technique each PLL settles to the desired carrier frequency of one of the bands at startup, and later the appropriate carrier frequency is selected by a multiplexer. Hence, there is no stringent settling requirement

**Fig. 2.7** Block diagram of a UWB frequency synthesizer based on multiple fixed-frequency PLLs



for the PLLs and the agility problem is reduced to the design of a multiplexer that performs the switching in less than 9.5 nS, which is a straightforward problem. An example of that is shown in Chap. 6. In addition, this technique does not require any SSB mixer, hence there is no mixer related linearity and spurious tone problem.

An example of this method to generate band center frequencies for Band Group One is presented in [11]. Other examples of this technique are the frequency synthesizers presented in [12, 13] that use three separate PLLs to generate band center frequencies for Band Groups one and three. Although this technique provides a solution for the agility problem of the conventional PLL, it usually leads to an increase in the die area and power consumption. Furthermore, using a dedicated conventional PLL for each band becomes very challenging for covering a larger number of bands due to practical issues of having several VCOs running simultaneously on one chip. The issues include VCO pulling and spurious tones due to different coupling mechanisms between the on-chip VCOs.

Another technique to overcome the agility requirement of UWB is based on using two fast-settling PLLs. In this technique while one PLL is providing the center frequency for that time slot, the other PLL is settling to the next carrier frequency. Consequently, the allotted time for PLL settling is increased to one symbol period, which is 312.5 nS. As a result, it is feasible to implement a phase detector and a charge-pump that operate at that slower speed, as shown in [9]. A simplified block diagram of this technique is shown in Fig. 2.8. As can be seen from this figure, this synthesizer does not require any SSB mixer, hence there is no mixer-related linearity and spurious tone issue.

An example of a UWB frequency synthesizer based on this technique is presented in [9] and is shown in Fig. 2.9. As can be seen from Fig. 2.9, this frequency synthesizer consists of two fast-settling PLLs, two frequency dividers (divide-by-two), and a four-to-one multiplexer. Although the frequency synthesizer of Fig. 2.9 generates seven band center frequencies for UWB, it is designed for an earlier frequency plan for MB-OFDM UWB [14] and only the PLL output frequencies after divide-by-two (3.432, 3.96, and 4.488 GHz) are compliant with the latest UWB frequency plan.

**Fig. 2.8** Block diagram of a frequency synthesizer for UWB based on two fast-settling PLLs

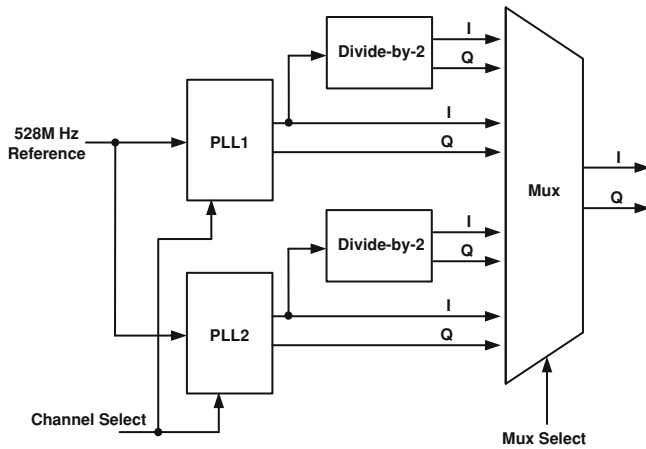
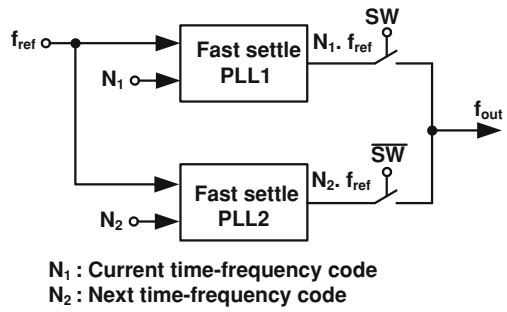


**Fig. 2.9** Block diagram of the UWB frequency synthesizer of [9]

The fast-settling PLLs used in the UWB frequency synthesizer of Fig. 2.8 are identical, and use a quadrature wideband VCO with a tuning range of 6.3–9.0 GHz, and a third-order on-chip active loop filter. Each PLL uses a reference frequency of 528 MHz, and is designed to achieve a bandwidth of 26 MHz with a phase margin of  $50^\circ$  for fast-settling [9]. The details of the circuit implementation of the PLLs of Fig. 2.8, including the design of the PFD and the charge-pump, are discussed in [9]. The frequency synthesizer of Fig. 2.9 was implemented in  $0.18\ \mu\text{m}$  CMOS technology and settles in 150 ns and achieves a phase noise of roughly  $-110\text{ dBc/Hz}$  at 1 MHz offset and sidebands of less than  $-52\text{ dBc}$ .

In short, the UWB frequency synthesizer of Fig. 2.8 overcomes the fast settling requirement for UWB and does not have the spurious tones of SSB mixer-based techniques. However, it is very difficult to expand this architecture to cover the entire UWB span since the VCOs cannot cover such a wide frequency range. On top of that, extending the frequency range of a VCO usually leads to degradation of its phase noise. Alternatively, multiple VCOs can be used to cover a wide range, but

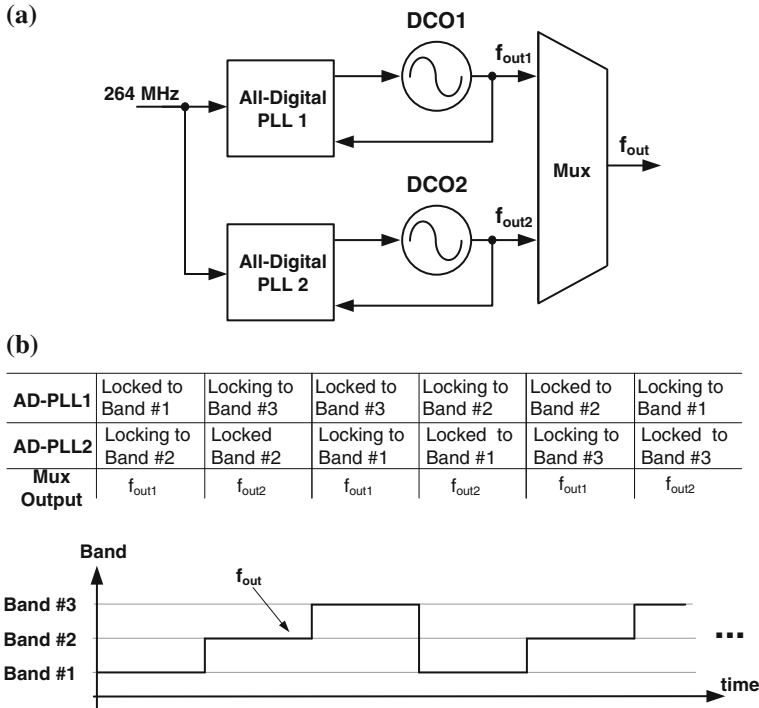


**Fig. 2.10** **a** Block diagram of the frequency synthesizer used for each Band Group in the UWB frequency synthesizer of [15], **b** principles of operation of this frequency synthesizer in achieving fast frequency switching

this demands a larger die area. As a result, it is not very easy to extend the coverage of the frequency synthesizer of Fig. 2.8 to cover the entire UWB frequency range.

### 2.2.3 Digital Phase-Locked Loop-Based Approaches

Another PLL-based architecture to implement a fast agile frequency synthesizer for UWB is presented in [15]. This architecture is similar to what was presented in [9]. However, a more compact implementation of the frequency synthesizer is achieved using All-Digital PLLs (AD-PLL). In addition to covering Band Group Three in [9], Band Groups One and Six are also supported in the frequency synthesizer of [15]. This synthesizer is compliant with WiMedia v1.2 UWB PHY that requires operation in Band Groups One, Three, and Six.

Figure 2.10a shows the implementation of the synthesizer for each Band Group. As can be seen in Fig. 2.10a, two integer-N AD-PLLs are used to generate the center frequency for each band. While one PLL is providing the center frequency for

that time slot, the other PLL is settling to the next carrier frequency. This technique increases the settling time budget for each PLL from 9.5 to 312 nS (time required for band switching) similar to what was discussed in the previous section. The outputs of two Digital PLLs are multiplexed and the appropriate output is selected at the time of frequency hopping. This switching can easily take place in less than 9.5 nS. The principles of operation of this frequency synthesizer within each Band Group is shown in Fig. 2.10b. The complete implementation of this UWB frequency synthesizer is shown in Fig. 2.11. As can be seen in this figure, the two-PLL frequency synthesizer of Fig. 2.10a is used for each Band Group and the entire frequency synthesizer consists of six AD-PLLs to cover three Band Groups. Using Digitally Controlled Oscillators (DCOs) helps to eliminate the inductors that are required inside VCOs. Each DCO consists of a cascade of four differential pair amplifiers and generates output signals with  $45^\circ$  phase shift, and achieves a phase noise of  $-84$  dBc/Hz at 1 MHz offset, while dissipating 8 mA at the highest frequency of operation [15].

In addition to the digital PLLs, an analog PLL is used to generate the reference frequency for the digital PLLs as well as the clock for the operation of the digital baseband. The reference frequency of 264 MHz enables the use of integer-N PLLs to implement the digital PLLs of Fig. 2.11, since all the LO frequencies in the UWB spectrum are integer multiples of 264 MHz. In addition, the fact that the LO frequencies are odd integer multiples of the reference frequency simplifies the implementation of the Time-to-Digital Converter (TDC) required for the digital PLLs [15]. Finally, the relaxed accuracy and phase noise requirement of the synthesized LO frequencies ( $\pm 20$  ppm and  $-100$  dBc/Hz at 1 MHz offset according to [16]) eases the implementation of digital PLLs. These features together facilitate a compact implementation of digital PLLs [15].

In spite of the poor phase noise performance of DCOs, using a wide PLL bandwidth helps to achieve acceptable phase noise for the operation of UWB radio.

Using multiple digital PLLs helps to eliminate the use of frequency dividers and SSB mixers to generate the center frequencies for different bands, as will be discussed in Sect. 2.2.8.

### 2.2.4 Sub-Harmonic Injection-Locking Techniques

One alternative way to overcome the PLLs settling time limitations in a UWB frequency synthesizer is by synchronizing an oscillator to an external signal [17, 18]. This synchronization can be done using a synchronization signal that has the same fundamental frequency as the target frequency, or by using a synchronization signal that is a subharmonic of the target frequency. The former approach is usually exploited to generate multiple phases of a carrier frequency to be used in mixers for frequency translation (up-conversion or down conversion), while the latter approach is often used for frequency multiplication, similar to using a PLL. It can be shown that in spite of the poor close-in phase noise of a free-running oscillator, an oscillator under injection-locking tracks the phase noise of the input source [19].

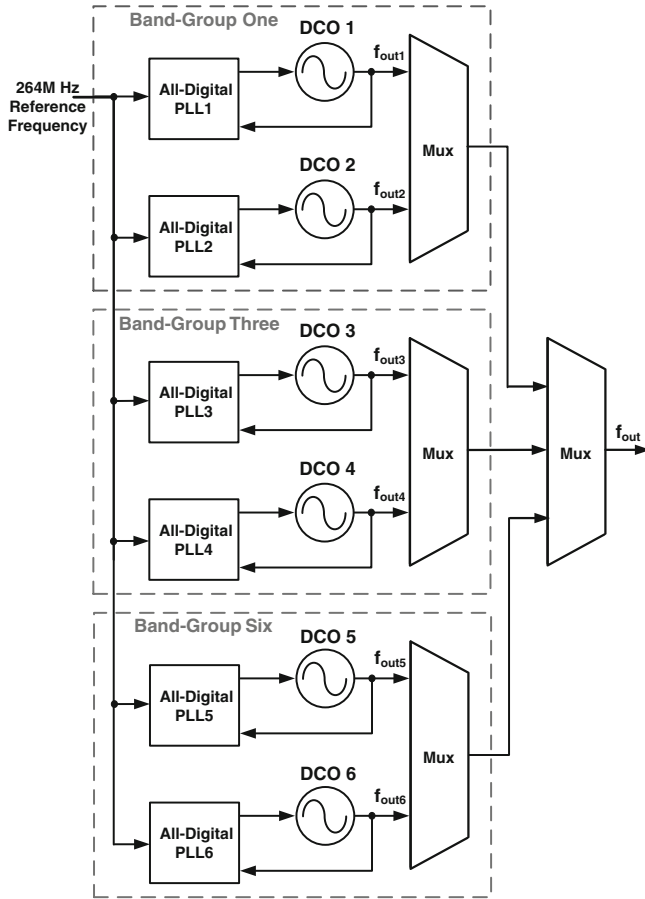
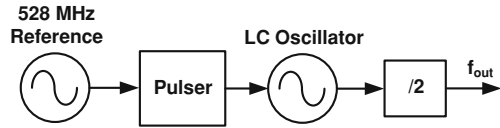


**Fig. 2.11** Block diagram of the complete implementation of the frequency synthesizer of [15]

Therefore, to obtain acceptable phase noise, it is sufficient to use a clean synchronization signal. When this technique is used in a wideband frequency synthesizer the oscillator must have a wide enough tuning range. Alternatively, multiple oscillators need to be used to cover the entire frequency span. In this case, the settling time constitutes of two components: (1) the time required for tuning the free running frequency of the oscillator through coarse tuning or any other mechanism, and (2) the locking time required for synchronization of the oscillator to the injection signal [20]. As discussed in [20], a proper design can easily meet the settling time requirement for a UWB frequency synthesizer.

An example of a frequency synthesizer for UWB based on subharmonic synchronization, or injection-locking, is presented in [17, 20]. This frequency synthesizer uses an LC oscillator with a wide tuning range to generate the center frequencies for Band Group Six. In this frequency synthesizer, the oscillator can be injection-locked to a low phase noise reference signal that is a subharmonic of the desired frequency.

**Fig. 2.12** Block diagram of a frequency synthesizers for UWB based on sub-harmonic injection-locking of an oscillator



A simplified block diagram of the frequency synthesizer of [17] is shown in Fig. 2.12. As can be seen from Fig. 2.12, an oscillator is followed by a divide-by-two to generate the quadrature output phases of the LO signal. Hence, the oscillator needs to oscillate at twice the frequency of the band center frequencies. In order to generate all the band center frequencies for Band Group Six, the fundamental frequency of the injection signal needs to be a common divisor of all these possible output frequencies. Considering the frequency synthesizer of Fig. 2.12, the greatest common divisor (GCD) for all the output frequencies is 528 MHz. Using a synchronization signal with a fundamental frequency that is the common divisor of all the output frequencies, but is not the GCD, can cause several practical issues: (1) a higher order harmonic of the fundamental frequency of this synchronization signal is needed to injection-lock the oscillator to the target desired frequency. It is more difficult to get the adequate power required for injection-locking at higher order harmonics. (2) Other harmonics will appear at the output through different coupling mechanisms, and will generate in-band spurious tones.

The synthesizer of [17] is implemented using an LC oscillator that uses capacitive coarse tuning to cover the frequency span required for Band Group Six. An external 528 MHz signal is used to generate the synchronization signal that injection-locks the oscillator. To create a synchronization signal that is rich in harmonics, this 528 MHz signal goes through a pulser circuit shown in Fig. 2.13. The synchronization can be done using both single-node and differential injection.

The frequency synthesizer of [17] achieves a very compact implementation on Silicon using a 90 nm CMOS technology and settles in less than 4 nS while it generates the band center frequencies for Band Group Six.

The frequency synthesizer of [18] uses similar principle for injection-locking a free-running oscillator. However, this frequency synthesizer uses a signal at the LO frequency or half the LO frequency for this purpose. Consequently, there is not an issue with spurious tones at the center frequencies of other UWB bands and it can achieve a much better spurious level. Figure 2.14 shows the top level block diagram of the frequency synthesizer of [18]. This frequency synthesizer generates the center frequencies for Band Groups One, Three, and Four. As can be seen from Fig. 2.14, a four-stage ring-oscillator is used to cover the entire frequency span of these three-Band Groups. Each delay cell consists of a differential pair and a programmable PMOS load through which the oscillator can be coarse tuned to different frequencies.

To achieve an acceptable phase noise at the output of the injection-locked ring-oscillator, the synchronization signal can be generated using a PLL with an LC-VCO. However, to overcome the fast settling time requirements, three separate PLLs are needed. These three PLLs are identical, and cover a frequency range of

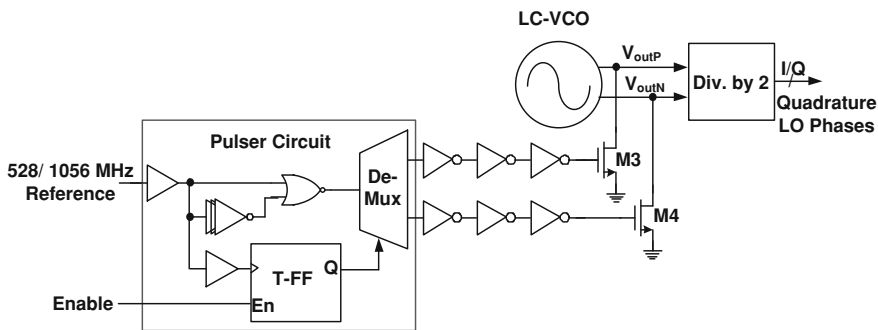


**Fig. 2.13** Block diagram of the pulser used in the UWB frequency synthesizer of [17] to generate high harmonic contents of the injection signal

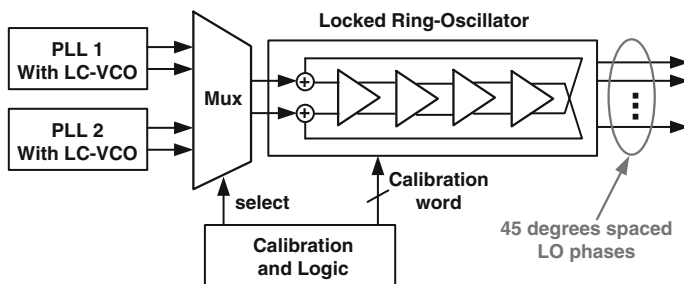


**Fig. 2.14** Block diagram of the UWB frequency synthesizer of [18]

3.300–4.620 GHz with a frequency step of 132 MHz, and they use a reference frequency of 66 MHz. Each PLL generates one of the band center frequencies of Band Group One for operation in Band Group One for operation in Band Group Three or Band Group Four, the PLLs generate the synchronization signal at half the center frequency for each band. The outputs of these three PLLs are multiplexed and are injected to the output of the first delay cell of the ring-oscillator.

The frequency synthesizer of [18] requires generation of  $45^\circ$  spaced LO phases, to be used in subharmonic mixers for upconversion and downconversion. Therefore, it exploits a four-stage ring-oscillator. However, as shown in [21], an  $N$ -stage ring-oscillator can only provide accurate phases at its free running frequency, and the spacing of its output phases will deviate from  $180/N$  when it is injection-locked to another frequency. This is a well-known issue and there are some solutions to maintain accurate output phase spacing across the locking range [21–23]. These solutions are mainly based on injecting the synchronization signal at multiple nodes using appropriate phases. These techniques will be studied in more detail in Chap. 4. As shown in [22], this phase error is directly proportional to the deviation from the free-running frequency. The frequency synthesizer of [18] uses digital calibration of the free-running frequency of the ring-oscillator to correct for this phase error. The details of this phase error correction are discussed in [22].

The injection strength needs to be large enough to achieve adequate phase noise suppression in the ring-oscillator and also to obtain a fast enough transient in the injection-locked oscillator. However, too large an injection signal will cause more systematic amplitude and phase error that can be beyond the level that can be corrected, while it will not provide any further phase noise suppression.

The synthesizer of [18] is implemented in a 90 nm CMOS technology and performs frequency hopping in approximately 4 nS. Using three separate PLLs to generate the synchronization signal at the LO frequency or half the LO frequency helps this frequency synthesizer to achieve a spurious tone level of better than  $-43$  dBc, compared to  $-19$  dBc of the frequency synthesizer of [17].

### 2.2.5 Delay-Locked Loop-Based Techniques

Another technique for frequency synthesis is based on combining the uniformly spaced phases of a reference signal obtained using a delay-locked loop (DLL) [24]. This technique achieves a frequency multiplication factor of  $N$  where  $N$  is the number of delay cells in the DLL. In this case, it is assumed that each delay cell introduces a phase shift of  $2\pi/N$ . This technique benefits from several advantages: first, in contrast to a PLL, a DLL is a first-order system; hence, it has no stability issue. Consequently, the bandwidth of a DLL can be chosen to be wide enough to meet the settling requirement. Second, the number of delay cells in the feedback loop can be programmable, hence a programmable frequency multiplication factor can be implemented. These two properties of a DLL-based frequency synthesizer make it a good candidate for fast agile applications, such as UWB. In addition to these properties, as discussed in [24], a DLL-based frequency synthesizer can achieve a better close-in phase noise when compared to a PLL-based frequency synthesizer. This comes from the fact that the power of the phase noise at the output of the delay cell is constant over the offset frequencies [25] while the phase noise power of a VCO is inversely proportional to the square of the offset frequency [26]. The superior close-in phase noise of a DLL-based frequency synthesizer may not be very important for UWB systems, but can be very important in many wireless communication standards.

A simplified block diagram of a DLL-based frequency synthesizer is shown in Fig. 2.15. As can be seen from this figure, the outputs of all the delay cells go to the edge combiner. The edge combiner combines all the  $N$  phases of the reference signal to generate a signal at  $N$  times the frequency of the reference signal. At low frequencies an edge combiner can be implemented using combinational logic gates including exclusive-OR (XOR) gates, similar to the clock multiplier presented in [27]. However, some analog techniques are required to implement an edge combiner at RF frequencies. A typical circuit implementation of such an edge combiner is shown in Fig. 2.16 [24]. As can be seen from Fig. 2.16, it consists of  $N$  open drain differential pairs that are all terminated to a tuned load.

The operation of a fifth-order DLL-based frequency synthesizer is studied in [28]. We present this analysis for the general case where the DLL consists of  $N$  stages.

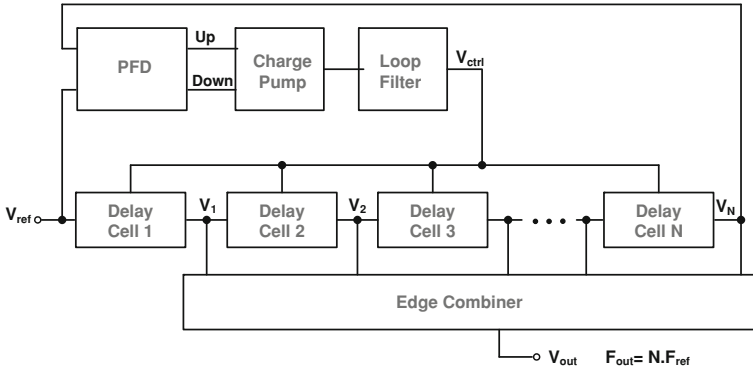
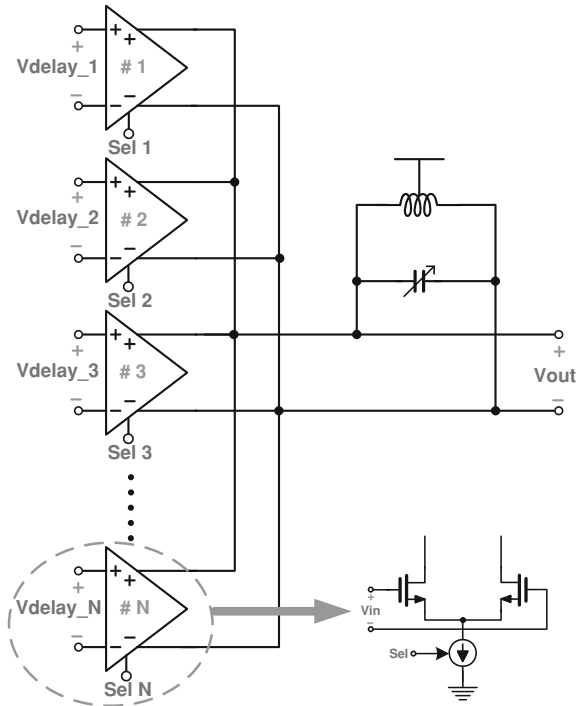


**Fig. 2.15** A general block diagram of a DLL-based frequency synthesizer

**Fig. 2.16** An example of circuit implementation for edge combiner for a DLL-based frequency multiplier



To do so, consider the  $N$ -stage DLL of Fig. 2.15. We assume that the gain of each delay cell is unity, i.e., the amplitude at the output of all delay cells are equal which is approximately true in large signal operation. Furthermore, we assume that the delay of all the delay cells are well matched. If this DLL is locked to a reference signal with a fundamental period of  $T_i$ , each delay cell introduces a delay of  $T_D$  where

$$T_D = T_i / N. \quad (2.8)$$

If the edge combiner sums the outputs of the delay cells, the transfer function of the edge combiner can be expressed as shown below.

$$H(s) = e^{-sT_D} + e^{-s2T_D} + e^{-s3T_D} + \dots + e^{-sNT_D} \quad (2.9)$$

If  $s \neq j2\pi k/T_D$ , where  $k$  is an integer number, (2.9) can be simplified as

$$H(s) = e^{-sT_i/N} \frac{1 - e^{-sT_i}}{1 - e^{-sT_i/N}}. \quad (2.10)$$

Summing the inputs to the delay cells instead of the outputs of the delay cells in the edge combiner only changes the phase of the transfer function of (2.10) while it keeps the magnitude of (2.10) intact. The magnitude of the transfer function of (2.10) is evaluated by substituting  $s$  with  $j\omega$  where  $\omega = 2k\pi/T_i$  ( $k = 1, 2, \dots, N$ ), as shown below.

$$|H(j\omega)| = 0 \quad \text{if } \omega = \frac{2\pi k}{T_i} \quad (k = 1, 2, \dots, N-1) \quad (2.11a)$$

$$|H(j\omega)| = N \quad \text{if } \omega = 0, \quad \frac{2\pi N}{T_i} \quad (2.11b)$$

As can be seen from (2.11a) and (2.11b), the transfer function of (2.10) has nulls at all the harmonics of  $2\pi/T_i$  except at the  $N$ th harmonic. Hence, it keeps the  $N$ th harmonic of the reference signal at the output of the edge combiner and serves as a frequency multiplier by a factor of  $N$ . The zeros in the magnitude of the transfer function of (2.10) are obtained when the delay cells are well matched, and presence of any mismatch between the delay cells leads to spurious tones at the output spectrum, as discussed in [24].

The frequency multiplication factor of a DLL-based frequency synthesizer is the number of its delay cells, i.e., the frequency multiplication factor of DLL-based synthesizer can be changed by changing the number of delay cells that are used in the feedback loop, or changing the delay per cell provided that each delay cell has enough programmability range to cover the required range of delay per cell at the reference frequency.

The UWB frequency synthesizer of [25] is a DLL-based frequency synthesizer (multiplier) that generates the center frequencies for Band Group One and achieves a band switching time of less than 9.5 nS with a phase noise of  $-120$  dBc/Hz at 1 MHz offset. The block diagram of this frequency synthesizer is shown in Fig. 2.17.

As discussed earlier, the bandwidth of a DLL can be chosen wide enough with no concern regarding the stability. However, the mathematical model of a DLL that is derived in [28] is only valid when the bandwidth of the DLL is only a fraction of the reference frequency of the DLL (the bandwidth is usually one tenth of the reference frequency) [28]. Consequently, a high reference frequency is needed to meet the

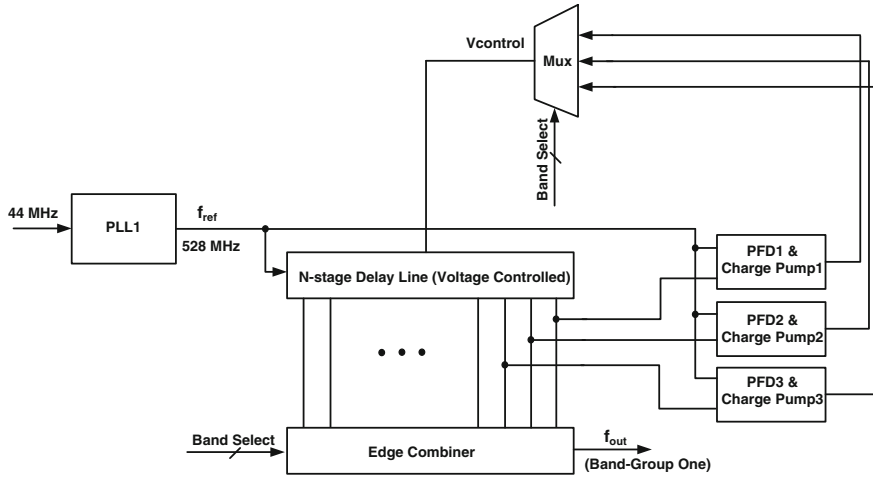


**Fig. 2.17** The block diagram of the DLL-based frequency synthesizer of [25] for Band Group One of UWB spectrum

settling requirement for a UWB frequency synthesizer. The frequency synthesizer of [25] uses a reference frequency of 528 MHz along with multiplication factors of 13, 15, and 17 and is followed by a divide-by-two to generate the quadrature outputs for the center frequencies of Band Group One.

As discussed in [25], switching between the delay cells, to achieve band switching, can introduce a “glitch” on the feedback signal, which will cause a longer settling time than required for UWB system. This issue is solved in the frequency synthesizer of [25] by using multiple PFDs and charge-pumps, as shown in Fig. 2.17.

### 2.2.6 Comb Generator Technique

Another technique that is used in fast hopping frequency synthesizers is comb generator [29]. In this technique, a “comb” of frequencies is generated and is present at all times. Therefore, the speed of frequency hopping is limited to the speed of switching. Combination of filtering and frequency nulling maybe required to suppress the unwanted frequencies.

To explain the operation of a comb generator, we consider an ideal comb of frequencies that contains all the harmonics of a fundamental frequency  $f_0$  with equal amplitudes of  $A_0/T_0$ . The output of this comb generator in frequency domain is represented as follows.

$$S(f) = \frac{A_0}{T_0} \sum_{k=-\infty}^{+\infty} \delta(f - kf_0) \quad (2.12)$$

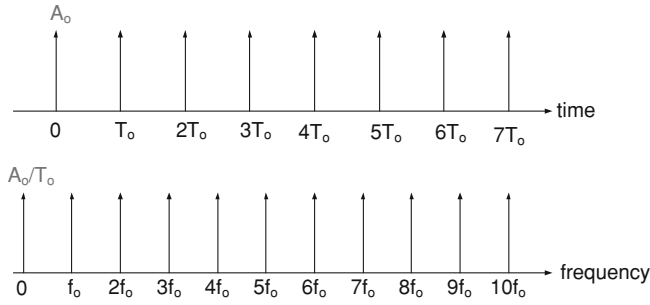


**Fig. 2.18** Representation of the output of an ideal comb generator with a fundamental frequency of  $f_0 = 1/T_0$  in time and frequency domains

It can be concluded from (2.12) that the output of the comb generator is an impulse train in time domain too, i.e.,

$$s(t) = A \sum_{n=-\infty}^{+\infty} \delta(t - nT_0) \quad (2.13)$$

where  $T_0 = 1/f_0$ . The output of an ideal comb generator with a fundamental frequency of  $f_0$ , or a fundamental period of  $T_0$ , in time and frequency domains is shown in Fig. 2.18.

It can be concluded from (2.13) that any circuit that can generate an impulse train at a given fundamental frequency of  $f_0$  can be used as a comb generator. To get one step closer to actual implementation of a comb generator, we assume that the impulse train in the time domain representation of the comb generator output shown in (2.13) is replaced by a pulse train with an amplitude of  $A_0$ , a pulse width of  $\tau$  and a period of  $T_0$ . In this case, a Fourier series representation can be used to demonstrate the output of this new comb generator in time domain as follows.

$$s(t) = \frac{A_0\tau}{T_0} \sum_{n=-\infty}^{+\infty} \text{sinc}(n\tau/T_0) e^{-j2\pi nt/T_0} \quad (2.14)$$

As can be seen from (2.14), in a realistic comb generator the  $\text{sinc}(\cdot)$  function leads to non-equal amplitudes for the harmonics of the fundamental frequency. In addition, the limited bandwidth of the circuitry that implements the comb generator also leads to non-flat spectrum over frequency. On the other hand, the  $\text{sinc}(\cdot)$  function in (2.14) can be used to null some of the undesired frequency components. The magnitude of the harmonics of the output of a comb generator that is described by (2.14) is shown in Fig. 2.19 for different values of  $\tau/T_0$ . As can be seen from (2.14) and Fig. 2.19 a narrow pulse width ( $\tau/T_0 \ll 1$ ) is required to achieve a wideband comb generator.

As mentioned earlier, a wideband comb generator requires generating very narrow pulses. Step recovery diodes [30] are often used to generate very narrow pulses or

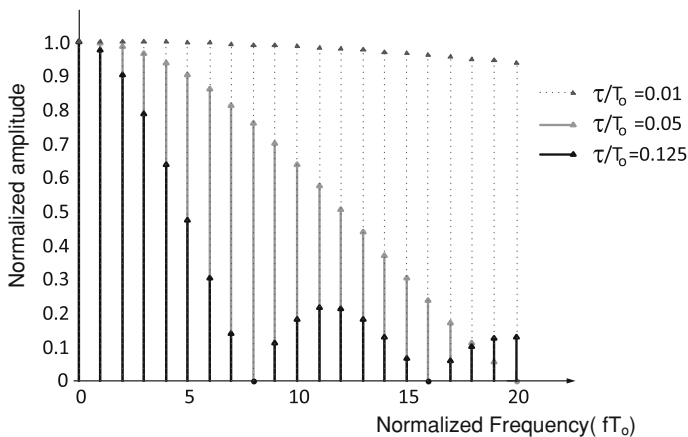


**Fig. 2.19** Spectrum of the output of a comb generator with a fundamental frequency of  $f_0 = 1/T_0$  and a pulse width of  $\tau$  for different values of  $\tau/T_0$

to approximate an impulse. A hybrid approach is also shown in [31] to generate a comb of frequencies for optical communications. However, other techniques need to be deployed for CMOS implementation of a comb generator.

A comb generator for UWB needs to generate integer multiples of 264 MHz. This fundamental frequency is used since it is the common divisor of all the UWB band center frequencies. In addition, it needs to provide a flat frequency response over the desired UWB frequency range. However, one of the main obstacles in implementing an integrated comb generator-based frequency synthesizer for UWB is the very high spurious level of the comb generator and the difficulty of on-chip implementation of a filter that provides adequate suppression of the spurious tones.

### 2.2.7 Techniques Based on Polyphase Filtering

The work presented in [32] uses a different technique to mitigate the agility requirement of a UWB frequency synthesizer. In the receiver of [32] a fixed frequency PLL (7.128 GHz) is used along with a fast switching polyphase filter and a high-speed Analog-to-Digital Converter (ADC).

The synthesizer used in the transceiver of [32] covers Band Group Three, and uses a fixed frequency PLL to generate the center frequency of band eight (7.128 GHz). The receiver of [32] uses quadrature phases of 7.128 GHz to downconvert the entire Band Group Three to DC. As a result, band eight is centered at DC, and bands seven and nine are centered at  $-756$  and  $+756$  MHz, respectively. The fast-switching polyphase filter of Fig. 2.20a is used to select a band of interest. The principles of operation of this polyphase filter is shown in Fig. 2.20b. As can be seen from Fig. 2.20b, the polyphase filter can pass the negative frequencies when

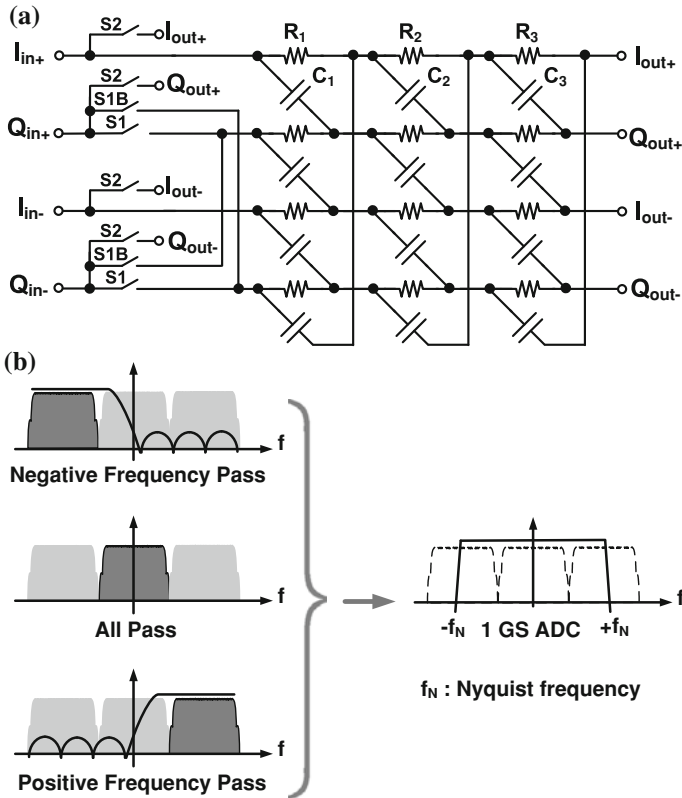


**Fig. 2.20** **a** Schematic of a fast-hopping polyphase filter, and **b** principles of operation. By feeding the appropriate quadrature sequence, the polyphase filter can pass the positive or negative frequency band

operating in band seven, or positive frequencies when operating in band nine by using the appropriate quadrature sequence (clockwise or counterclockwise) at its input. The polyphase filter can also be bypassed for operation in band eight. The polyphase filter of Fig. 2.20a achieves a switching time of 5 ns. The polyphase filter is followed by a 1 GS/sec ADC. Final downconversion to DC for bands seven and nine will be done after analog-to-digital conversion in digital domain.

The synthesizer presented in [32] eliminates the use of SSB mixers in the synthesizer path and does not suffer from mixing spurious tones. However, due to limitations in the speed and dynamic range of the ADCs, it is very difficult for this architecture to cover more than one Band Group using a single fixed frequency PLL.

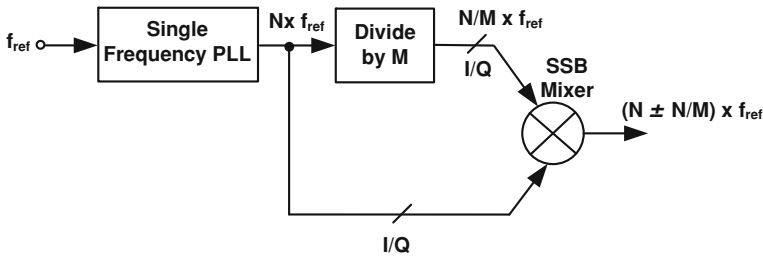


**Fig. 2.21** Block diagram of a frequency synthesizers for UWB based on frequency division and mixing

### 2.2.8 Frequency Synthesis by Frequency Division and Mixing

Another architecture that can be used to implement a frequency synthesizer is a hybrid architecture shown in Fig. 2.21 where a combination of fixed frequency PLLs, frequency dividers, and SSB mixers are required. In this architecture, all the MB-OFDM UWB carriers are generated by combination of frequency division and SSB mixing. Accordingly, this technique alleviates the challenging settling requirement of the PLL.

The architecture of Fig. 2.21 can achieve a very fast switching time, as low as 1 nS. In addition, it can cover the entire UWB frequency range. However, the main drawback of this architecture is the use of SSB mixers. As was discussed in Sect. 2.2.1, using SSB mixers requires the availability of quadrature phases of both mixing signals, with sufficient amplitude and phase accuracy to achieve adequate side-band cancellation. On the other hand, every mixing stage introduces multiple spurious tones. Mixer linearization, and appropriate frequency planning can minimize spurious responses. However, mixer linearization reduces the conversion gain, and it will require additional power consumption to achieve same voltage swing at the mixer output.

A generalization to the architecture of Fig. 2.21 using two SSB mixers is shown in Fig. 2.22. The main challenge in implementing the architecture of Fig. 2.22 is to achieve a hardware efficient design compatible with digital CMOS technology.

For instance, the synthesizer of [33] requires two PLLs and one level of SSB mixing to generate the center frequencies of Band Group One. Another example is the frequency synthesizer of [34] which uses two separate PLLs and one SSB mixer to generate seven band center frequencies of MB-OFDM UWB (based on the earlier band allocation [14]).

The architecture of Fig. 2.22 is used in the majority of UWB frequency synthesizers that cover more than one Band Group, and in most of the 14-band UWB frequency synthesizers due to flexibilities in covering a wide frequency range and overcoming the settling issues [35–37].

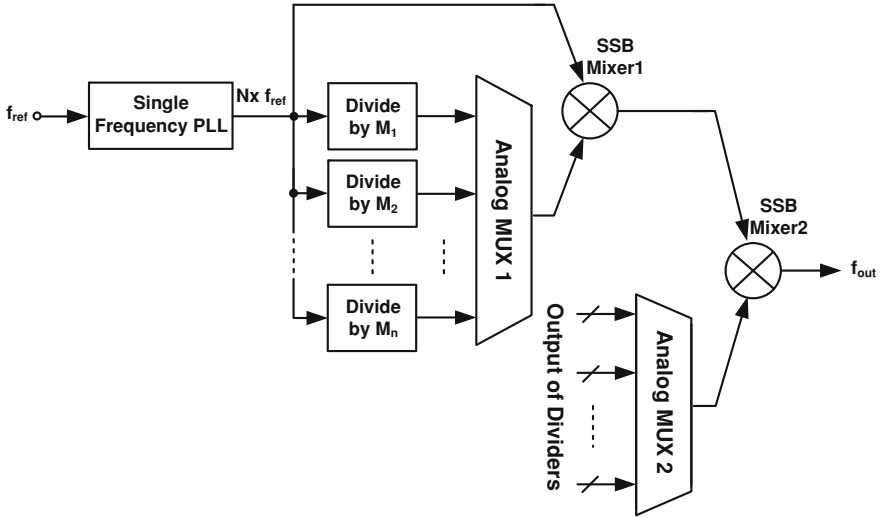


**Fig. 2.22** Generalized architecture for a UWB frequency synthesizers that is based on frequency division and mixing with two levels of SSB mixing

### 2.2.9 Comparison of Different Frequency Synthesizer Solutions

Among all the different architectures for a UWB synthesizer, the all digital PLL presented in Sect. 2.2.3 and the hybrid architecture based on frequency division and mixing presented in Sect. 2.2.8 seem to meet the agility requirement of UWB, and at the same time cover the entire UWB span with a manageable power consumption and die area. The use of an AD-PLL-based architecture for a UWB frequency synthesizer eliminates the use of SSB mixers and frequency dividers required for frequency mixing. Hence, there will be no concern with mixer nonlinearity. However, the objective of this work was to explore the inductor-less implementation of frequency division and mixing techniques in a CMOS-only technology. In addition, although it is not required by WiMedia v1.5 [16, 38], but our focus in choosing an architecture for a UWB frequency synthesizer was to cover the entire fourteen-band span of UWB. In addition, the ability to perform frequency hopping among different Band Groups is also considered. The architecture that is proposed in Sect. 2.3 meets all these specifications.

## 2.3 Proposed Fourteen-Band Frequency Synthesizer for UWB

The latest Band Groups and band allocation for MB-OFDM UWB was formerly shown in Fig. 1.9. Based on that, a frequency plan that can generate the band center frequencies for all fourteen bands and perform frequency hopping within each of the six Band Groups is shown in Fig. 2.23.



The proposed frequency synthesizer uses a single external 14.256 GHz input. Quadrature signals for the operation of the first mixing stage are generated from a ring-oscillator-based Injection Locked Frequency Divider (ILFD) whose output is the center frequency of Band 8. The channel spacing of 528 MHz is obtained from the output of two additional regenerative dividers, with total division ratio of 13.5. A cascade of quadrature output divide-by-2.25, divide-by-2, and divide-by-3 circuits provides the required frequencies. As shown in Fig. 2.24, two similar frequency dividers are used to generate quadrature phases at 7.128 GHz. This scheme avoids any I/Q amplitude or phase mismatch induced due to routing, compared to the case when only one frequency divider is used.

### ***2.3.1 Frequency Dividers with Quadrature Outputs***

As can be seen from Fig. 2.24 the inductor-less implementation of this technology relies on using SSB mixers and the availability of quadrature phases of all the signals used in the mixers. Therefore, frequency dividers that generate quadrature outputs and can be implemented in digital CMOS technology while running at several GHz frequencies are of interest. Ring-oscillator-based frequency dividers that can meet these goals are studied in Chap. 3.

### ***2.3.2 Fractional Frequency Dividers***

The architecture of Fig. 2.24 requires some frequency dividers that can obtain fractional division ratios while providing quadrature outputs. Several techniques to implement a fractional division ratio are studied in Chap. 4 and the appropriate architecture that can be implemented in digital CMOS is chosen. In addition, the operation and the stability of this divider is studied.

### ***2.3.3 Broadband High-Gain Interface Amplifiers***

The UWB frequency synthesizer of Fig. 2.24 uses up to two levels of SSB mixing to generate the desired carrier frequency. The output of the first SSB mixer needs to be amplified to be used in the second SSB mixer. Different amplifiers that can be used for this purpose in the frequency synthesizer of Fig. 2.24 are studied in Chap. 5 and an architecture that can be implemented in a low-voltage digital CMOS technology is chosen.

## References

1. Uusikartano R, Niittylahti J (2001) A periodical frequency synthesizer for a 2.4-GHz fast frequency hopping transceiver. *IEEE Trans Circuits Syst II: Analog Digit Signal Process* 48(10):912–918
2. Kroupa V (1999) Direct digital frequency synthesizers. IEEE Press, New York
3. Sandner C, Derksen S, Draxelmayer D, Ek S, Filimon V, Leach G, Marsili S, Matveev D, Mertens K, Michl F et al (2006) A WiMedia/MBOA-compliant CMOS RF transceiver for UWB. *IEEE J Solid-State Circuits* 41(12):2787–2794
4. Perrott MH (1997) Techniques for high data rate modulation and low power operation of fractional-N frequency synthesizers. Ph.D. dissertation, Massachusetts Institute of Technology
5. Gardner F (2005) Phaselock techniques. Wiley-Blackwell, New York
6. Gardner F (1980) Charge-pump phase-lock loops. *IEEE Trans Commun* 28(11):1849–1858
7. Bishop R, Dorf R (2004) Modern control systems. Addison wesley publishing company, Reading
8. Rategh H, Lee T (2001) Multi-GHz frequency synthesis and division: frequency synthesizer design for 5 GHz wireless LAN systems. Springer
9. Tak G, Hyun S, Kang T, Choi B, Park S (2005) A 6.3–9-GHz CMOS fast settling PLL for MB-OFDM UWB applications. *IEEE J Solid-State Circuits* 40(8):1671–1679
10. Vaucher C (2000) An adaptive PLL tuning system architecture combining high spectral purity and fast settling time. *IEEE J Solid-State Circuits* 35(4):490–502
11. Razavi B, Aytur T, Lam C, Yang F, Li K, Yan R, Kang H, Hsu C, Lee C (2005) A UWB CMOS transceiver. *IEEE J Solid-State Circuits* 40(12):2555–2562
12. Bergervoet J, Harish K, Lee S, Leenaerts D, van de Beek R, van der Weide G, Roovers R (2007) A WiMedia-compliant UWB transceiver in 65 nm CMOS. *IEEE ISSCC digest of technical papers*, pp 112–590, Feb 2007
13. Stadius K, Rapinaja T, Kaukuvuori J, Ryyanen J, Halonen K (2007) Multitone fast frequency-hopping synthesizer for UWB radio. *IEEE Trans Microw Theory Tech* 55(8):1633–1641
14. Multi-Band OFDM Physical Layer Proposal, July 2003, doc IEEE 802.15- 03/267r5
15. Leenaerts D, van de Beek R, Bergervoet J, Kundur H, van der Weide G, Kapoor A, Pu T, Fang Y, Wang Y, Mikkala B et al (2009) A 65 nm CMOS inductorless triple Band Group WiMedia UWB PHY. *IEEE J Solid-State Circuits* 44(12):3499–3510
16. Standard ECMA-368: High Rate Ultra Wideband PHY and MAC Standard, 3rd edition, Dec. 2008. <http://www.ecma-international.org/publications/standards/Ecma-368.htm>
17. Dal Toso S, Bevilacqua A, Tiebout M, Marsili S, Sandner C, Gerosa A, Neviani A (2008) UWB fast-hopping frequency generation based on sub-harmonic injection locking. *IEEE J Solid-State Circuits* 43(12):2844–2852
18. Mazzanti A, Vahidfar M, Sosio M, Svelto F (2009) A reconfigurable demodulator with 3-to-5GHz agile synthesizer for 9-band WiMedia UWB in 65nm CMOS. *IEEE ISSCC digest of technical papers*, pp 412–413, 413a, Feb 2009
19. Verma S, Rategh H, Lee T (2003) A unified model for injection-locked frequency dividers. *IEEE J Solid-State Circuits* 38(6):1015–1027
20. Dal Toso S, Bevilacqua A, Tiebout M, Marsili S, Sandner C, Gerosa A, Neviani A (Dec. 2008) UWB fast-hopping frequency generation based on sub-harmonic injection locking. *IEEE J Solid-State Circuits* 43(12):2844–2852
21. Chien J, Lu L (2007) Analysis and design of wideband injection-locked ring-oscillators with multiple-input injection. *IEEE J Solid-State Circuits* 42(9):1906–1915
22. Mazzanti A, Vahidfar M, Sosio M, Svelto F (2010) A low phase-noise multiphase LO generator for wideband demodulators based on reconfigurable subharmonic mixers. *IEEE J Solid-State Circuits* 45(10):2104–2115
23. Farazian M, Gudem P, Larson L (2010) Stability and operation of injectionlocked regenerative frequency dividers. *IEEE Trans Circuits Syst I: Regul Pap* 57(8):2006–2019

24. Chien G, Gray P (2000) A 900-MHz local oscillator using a DLL-based frequency multiplier technique for PCS applications. *IEEE J Solid-State Circuits* 35(12):1996–1999
25. Lee, T-C, Hsiao K-J (2006) The design and analysis of a DLL-based frequency synthesizer for UWB application. *IEEE J Solid-State Circuits* 41(6):1245–1252
26. Hajimiri A, Lee T (1998) A general theory of phase noise in electrical oscillators. *IEEE J Solid-State Circuits* 33(2):179–194
27. Chin-Chieh C, Chao-Ping S, Yen-Kuang C (2005) Clock multiplier. US Patent US 6977536 B2, 20 Dec 2005
28. Chien G (2000) Low-noise local oscillator design techniques using a DLL-based frequency multiplier for wireless applications. Ph.D. dissertation, University of California, Berkeley
29. Adkins L (1976) Fast frequency hopping with Surface Acoustic Wave (SAW) frequency synthesizers. *Proceedings of the 30th annual symposium on frequency control*, pp 367–370
30. Moll J, Hamilton S (1969) Physical modeling of the step recovery diode for pulse and harmonic generation circuits. *Proc IEEE* 57(7):1250–1259
31. Tsao S, Wu J, Chen T, Wu J (2002) Phaselocked tunable subcarrier comb generator. *Electron Lett* 30(24):2059–2060
32. Tanaka A, Numata K, Kodama H, Ishikawa H, Oshima N, Yano H (2009) A 2.88Gb/s digital hopping UWB transceiver. *IEEE ISSCC digest of technical papers*, pp 318–319, Feb 2009
33. Leenaerts D, van de Beek R, Van der Weide G, Bergervoet J, Harish K, Waite H, Zhang Y, Razzell C, Roovers R, Res P et al (2005) A SiGe BiCMOS 1ns fast hopping frequency synthesizer for UWB radio. *IEEE ISSCC digest of technical papers*, pp 202–593, Feb 2005
34. Lee J, Chiu D (2005) A 7-band 3–8 GHz frequency synthesizer with 1 ns bandswitching time in 0.18  $\mu\text{m}$  CMOS technology. *IEEE ISSCC digest of technical papers*, pp 204–593, Feb 2005
35. Ismail A, Abidi A (2005) A 3.1- to 8.2-GHz zero-IF receiver and direct frequency synthesizer in 0.18 $\mu\text{m}$  SiGe BiCMOS for mode-2 MB-OFDM UWB communication. *IEEE J of Solid-State Circuits* 40(12):2573–2582
36. Liang C-F, Liu S-I, Chen Y-H, Yang T-Y, Ma G-K (2006) A 14-band frequency synthesizer for MB-OFDM UWB application. *IEEE ISSCC digest of technical papers*, pp 428–437, Feb 2006
37. Lu T-Y, Chen W-Z (2008) A 3-to-10GHz 14-band CMOS frequency synthesizer with spurs reduction for MB-OFDM UWB system. *IEEE ISSCC digest of technical papers*, pp 126–601, Feb 2008
38. Multiband OFDM physical layer specification PHY Specification: Final Deliverable 1.5 (2009). [http://www.wimedia.org/en/docs/10003r02WM\\_CRB-WiMedia\\_PHY\\_Spec\\_1.5.pdf](http://www.wimedia.org/en/docs/10003r02WM_CRB-WiMedia_PHY_Spec_1.5.pdf)
39. Lin C, Wang C-K (2005) A regenerative semi-dynamic frequency divider for mode-1 MB-OFDM UWB hopping carrier generation. *IEEE ISSCC digest of technical papers*, vol 1. pp 206–207, Feb 2005

<http://www.springer.com/978-1-4614-0489-7>

Fast Hopping Frequency Generation in Digital CMOS

Farazian, M.; Larson, L.E.; Gudem, P.S.

2013, XIV, 154 p., Hardcover

ISBN: 978-1-4614-0489-7