An FMDLL Based Dual-Loop Frequency Synthesizer for 5GHz WLAN Applications

Ting Wu, Pavan Kumar Hanumolu, Un-Ku Moon, and Kartikeya Mayaram
Department of Electrical and Computer Engineering
Oregon State University
Owen Hall 220, Corvallis, OR, 97331
E-mail: wuti@ece.orst.edu

Abstract -- **We present a new dual-loop frequency synthesizer for 5GHz wireless local-area network (WLAN) applications. In line with the IEEE 802.11a standard, the output frequency is targeted at 5.15G to 5.35GHz, with a frequency step of 5MHz. To make the use of ring-type VCO feasible for this application, we adopted a dual-loop frequency synthesizer architecture similar to [1]. However, it is critical that the primary loop of such a dual-loop frequency synthesizer is supported by a low-noise reference (by the peripheral loop). To address this problem, we propose a new fractional frequency multiplying delay-locked loop (FMDLL) working as the peripheral loop. MATLAB simulation results demonstrate the phase noise improvement by the use of the proposed FMDLL compared to the conventional design.**

I. INTRODUCTION

Today's rapid advancements in wireless communication technology and the continued scaling of low-cost CMOS processes have made integrated single-chip 5GHz 802.11a WLAN receivers an efficiently feasible reality. This is spurring a great demand for high-performance and low-cost frequency synthesizers that can be run at 5GHz range with a small frequency step size (e.g. 5MHz). In this paper, we propose the design of a low-cost frequency synthesizer, having an output frequency of 5.15GHz to 5.35GHz with a frequency step of 5MHz. Given the IEEE 802.11a standard (specification), our main focus is to develop new techniques to reduce the design cost.

II. CONVENTIONAL DESIGN

Most of the conventional frequency synthesizers for 5GHz WLAN applications are of integer-N type [2, 3]. As such, the reference frequency has to be equal to the frequency resolution of 5MHz, and the division ratio N becomes an unacceptably large number. Thus, most such implementations incorporate a 8/9 or a 16/17 prescaler. To satisfy the phase noise specification, the designed frequency synthesizers commonly have a low bandwidth. As a consequence, a high-performance VCO, e.g. LC-tank VCO, is needed since the VCO noise is high-pass transferred to the output phase noise. Another solution is the use of a $\Delta\Sigma$ fractional-N frequency synthesizer [4]. The reference frequency could be larger than 5MHz. However, the bandwidth has to be kept low enough to filter out the fractional spurs and thus a LC-tank VCO is still required to maintain low phase noise. A lower bandwidth also results in a longer settling time. In addition to this, a LC-tank VCO is not able to generate orthogonal I-Q phases. Therefore, some have designed the synthesizer (the VCO) to oscillate at 10GHz, followed by a divide-by-2 divider to produce orthogonal I-Q phases [4]. It not only further complicates the design of a higher frequency 10GHz VCO, but also requires high-performance and high-speed dividers as well.

An alternative method is the dual-loop frequency synthesizer, for example [5], as shown in Fig. 1. It incorporates two integer-N PLLs. The primary loop, with a fixed division number N_1 and a tunable reference f_{ref1}, generates the output frequency. The peripheral loop, with a tunable division number N_2 and a fixed reference f_{ref2}, provides the reference frequency f_{ref1} for the primary loop. The output frequency f_0 and resolution Δf_0 are given as

$$f_0 = f_{ref\,1} * N_1 = f_{ref\,2} * N_2 * N_1$$
$$\Delta f_0 = N_1 * \Delta f_{ref\,1} = N_1 * f_{ref\,2} \quad \textbf{(1)}$$

Compared with the single-loop design, the dual-loop architecture provides more flexibility in dealing with the trade-off between the loop bandwidth, high frequency output, and small frequency resolution. In the dual-loop design, the output frequency can be tuned by changing f_{ref1}, which can be fulfilled by altering N_2 in the peripheral loop. By doing so, a fixed and smaller division number N_1 is realizable and thereby a high loop bandwidth can be achieved. Therefore, a 4-stage ring-type VCO was implemented in the primary loop [5], enabling I-Q phases to be directly generated from the 5GHz VCO.

Increasing the primary loop bandwidth reduces the noise contribution from the VCO, however, it increases the noise contribution from its reference clock f_{ref1}. Therefore, the difficulty remains in the peripheral loop which needs to produce a clean output f_{ref1} with an even finer frequency resolution. As N_1 is 4, to maintain a 5MHz frequency step at the output, the reference frequency f_{ref2} should be as small as 1.25MHz. Therefore, a large number N_2 is required. An off-chip LC-tank VCO was used [5] for the peripheral loop in order for the primary loop to have a clean reference clock. In addition, the bandwidth of the peripheral loop is lower, becoming more critical in the design of the entire frequency synthesizer.

0-7803-8834-8/05/$20.00 ©2005 IEEE.

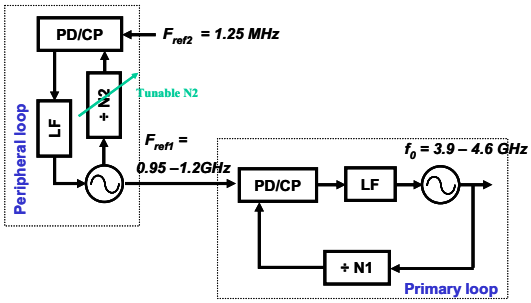


Figure 1. Dual-loop frequency synthesizer [5].

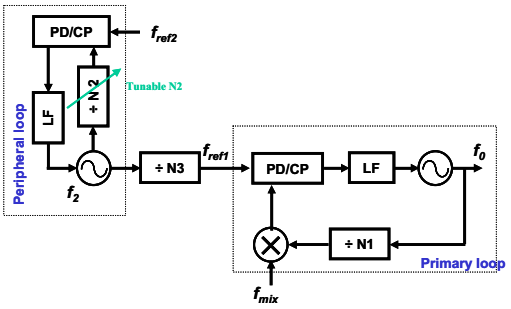


Figure 2. Another dual-loop frequency synthesizer [1].

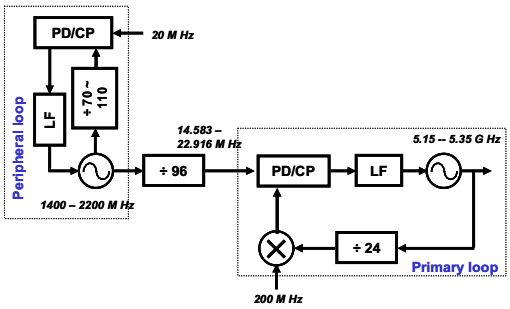


Figure 3. Implementation example of Fig. 2.

The non-ideal design parameters, such as small f_{ref2}, low bandwidth of peripheral loop, and a large division number N_2, are all due to the smaller frequency resolution requirement at f_{ref1}. If an additional division is inserted between the two loops, then the same frequency resolution at f_{ref1} can be achieved with a larger f_{ref2}, a higher bandwidth of the peripheral loop, and a smaller division number N_2. This reveals the idea of another dual-loop frequency synthesizer [1]. This is shown in Fig. 2. Similarly, the peripheral loop is a traditional integer-N PLL with a programmable N_2. The inserted divider not only reduces the frequency resolution at f_{ref1} but the magnitude of f_{ref1} as well. To compensate for this effect in the primary loop, an image rejection mixer is added before the phase detector (PD). The other input of the mixer is f_{mix}. After image-rejection mixing, only the low frequency image is left and compared with f_{ref1} in the PD. The output frequency f_0 and the resolution Δf_0 are given as,

$$f_0 = (f_{ref1} + f_{mix}) * N_1 = (N_2 * f_{ref2} / N_3 + f_{mix}) * N_1$$
$$\Delta f_0 = N_1 * \Delta f_{ref1} = N_1 * f_{ref2} / N_3 \quad \textbf{(2)}$$

Comparing Equation (2) with (1), one can see that adding a third divider between the two loops reduces the frequency resolution by a factor of N_3. In other words, given a specific frequency step, f_{ref2} can be enhanced by a factor of N_3. A higher loop bandwidth of the peripheral loop also becomes possible. Thus, a high-performance off-chip LC-tank VCO is no longer needed for the peripheral loop. Although a mixer adds design cost and may introduce additional noise, it provides a distinct advantage of high performance with a ring-based VCO. For instance, such a dual-loop frequency synthesizer was designed for GSM application in [9].

We adopt this dual-loop architecture for 5GHz WLAN. An implementation example of such a frequency synthesizer is shown in Fig. 3. The reference clock f_{ref2} is 20MHz and the mixer input f_{mix} is 200MHz. N_1 is 24, making a high loop bandwidth and ring-type VCO in the primary loop possible. With a tunable N_2 from 70 to 110 and fixed N_3 equal to 96, the peripheral loop produces an output frequency of 1400 to 2200 MHz with a frequency step of 20MHz. The output of the primary loop is 5.15GHz to 5.35GHz with a frequency step of 5MHz.

III. FRACTIONAL FREQUENCY MULTIPLYING DLL

Since a clean reference clock f_{ref1} for the primary loop is important for the overall performance of the dual-loop frequency synthesizer, we propose to further improve the performance of the peripheral loop. A PLL traditionally achieves the frequency multiplying function. However, it suffers from the inherent jitter accumulation of the VCO. To address this problem, DLL based frequency synthesizers have been developed [6, 7]. In these DLL frequency synthesizers, the reference clock goes through the delay line at every reference clock cycle. Thus, only a limited amount of phase error accumulation occurs. Given a clean crystal reference and an identical noisy environment, a DLL performs better than a PLL.

The edge-combiner based frequency multiplying DLL [6] is limited by mismatches between the delay elements and non-ideal effects in the edge combiner. In order to mitigate this problem, a multiplying DLL (MDLL) [7] was proposed to eliminate the use of an edge combiner. A frequency multiplying function can be realized by enabling PD every N cycle. Fig. 4 shows the architecture as well as the timing diagram for the case N equal to 4. A MUX is inserted in the voltage controlled delay line (VCDL). The same selection bit used in the PD is applied to the MUX. When the selection is disabled, the MUX selects the VCDL output. The net result is that the VCDL oscillates, behaving like a VCO. When the selection is enabled, the clean reference clock enters the delay line, thus the phase error accumulation is reduced. By assigning a different selection logic, programmability of the frequency multiplying ratio N is easy to incorporate.

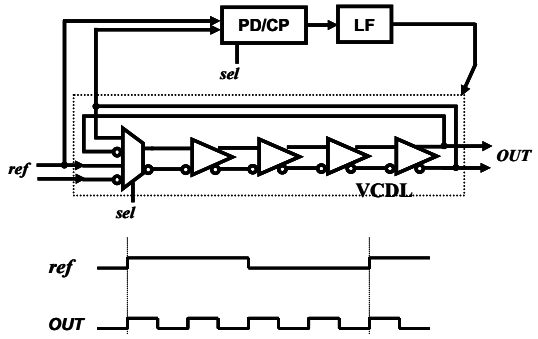


Figure 4. MDLL architecture and timing diagram [7].

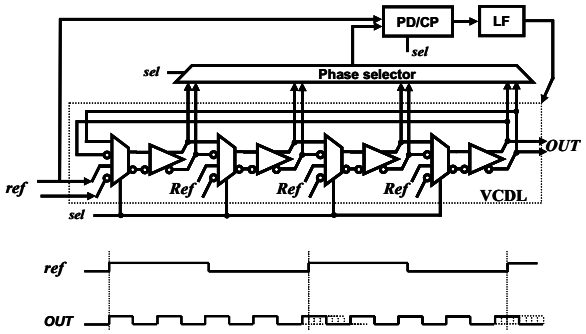


Figure 5. Proposed fractional MDLL architecture and timing diagram.

Although the above MDLL achieves frequency synthesis, the multiplication ratio is only an integer number. It is desirable to realize a fractional frequency multiplication for other applications. In the dual-loop frequency synthesizer, the peripheral loop will ideally have a larger reference frequency f_{ref2} and a higher loop bandwidth. To address this issue, a novel MUX based fractional MDLL (FMDLL) is proposed. Fig. 5 shows the architecture of the new FMDLL, assuming 4 delay elements are used in the VCDL. In this new DLL, a 2-to-1 MUX is inserted between each two successive delay elements. For each MUX, the clean reference clock is one of the inputs, while the other input is the output of the preceding delay element. If the selection bit is '1', the reference clock is chosen by the MUX, and the clean clock goes through the delay line as in the MDLL. If the selection bit is '0', the preceding delay element output is chosen by the MUX, and the delay elements propagate the oscillation.

Fractional frequency multiplication can be realized by altering the MUX-selection bus. For instance, if a frequency multiplication ratio is specified with an integer number N equal to 4 and a fractional number F equal to 1/8, the MUX-selection bus can be assigned as follows, "1000", "0000", "0000", "0000", "0100", "0000", and so on. By doing so, for example, the reference clock goes through the VCDL via the first MUX. After one reference clock cycle (more than 4 VCO cycles), the reference clock goes through the VCDL via the second MUX, and so on.

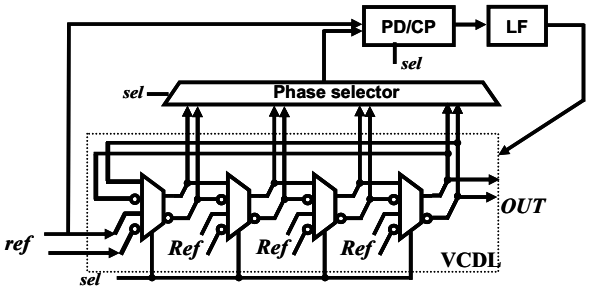


Figure 6. Simplified new fractional MDLL (FMDLL).

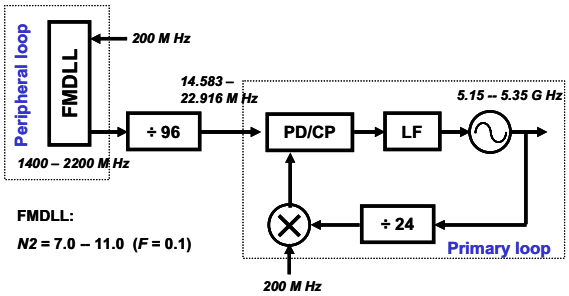


Figure 7. Dual-loop frequency synthesizer using FMDLL.

At the same time, the MUX-selection bus is also applied to the phase selector to choose the proper phase to feed into the PD. The PD is enabled every reference clock cycle. In other words, the PD fulfills the phase comparison at the same time when a reference clock goes through the VCDL. Fig. 5 shows the timing diagram of this example.

Similar to the MDLL, programmability of the frequency multiplying ratio is easy to implement. Furthermore, to simplify the architecture, each MUX and the following delay element can be combined to one MUX based delay element, as shown in Fig. 6. This new FMDLL has more balanced delay elements than those in the MDLL. However, most MDLLs are built similarly with dummy devices for balance. Thus the elements of the FMDLL have very little or no overhead compared to the MDLL.

One of the main drawbacks in the new FMDLL is that it might have fixed pattern jitter due to delay mismatches between the delay elements in the DLL. To minimize the effect of delay mismatch, a self-calibration technique [8] can be employed.

Fig. 7 shows an example of a dual-loop frequency synthesizer architecture using our proposed fractional frequency multiplying DLL. Compared with the conventional design given in Fig. 3, the reference clock frequency of the peripheral loop is increased by a factor of 10 and the division number N_2 is reduced by a factor of 10 as well. The fractional factor of 1/10 can be realized by a 5-stage VCDL. Note that using a higher frequency f_{ref2} does not add further cost, as it is the same as the frequency *fmix*. This 200MHz frequency could be derived from a crystal directly or generated by a simple PLL using a low frequency crystal.

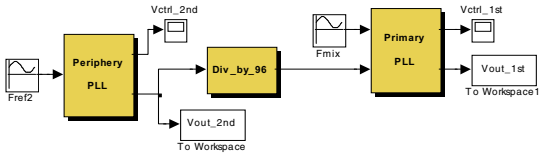


Figure 8. SIMULINK model of dual-loop simulation.

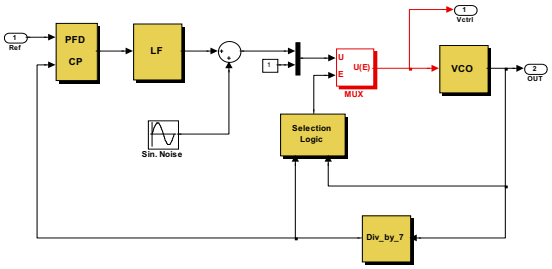
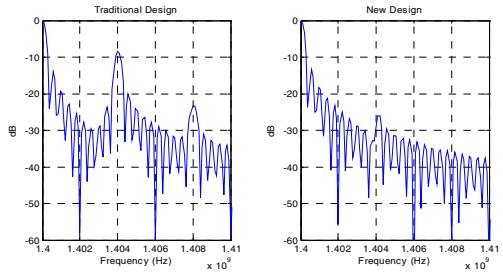


Figure 9. Equivalent model of the new peripheral DLL loop.

IV. SIMULATION

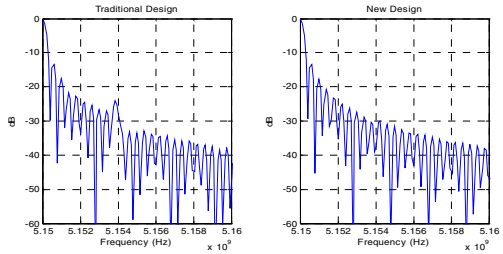
In order to demonstrate our proposed new fractional DLL, a traditional dual-loop using a peripheral PLL and a new dual-loop using our proposed DLL have been simulated. MATLAB SIMULINK was employed to create the model for simulation. Fig. 8 shows the dual-loop architecture, corresponding to Fig. 3 and Fig. 7. Both primary loops are assumed to be noiseless, generating a 5.15GHz frequency output. The peripheral loops produce a 1400MHz output. To evaluate the phase noise performance, a small-amplitude sinusoidal phase modulation is applied to both peripheral loops at an offset frequency of 4MHz. The traditional peripheral PLL has a f_{ref2} of 20MHz and a divider with ratio of 70. The new peripheral DLL is created using an equivalent PLL model as shown in Fig. 9. It has a f_{ref2} of 200MHz and an effective division ratio of 7. Therefore, every 7 VCO cycles correspond to a reference clock cycle. When the PD compares the reference with the feedback clock, the VCO control voltage is also reset to the correct value. By doing so, the VCO phase error accumulation is reduced. This is essentially equivalent to the fact that a precise reference clock phase goes through the VCDL at every reference clock cycle, which is exactly so in the new fractional DLL. This modeling method is not only easy to realize, but also makes the comparison fair by using nearly the same components in a traditional PLL. In our simulations, the bandwidth of the peripheral loop is set to 2MHz and 20MHz for the traditional and new designs, respectively.

Fig. 10 and Fig. 11 show the Fourier transforms of both peripheral loop outputs and both primary loop outputs, respectively. The simulated phase-noise side-bands of traditional designs (a) and new designs (b) are shown. It can be seen that the use of our new peripheral loop reduces the phase noise at an offset of 4MHz, by 15dB at the peripheral loop output and by 5dB at the primary loop output.



(a) (b)

Figure 10. Comparison of the spectrum at peripheral loop output. (a) Traditional peripheral PLL. (b) New peripheral FMDLL



(a) (b)

Figure 11. Comparison of the spectrum at primary loop output. (a) Traditional dual-loop. (b) New dual-loop.

V. CONCLUSIONS

A fractional frequency multiplying delay-locked loop (FMDLL) was described. The FMDLL is used as the peripheral loop in a dual-loop architecture for 5GHz WLAN applications. Compared to the non-dual-loop conventional design, a lower cost can be realized by replacing the LC-tank VCO with a ring-type oscillator in the primary loop. Jitter accumulation in the peripheral loop is minimized by use of the FMDLL.

REFERENCES

[1] T. Aytur and J. Khoury, "Advantages of dual-loop frequency synthesizers for GSM applications," *IEEE International Symposium on Circuits and Systems*, pp. 17-20, May 1997.

[2] M. Terrovitis, M. Mack, K. Singh, M. Zargari, "A 3.2 to 4GHz, 0.25um CMOS frequency synthesizer for IEEE 802.11a/b/g WLAN," in *Dig. Tech. Papers IEEE ISSCC*, Feb. 2004.

[3] D. Su, *et al.*, "A 5GHz CMOS transceiver for IEEE 802.11a wireless LAN," in *Dig. Tech. Papers IEEE ISSCC*, Feb. 2002.

[4] L. Perraud, *et al.*, "A dual-band 802.11a/b/g radio in 0.18um CMOS," in *Dig. Tech. Papers IEEE ISSCC*, Feb. 2004.

[5] T. Schwanenberger, M. Ipek, S. Roth, H. Schemmann, "A multi standard single-chip transceiver covering 5.15 to 5.85GHz," in *Dig. Tech. Papers, IEEE ISSCC*, Feb. 2003.

[6] G. Chien and P. Gray, "A 900MHz local oscillator using a DLL-based frequency multiplier technique for PCS applications," in *Dig. Tech. Papers IEEE ISSCC*, Feb. 2000

[7] R. Farjad-Rad, *et al.*, "A low-power multiplying DLL for low-jitter multigigahertz clock generation in highly integrated digital chips," *IEEE J. Solid-State Circuits*, vol. 37, pp. 1804-1811, Dec. 2002.

[8] C-H Park, O Kim and B. Kim, "A 1.8-GHz self-calibrated phase-locked loop with precise I/Q matching," *IEEE J. Solid-State Circuits*, vol. 36, pp. 777-783, May 2001.

[9] W. S. T. Yan and H. C. Luong, "A 2-V 900-MHz monolithic CMOS dual-loop frequency synthesizer for GSM receivers," *IEEE J. Solid-State Circuits*, vol. 36, pp. 204-216, Feb. 2001.