Fast-lock Hybrid PLL Combining Fractional- & Integer-N Modes of Differing Bandwidths

Kyoungho Woo, Yong Liu, and Donhee Ham
School of Engineering & Applied Sciences, Harvard University, Cambridge, MA 02138

Abstract

We report a single-loop PLL that operates in a wide-bandwidth, fractional-N mode (with no fractional spur reduction scheme) during transient but in a narrow-bandwidth, integer-N mode during phase lock. This *hybrid* PLL, as an extension of the conventional fast-lock PLL (that shifts only its bandwidth while retaining the same frequency division mode), simultaneously achieves the fast-lock advantage of the fractional-N PLL and design simplicity of the integer-N PLL, bringing unique benefits. It also enables a new, more digital protocol to execute bandwidth switching. The hybrid PLL concept is affirmed by a 2.4-GHz, 1-MHz resolution CMOS hybrid PLL prototype of integer-N-level design-simplicity, which exhibits a 20-μs lock time for a 64-MHz frequency step, outperforming its fixed integer-N operation by a factor of 4.

Introduction

In PLL frequency synthesizers, bandwidths exert key influences on their dynamic characteristics, such as lock time and output spectrum. To minimize the lock time, the bandwidth should be maximized (the upper bound is about 10% of the reference frequency, f_{REF}, in the widely-used charge pump PLLs [1]). Too large a bandwidth, however, brings more spurs and component noise (except VCO noise) into the PLL dynamics, and hence the bandwidth for optimal spectrum is to be usually smaller than $0.1f_{REF}$. Therefore, maximizing the bandwidth for fast locking contradicts the need for a smaller bandwidth for optimum spectrum.

To circumvent this tradeoff, the variable-bandwidth PLL scheme [2] has been widely used. In this approach, a wider bandwidth is used during transient to reduce lock time, but once a phase lock is attained, the bandwidth is shifted to a smaller value for optimum spectrum. This scheme exploits the fact that lock time matters only during transient while spectrum matters only in steady states.

This paper generalizes this variable-bandwidth scheme, introducing a single-loop PLL that changes not only bandwidth but also frequency division mode between transient & steady states. Specifically [Fig. 1], this PLL operates in a narrow-bandwidth, integer-N mode during phase lock but in a wide bandwidth, fractional-N mode (with no fractional spur reduction circuit) during transient. The rationale, architecture, & measurements (CMOS IC) of this *hybrid* PLL will be presented (we reported only the theory part in [3]).

Rationale

To elucidate the rationale behind the hybrid PLL, we first consider the fractional-N & integer-N PLL comparatively. For a given frequency resolution, the former has a larger reference frequency than the latter, and hence, the loop bandwidth limited to 10% of the reference frequency can be set larger in the fractional-N PLL than in the integer-N PLL. As a result, the former assumes a faster locking. This speed advantage of the fractional-N PLL, however, comes at the price of increased design complexity. This is because the fractional-N operation in steady state requires fractional spur reduction circuits (*e.g.*, high-order $\Sigma\Delta$ modulators) whose quantization noise folds into the PLL spectrum via loop nonlinearities, demanding more significant design efforts to minimize the loop nonlinearities [1], [4]. On the contrary, integer-N PLLs involve much less design complexity due to the absence of fractional spurs.

Our hybrid PLL, which operates in a narrow-bandwidth integer-N mode during phase lock, but in a fractional-N mode with a wider bandwidth (as mentioned, the fractional-N PLL accommodates a wider bandwidth) during transient [Fig. 1], simultaneously achieves the fast-locking advantage of the fractional-N PLL and design simplicity of the integer-N PLL. Fast locking is a natural outcome of the wider-bandwidth fractional-N operation during transient. The design simplicity is attained because no fractional spur reduction scheme is needed in the fractional-N mode as spurs matter only in steady state. Due to the absence of such spur reduction circuits, the mode/bandwidth switching is executed via a simple alteration of what is almost a normal integer-N loop.

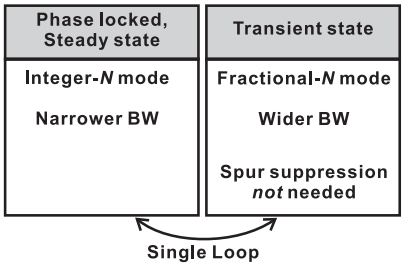

Fig. 1. Conceptual illustration of the hybrid PLL operation.

The hybrid PLL is essentially an integer-N PLL that is made faster than the normal integer-N PLL by borrowing the speed of the fractional-N PLL during transient. Therefore, the hybrid PLL can be valuable when design simplicity is a major priority and hence an integer-N PLL is preferred, but at the same time, the target frequency resolution is high (*e.g.*, GSM, Bluetooth, WLAN) so that normal integer-N PLLs with correspondingly small bandwidths have speed handicap. Since the integer-N PLL in steady state has inherently worse phase noise than the fractional-N PLL in steady state, the hybrid PLL would not be an optimal design choice when absolute phase noise minimization is desired. However, in certain applications (*e.g.*, GSM, Bluetooth, and WLAN), integer-N PLLs can still meet target phase noise specifications [5]- [7]. In such cases, the hybrid PLL will be a valuable choice for the reasons stated above.

Another distinctive feature of the hybrid PLL lies in its new, more digital protocol to execute the bandwidth switching (next section).

Architecture & Operation

Combining an integer- & fractional-N mode of differing bandwidths in a single loop, the hybrid PLL is formed as in Fig. 2.

Frequency Division Mode Switching

When the PLL of Fig. 2 enters a transient state, the two static divide-by-M blocks are disabled and screened out by the two multiplexers, and the crystal oscillator signal $x(t)$ [frequency: f_0] and the prescaler output, $d_1(t)$, are fed to the PFD. This corresponds to a standard fractional-N PLL with no fractional spur reduction circuit. The standard prescaler-accumulator combination inside the dashed box of Fig. 2 provides a fractional division ratio of $N_d = N + k/M$ through $d_1(t)$. The reference frequency is f_0.

When the PLL of Fig. 2 attains a phase lock, both divide-by-M blocks are enabled and their outputs, $x_1(t)$ and $d_2(t)$, are now fed to the PFD through the two multiplexers. This reconfigured loop is an integer-N PLL where the reference frequency is f_0/M and the frequency division ratio provided through $d_2(t)$ by the combination of the accumulator, divide-by-M, and prescaler inside the dashed box of Fig. 2 is an integer number, $N_d = NM + k$. The set of output frequencies and the frequency resolution are the same as in the fractional-N mode. The integer-divider [dashed box, Fig. 2]

may be unfamiliar as the conventional integer-divider uses a swallow counter instead of the accumulator [1]. To see the circuit inside the dashed box indeed provides integer division, refer to Fig. 3.

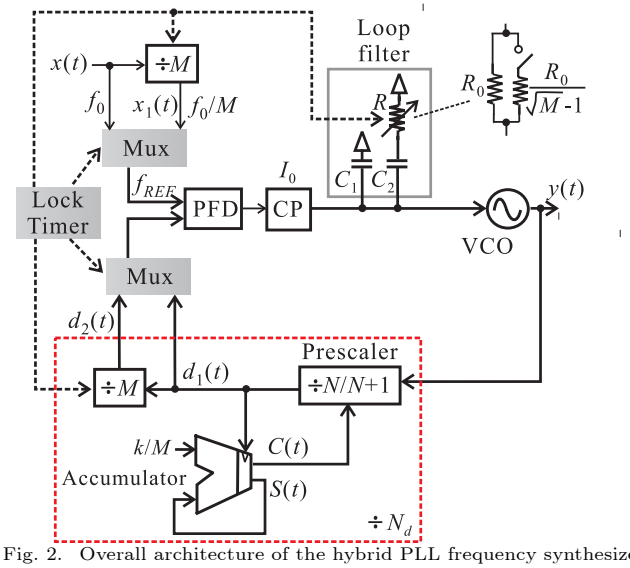

Fig. 2. Overall architecture of the hybrid PLL frequency synthesizer

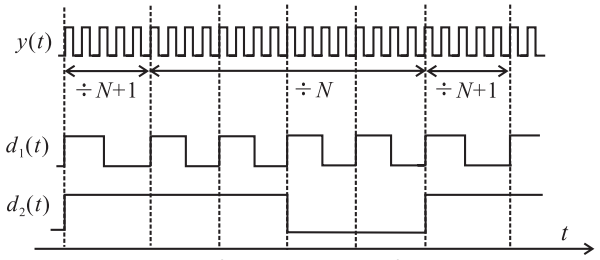

Fig. 3. Integer-N mode of the hybrid PLL of Fig. 2. N=4, M=5, k=1

Note that the simple switching between the two modes is enabled by the absence of fractional spur reduction circuits, and that the overall hybrid PLL of Fig. 2 is almost the normal integer-N PLL, with only 3 additional, simple digital components (shaded areas).

Loop Bandwidth Switching

Execution of the mode switching in sync with bandwidth switching is at the heart of the hybrid PLL operation. Using the well-known bandwidth-switching principle [2], one can show for a charge-pump PLL with a 2nd-order loop filter (gray box, Fig. 2) that to increase the loop bandwidth by a factor of α while keeping the same phase margin at the onset of a transient, the following adjustments of loop parameters must be performed together (I_0: charge-pump current, R: loop filter resistance, and N_d: frequency division ratio):

$$\begin{cases} \textit{Adjustment A:}\ \text{Reduction of } R \text{ by a factor of } \alpha. \\ \textit{Adjustment B:}\ \text{Increase of } I_0/N_d \text{ by a factor of } \alpha^2. \end{cases}$$

In conventional variable-bandwidth PLLs (fixed N_d), *Adjustment B* is performed only by I_0. In our hybrid approach, N_d is automatically reduced by a factor of M with the integer-N mode's shifting to the fractional-N mode at the onset of a transient, which can serve as an additional parameter to execute *Adjustment B*. This feature permits exploration of a larger design space for bandwidth switching, *e.g.*, when α is large, this new protocol can lessen the burden of the large I_0-increase, as the N_d-reduction can also contribute.

One especially interesting usage of the new bandwidth-switching protocol is to rely solely on the automatic N_d-reduction while keeping the same I_0 for *Adjustment B*: we then automatically have $\alpha = \sqrt{M}$. *Adjustment A* then can be executed by implementing a switch-bearing R [top right, Fig. 2] and closing the switch at the onset of a transient to reduce R by a factor of $\sqrt{M}$. Our CMOS IC uses this bandwidth switching scheme. Note that this represents a more digital fashion of bandwidth switching, and a polar opposite of the conventional bandwidth shifting where I_0 must be altered.

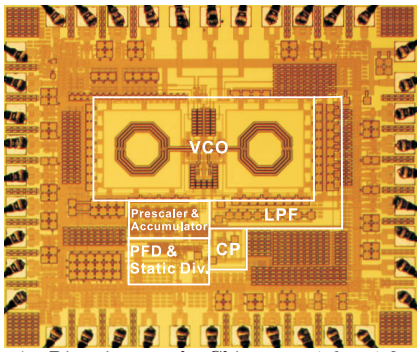

Fig. 4. Die micrograph. Chip area: 1.6 × 1.3 mm^2.

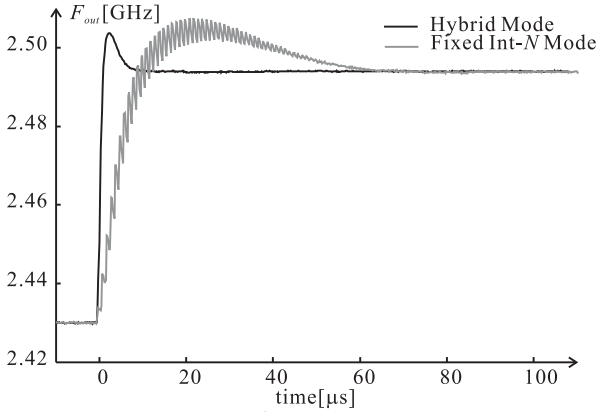

Fig. 5. Measured frequency settling transients.

Two points warrant comments. First, if the mode/bandwidth switching moment is well chosen, there will be no phase error glitch at the switching moment [3]. Experiments in the next section justify this statement. Second, the PLL in [8] uses a similar frequency division mode switching, but it *deliberately* maintains the same bandwidth, not fully exploiting the speed advantage of the hybrid approach. To preserve the same bandwidth, [8] uses what is topologically and operationally in striking contrast with our hybrid PLL.

Experimental Results

For proof of concept, the hybrid PLL [Fig. 2] was designed in TSMC 1.8-V, 0.18μm CMOS technology [Fig. 4: die micrograph]. The only off-chip components are the crystal oscillator, lock timer, and most of the loop filter. The IC is mounted on a 10-mm LQFP 44-pin package for on-board characterization. The target output frequencies are 2.368 $\sim$ 2.496 GHz (frequency resolution: 1 MHz).

Figure 5 shows the measured locking transients with a 64 MHz frequency jump for the hybrid PLL (black line) and for the fixed integer-N mode PLL (gray line) derived from the hybrid PLL by constantly operating it in its integer-N mode. In the hybrid operation, the switching from the fractional-N (bandwidth: 400 kHz) to the integer-N mode (bandwidth: 50 kHz) is executed at t=10 μs, and the measured lock time is about 20 μs. In the fixed integer-N operation (bandwidth: 50 kHz), the measured lock time is about 80 μs. This comparative characterizations clearly confirm the fast-locking advantage of the hybrid PLL in its almost integer-N architecture with the digital bandwidth switching. The total power dissipation excluding the output buffer is 16 mW. The measured phase noise of the hybrid PLL at its steady-state integer-N mode is -111.3 dBc/Hz at 1MHz offset. The 1 MHz reference spur is -54dBc.

References

[1] B. Razavi, *RF Microelectronics.*, Prentice Hall, 1998.
[2] Crowley, *US Patent* 4,156,855, May 29th, 1979.
[3] K. Woo and D. Ham, *IEEE MWSCAS.*, Aug. 2006.
[4] T. A. D. Riley, et al., *JSSC.*, vol. 28, no. 5, pp. 553-559, May. 1993.
[5] E. Duvivier, et al., *JSSC.*,vol. 38, no. 12, pp. 2249-2257, Dec. 2003.
[6] A. Ajjikuttira, et al., *IEEE ISSCC.*, pp. 198-199, Feb. 2001.
[7] S. Mehta, et al., *IEEE ISSCC.*, pp. 94-95, Feb. 2005.
[8] B. Memmler, et al., *IEEE ESSCIRC.*, pp. 468-471, Sep. 2000.
[9] D. Banerjee, *PLL perf., simul., and design.*, National Semi., 2005.