A Sub-Picosecond Hybrid DLL for Large-Scale Phased Array Synchronization

Matan Gal-Katziri and Ali Hajimiri
Department of Electrical Engineering
California Institute of Technology
Pasadena, CA 91125, USA
Email: mgal@caltech.edu

Abstract—A large-scale timing synchronization scheme for scalable phased arrays is presented. This approach utilizes a DLL co-designed with a subsequent 2.5GHz PLL. The DLL employs a low noise, fine/coarse delay tuning to reduce the in-band rms jitter to 323fs, an order of magnitude improvement over previous works at similar frequencies. The DLL was fabricated in a 65nm bulk CMOS process and was characterized from 27MHz to 270MHz. It consumes up to 3.3mW from a 1V power supply and has a small footprint of 0.036mm².

Keywords—*CMOS integrated circuits, phased-arrays, radio frequency, tracking loops, delay-lines, phase locked loops, phase noise.*

I. INTRODUCTION

Phased arrays are extensively used in radar, sensing, and communication systems due to their electronic beam steering capabilities combined with the added directivity and enhanced SNR/SIR which scale with number of array elements [1]. Example applications are 5G networks, which are currently targeting hundreds to thousands of elements, and very large-scale arrays, sometimes referred to as million-element arrays [2]. The practical implementation of such systems necessitates a broad range of architectural and technological innovations, such as scalable structures and highly-integrated silicon-based RFICs [2-5]. In this scalable array transceiver architecture, a single low-frequency reference clock is distributed to identical blocks (tiles), where high-frequency signals are synthesized (often using an integrated PLL-based frequency synthesizer) and used for coherent RF signal generation and reception in concert with the other array elements, as shown in Fig. 1.
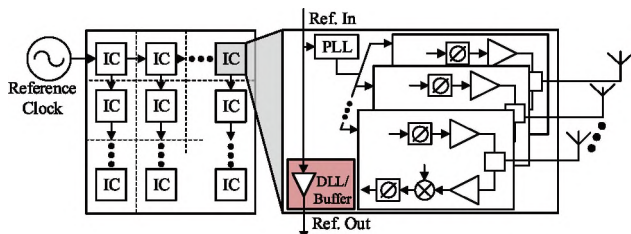

Fig. 1. Clock distribution to CMOS driven phased array

One major challenge with this architecture is maintaining the timing accuracy of the reference signal in the distribution process. A central star or H-tree distribution is impractical in the case of a large scale array as the number of traces and the

electrical load of all the driven elements become prohibitively large. On the other hand, sequential buffering of the reference suffers from large accumulated timing deviations due to variations in the supply, temperature, and the driven load. These challenges can be mitigated utilizing a delay-locked loop (DLL) in the repeater buffer. While fundamentally sound, this approach presents new challenges since the low reference frequency, usually a few tens of MHz, necessitates a relatively large delay which can lead to unacceptable timing jitter. We propose a hybrid DLL architecture that utilizes several noise reduction techniques as well as a novel semi-digital loop control scheme with a single phase detection path. Moreover, the co-design of the DLL with the subsequent PLL-based synthesizer is exploited to further reduce the overall timing jitter by proper alignment of the two phase noise transfer functions; one loop provides rejection over the frequencies where the other has a large noise contribution. This approach opens the design space, leading to superior overall performance.
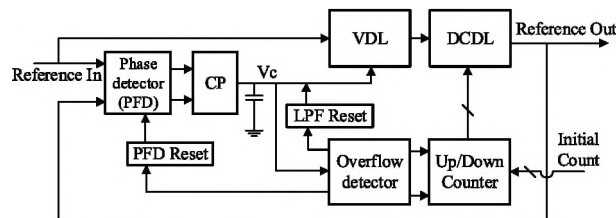
II. HYBRID DLL

Fig. 2. Hybrid DLL block diagram

In a DLL, the output signal must practically be delayed by at least half a clock period compared to the reference in order to correct both negative and positive timing errors. A standard implementation does so with a single continuous delay line, which is usually the main noise contributor due to the large delay range it needs to cover. A hybrid DLL can solve this problem by using two different sets of delay elements, as shown in Fig. 2. A digitally controlled delay line (DCDL) composed of low-noise fixed-delay elements is used for coarse delay tuning, while a short, continuously variable delay line (VDL) is used to fine tune within the digital segments.

In order to achieve delay lock, we use an analog DLL architecture and continuously monitor its charge pump (CP) output control voltage (Vc) to adjust the required DCDL value.
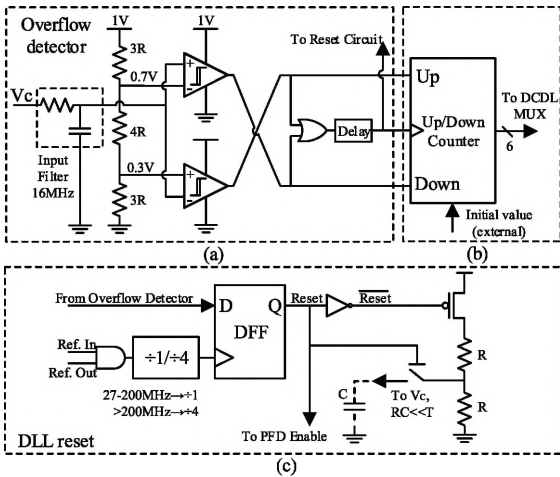
Fig. 3. (a) Overflow detect/actuate circuit (b) DCDL MUX set (c) reset circuit for hybrid operation

Initially, the up/down counter of Fig. 3b is set to fix the DCDL state, and the DLL loop of Fig. 2 continuously controls the VDL. If an unattainable VDL tuning value is required, the control voltage Vc will rail, crossing some lower or upper threshold along the way. This activates the overflow detector of Fig. 3a to pause the continuous control loop, initiate a single increase/decrease of a DCDL cell, and restart VDL tracking. Unlike [6], we are not changing the continuous delay range by flipping a state machine to set discrete phase states but are instead adding or removing fixed amount of low noise delay as required. This significantly improves the noise performance. In addition, we are tracking the same edge in a monotonous, continuous, and overlapping manner—which, when combined with the fact that the DLL is a first-order control loop, guarantees its stability. The reset circuitry in Fig. 3c is crucial to temporarily disable the phase detector and force Vc to mid-supply when a DCDL shift occurs and is synchronized such that the phase detector starts at a consistent state once the VDL tracking restarts. The noise-optimized, pseudo-differential delay elements of Fig. 4 also allow tracking of the falling edge of the output clock, which effectively reduces the minimum delay required by $T/2$ and enables usage of the same delay line at lower reference frequencies.
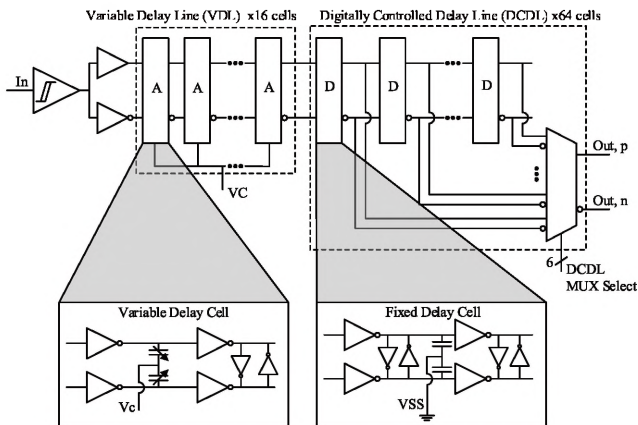

Fig. 4. Hybrid delay line structure

This architecture offers enhanced robustness because (1) it necessitates neither lock detect indication nor dual phase detection circuitry as in [7][8], (2) the small signal gain is identical for all DCDL values, and (3) the DCDL state changes in single up/down steps. The latter indicates that subsequent VDL tracking starts from a well-defined, nearby position, unlike a digital controller with automatic delay step adjustment. Our implementation favors clock distribution applications where lock time is not a major consideration. If necessary, fast lock is achievable with an *a priori* estimate of the DCDL delay step values and external programming of the up/down counter state.
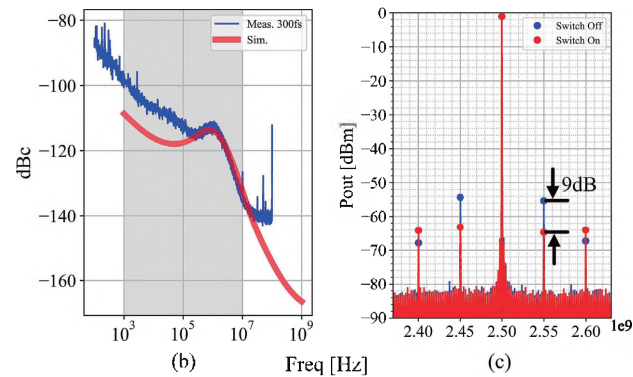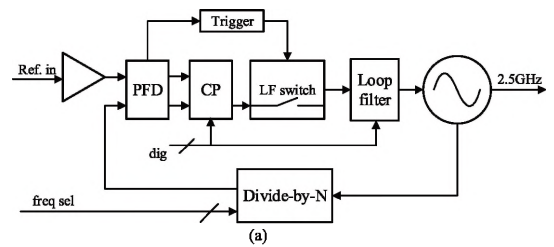
III. PLL DESIGN

Fig. 5. (a) Application PLL (b) PLL phase noise and rms jitter at 1kHz-10MHz (c) reference spurs with and without reduction

The hybrid DLL was co-designed with its intended load PLL (Fig. 5) for a 50MHz clock distribution of an existing RF phased array application. The PLL itself is fully integrated and operates at an output frequency of 2.5GHz with a loop bandwidth of 1MHz. It contains a mechanism similar to [9] to reduce its reference spurs, which, when present at the output of a large-scale transmitter array, might become a significant spectral disturbance. In order to minimize the DLL in-band noise, its loop filter bandwidth was optimized to be around 1MHz in order to sufficiently reject the delay line noise while maintaining a relatively flat noise shape around the PLL loop filter knee frequency.

IV. MEASUREMENT RESULTS

Both the DLL and the PLL were fabricated in a 65nm bulk CMOS process (Fig. 6). They occupy 0.036mm^2 and 0.4mm^2 of active area, respectively, and their joint operation was characterized at an output frequency of 2.5GHZ with the input reference ranging from 27MHz-270MHz.
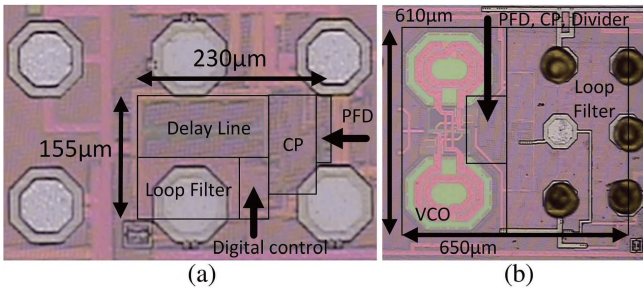
Fig. 6. Die micrographs of the (a) DLL and (b) the driven PLL

Fig. 7 shows the delay locking mechanism while the DLL drives either 50Ω or 10pF loads. The control voltage Vc in Fig. 7a overflows and resets until it reaches the necessary DCDL value, while fine-tuning persists indefinitely. The delay between the output and reference signals (Fig. 7b) was calculated from the waveforms' zero-crossing points, emphasizing how proper sizing of the overlapping DCDL step size and VDL range allow for proper operation of the circuit.
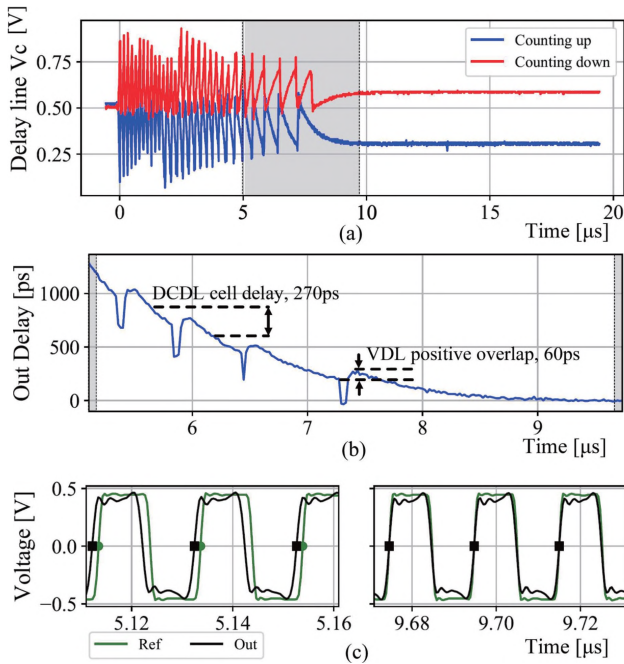

Fig. 7. Hybrid lock process at different time scales. (a) Loop filter control voltage, (b) time delay between reference and output clocks, and (c) time domain waveforms (adjusted)

In our phased-array application, the expected temperature fluctuation is less than 10°C in steady state, and the measured closed-loop control voltage tracks the temperature at a rate of 2.4mV/°C. The nominal control voltages for locking are 340mV and 660mV when counting up and down, respectively, and the overflow detector has a nominal hysteresis of 30mV. Therefore, temperature variations are not expected to toggle the digital counter and add additional, unaccounted noise. In our clock distribution scheme, static buffer phase offset is programmatically removed when the array is calibrated and therefore not of a major concern.

Fig. 8 shows how the DLL degrades the noise performance of a reference clock source by examining the phase noise spectral density profile of the cascaded application blocks, measured using a Keysight PXA N9030B signal analyzer. Notably, the frequency band of interest is above 1kHz, where phase errors are presumably correctable by external phased array adjustment algorithms, and below 10MHz, far away from the load PLL loop filter knee frequency.
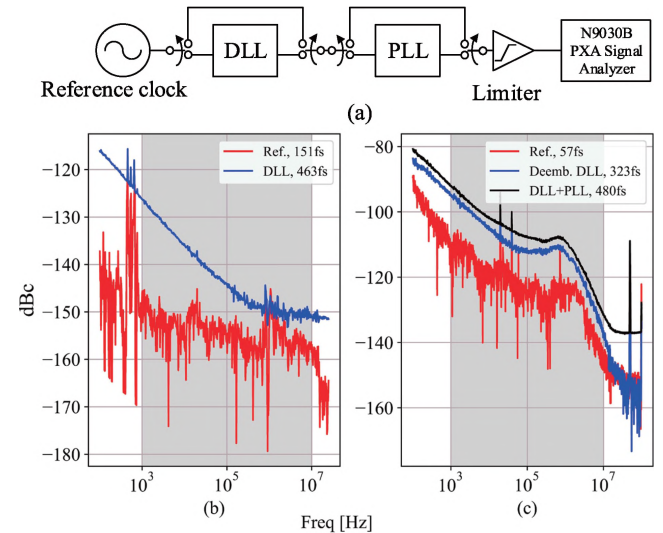

Fig. 8. (a) Phase noise test setup. Blocks are taken off when not measured. (b) 50MHz DLL phase noise and rms jitter (c) 2.5GHz post PLL phase noise and rms jitter. The red curves are the rms measurement uncertainty

Figs. 8b and 8c clearly show how the PLL loop filer rejects most of the DLL noise and thus brings it to contribute as little as 323fs rms jitter in the relevant frequency band.
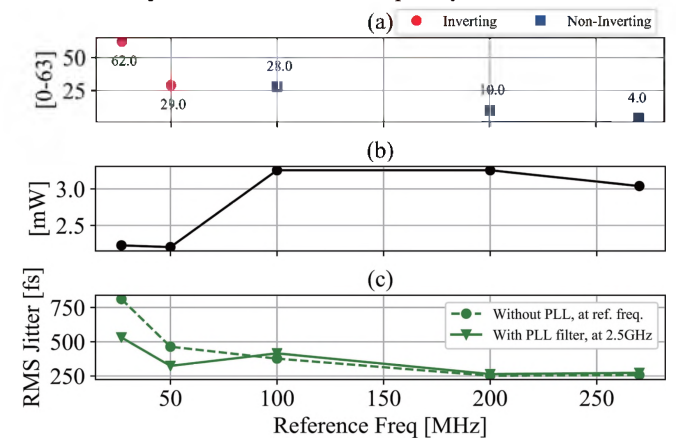

Fig. 9. Performance vs. frequency. (a) Participating DCDL cell count, (b) power consumption of participating DLL blocks, and (c) rms jitter within the 1kHz - 10MHz band

These measurements were repeated at different frequencies between 27MHz and 270MHz, and a summary is shown in Fig. 9. The lower and upper frequency ranges are limited by the maximum DCDL delay and overflow actuation timing accuracy, respectively. Fig. 9b demonstrates how this DLL is

advantageous in that an increase in the frequency of operation decreases the number of DCDL elements that participate in the delay chain, and thus the power consumption remains roughly constant. Fig 9c emphasizes how the system is optimized for 50MHz operation. At lower frequencies, the high DCDL count adds more noise to the output, while at higher frequencies the subsequent PLL loop filter has little effect on noise rejection.

Because the end goal is the phased array reference distribution scheme of Fig. 1, noise performance was characterized for several, cascaded DLLs. If the noise of each stage is uncorrelated with the others, the total noise measured at the output of an N DLL cascade is expected to be:

$$n^2_{total} = n^2_{ref} + n^2_{meas} + N \cdot n^2_{DLL} \qquad (1)$$

where n_{meas} is the measuring instrument noise, n_{ref} is the reference noise, and the single device noise can be estimated from the slope of the linear fit. Fig. 10 shows the linear behaviour of the DLL cascade jitter variance at different frequencies and the resulting rms jitter is summarized in Table I, showing good agreement with the single device measurements of Figs. 8b and 9c.
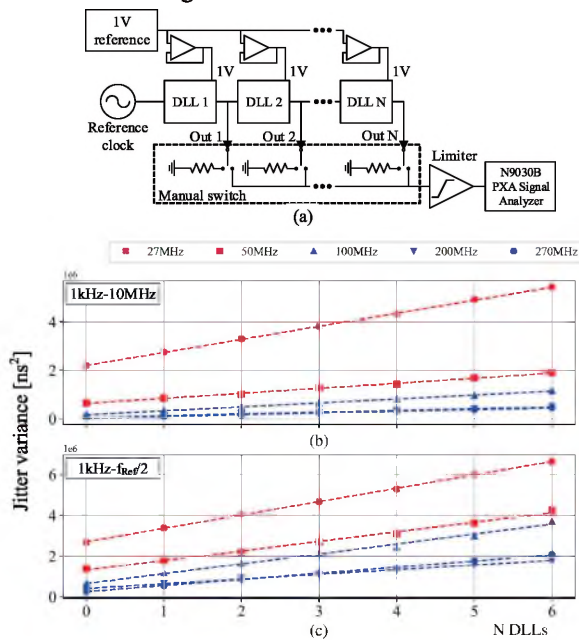

Fig. 10. Cascaded DLL jitter (a) test setup, (b) 1kHz - 10MHz measurement, and (c) 1kHz-f_{Ref}/2 measurement. Red and blue curves indicate locking to inverted/non-inverted output, respectively.

DLL NOISE PERFORMANCE, BASED ON FIG. 10

Reference Frequency [MHz]	27	50	100	200	270
RMS Jitter, 1kHz-10MHz [fs]	733	456	402	268	261
RMS Jitter, 1kHz-f_{Ref}/2 [fs]	809	685	698	481	549

V. CONCLUSIONS

The task of distributing a low noise reference to very large-scale phased arrays is challenging because it does not enjoy the shorter period times of GHz range clocks. Table II shows a performance comparison of the hybrid DLL/PLL scheme with prior art at similar frequency ranges, and demonstrates how combining new circuit architectures with application-aware design can result in an order-of-magnitude improvement over the state-of-the-art.

REFERENCES

[1] W. L. Stutzman and G. A. Thiele, Antenna Theory and Design, 2nd ed. New York: Wiley, 1998.

[2] S. Jeon et al., "A Scalable 6-to-18 GHz Concurrent Dual-Band Quad-Beam Phased-Array Receiver in CMOS," in IEEE JSSC, vol. 43, no. 12, pp. 2660-2673, Dec. 2008.

[3] Xiang Guan, H. Hashemi and A. Hajimiri, "A fully integrated 24-GHz eight-element phased-array receiver in silicon," in IEEE JSSC, vol. 39, no. 12, pp. 2311-2320, Dec. 2004.

[4] A. Natarajan, A. Komijani and A. Hajimiri, "A fully integrated 24-GHz phased-array transmitter in CMOS," in IEEE JSSC, vol. 40, no. 12, pp. 2502-2514, Dec. 2005.

[5] F. Bohn, B. Abiri and A. Hajimiri, "Fully integrated CMOS X-Band power amplifier quad with current reuse and dynamic digital feedback (DDF) capabilities," 2017 IEEE RFIC 2017, pp. 208-211.

[6] Byung-Guk Kim and Lee-Sup Kim, "A 250-MHz-2-GHz wide-range delay-locked loop," in IEEE JSSC, vol. 40, no. 6, pp. 1310-1321, June 2005.

[7] Yeon-Jae Jung et al., "A dual-loop delay-locked loop using multiple voltage-controlled delay lines," in IEEE JSSC, vol. 36, no. 5, pp. 784-791, May 2001.

[8] K. H. Cheng and Y. L. Lo, "A Fast-Lock Wide-Range Delay-Locked Loop Using Frequency-Range Selector for Multiphase Clock Generator" in IEEE TCAS-II, vol. 54, no. 7, pp. 561-565, July 2007.

[9] B. Zhang, P. E. Allen and J. M. Huard, "A fast switching PLL frequency synthesizer with an on-chip passive discrete-time loop filter in 0.25-μm CMOS," in IEEE JSSC, vol. 38, no. 6, pp. 855-865, June 2003.

[10] C. N. Chuang and S. I. Liu, "A 20-MHz to 3-GHz Wide-Range Multiphase Delay-Locked Loop," in IEEE TCAS-II, vol. 56, no. 11, pp. 850-854, Nov. 2009.

[11] D. Zhang et al., "A fast-locking digital DLL with a high resolution time-to-digital converter," Proceedings of the IEEE 2013 CICC, San Jose, CA, 2013, pp. 1-4.

TABLE II

PERFORMANCE SUMMARY AND COMPARISON TO PRIOR WORK

	This work		JSSC 05' [6]	TCASII 07' [8]	TCASII 09' [10]	CICC 13' [11]
Frequency range	27-270MHz		0.25-2GHz	32-320MHz	0.02-3GHz	80-450MHz
Comparison frequency	50MHz	270MHz	250MHz	200MHz	50MHz	180MHz
RMS jitter [ps]	0.685	0.55	5.25	4.44	7 (approx.)	2.3
In band RMS jitter	0.33ps	0.26ps	NA	NA	NA	NA
Power consumption [mW]	2.25	3	1.2	15 (320MHz)	0.4-3.6	26
Supply voltage [V]	1		1.8	2.5	1	1.5
Technology process [CMOS]	65nm		180nm	250nm	90nm	130nm
Die area [mm²]	0.036		0.046	0.07	0.005	0.08