

A Highly Digital MDLL-Based Clock Multiplier That Leverages a Self-Scrambling Time-to-Digital Converter to Achieve Subpicosecond Jitter Performance

Belal M. Helal, *Member, IEEE*, Matthew Z. Straayer, *Student Member, IEEE*, Gu-Yeon Wei, and Michael H. Perrott

Abstract—This paper presents a mostly digital multiplying delay-locked loop (MDLL) architecture that leverages a new time-to-digital converter (TDC) and a correlated double-sampling technique to achieve subpicosecond jitter performance. The key benefit of the proposed structure is that it provides a highly digital technique to reduce deterministic jitter in the MDLL output with low sensitivity to mismatch and offset in the associated tuning circuits. The TDC structure, which is based on a gated ring oscillator (GRO), is expected to benefit other PLL/DLL applications as well due to the fact that it scrambles and first-order noise shapes its associated quantization noise. Measured results are presented of a custom MDLL prototype that multiplies a 50 MHz reference frequency to 1.6 GHz with 928 fs rms jitter performance. The prototype consists of two 0.13 μm integrated circuits, which have a combined active area of 0.06 mm^2 and a combined core power of 5.1 mW, in addition to an FPGA board, a discrete DAC, and a simple RC filter.

Index Terms—Correlated double sampling, correlation, deterministic jitter, first-order noise shaping, gated ring oscillator (GRO), multiplying delay-locked loop (MDLL), reference spur, scrambling, time-to-digital converter (TDC).

I. INTRODUCTION

AS TECHNOLOGY has advanced, on-chip clock multiplication has become a necessity for nearly all digital integrated circuits (ICs) in order to realize high-speed clock signals from lower speed external sources such as crystal oscillators. The typical approach to achieve such clock multiplication is to employ a phase-locked loop (PLL) circuit consisting of a phase detector, analog loop filter, frequency divider, and voltage-controlled oscillator (VCO). Unfortunately, the analog content of PLLs prevents their design from easily fitting into a typical digital design flow.

Multiplying delay-locked loops (MDLL) [1], [2] have been introduced recently as an alternative to PLLs for clock multiplication. Although similar in concept to [3], MDLLs offer much better jitter performance. As seen in Fig. 1, the MDLL operates by replacing every N th edge of a naturally running ring

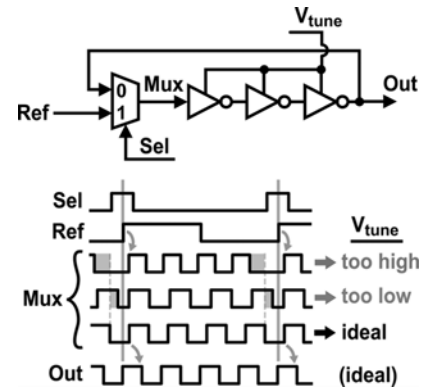


Fig. 1. Conceptual MDLL clock multiplier and impact of tuning voltage on its associated signals.

oscillator VCO with a reference frequency edge, where N corresponds to the frequency multiplication factor. This has been shown to allow significant suppression of jitter caused by phase noise of the VCO [2]. However, as shown in the figure, an incorrect setting of the V_{tune} voltage on the VCO (which tunes its corresponding frequency) leads to undesired “deterministic jitter” due to corresponding periodic changes in the output period. Reduction of such deterministic jitter is the focus of this paper, with the aim of achieving such reduction with a highly digital implementation [11] that is insensitive to such analog issues as mismatch and offset between circuit components.

An overview of the paper is as follows. Section II provides background of previous approaches to MDLL tuning and highlights their sensitivity to analog nonidealities. Section III presents the proposed MDLL tuning structure, which achieves low jitter with a mostly digital structure and low sensitivity to analog nonidealities. The proposed approach relies heavily on a new time-to-digital (TDC) structure based on a gated ring oscillator (GRO), which is described in detail in Section IV. The overall implementation is presented in Section V, while key circuit blocks are described in Section VI. Section VII presents measured results and, finally, conclusions are presented in Section VIII.

II. BACKGROUND

Fig. 2 shows the classical feedback approach used for adjustment of V_{tune} . The key idea of this approach is to use a phase detector to measure the difference in time Δ between two appropriate edges in the system and then use a charge pump and

Manuscript received August 27, 2007; revised November 4, 2007.

B. M. Helal, M. Z. Straayer, and M. H. Perrott are with the Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: bhelal@mit.edu; straayer@mit.edu; perrott@mit.edu).

G.-Y. Wei is with Harvard University, Cambridge, MA 02138 USA (e-mail: gyeon@eecs.harvard.edu).

Digital Object Identifier 10.1109/JSSC.2008.917372

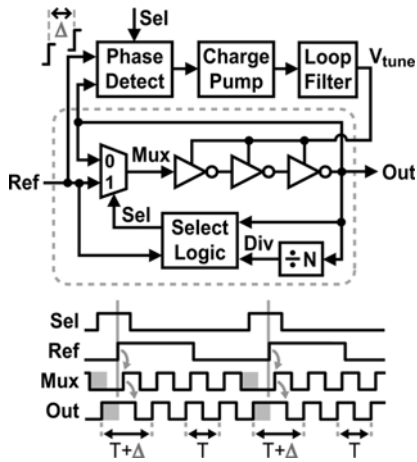


Fig. 2. Classical approach to MDLL tuning.

loop filter to integrate the resulting error signal to form V_{tune} . Ideally, V_{tune} will then be adjusted by the feedback loop until Δ goes to zero, which would ideally lead to zero deterministic jitter under steady-state conditions.

Unfortunately, practical circuit implementations for the traditional MDLL tuning approach are sensitive to nonidealities which cause the V_{tune} feedback loop to settle to a nonzero value of Δ , such that a substantial amount of deterministic jitter is introduced into the MDLL output [1], [2], [4]–[6]. The major nonidealities are path mismatch in the multiplexer and phase detector, mismatch between the currents of the charge pump, and finite dc output impedance of the charge pump output. While several techniques have been recently proposed to reduce the impact of these nonidealities [4], [5], the relatively high analog design effort required by these approaches makes them less amenable to inclusion within standard digital design flows. Therefore, it is attractive to develop an MDLL tuning architecture that is insensitive to such analog nonidealities and which requires minimal custom analog design effort for its implementation.

III. PROPOSED HIGHLY DIGITAL MDLL ARCHITECTURE

The proposed MDLL tuning architecture, which is shown in simplified form in Fig. 3, dramatically reduces the impact of path mismatch by avoiding comparison of *two different* edge signals as pursued in classical structures. Instead, *one* signal is examined, *Enable*, whose pulsewidth alternates twice every reference cycle between the free running period of the oscillator, T , and the period of the error-affected cycle, $T + \Delta$. By doing a relative comparison of each consecutive pulse period of the *Enable* signal, the value of Δ can be obtained in a manner such that the issue of mismatch is greatly mitigated since only one signal is being examined. This technique is referred to as correlated double-sampling when used in analog circuits and is commonly applied in applications such as imagers and switched capacitor circuits to reduce dc offset and $1/f$ noise [7], [8].

While classical analog applications of the correlated double-sampling technique leverage switches, sampling capacitors, and operational amplifiers in their implementation, the goal for the MDLL application examined here is to avoid such analog blocks

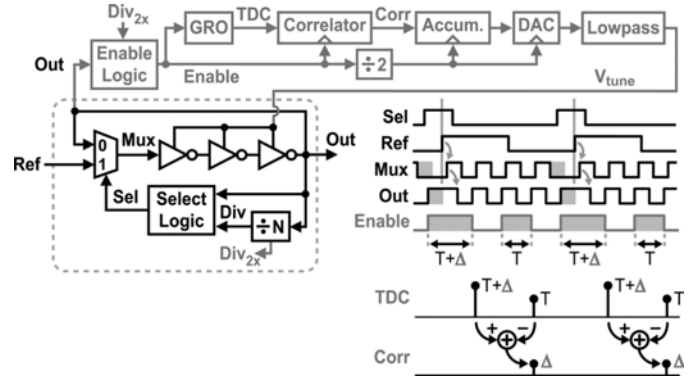


Fig. 3. Proposed MDLL tuning method leveraging a TDC and correlated double-sampling.

and instead seek a highly digital implementation. As shown in Fig. 3, this goal can be realized by leveraging a TDC structure, which in this case is based on a GRO, to measure the periods in the *Enable* signal. The GRO is discussed in more detail in the following section. As shown in Fig. 3, this block outputs a digital signal, *TDC*, which is updated at the end of each *Enable* pulse and corresponds to a quantized measurement of the corresponding *Enable* pulse period (i.e., T or $T + \Delta$). A digital correlator circuit simply subtracts consecutive pairs of the *TDC* samples to yield a stream of samples, *Corr*, which correspond to quantized estimates of Δ . By passing these Δ samples into a digital accumulator, the quantization error of the Δ samples is reduced by the averaging effect of the accumulation operation (assuming that the quantization error varies in an appropriately random fashion). The accumulator output adjusts V_{tune} through a DAC with an accompanying RC low-pass filter until Δ reaches zero at steady state.

There are several advantages of using the proposed MDLL tuning structure over previous approaches. First, the only analog blocks required by this tuning approach are a DAC and a simple RC low-pass filter, so that custom analog design effort for the proposed MDLL implementation is reduced compared with competing approaches given that an appropriate DAC structure is available to the designer. In the case of the prototype system in this paper, first-order Sigma-Delta modulation of 8 MSB inputs of a 16-bit DAC (with the 8 LSB inputs being ignored) running at 50 MHz, along with a passive RC filter pole at 3 MHz, proved sufficient to achieve subpicosecond jitter levels at the MDLL output. Second, the architecture is inherently insensitive to analog mismatch and offset due to the use of a correlated double-sampling technique. Third, the digital accumulator structure has infinite dc gain, which is in stark contrast to the limited dc gain of analog integrators, so that the full range of V_{tune} can be achieved without any secondary impact on the steady-state value of Δ . In addition, the compact area of the digital accumulator allows easy integration of a low-bandwidth tuning loop without concern for large capacitive area or degraded leakage characteristics. Finally, the highly digital architecture, which is insensitive to analog mismatch and offset, should greatly improve portability of the design between different CMOS technologies.

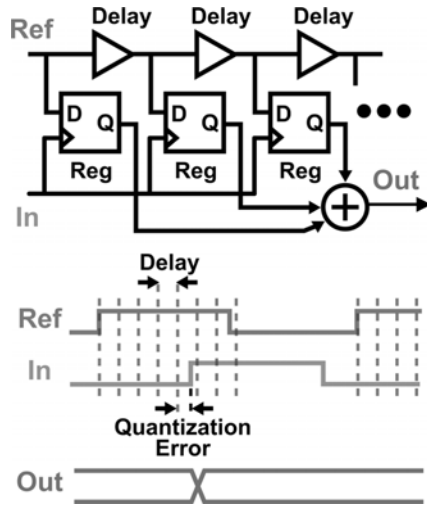


Fig. 4. Classical time-to-digital structure and associated signals.

Of course, there are some subtle issues associated with the design that must be addressed to achieve high performance. One of the key issues is achieving a TDC structure that has subpicosecond effective resolution, which is required for corresponding reduction of the MDLL deterministic jitter to subpicosecond levels. This issue is addressed in the next section as the GRO structure is discussed in more detail. Another important issue is the sensitivity to power supply noise, especially if the same supply is shared among various blocks or if noisy digital blocks are in close vicinity. In such case, it is essential to adequately decouple, and possibly regulate, the supplies of the VCO, DAC, and any blocks through which the reference signal passes. However, the GRO and the Enable Logic blocks are less critical since the correlation operation and the low bandwidth of the loop suppress their supply noise more effectively [6], though proper decoupling and regulation is certainly advised for these blocks, as well.

IV. PROPOSED GRO TDC

Here, we provide a brief overview of the classical TDC structure and highlight its limitations in achieving high resolution. We then present the proposed GRO structure, explain its benefits in achieving the desired resolution performance with minimal complexity, and then describe some key issues related to its implementation.

A. Classic TDC Structure

Fig. 4 displays the classical delay chain TDC structure, which uses delay cells and synchronized registers to detect the time difference between two input edges [9]. The quantization error for this approach is set by the delay cell, which typically corresponds to an inverter delay. Since inverter delays are currently greater than 10 ps in modern CMOS processes [9], the classical structure falls over an order of magnitude short in meeting the desired subpicosecond resolution that we seek. In the MDLL application, this limitation could be overcome through averaging of many TDC measurements (by the digital accumulator) if the quantization noise of the classical structure was sufficiently random [10]. Unfortunately, the classical TDC always

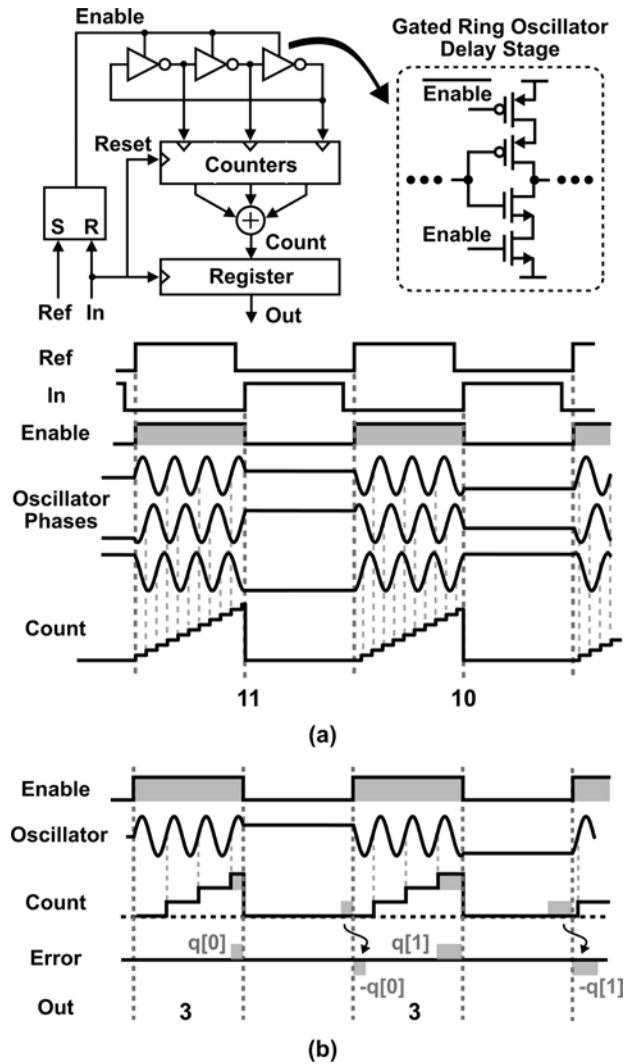


Fig. 5. (a) Proposed GRO time-to-digital structure and associated signals. (b) Closer view of quantization noise in GRO structure (shown for one oscillator phase).

yields the same quantization error for a fixed time difference between the input edges, so that averaging is not helpful for increasing the effective resolution. In practice, the time differences measured by the TDC will vary according to cycle-to-cycle jitter in the MDLL output, but the goal of achieving subpicosecond jitter performance eliminates the possibility of this jitter being an adequate dithering source for achieving higher effective TDC resolution through averaging.

B. Proposed GRO TDC Structure

As shown in Fig. 5(a), the proposed GRO structure achieves time measurement by counting edges occurring in each stage of a ring oscillator that is only turned on during the measurement interval. The gating circuitry for the oscillator, which consists of nMOS and pMOS enable devices, is designed to *hold the state* of the oscillator between measurements, so that the ring oscillator starting phase of a given measurement interval corresponds to the stopping phase of the previous measurement interval. The variable starting phase of the oscillator effectively scrambles the quantization noise across the different measurement intervals,

so that averaging of the GRO measurements yields an improved resolution signal.

In addition to the benefit of scrambling, the GRO structure also provides noise shaping of its quantization noise. This property is illustrated in Fig. 5(b), which shows a closer view of the start and stop phase progression across a few different measurement intervals. As revealed in the figure, the impact of holding the phase state between measurement intervals is to create quantization noise, $\text{Error}[k]$, that is a function of the previous quantization error, $q[k-1]$, and the current quantization error, $q[k]$, as

$$\text{Error}[k] = q[k] - q[k-1]. \quad (1)$$

Equation (1) reveals that the quantization noise of the GRO is, in fact, first-order noise shaped. Therefore, the GRO not only scrambles its quantization noise, but also shapes it to higher frequencies. Since averaging of the GRO output effectively acts like a low-pass filter on the quantization noise, the first-order noise shaping provides improved effective resolution for a given amount of averaging compared to having a white quantization noise profile. The noise-shaping property of the GRO will be confirmed in the measured results section of this paper.

C. Issues Related With the GRO Structure

There are a few issues to be aware of in using the GRO as the TDC within the MDLL tuning structure. First, while the noise-shaping property of the GRO is expected to be very useful in a variety of other PLL/DLL applications, it does not provide any benefit to the MDLL application that is our current focus. The issue at stake, which is illustrated in Fig. 3, is that the correlated double-sampling technique used in the MDLL tuning loop requires a relative comparison of alternate samples of the GRO output, which effectively leads to multiplication of the GRO samples by the alternating sequence $\{\dots, 1, -1, 1, -1, \dots\}$. This multiplication operation, in turn, causes mixing of the higher frequency GRO quantization noise down to lower frequencies, thereby removing the noise-shaped characteristics in the original GRO signal. Fortunately, due to the relatively low bandwidth (less than 10 kHz) of the MDLL tuning loop (and the high degree of averaging that it provides), the scrambling action of the GRO quantization noise is sufficient without noise shaping to achieve the effective subpicosecond resolution that we desire. Therefore, the benefit that the GRO brings to the MDLL application is not its noise shaping characteristic, but rather its inherent operation of scrambling the quantization noise so that averaging can improve the effective measurement resolution. The noise-shaping property was mentioned here simply to point out its potential benefit to other PLL/DLL applications.

The other GRO issue worth mentioning is that the current implementation, shown in Fig. 5(a), exhibits dead zones in the transfer characteristic. The key issue is that the state of the oscillator is not transferred perfectly from one measurement to the next, which introduces additional error to the measurement. This additional error is a nonlinear function of when the delay stage transitions are gated, implying that some states resume oscillation more quickly than others. The deadzone, which is an

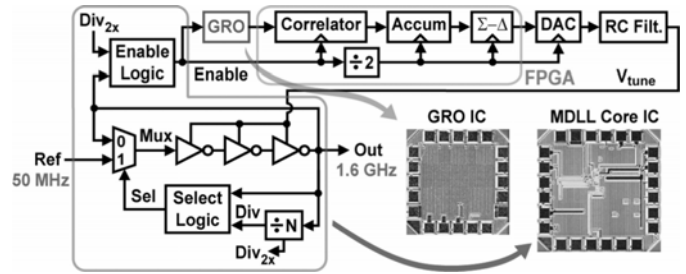


Fig. 6. Overall MDLL prototype.

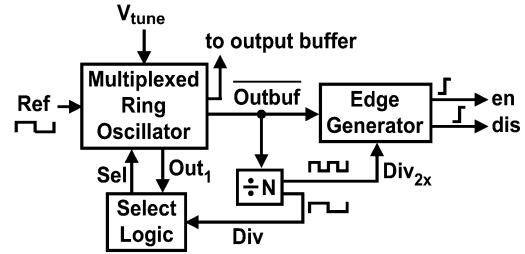


Fig. 7. MDLL core logic.

effect of this additional error, occurs when the input time measurement intervals are repeatedly close to an integer multiple of the nominal delay per TDC element. In this case, the additional nonlinear error term will pull the oscillator phase towards a preferred state, effectively forming a trap for the error accumulation. Once this equilibrium is reached, the TDC will output a constant value for each quantization without scrambling. Fortunately, the dead zones have a predictable and limited region of influence in the transfer characteristic, so simple hand tuning of the GRO (by proper adjustment of its supply voltage) is sufficient to alleviate this issue when testing the MDLL prototype. Elimination of the GRO dead zone behavior is a topic of current research in our group.

V. OVERALL MDLL IMPLEMENTATION

The overall MDLL prototype, which is shown in simplified form in Fig. 6, consists of two custom integrated circuits that implement the GRO and MDLL core logic, an FPGA board that implements the correlator, accumulator, a first-order digital Σ - Δ modulator, and other basic logic operations, an off-chip, low-noise, 100-MHz reference source, and a commercially available 16-bit DAC. The 100 MHz reference frequency is internally divided on the MDLL core IC to a range of possible reference frequencies including 12.5, 25, 50, and 100 MHz. The custom GRO and MDLL core chips were wire bonded directly to a gold-plated test board, to which the commercially available FPGA board was connected using surface mount connectors. The gain of the tuning loop was adjusted by bit shifting in the FPGA, and all settings were controlled from a PC through a USB connection. While 16-bits are available for the DAC, only 8 bits are used in conjunction with a first-order Σ - Δ modulator. The RC filter consisted of a discrete resistor and capacitor with a pole location of 3 MHz.

Fig. 6 shows die photographs of the MDLL core and the GRO TDC chips, which were fabricated in a 0.13 μm CMOS process. The active size of the MDLL core chip is 150 μm \times 250 μm out

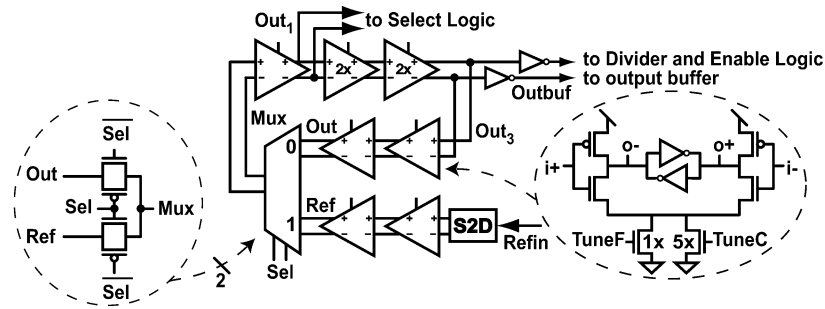


Fig. 8. Multiplexed ring oscillator.

of a total die size of $1.2 \text{ mm} \times 1.2 \text{ mm}$, and the active size of the GRO chip is $120 \mu\text{m} \times 172 \mu\text{m}$ out of a total die size of $1 \text{ mm} \times 1 \text{ mm}$.

VI. MDLL CORE CIRCUIT DESIGN

Here, we turn our attention to the various subblocks of the MDLL core logic which are shown in Fig. 7. The first subsection focuses on the multiplexed ring oscillator. The next subsection provides details on the Select Logic block which controls the multiplexer operation of replacing every N th edge of the ring oscillator with a clean reference edge. Finally, we turn our attention to the Edge Generator, which is the key circuit in the Enable Logic. It generates two signals, *en* and *dis*, that drive *Ref* and *In*, respectively, of the GRO TDC to produce its *Enable* signal [as shown in Fig. 5(a)].

A. Multiplexed Ring Oscillator

Fig. 8 shows the multiplexed ring oscillator and its constituent delay cells. The delay cells are similar to [12], except that only a single-ended nMOS bias is used for frequency tuning in order to improve phase noise, increase speed and reduce complexity. Separate coarse and fine tuning ports, TuneC and TuneF, which are implemented by different size nMOS devices as shown in the figure, are used to achieve a wide frequency range and a relatively low K_v value for the MDLL tuning feedback loop (which feeds only the fine tune port). The narrowed tuning range offered by the fine tune port both reduces the impact of noise from the MDLL tuning loop and also helps prevent subharmonic locking during the tuning process. In the prototype, we currently tune the voltage on the coarse port by hand to achieve the appropriate frequency range for the fine-tune port.

An important design consideration of the multiplexed ring oscillator is to match the slope of the edges of the two inputs feeding into the multiplexer, which correspond to the output of the last delay cell and the reference input, in order to minimize deterministic jitter [5]. In addition, care must be taken to avoid influence of the *Sel* signal on the edges running through the multiplexer since such influence would also lead to increased deterministic jitter [1]. To deal with the first issue, the reference input signal is buffered using two delay cells that are identical in design and tuning to the oscillator delay stages, as shown in Fig. 8. Each of these delay stages is placed in close proximity to each other in the chip layout in order to achieve good matching between them. As for the second issue, the impact of *Sel* is sought

to be minimized by striving for fast edges going through the multiplexer [1] so that there is a smaller time window for *Sel* to influence them. To this end, the number of delay cells is chosen to be as large as possible while still supporting the desired frequency range of the oscillator, which leads to less delay per stage and, therefore, faster edges. It should be noted that increasing the number of delay stages does not have a significant impact on phase noise [13]. In this $0.13 \mu\text{m}$ CMOS design, the choice of five delay stages allows oscillation frequencies high enough to achieve our 1.6 GHz target with a comfortable design margin. Note that the second and third delay stages were doubled in size in order to drive external blocks. Also, edges going in and out of the multiplexer are kept sharp by eliminating external loading on its input and output.

Differential load balancing in the oscillator delay stages produces a more symmetric waveform, and thus, lowers $1/f$ noise [13]. To achieve that goal, care was taken to provide matching loads for all delay cells. To that effect, the output of the first delay cell after the multiplexer, *Out₁*, drives two identical gates in the select logic. Similarly, *Out₃* drives two identical inverters, with one output feeding the output buffer, *Outbuf*, and another, *Outbuf*, that drives the divider and the edge generator.

Proper design of the MUX is required to avoid mismatch between the *Out* and *Ref* edges while they pass through it. This issue is especially problematic for architectures that detect the error by comparing the edges of the two MUX inputs since the measurement circuitry will not be able to detect any error due to path mismatch in the MUX which occurs after the observation nodes. Fortunately, this issue is significantly mitigated in the proposed architecture since the single-path detection method that is employed will detect the error regardless of its source. Nevertheless, care was taken to match the two paths of the MUX, and to minimize its propagation delay so that the impact of any remaining mismatch would be reduced.

B. Select Logic

The select logic circuit and its timing diagram are illustrated in Fig. 9. The main goal of this block is to generate a select signal, *Sel*, with sharp edges that are sufficiently separated in time from the falling *Out* and the rising *Ref* edges. Also, care must be taken to generate the *Sel* signal edges in approximately the middle of the ring oscillator transitions. This is important in order to minimize the influence of multiplexing and the

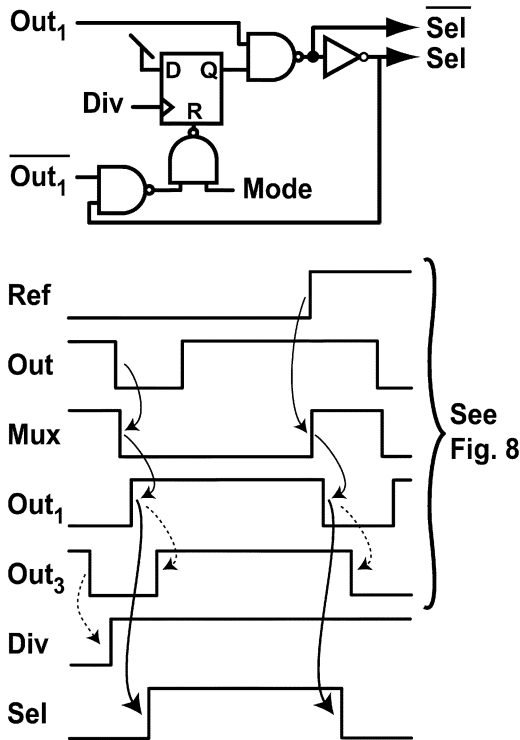


Fig. 9. Select logic.

feedthrough from the *Sel* signal on the edges passing through the multiplexer around the time of its switching.

In normal operation, the select logic is enabled (by pulling the signal *mode* high), and the select signal is generated as follows: the last falling edge of *Out₃* (before multiplexing *Ref*) causes the divider output, *Div*, to rise and trigger a D-flip-flop with reset (DFFR), which allows the NAND gate and the subsequent inverter to generate a rising *Sel* edge after a rising *Out₁* edge. The falling *Out₁* edge causes *Sel* to fall and resets the DFFR to make it ready for the next select cycle. Note that signal *Out₁* is chosen to drive the select logic in order to guarantee that the multiplexer switches at a time that is approximately in the middle of the *Out* transition. This design alleviates the challenge of optimal positioning of the *Sel* signal as posed in [5].

Note that all NAND gates used in the select logic circuitry are identical so that the load is symmetric on the *Out₁* branch of the ring oscillator. In addition, the design is almost entirely based on standard cells for ease of design and portability.

C. Edge Generator

The Edge generator circuit and its timing diagram are illustrated in Fig. 10. This is the first stage in generating the *Enable* signal that drives the GRO, and it generates two signals, *en* and *dis*, whose relative delay captures the period of two MDLL output cycles every reference cycle, namely T and $T + \Delta$.

As shown in Fig. 10, the edge generator has two inputs. The first input, *Outbuf*, is an inverted version of *Out₃* (the output of the third delay cell in the ring oscillator), and it carries the period information. The second input, *Div_{2x}*, is the divider output that runs at twice the reference frequency, and it selects the proper

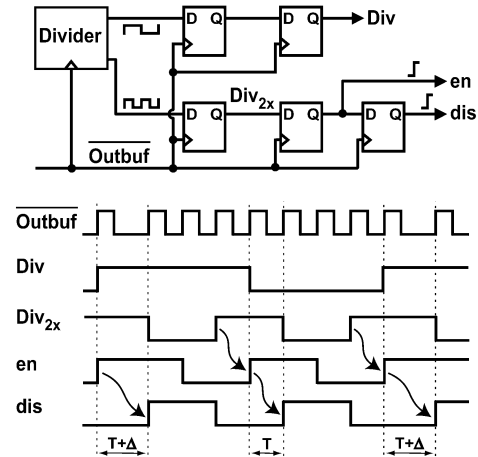


Fig. 10. Edge generator.

period to sample; however, its exact edge location does not affect the measurement. *Div_{2x}* is retimed twice by *Outbuf* to generate two signals, *en* and *dis*, which have the desired property of being separated in delay by the corresponding MDLL output period. These two signals go off-chip to the GRO TDC to form the *Enable* signal.

A retiming stage is used in the *Div_{2x}* path to increase immunity to metastability for the retiming DFF that generates the *en* signal, as shown in Fig. 10. The divider output, *Div*, is sent through two retiming stages. One stage is used in order to relax the timing margins for the select logic, while the second is needed to delay the rising edge of *Div* for one output cycle in order to synchronize it with the rising edge of *en*, such that the period that includes the error, Δ , is captured.

Note that any offset or nonideality that affects the generation of the *en* and *dis* signals is consistent between samples and, hence, will be canceled by the subtraction operation in the digital correlator. Furthermore, the design is simple and robust, and the DFF cells do not need to be custom-designed, which makes it amenable to porting between technologies.

VII. MEASURED PERFORMANCE

Here, we present measured performance of the custom prototype. Although our primary attention will be given to demonstration of the overall MDLL jitter performance, we begin by verifying the noise shaping property of the GRO TDC as discussed in Section IV.

Fig. 11 shows the GRO test setup, which consists of a square-wave reference generator whose output is sent through a variable delay block in order to create two edge signals with varying time offset between them. These edge signals are sent into the GRO IC, which is designed to have a 10-bit range and a raw timing resolution corresponding to the delay of the gated inverter stages, of which 15 are used. Using this test setup in conjunction with Matlab postprocessing scripts, the raw resolution of the GRO was measured to be approximately 45 ps at a power supply of 1.2 V, which yields an overall time measurement range of 200 ps to 45 ns. The maximum reference frequency supported by the GRO prototype is roughly 250 MHz. Total current consumption of the GRO from a 1.2 V supply is a linear function

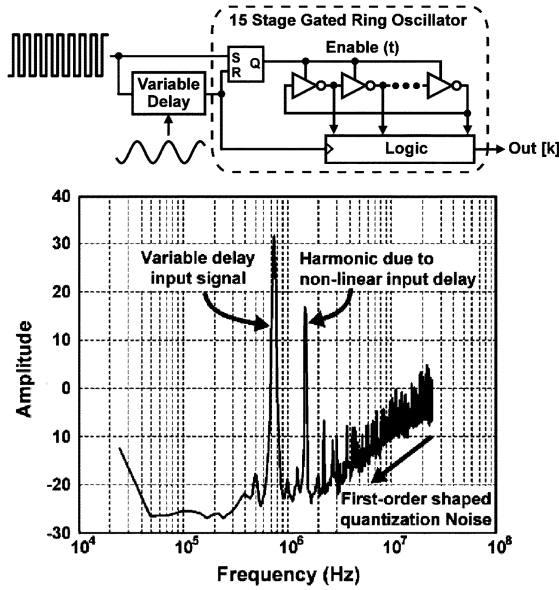


Fig. 11. Test setup and measured GRO output spectrum illustrating first-order noise shaping.

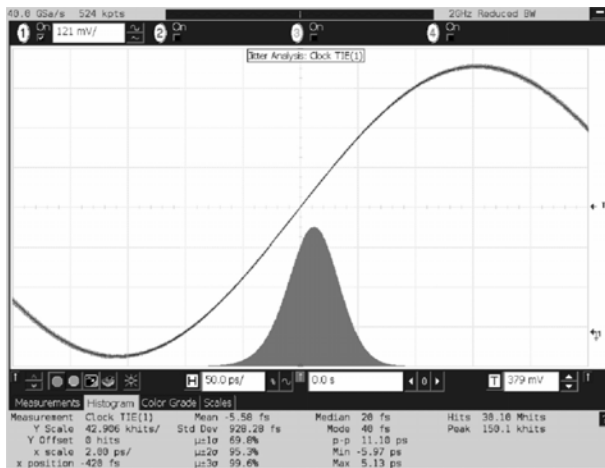


Fig. 12. Measured overall jitter (1.6 GHz MDLL output with a 50 MHz reference).

of the duty cycle of the input and ranges from 1.7 mA with 2% activity to 4.4 mA at 80% (i.e., 2.0 to 5.3 mW).

Fig. 11 shows a measured output spectrum of the GRO output when the variable delay is modulated with a 780 kHz sine wave. As seen in the figure, the first-order noise shaping of the GRO quantization noise is clearly visible. The large second harmonic is due to the nonlinear voltage-to-delay characteristic of the variable delay, which is implemented as a digital buffer gate with varying supply voltage. Note that these harmonics due to the nonlinear voltage-to-delay characteristic are inconsequential to the operation of the GRO *within the MDLL* since the variable delay element is only used for testing the GRO in isolation.

We now focus on the overall MDLL structure, for which a simplified diagram of its prototype was previously shown in Fig. 6. Unless otherwise noted, all measurement results presented here were performed with a reference frequency of

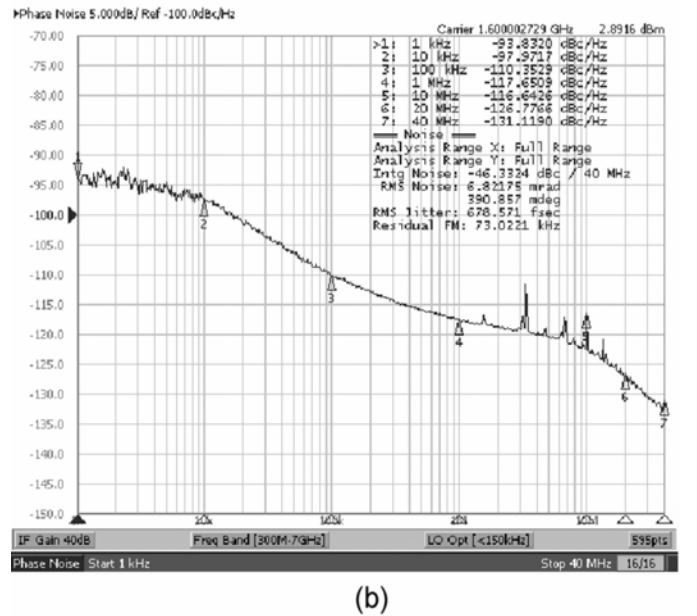
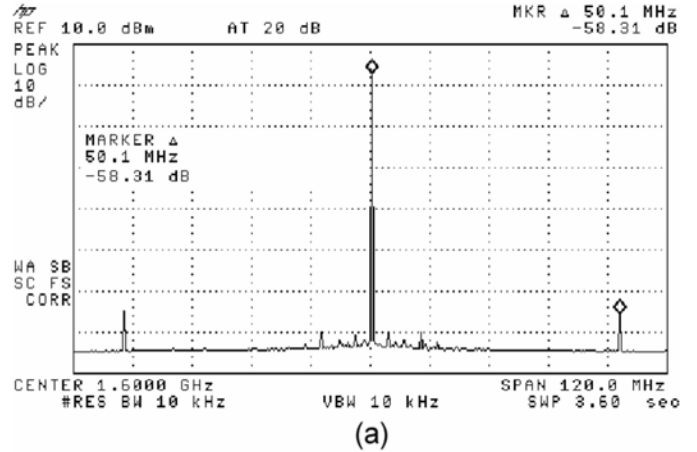


Fig. 13. Measured (a) reference spur and (b) phase noise.

50 MHz; detailed results with other reference frequencies are described in [6]. At a 1.2 V supply, the power consumption of the MDLL core and the GRO TDC chips (excluding output buffers) is 3.9 and 1.2 mW, respectively. Since the DAC is an off-chip component in this prototype, an estimate of its power and area when integrated on-chip is found by examining recent published work on such components. An 8-bit DAC in a similar 0.13 μm process is shown in [14] to consume 3.1 mW with a 100 MHz clock and occupies less than 0.7 mm^2 of area. As for digital functions performed by the FPGA, simulations indicate that they would consume less than 1 mW and occupy less than 0.01 mm^2 .

To demonstrate the subpicosecond jitter performance of the MDLL prototype, the overall jitter was measured using an Agilent DSO81204B high performance oscilloscope. Fig. 12 shows a measured overall jitter of 928 fs (rms) and 11.1 ps (peak–peak) based on 30.1 million samples and a reference frequency of 50 MHz. Measured overall jitter for the case of reference frequencies equal to 25 and 12.5 MHz were 1.23 and 1.92 ps (rms), respectively.

TABLE I
MEASURED PERFORMANCE COMPARISON

	[1]	[4]	[5]	This work
Output Frequency (GHz)	2.0	1.216	0.176	1.6
Reference Frequency (MHz)	250	64	8	50
Reference Spur (dBc)	-37	-46.5	-70 (estimated)	-58.3
Deterministic Jitter (ps pp) estimated from meas. Spurs (Figure-of-merit)	7.06 (reported DJ: 12)	3.89	1.80	0.76
Random Jitter (ps rms) from integrated phase noise	N/A	N/A	5 (1.8 simulated) (1 kHz to 10 MHz)	0.68 (1 kHz to 40 MHz)
Overall Jitter	1.62 ps (rms) 13.11 ps (p-p) 25 khits	((@2.16 GHz) 1.6 ps (rms) 12.9 ps (p-p) 12.2 khits	N/A	0.93 ps (rms) 11.1 ps (p-p) 30.1 Mhits
Technology (CMOS)	0.18 μm	0.18 μm	0.18 μm	0.13 μm

The measured overall jitter includes both random and deterministic jitter components. Since the focus of this paper is primarily on achieving low deterministic jitter, it is worthwhile to seek a means of measuring it apart from the random component. To do so, it is helpful to look at a frequency domain view of the jitter rather than the time-domain view shown by the oscilloscope. In particular, since deterministic jitter occurs periodically at the reference frequency rate, it will show up in the frequency domain as a spurious noise signal with a fundamental frequency offset that corresponds to the reference frequency.

The following equation provides an expression [6], based on Fourier analysis, which can be used to estimate deterministic jitter, Δ , from reference spurs in the measured spectrum:

$$\Delta \approx T_{\text{out}} \times 10^{\text{Spur(dBc)}/20}. \quad (2)$$

In the above expression, T_{out} is the ideal output period, while *Spur* is the level of the reference spur, measured in units of dBc, that corresponds to the difference between the peak of the carrier frequency (at 1.6 GHz) and the reference spur (at 50 MHz offset).

As shown in Fig. 13(a), measurement of the MDLL output with a HP8595E spectrum analyzer reveals a reference spur of -58.3 dBc. Using (2), the corresponding deterministic jitter is estimated to be 0.76 ps (peak-to-peak). This result validates the proposed technique's ability to achieve sub-picosecond deterministic jitter.

As an additional measure of the performance, the phase noise of the MDLL output was measured using an Agilent E5052A signal source analyzer, as shown in Fig. 13(b). The random jitter was estimated by integrating the measured phase noise from 1 kHz to 40 MHz, and was found to be 679 fs (rms).

Table I compares the performance of the proposed MDLL architecture to previous works. The comparison is limited to edge-multiplexing MDLL architectures, which some sources refer to as recirculating DLLs. The key figure of merit we propose for this type of architecture is the deterministic jitter as estimated from the measured reference spurs, using (2).

The comparison clearly shows that the proposed architecture achieves the lowest jitter—both random and deterministic—compared with previous works. In addition, the proposed architecture is unique in its highly digital tuning approach as compared to the primarily analog approaches used in previous works.

VIII. CONCLUSION

This paper presented a low-jitter highly digital MDLL architecture that leverages time-to-digital conversion and a correlated double-sampling technique to achieve subpicosecond jitter with a 1.6 GHz output frequency and 50 MHz reference. The key benefit of the correlated double-sampling method is that it avoids comparison of two different signals and the associated issues of mismatch and offset that accompany such a comparison. Instead, relative comparison is done on samples obtained by measuring relevant periods of the MDLL output through the use of a TDC. In order to achieve subpicosecond effective resolution for the TDC, a self-scrambling GRO structure was proposed which allows the desired resolution to be achieved through averaging. The proposed GRO TDC was also shown to provide first-order noise shaping of its quantization noise, which is expected to be of high value in other PLL/DLL applications. In summary, the low analog complexity and excellent jitter performance of the presented MDLL structure make it an attractive consideration for future clock multiplication applications.

REFERENCES

- [1] R. Farjad-Rad, W. Dally, N. Hiok-Tiaq, R. Senthinathan, M.-J. E. Lee, R. Rathi, and J. Poulton, "A low-power multiplying DLL for low-jitter multi-gigahertz clock generation in highly integrated digital chips," *IEEE J. Solid-State Circuits*, vol. 37, no. 12, pp. 1804–1812, Dec. 2002.
- [2] S. Ye, L. Jansson, and I. Galton, "A multiple-crystal interface PLL with VCO realignment to reduce phase noise," *IEEE J. Solid-State Circuits*, vol. 37, no. 12, pp. 1795–1803, Dec. 2002.
- [3] A. Waizman, "A delay line loop for frequency synthesis of de-skewed clock," in *IEEE ISSCC Dig. Tech. Papers*, Feb. 1994, pp. 298–299.
- [4] Q. Du, J. Zhuang, and T. Kwasniewski, "A low-phase noise, anti-harmonic programmable DLL frequency multiplier with period error compensation for spur reduction," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 53, no. 11, pp. 1205–1209, Nov. 2006.
- [5] P. Maulik and D. Mercer, "A DLL-based programmable clock multiplier in 0.18- μm CMOS with -70 dBc reference spur," *IEEE J. Solid-State Circuits*, vol. 42, no. 8, pp. 1642–1648, Aug. 2007.
- [6] B. M. Helal, "Techniques for low jitter clock multiplication," Ph.D. dissertation, MIT, Cambridge, MA, Feb. 2008.
- [7] K. H. White, D. R. Lampe, F. C. Blaha, and I. A. Mack, "Characterization of surface channel CCD image arrays at low light levels," *IEEE J. Solid-State Circuits*, vol. SSC-9, no. 1, pp. 1–14, Feb. 1974.
- [8] C. C. Enz and G. C. Temes, "Circuit techniques for reducing the effects of op-amp imperfections: Autozeroing, correlated double sampling, and chopper stabilization," *Proc. IEEE*, vol. 84, no. 11, pp. 1584–1614, Nov. 1996.

- [9] R. B. Staszewski, J. L. Wallberg, S. Rezek, C.-M. Hung, O. E. Eliezer, S. K. Vemulapalli, C. Fernando, K. Maggio, R. Staszewski, N. Barton, M.-C. Lee, P. Cruise, M. Entezari, K. Muhammad, and D. Leipold, "All-digital PLL and transmitter for mobile phones," *IEEE J. Solid-State Circuits*, vol. 40, no. 12, pp. 2469–2482, Dec. 2005.
- [10] I. Nissinen, A. Mantyniemi, and J. Kostamovaara, "A CMOS time-to-digital converter based on a ring oscillator for a laser radar," in *Proc. ESSCIRC*, Sep. 2003, pp. 469–472.
- [11] B. M. Helal, M. Z. Straayer, G.-Y. Wei, and M. H. Perrott, "A low jitter 1.6 GHz multiplying DLL utilizing a scrambling time-to-digital converter and digital correlation," in *Symp. VLSI Circuits Dig. Tech. Papers*, Jun. 2007, pp. 166–167.
- [12] L. Dai and R. Harjani, "Comparison and analysis of phase noise in ring oscillators," in *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS)*, May 2000, vol. 5, pp. 77–80.
- [13] A. Hajimiri, S. Limotyrakis, and T. H. Lee, "Jitter and phase noise in ring oscillators," *IEEE J. Solid-State Circuits*, vol. 34, no. 6, pp. 790–804, Jun. 1999.
- [14] N. Ghittori, A. Vigna, P. Malcovati, S. D'Amico, and A. Baschiroto, "1.2-V low-power multi-mode DAC+filter blocks for reconfigurable (WLAN/UMTS, WLAN/Bluetooth) transmitters," *IEEE J. Solid-State Circuits*, vol. 41, no. 9, pp. 1970–1982, Sep. 2006.



Belal M. Helal (S'92–M'06) received the B.S. degree (with the highest honors) in electrical engineering from King Abdulaziz University, Jeddah, Saudi Arabia, in 1995, and the M.S. and Ph.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, in 2002 and 2008, respectively.

His research interests include digital clock multiplication and frequency-synthesis techniques, wireless communications circuits and systems, and digital signal processing.



Matthew Z. Straayer (S'05) received the B.S. and M.S. degrees in electrical engineering from the University of Michigan, Ann Arbor, in 2000 and 2001, respectively. He is currently working toward the Ph.D. degree at the Massachusetts Institute of Technology, Cambridge. His dissertation focuses on the use of ring oscillators as building blocks for high-performance converters.

From 2001 to 2003, he was with Integrated Sensing Systems, Ypsilanti, MI, designing wireless readout ASICs for MEMS sensors. From 2003 to the present, he has been a member of staff at Lincoln Laboratory, Lexington, MA.



Gu-Yeon Wei received the B.S., M.S., and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, in 1994, 1997, and 2001, respectively.

He is currently an Associate Professor of Electrical Engineering with the School of Engineering and Applied Sciences, Harvard University, Cambridge, MA. After a brief stint as a Senior Design Engineer at Accelerant Networks, Inc., Beaverton, OR, he joined the faculty at Harvard as an Assistant Professor in January 2002. His research interests span several areas:

high-speed low-power link design, mixed-signal circuits for communications, ultralow-power hardware for wireless sensor networks, and co-design of circuits and computer architecture for high-performance and embedded processors to address PVT variability and power consumption that plague nanoscale CMOS technologies.



Michael H. Perrott received the B.S. degree in electrical engineering from New Mexico State University, Las Cruces, in 1988, and the M.S. and Ph.D. degrees in electrical engineering and Computer Science from the Massachusetts Institute of Technology (MIT), Cambridge, in 1992 and 1997, respectively.

From 1997 to 1998, he was with Hewlett-Packard Laboratories, Palo Alto, CA, where he was involved with high-speed circuit techniques for sigma-delta synthesizers. In 1999, he was a visiting Assistant Professor with the Hong Kong University of Science and Technology and taught a course on the theory and implementation of frequency synthesizers. From 1999 to 2001, he was with Silicon Laboratories, Austin, TX, where he developed circuit and signal processing techniques to achieve high-performance clock and data recovery circuits. He is currently an Associate Professor in electrical engineering and computer science with MIT, and his research focuses on high-speed circuit and signal processing techniques for data links and wireless applications.