Mitigation of Sampling Errors in VCO-Based ADCs

Vishnu Unnikrishnan and Mark Vesterbacka

Journal Article

N.B.: When citing this work, cite the original article.

©2016 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

http://dx.doi.org/10.1109/TCSI.2017.2670058
Postprint available at: Linköping University Electronic Press
http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-137327
Mitigation of Sampling Errors in VCO-Based ADCs

Vishnu Unnikrishnan, Student Member, IEEE, and Mark Vesterbacka, Senior Member, IEEE

Abstract—VCO-based ADC is a scaling-friendly architecture to build ADCs in fine-feature CMOS processes. Lending itself to an implementation with digital components, such a converter enables design automation with existing digital CAD hence reducing design and porting costs compared to a custom design flow. However, robust architectures and circuit techniques that reduce the dependence of performance on component accuracy are required to achieve good performance while designing converters with low accuracy components like standard cells in deeply-scaled processes. This work investigates errors resulting from sampling of a fast switching multi-phase ring oscillator output. A scheme employing ones-counters is proposed to encode the sampled ring oscillator code into a binary representation, which is resilient to a class of sampling induced errors modeled by temporal reordering of the transitions in the ring. In addition to correcting errors caused by deterministic reordering, proposed encoding suppresses conversion errors in the presence of arbitrary reordering patterns that may result from automatic place-and-route in wire-delay dominated processes. The error suppression capability of the encoding is demonstrated using MATLAB simulation. The proposed encoder reduces the error caused by random reordering of six subsequent bits in the sampled signal from 31 to 2 LSBs for a 31-stage oscillator.

Index Terms—Analog-to-digital, VCO-based, ADC, sampling error, time-domain, synthesis, deep submicrometer technology, error correction, encoding, ring oscillator, ones-counter, non-ideal sampling, synthesis, place-and-route, sigma-delta.

I. INTRODUCTION

MIXED-SIGNAL functions can be implemented with standard digital components by employing a time-domain signal representation [1]–[9], which is an interesting approach to build high performance data converters in ultra deep submicrometer (UDSM) CMOS processes. While conventional analog data converter architectures are difficult to design in modern processes, implementation with digital components benefits from the excellent time resolution provided by deep submicrometer technologies. Also, it is expected that the performance, energy efficiency, and area efficiency of digital architectures improve as the feature size is scaled down. Besides being scaling-friendly, such architectures enable automated design and synthesis of mixed-signal circuits using digital CAD tools, leading to fast and low cost development of technology portable system-on-chip solutions in fine process nodes. However, robust architectures and circuit techniques are required to reduce the dependency of performance on component accuracy, while designing data converters with digital components like standard cells in deeply-scaled processes.

A voltage controlled oscillator (VCO) based analog-to-digital converter (ADC) [10]–[14], which is a specific case of open loop sigma-delta modulators [15]–[17], is attractive for building ADCs with digital components [18]–[20] due to favorable properties like first-order noise-shaping and inherent anti-alias filtering [14]. A basic VCO-based converter built around a voltage controlled ring oscillator and a corresponding model are shown in Fig. 1. The oscillator works as a signal integrator since its phase is the integral of its frequency. The multi-phase square-wave output of the oscillator provides an inherent quantization of the phase. The quantized phase of the oscillator is sampled with the converter clock and is encoded to a number. A time-discrete first-order differentiation of the resulting sample sequence generates the converter output. Note that proper operation requires the subtraction to be performed in a cyclic manner within the code range of the encoder. The signal integration followed by quantization and subsequent differentiation result in first-order shaping of the quantization error. Assuming a linear voltage-frequency conversion by the VCO, the converter is functionally equivalent to a continuous-time sampling first-order sigma-delta ADC [10]. It is important to utilize all the transitions at the multi-phase output of the oscillator in order to maximize the resolution of the converter.

In contrast to the pre-quantization sampling of voltage signals in conventional ADCs, a VCO-based converter performs a post-quantization sampling of a multi-bit time-domain signal. The sampling is performed using a register. Further, the transitions in the phase signal are asynchronous with the sampling clock resulting in metastability. Hence, sampling of the phase signal requires careful consideration to avoid large errors at the converter output due to a variety of sampling induced errors.

This work investigates sampling induced errors in VCO-based ADCs. Sampling errors are modeled as temporal reordering of the transitions in the ring. A reference design is shown in Fig. 1. The oscillator works as a signal integrator since its phase is the integral of its frequency. The multi-phase square-wave output of the oscillator provides an inherent quantization of the phase. The quantized phase of the oscillator is sampled with the converter clock and is encoded to a number. A time-discrete first-order differentiation of the resulting sample sequence generates the converter output. Note that proper operation requires the subtraction to be performed in a cyclic manner within the code range of the encoder. The signal integration followed by quantization and subsequent differentiation result in first-order shaping of the quantization error. Assuming a linear voltage-frequency conversion by the VCO, the converter is functionally equivalent to a continuous-time sampling first-order sigma-delta ADC [10]. It is important to utilize all the transitions at the multi-phase output of the oscillator in order to maximize the resolution of the converter.

In contrast to the pre-quantization sampling of voltage signals in conventional ADCs, a VCO-based converter performs a post-quantization sampling of a multi-bit time-domain signal. The sampling is performed using a register. Further, the transitions in the phase signal are asynchronous with the sampling clock resulting in metastability. Hence, sampling of the phase signal requires careful consideration to avoid large errors at the converter output due to a variety of sampling induced errors.

This work investigates sampling induced errors in VCO-based ADCs. Sampling errors are modeled as temporal reordering of the transitions in the ring. A reference design is synthesized and simulated in a UDSM process to demonstrate the necessity of error suppression while designing VCO-based converters in deep submicron technologies. An error suppres-
sion encoding scheme is proposed that encodes the sampled ring oscillator code into a binary mapping, which is resilient to conversion errors resulting from reordering of transitions. In addition to correcting error patterns due to deterministic reordering, the proposed scheme suppresses many general conversion errors that may result from automatic place-and-route in wire-delay dominated UDSM processes. The effectiveness of the scheme is demonstrated using simulation in MATLAB.

The remaining part of this paper is organized as follows. Section II reviews sampling errors and their mitigation in VCO-based converters employing counters for phase accumulation. Section III investigates mechanisms that cause sampling errors when a fast switching multi-phase ring oscillator output is sampled by a register. Section IV presents the details of the proposed encoding scheme that suppresses large conversion errors resulting from many arbitrary reordering patterns, in addition to correcting an identified deterministic pattern. Section V concludes the discussion.

II. SAMPLING ERRORS IN COUNTER-BASED PHASE ACCUMULATION

A simple VCO-based ADC with the architecture shown in Fig. 1 performs a sub-cycle accumulation of the oscillator phase during a sampling period. This requires a sampling rate higher than the highest oscillator frequency in order to retain complete information about the phase progression of the VCO. The constraint on the sampling rate can be relaxed by the use of counters to track integer cycles of oscillator phase progression. Even though the focus of the paper is direct sampling of the oscillator phase, for the sake of completeness, this section reviews some of the error mechanisms and the respective solutions proposed in the literature for the case where counters are employed for phase accumulation.

A. Counter array phase accumulation

One solution to accumulate the phase progression of the oscillator utilizing all the transitions at its multi-phase output is to use an array of counters as shown in Fig. 2. Rising and falling edge sensitive counters are used at each phase tap to achieve maximal phase resolution. A dominant error source while sampling the output of a multi-bit counter is partial sampling of the output when the sampling instance coincides with the counter update, which can cause large errors at the output. This error can be mitigated by using Gray encoded counters as shown in Fig. 2, thereby limiting the conversion error resulting from partial sampling and metastability to one LSB [19].

B. Coarse-fine phase accumulation

A more energy-efficient solution to accumulate phase progression of the oscillator is to use a coarse-fine quantizer [13] as shown in Fig. 3. The circuit tracks the integer phase progression using a counter and the fractional progression by sampling the state of the ring oscillator. A major source of error is the asynchrony in the sampling of the integer and the fractional phase measurements, which can cause large errors at the converter output. The solution depicted in Fig. 3 mitigates this problem by sampling the counter output more safely using an appropriate tap of the oscillator [13]. A suitable phase tap can be chosen based on the assumption that the counter settles within a certain interval in relation to the maximum VCO frequency. The converter clock then resamples the counter output as well as its sampled copy. One among these resampled values, at the most, can be corrupted due to the asynchrony between the sampling clock and the VCO oscillation. The correct value can be chosen utilizing the redundancy between the integer and fractional measurements due to the shared phase tap, such that incoherence between the sampled integer and fractional phase progression is avoided.

III. DIRECT SAMPLING OF THE RING OSCILLATOR OUTPUT

The circuit discussed in section II-B assumes that a register samples the multi-phase output of a ring oscillator with rapidly switching nodes without errors. This is however not always possible, especially in modern processes, as elaborated below. The focus of this section is to discuss direct sampling of the fast switching multi-phase output of a ring oscillator using a register. We will thus focus on the architecture in Fig. 1 henceforth, where no counters are used. As an example, we consider the seven-stage ring oscillator with a register sampling its output shown in Fig. 4. The choice of seven delay stages is made to allow sufficient length in the ring to illustrate the relevant sampling error patterns. The oscillator output nodes as well as the respective codes are denoted by \( R = \{ r_6, r_5, r_4, r_3, r_2, r_1, r_0 \} \) and the sampled output is denoted as \( S = \{ s_6, s_5, s_4, s_3, s_2, s_1, s_0 \} \). The valid states that can be observed at the output of the ring oscillator and a decimal mapping are provided in Table I.
A. Temporal reordering of transitions

The transitions at the multi-phase output of a ring oscillator are temporally ordered according to the relationship between the different phase taps of the ring oscillator. However, the temporal order of transitions seen by the register can be different from the correct order due to various non-ideal circuit conditions, as illustrated with the example in Fig. 5. Such a reordering of transitions causes non-ideal sampling of the multi-phase output of the oscillator, leading to invalid patterns (other than the sequence listed in Table I) at the input of the encoder. An encoder designed to encode only valid patterns into a predefined decimal mapping generates undefined output for invalid input patterns, possibly causing large errors at the converter output. Some of the mechanisms that cause reordering of transitions are described in the following subsections.

B. Deterministic reordering due to switching characteristics

The switching characteristics in deeply-scaled technologies cause reordering of transitions. Fig. 6 shows the output waveforms of a supply-controlled ring oscillator in a 28 nm FinFET process for supply voltages of 1.1 V and 0.6 V. The input and the output voltages ($v_D$ and $v_Q$) of a standard cell flip-flop from the same process, working from the nominal supply of 1 V, is plotted to the right showing the switching threshold voltage of the flip-flop. The correct temporal order of the transitions in the ring holds only in a limited voltage range, shown as shaded in Fig. 6. Above and below this range, the switching waveforms may exhibit an incorrect temporal order of transitions. Hence, misalignment of the switching threshold of the flip-flop outside the shaded voltage range leads to reordering of transitions as seen by the register. The pattern of reordering, as can be observed from Fig. 6, is deterministic making it possible to correct it.

Note that a precursor to this problem, the non-uniform quantization steps resulting from the same phenomenon, is discussed and solved in [18] using a VCO with differential output. We focus on the case where the transitions are reordered in time. While the solution in [18] reduces the likelihood of reordering due to switching characteristics, it may not be sufficient to avoid reordering due to mismatches and place-and-route described later in Section III-C.

A first-order model of the pattern of reordering is discussed below. The transitions in the multi-phase output of a ring oscillator are assumed to be linear as illustrated in Fig. 7.
Rising and falling transition times are assumed to be equal (= \(t_{\text{trans}}\)). Low-to-high and high-to-low propagation delays are assumed to be equal (= \(t_p\)). The ideal switching threshold is denoted as \(V_{SW}\) (\(V_{SW} = V_{DD}/2\), where \(V_{DD}\) is the supply voltage). \(T_i\) denotes the consecutive transitions in the ring where \(i\) indicates the order. When the switching threshold of the flip-flop, \(V_{SW}\), is within \(V_{SW} \pm \Delta V_0\), the register sees the correct temporal order of transitions in accordance with the indexing. \(\Delta V_0\), \(\Delta V_1\), and \(\Delta V_2\) are the voltage thresholds above and below \(V_{SW}\) at which the transitions cross each other as shown in Fig. 7. When \(V_{SW}'\) deviates from \(V_{SW}\) by a voltage more than \(\Delta V_0\), the transitions are reordered in time according to the following expressions.

\[
\begin{align*}
V_{SW} - \Delta V_0 &< V_{SW}' < V_{SW} + \Delta V_0 : \\
& \quad \cdots T_{-4} T_{-3} T_{-2} T_{-1} T_0 T_1 T_2 T_3 T_4 \cdots \\
V_{SW} + \Delta V_0 &< V_{SW}' < V_{SW} + \Delta V_1 : \\
& \quad \cdots T_{-4} T_{-3} T_{-2} T_{-1} T_0 T_3 T_2 T_5 T_4 \cdots \\
V_{SW} - \Delta V_1 &< V_{SW}' < V_{SW} - \Delta V_0 : \\
& \quad \cdots T_{-4} T_{-5} T_{-2} T_{-3} T_0 T_{-1} T_2 T_1 T_4 \cdots \\
V_{SW} + \Delta V_1 &< V_{SW}' < V_{SW} + \Delta V_2 : \\
& \quad \cdots T_{-4} T_{-1} T_{-2} T_{-3} T_0 T_5 T_2 T_7 T_4 \cdots \\
V_{SW} - \Delta V_2 &< V_{SW}' < V_{SW} - \Delta V_1 : \\
& \quad \cdots T_{-4} T_{-7} T_{-2} T_{-5} T_0 T_{-3} T_2 T_{-1} T_4 \cdots \\
& \quad \text{(1)}
\end{align*}
\]

For the cases \(V_{SW} + \Delta V_0 < V_{SW}' < V_{SW} + \Delta V_1\) and \(V_{SW} - \Delta V_1 < V_{SW}' < V_{SW} - \Delta V_0\), one transition preceding a reference transition can at the most be reordered to appear as

succeeding the reference transition (or vice versa). We refer to this case as first-order reordering. Similarly for \(V_{SW} + \Delta V_1 < V_{SW}' < V_{SW} + \Delta V_2\) and \(V_{SW} - \Delta V_2 < V_{SW}' < V_{SW} - \Delta V_1\), two preceding transitions can at the most be reordered to appear succeeding, denoted as a second-order reordering.

If \(\alpha\) is the angle made by the rising transition with the vertical as shown in Fig. 7, \(\tan \alpha = t_{\text{trans}}/0.8V_{DD}\) where \(t_{\text{trans}}\) is the 10%-90% transition time. \(\alpha\) is a measure of the transition time in the technology. \(\alpha\) equals zero for an ideal switching which takes no time to finish. The voltage thresholds \(\Delta V_0\), \(\Delta V_1\), and \(\Delta V_2\) can be expressed as

\[
\begin{align*}
\Delta V_0 &= \frac{1}{2} \frac{t_p}{\tan \alpha} = \frac{1}{2} \frac{0.8V_{DD}t_p}{t_{\text{trans}}} \quad (2) \\
\Delta V_1 &= \frac{3}{2} \frac{t_p}{\tan \alpha} = \frac{3}{2} \frac{0.8V_{DD}t_p}{t_{\text{trans}}} \quad (3) \\
\Delta V_2 &= \frac{5}{2} \frac{t_p}{\tan \alpha} = \frac{5}{2} \frac{0.8V_{DD}t_p}{t_{\text{trans}}} \quad (4)
\end{align*}
\]

and for a general case,

\[
\Delta V_i = \frac{1 + 2i}{2} \frac{t_p}{\tan \alpha} = \frac{1 + 2i}{2} \frac{0.8V_{DD}t_p}{t_{\text{trans}}} \quad (5)
\]

Equations (2)-(5) suggest that a reduction in propagation delay \(t_p\) in relation to the transition time \(t_{\text{trans}}\) leads to smaller values of \(\Delta V_i\), thereby increasing the probability of reordering. This is likely when using fast transistor flavors to maximize time resolution, especially in deeply-scaled processes.

One case where the mechanism described above is likely to cause a conversion error is when using a supply controlled ring oscillator as the VCO, where the \(V_{SW}\) and \(\Delta V_i\) of the ring oscillator vary in relation to the register threshold \(V_{SW}'\) leading to reordering errors. Also when using a VCO with fixed rail-to-rail output, momentary supply and ground glitches at the VCO or the register may lead to an error due to the same mechanism. This is more likely to happen with the narrow \(\Delta V_i\) that results when supply voltage scaling is employed to reduce power. In order to verify that the mechanism causes large conversion errors, a VCO-based converter with the architecture shown in Fig. 8 is designed and synthesized using a vendor supplied standard cell library in a 28 nm FD-SOI process. A pseudo-differential topology is used where each sub-ADC has the same architecture as that of the converter in Fig. 1. The VCO is a 31-stage supply controlled ring oscillator. The encoder is a combinational logic that maps the ring oscillator code to the decimal mapping given in Table I. An invalid input code (any pattern not listed in Table I) results in an undefined output code. A linearization block is used to correct for the nonlinearity of the VCO. It is realized as an inverse polynomial derived from the polynomial approximation of the ramp response of the converter.

A schematic simulation of the reference design using the Spectre simulator (no layout parasitics included) reveals large errors at the converter output. The data from the sampling register recorded for long runs of simulation shows that all the errors belong to the first-order and second-order reordering patterns predicted from the model described above. The error patterns in the sampled data are illustrated with the example of a seven-stage oscillator in Fig. 9. The transiting phase tap

\[
\begin{align*}
\text{voltage} & \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \ quad
\end{align*}
\]
between each pair of adjacent codes is marked. The transitions that are sampled and missed by the register as shown toward the right side of Fig. 9 result in the error code shown in the examples. Both first-order and second-order error patterns are shown. The pattern creates a range spanning a few codes shown shaded in Fig. 9, any of which can be the correct code. We correct the error to the code in the middle of the range, as shown.

The response of the converter to a ramp and a sinusoid input is shown in Fig. 10 where large errors resulting from sampling errors are visible. The converter is redesigned with an encoder that corrects the first-order and second-order reordering patterns identified above. The response from the redesigned converter is also shown Fig. 10 where all the error spikes are seen suppressed. The spectra of the converter output for a sinoid input with and without reordering errors are shown in Fig. 11. It can be seen that the error spikes cause high noise power in low frequencies thereby offsetting the benefits of the converter’s quantization error shaping. By correcting the first-order and second-order reordering patterns, the expected response is obtained as shown in Fig. 11.

The demonstration above shows that sampling errors that can be modeled by temporal reordering of transitions occur in VCO-based converters implemented in advanced technology nodes. Hence, a robust error suppression encoding is necessary to achieve good performance.

C. Reordering due to mismatches and place-and-route

In addition to the error mechanism discussed in the previous subsection, sampling errors that can be modeled by reordering of transitions can also result from delay mismatch among multiple phase taps of the VCO. Fig. 12 shows the mismatches associated with sampling of the multi-phase output of a ring oscillator with a register. They include the imbalance in low-to-high and high-to-low propagation delays ($t_{PLH}$ and $t_{PHL}$) of the inverters, the drive mismatch among the inverters, the wire-load mismatch among the phase taps, and the clock skew among the flip-flops of the register. The overall effect is a net delay mismatch among the phase taps. When the magnitude of the delay mismatch approaches the inverter delay in the technology, the register may see transitions in a different temporal order than how they occur in the ring.

Further, the automatic place-and-route with CAD tools adds to the mismatches, resulting in complex reordering patterns that are difficult to predict. The effect is expected to worsen as the technology scales down leading to wire-delays dominating the gate delays.
IV. PROPOSED ERROR SUPPRESSION ENCODING

It follows from the discussion in the previous section that a robust error suppression encoding technique is necessary to achieve good performance when designing VCO-based ADCs in advanced processes. In addition to correcting for the deterministic error patterns described in section III-B, it is desired to suppress large conversion errors resulting from reordering due to factors mentioned in section III-C. Below we propose an encoding scheme that utilizes the redundancy inherent to the ring oscillator code in order to achieve robust error suppression even in the presence of arbitrary reordering patterns.

A. Translation to Johnson code

If every other bit of the ring oscillator code is inverted, we obtain a Johnson code as shown in Table II. This translation presents properties that are useful in error correction as described in further subsections. Let \( J = [j_6 \ j_5 \ j_4 \ j_3 \ j_2 \ j_1 \ j_0] = [r_6 \ \overline{r_5} \ r_4 \ \overline{r_3} \ r_2 \ \overline{r_1} \ r_0] \) denote the Johnson code for the example considered.

B. Reordering errors appear as bubble errors

The valid ring oscillator codes from the earlier example as well as the first and second-order reordering patterns identified in section III-B are listed in Table III. The patterns obtained when these are translated to Johnson code are also listed. Let \( J' = [s_6 \ s_5 \ s_4 \ s_3 \ s_2 \ s_1 \ s_0] \) denote the code obtained when the sampled ring oscillator code \( S \) is translated to Johnson code. A decimal mapping for the error codes is also given. It can be seen that the sampling errors appear as bubble errors in the Johnson code. The errors listed in Table III covers only two bubble patterns that correspond to the first and second-order reordering shown in Fig. 9. However, more complex error patterns that may occur due to factors discussed in section III-C would also appear as bubble errors after translation to Johnson code. The resulting bubble patterns as well as the number of phase taps over which the bubbles span may differ from what is covered in Table III.
Another property of the bubble errors in the Johnson code is that they wrap around the boundary between the first and the last phase tap in a cyclic fashion, twice within the code span. It may also be noted from Table III that the bubbles that wrap across are inverted.

C. Ones-counter

An elegant approach to bubble suppression in unary codes is to use a ones-counter, which counts the ones in the unary code [21], [22]. A ones-counter, in effect, groups the ones and zeros in the code such that the bubbles are removed. The merit of the approach lies in that it works well irrespective of the complexity of bubble patterns, exploiting the redundancy in counting ones. Further, all the bits are given equal weights avoiding an emphasis on the correctness of any specific bit thereby improving error suppression properties.

D. Ones-counter encoding of ring oscillator code

It can be observed from Table II that a ones-counter operating on the Johnson code \( J \) produces a non-unique output with multiple ring oscillator codes mapped to the same output code. This is illustrated in Fig. 13 where the output from a ones-counter operating on \( J \) as well as the desired output are plotted against the oscillator phase for the case of a seven-stage ring oscillator. The ones-counter output can be unwrapped to obtain a unique encoding by assigning an increased weight to a specific bit. The value of this pivotal bit is used to separate the ring oscillator codes that are mapped to a common ones-counter output. For example, the bit \( j_0 \) in Table II can be used as the pivotal bit for encoding. The desired decimal output is then obtained as

\[
D = j_{N-1}.(\text{ones}(J) - 1) + j_{N-1}.(2N - \text{ones}(J) - 1) \quad (6)
\]

where \( D \) is the decimal mapping, \( N \) is the number of inverters in the ring oscillator, \( J=[j_{N-1} j_{N-2} \ldots j_1 j_0] \) is the Johnson translated ring oscillator code and ones is a function that returns the number of ones in the input code. The excessive weight given to the pivotal bit does however lead to limitations in error suppression capabilities (section IV-F).

E. Error correction with ones-counter encoding

In addition to encoding the valid ring oscillator codes in Table II, Eqn. 6 achieves encoding of most of the error codes listed in Table III to the desired corrected decimal mapping. The error given by the difference between the encoder output while using Eqn. 6 and the desired decimal mapping is given in Table III. It can be seen that the error is zero except for a few codes.

F. The problem of wrapped bubbles

It can be observed from Table III that the non-zero encoding errors correspond to the case where bubbles wrap across from \( j_0 \) to \( j_0 \), the pivotal phase tap chosen for Johnson encoding. Hence, Eqn. 6 achieves the desired bubble correction except for the case where bubbles wrap across the pivotal phase tap. The large errors are due to the large weight given to a particular phase tap compared to others. If the oscillator output is sampled with reordering errors when the transiting phase tap of the oscillator is near the pivotal phase tap, there is a risk of large errors. Note that the transitions in the ring pass through the pivotal phase tap twice within an oscillation cycle. For the example considered in Fig. 13, this corresponds to the points where the ones-counter output changes direction (near 0 and \( \pi \) rad).

The errors due to wrapped bubbles follows a deterministic pattern as illustrated in Table III, and hence can be corrected. The circuit shown in Fig. 14 corrects all the encoding errors in Table III. The errors are grouped into four groups as given in the last column of Table III for the convenience of error identification. Signal \( S \) represents the sampled output from a ring oscillator with \( N \) phase taps. Two highest index bits of \( S \) goes through a conditional swap block that swaps the incoming bits when enabled, before being fed to the encoder. The encode block implements Eqn. 6. The encoded output passes through two conditional arithmetic blocks which add/subtract a constant to/from the encoder output. The parenthesis of the add/subtract blocks denotes that the arithmetic must be performed within a finite integer ring in a cyclic manner. The blocks marked with a group and a number are logic circuits to detect if an error belonging to the respective group has occurred. The correction circuit works for a general number of taps \( N \).

Solutions like the one described above can be designed for deterministic error patterns. However, a generic solution that works for arbitrary error patterns is desired. A robust and generic solution to handle the wrapped bubbles is proposed in the following subsection.
We propose a generic solution to the problem of wrapped bubbles by using two ones-counters with their pivotal phase taps skewed from each other. The concept is illustrated in Fig. 15 with the case of a 31-stage ring oscillator. A 31-stage oscillator is chosen to illustrate the discussion. The first encoder has a mapping similar to the example in Fig. 13 with its sensitive range near 0 and $\pi$ rad. The second encoder is configured such that its pivotal phase tap for Johnson encoding is skewed away from that of the first encoder. This shifts the sensitive region of the second encoder away from that of the first encoder. This shifts the sensitive region of the second encoder away from that of the first encoder as shown in Fig. 15. In order to maximize the distance between the sensitive points of the two encoders thereby maximizing the error resilience, the pivotal tap of second encoder is offset by ceil($N/2$) taps from that of the first encoder (corresponding to an approximate phase shift of $\pi/2$). ceil returns the next integer value. In relation to the first encoder’s reference, this places the sensitive point of the second encoder near $\pi/2$ and $3\pi/2$ rad as shown in Fig. 15.

Assuming that the spread of reordering is limited to less than $N/2$ phase taps (ensured with a carefully designed $N$), at least one encoder encodes the input correctly without errors due to wrapped bubbles. The correct output can then be chosen by detecting whether the position of the switching phase tap in the ring is near the pivotal tap of the first encoder or the second at the sampling instant. The block diagram of the solution is shown in Fig. 16. The two encoders implement the encoding represented by Eqn. 6 with their respective pivotal taps. Note that both the encoders are connected to all the phase taps of the oscillator and encodes the complete state of the ring. The third block is connected to a subset of the phase taps, detecting the coarse position of the phase tap that was transiting at the sampling instant. The logic block then computes the proper output based on the position information as well as the two encoder outputs.

An example implementation is shown in Fig. 17. The pivotal tap of the second encoder $s_K$ is positioned at an offset ceil($N/2$) from that of the primary encoder $s_N$. In this work, we implement the detection of the coarse position of the transiting tap as follows. $W$ phase taps around the pivotal tap of the first encoder is used for the detection as shown in Fig. 17. If the adjacent bits of the sampled output from the chosen $W$ taps are in opposite logic states, we consider that the tap of the ring that was switching at the sampling instant is outside the range of the chosen $W$ taps around the pivotal tap. Then we use the output from the first encoder as the final output. Otherwise, the output of the second encoder is used after remapping it to be consistent with the mapping of the first encoder. The mapping is illustrated in Fig. 15. The output of the second encoder can be mapped to the code space of the first encoder by subtracting the offset.

The encoding scheme described above encodes all error patterns in Table III without errors. In order to evaluate the robustness of the scheme to arbitrary reordering errors, further simulations are performed as discussed in the following subsection.
H. Robustness against arbitrary reordering patterns

The proposed dual ones-counter scheme is tested for its robustness to sampling errors resulting from random temporal reordering of transitions, which would be the case where place-and-route induced wire-delay mismatches dominate over other sources of sampling errors. The test setup shown in Fig. 18 is modeled in MATLAB, which evaluates a large number of reordering patterns with reasonable simulation time.

The test vector consists of a large number of samples with codes expected at the output of a 31-stage ring oscillator ($N = 31$). A ramp covering all the $2N$ valid codes is repeated resulting in a uniform distribution of input codes. An ideal encoder operating on the test vector produces an expected output vector. Random reordering errors are injected into the test vector over $U$ adjacent taps near the switching tap. The error injection function first identifies the position of transiting bit (corresponding to the switching phase tap) in each sample of the test vector. A range spanning $U$ bits around the transiting bit is chosen for error injection in each sample. Then the values of the bits in the range are reordered in a random manner. The resulting vector is then encoded using the proposed encoder as well as using a reference encoder without error suppression. The reference encoder produces the desired output for valid ring oscillator codes. For invalid input codes, it generates an undefined code from the set of valid encoder output codes.

The average error at the encoder output for the reference encoder as well as the proposed encoder as a function of the width of the error site ($U$) is shown in Fig. 19. It can be seen that the proposed encoding generates far lower average error than the reference encoder for a wide range of $U$. However, averaging errors filters out large instantaneous errors at the converter output. Another measure is the maximum error present in the encoded output vector, which is plotted in Fig. 20 as a function of $U$. It can be seen that the proposed encoding maintains a much lower maximum error magnitude than the uncorrected encoder. While the reference encoder generates conversion errors as large as 31 LSBs (in comparison with the encoder range of $2N = 62$ LSBs) for a random reordering of three or more taps, the largest error caused by the proposed encoding is two LSBs for a random reordering of up to six taps. Note that we use seven phase taps around $s_N$ to detect the coarse position of the switching tap ($W = 7$) in the examples above.

I. Hardware cost

In order to estimate the relative increase in hardware cost due to the proposed encoding scheme, synthesis results of four converter implementations are compared. The first implementation has an encoder with no error correction. The second case corrects first-order reordering patterns (section III-B). The third case corrects first-order as well as second-order reordering patterns. The fourth implementation has an encoder with the proposed encoding scheme that is robust to random reordering errors. The synthesizer estimates of the power consumed while sampling at 2.5 GHz and the area required are summarized in Table IV. When compared to the case without error suppression, the proposed encoding roughly doubles the area required and increases the power consumption by 11%.

<table>
<thead>
<tr>
<th>TABLE IV</th>
<th>ESTIMATE OF HARDWARE COST FROM SYNTHESIZER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost measure</td>
<td>No error correction</td>
</tr>
<tr>
<td>Power (mW)</td>
<td>1.25</td>
</tr>
<tr>
<td>Area ($\mu$m$^2$)</td>
<td>442</td>
</tr>
</tbody>
</table>
V. CONCLUSION

The work investigates conversion errors in VCO-based ADCs due to non-ideal sampling of the quantized phase signal. Errors associated with the sampling of a fast switching multi-phase output of a ring oscillator, which can lead to large conversion errors, are modeled by temporal reordering of transitions in the ring. Spectre simulation of a reference design in a UDSM CMOS process shows that such errors are present in converters designed in deeply-scaled processes. An error suppression encoding scheme is proposed to encode sampled ring oscillator output into a binary representation, which is resilient to reordering errors. In addition to correcting the error pattern caused by a deterministic reordering mechanism, the proposed encoding suppresses large conversion errors in the presence of arbitrary reordering patterns that may result from automatic place-and-rout in wire-delay dominated advanced processes. The effectiveness of the scheme is demonstrated using MATLAB simulations. The proposed encoder reduces the largest error caused by random reordering of six subsequent bits in the sampled signal from 31 LSBs to 2 LSBs for the case of a 31-stage oscillator. The proposed correction is useful to suppress large errors when a fast incrementing cyclic unary weighted signal is sampled with a clock that is not synchronized with the signal. For example, the proposed encoding may also be useful in ADPLLs employing ring oscillator based Vernier TDCs, in addition to its application in VCO-based ADCs.

REFERENCES