

FORMAL VERIFICATION AND IN-SITU TEST
OF ANALOG AND MIXED-SIGNAL CIRCUITS

A Dissertation

by

LEYI YIN

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Peng Li
Committee Members,	Gwan Choi
	Jose Silva-Martinez
	Duncan M. Walker
Head of Department,	Costas N. Georghiades

December 2012

Major Subject: Computer Engineering

Copyright 2012 LEYI YIN

ABSTRACT

As CMOS technologies continuously scale down, designing robust analog and mixed-signal (AMS) circuits becomes increasingly difficult. Consequently, there are pressing needs for AMS design checking techniques, more specifically design verification and design for testability (DfT). The purpose of verification is to ensure that the performance of an AMS design meets its specification under process, voltage and temperature (PVT) variations and different working conditions, while DfT techniques aim at embedding testability into the design, by adding auxiliary circuitries for testing purpose. This dissertation focuses on improving the robustness of AMS designs in highly scaled technologies, by developing novel formal verification and in-situ test techniques.

Compared with conventional AMS verification that relies more on heuristically chosen simulations, formal verification provides a mathematically rigorous way of checking the target design property. A formal verification framework is proposed that incorporates nonlinear SMT solving techniques and simulation exploration to efficiently verify the dynamic properties of AMS designs. A powerful Bayesian inference based technique is applied to dynamically trade off between the costs of simulation and nonlinear SMT. The feasibility and efficacy of the proposed methodology are demonstrated on the verification of lock time specification of a charge-pump PLL.

The powerful and low-cost digital processing capabilities of today's CMOS technologies are enabling many new in-situ test schemes in a mixed-signal environment. First, a novel two-level structure of GRO-PVDL is proposed for on-chip jitter testing of high-speed high-resolution applications with a gated ring oscillator (GRO) at the first level to provide a coarse measurement and a Vernier-style structure at the second level to further measure the residue from the first level with a fine resolution.

With the feature of quantization noise shaping, an effective resolution of 0.8ps can be achieved using a 90nm CMOS technology. Second, the reconfigurability of recent all-digital PLL designs is exploited to provide in-situ output jitter test and diagnosis abilities under multiple parametric variations of key analog building blocks. As an extension, an in-situ test scheme is proposed to provide online testing for all-digital PLL based polar transmitters.

To my wife, Yongfeng, and my parents

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 1117660, and the Semiconductor Research Corporation and Texas Analog Center of Excellence under Contract 2008-HJ-1836. The support of the C2S2 Focus Center, one of six research centers funded under the Focus Center Research Program (FCRP), a Semiconductor Research Corporation subsidiary, is also gratefully acknowledged.

My advisor Dr. Peng Li has been continuously tutoring and guiding me with my research work. The first thing I learned from him is the way of thinking which is both creative and rigorous. Simple as it sounds, it is never easy to do. Thanks to him, I have done a list of research works, and most importantly I have built up my confidence in research gradually in the last four years. As for detailed research projects, I also benefited a lot from his insightful ideas and suggestions. Therefore, I would like to express my gratitude and respect to him, not just as a graduating PhD student to his advisor but also as a growing young man to his tutor in life.

Though students come and go, I always feel our research group is a big, warm family, whose members help each other on work as well as on life. They made my PhD life no longer a lonely stressful march, but an exciting joyful group trip. So I would like to thank all the colleagues in Dr. Li's research group, especially Yue Deng and Yongtae Kim, who have directly cooperated with me on research.

Also, I would like to thank my wife, Yongfeng, who has sacrificed so much and supported me so much. Her optimistic life attitude has influenced me to be courageous when dealing with challenges in work or life.

Last but not least, thanks to my parents who made what I am now. Now when I look back to my life, their love is the deepest love to me ever.

TABLE OF CONTENTS

CHAPTER		Page
I	INTRODUCTION	1
	A. Formal verification of AMS circuits	4
	B. High-resolution on-chip jitter measurement	5
	C. In-situ test of all digital PLLs and polar transmitters	6
II	SMT-BASED FORMAL VERIFICATION OF AMS CIRCUITS	8
	A. Introduction	8
	B. Hybrid systems	12
	C. NL-SMT based verification	14
	1. Formulation of NL-SMT constraints	15
	a. Initial space constraints	15
	b. Hybrid dynamics constraints	16
	c. Other constraints	19
	2. Basic NL-SMT approach	19
	a. State-space discretization	19
	b. Box mergence	21
	c. Basic flow of invoking NL-SMT solver	22
	D. Simulation-assisted NL-SMT	25
	1. Simulation-assisted NL-SMT flow	26
	2. Stop condition	28
	3. Statistical framework	32
	4. Bayesian inference	34
	a. Principle of Bayesian inference	34
	b. Bayesian inference for θ	35
	5. Computation of $\mathbf{E}[N_{new}^{(k)} H]$	37
	E. PLL lock time verification	40
	1. Charge pump PLL	41
	2. SMT constraints for lock time	43
	3. Fast forwarding	45
	F. Experimental results	48
	G. Summary	53
	H. Appendix	54

CHAPTER	Page
III	HIGH-RESOLUTION ON-CHIP JITTER MEASUREMENT 59
	A. Introduction 59
	B. Proposed structure 62
	1. Vernier delay line 63
	2. Gated ring oscillator 64
	3. The proposed GRO-PVDL structure 68
	C. Circuit implementation 71
	1. GRO 71
	2. Counters 76
	3. PVDL 80
	4. DFFs 81
	5. GRO-PVDL 82
	6. DSP unit 86
	a. Coarse code generator 88
	b. VDL decoder 89
	c. Fine code generator 90
	D. Experimental results 93
	1. Delay mismatch analysis 97
	2. Specification comparison 101
	E. Summary 103
IV	IN-SITU TEST OF ALL DIGITAL PLLS 104
	A. Introduction 104
	B. Principle of jitter estimation and diagnosis 106
	1. Noise model 106
	2. Transfer function analysis 108
	C. BIST scheme 113
	1. Reconfigurable loop filters 113
	2. TDC calibrator 115
	3. Hardware overhead 117
	D. Simulation results 118
	1. Setup of simulation environment 118
	2. Monte Carlo analysis 119
	E. Summary 123
V	IN-SITU TEST AND CALIBRATION OF ALL DIGITAL POLAR TRANSMITTERS 124
	A. Introduction 124

CHAPTER	Page
B. ADPLL	127
1. Architecture	127
2. Two-point modulation	127
3. Requirements of WCDMA	129
C. RF BIST for EVM	129
1. z -domain model	130
2. Noise analysis	132
3. BIST principle	134
4. BIST scheme	137
5. The optimization of the branch filter	138
D. DCO gain calibration	139
1. DCO gain mismatch	140
2. DCO gain calibration	141
E. Simulation results	143
1. Simulation platform	143
2. Simulation results for DCO gain calibration	144
3. Simulation results for EVM BIST	145
F. Implementation issues	148
G. Summary	149
VI CONCLUSIONS AND FUTURE DIRECTIONS	150
A. Conclusions	150
B. Future directions	150
REFERENCES	152

LIST OF TABLES

TABLE		Page
I	Valid locations of state variables	43
II	Effective resolutions for different k_{mis}	101
III	Comparison of Specifications.	102
IV	Transfer functions from noise sources to output phase noise and digital signatures	110
V	An example of configuration setup. (LF1 is bypassed for Config. 3) .	115
VI	Hardware overhead	118
VII	Transfer functions from noise sources to output phase noise and digital signatures	132
VIII	The comparison of the noise contributions (low frequency range: 100Hz-100kHz; high frequency range: 100kHz-13MHz).	134
IX	The systematic errors and the EVM sensitivities to the digital signatures.	139
X	EVM degradation due to DCO gain mismatch (PM path only, no random noise).	141
XI	Hardware overhead estimation using 90nm CMOS technology	148

LIST OF FIGURES

FIGURE	Page
1	Power supply voltage of future CMOS technologies predicted by ITRS. 2
2	Product stage vs. checking techniques. 2
3	Organization of the subtopics. 3
4	1-bit $\Delta - \Sigma$ ADC. 13
5	Hybrid automaton of 1-bit $\Delta - \Sigma$ ADC. 13
6	Transient trajectories of 1-bit $\Delta - \Sigma$ ADC. 14
7	Step-by-step evolution of hybrid automaton. 16
8	The split of NL-SMT problem. 20
9	Box mergence. 21
10	Search of reachable boxes. 26
11	Flow of simulation-assisted approach. 27
12	Simulation exploration and NL-SMT check. 27
13	Statistical view of simulation exploration. 33
14	$\mathbf{E}[N_{new}^{(k)} \mathbf{H}]$ vs k ($m_{B^*}=100$, $m=1,000$, $\alpha_0=0.001$): (a) $n=100$ (b) $n=100,000$ 40
15	Block diagram of charge pump based PLL. 41
16	Timing diagram of PFD. 42
17	PFD hybrid automaton. 42
18	Timing diagram of fast forwarding. 46

FIGURE	Page
19	Flow of fast forwarding. 47
20	From flow to constraints. 48
21	Reachable space of ϕ_d computed by the proposed reachability analysis. 49
22	Runtime vs sampling density. 50
23	The numbers of samples of simulations for dynamic and static <i>Stop</i> conditions. 51
24	The numbers of samples of simulations for dynamic <i>Stop</i> condition with different k 52
25	Runtime vs initial space (with simulation assistance and Bayesian inference). 53
26	Block diagram of GRO-PVDL structure. 62
27	VDL: (a) structure, (b) timing diagram. *START(i): START delayed by $i \cdot \tau_1$, STOP(i): STOP delayed by $i \cdot \tau_2$ 63
28	Equivalence of VDL: (a) structure, (b) timing diagram. 64
29	GRO: (a) structure, (b) timing diagram. 66
30	Equivalent timing diagram of GRO. 67
31	Frequency spectra of without/with quantization noise shaping. . . . 68
32	The proposed GRO-PVDL structure. 69
33	The timing diagram of GRO-PVDL. 69
34	GRO phase shift: (a) gated inverter, (b) simplified model, (c) waveform. 72
35	Simulated phase shift <i>vs</i> φ_{disab} of a three-stage inverter-based GRO. 72

FIGURE	Page
36	Gating phase shift of three-stage inverter-based GRO. 73
37	Gated delay cell with CCI: (a) symbol, (b) schematic. 74
38	Simulated phase shift <i>vs</i> φ_{disab} of a three-stage CCI-cell-based GRO. 74
39	Simulated phase shift <i>vs</i> EN/nEN rising/falling time of a three-stage inverter-based GRO. 75
40	Asynchronous counter. 76
41	Phase tracking based counting structure (single-ended version). . . . 77
42	Phase transition of three-stage GRO. 78
43	Over/under-counting due to counter/decoder input mismatch. . . . 79
44	GRO counting structure with latch sharing. 79
45	Glitch illustration. 80
46	Differential delay cell: (a) symbol and gate-level schematic (b) schematic. 81
47	(a) Gate-level schematic of differential DFFs (b) schematic of differential latch. 82
48	Schematic of PVDL and DFFs. 83
49	Uneven stage delays of the GRO, and the effective timing diagram. . 84
50	Timing diagram of the GRO-PVDL with d_v 85
51	Block diagram of the DSP unit. 87
52	Implementation of coarse code generator. 88
53	VDL decoder: (a) implementation (b) bubble suppression. 91
54	Implementation of fine code calibration. 93
55	Layout of the entire GRO-PVDL structure. 94

FIGURE	Page
56	GRO-PVDL measurement for 0.5ps_{pp} sin. input: (a) PSD (b) transient view (after low-pass filtering). 96
57	PSD of the measurement for 0.5ps_{pp} sin. input (transistor noise and power supply noise are NOT included in the simulation). 97
58	The histograms of (a) input jitter (b) measurement result (after low-pass filtering). 98
59	Delay mismatch of the GRO and the PVDL. 99
60	Simulated PSD with different delay mismatches ($k_{\text{mis}}= 3\%, 5\%, 10\%$). 100
61	All-digital PLL block diagram including BIST. 105
62	The phase noise spectrum of a typical oscillator. 108
63	s -domain model of ADPLL including noise sources. 109
64	BIST block diagram. 114
65	The composition of the signature power spectral density. 116
66	TDC resolution calibration. 117
67	BIST estimation VS. directly measurement. 120
68	Estimated jitter compares with measured jitter 121
69	Relative estimation error. Error is averaged in each 1ps interval of the output jitter. 121
70	The proposed BIST scheme VS. the BIST scheme in [1]. 122
71	Average error of the diagnosis of the four main noise sources. 122
72	Diagram of an all-digital polar RF modulator. 125
73	The ADPLL architecture with two-point modulation. 128

FIGURE	Page
74	The frequency deviation of a typical WCDMA modulation in one WCDMA slot ($667\mu s$) with the bandwidth reduction technique. 130
75	The z -domain model of the ADPLL including noise sources. 131
76	The composition of the noise. 133
77	The relationship between the EVM and the phase noise. 135
78	BIST block diagram. 138
79	DCO core with LC tank and biasing network in [12]. 140
80	The block diagram of the calibration scheme. 142
81	The relationship between $\Delta FCW/\Delta NDTW$ and ΔDTW 143
82	The event-driven simulation platform for the ADPLL. 144
83	The constellation graphs for the worst mismatch case. 145
84	The output EVM versus the sample step. 146
85	Estimated EVM vs. simulated EVM. 147
86	Pass/fail test. 147

CHAPTER I

INTRODUCTION

As CMOS technologies continuously scale down, designing robust analog and mixed-signal (AMS) circuits becomes increasingly difficult. Consequently, there are pressing needs for AMS design checking techniques.

Although CMOS technology scaling is beneficial to achieving higher speed and lower power for digital circuits, it is decreasing the reliability of AMS circuits. Because the uncertainties of circuit electrical characteristics are increasing along with the technology scaling, circuit performances are more likely to be statistically distributed than to be deterministic values [2]. Also, reduced power supply voltage shrinks the dynamic range of AMS circuits and makes circuit design more challenging. Fig. 1 illustrates the future power supply voltage predicted by the International Technology Roadmap for Semiconductors (ITRS) [3]. Moreover, the ongoing design trends, such as the proliferation of consumer electronic systems, move towards integration of more functionalities on the same chip, requiring AMS modules to work in multiple modes, and have more complex control interface and lower power consumption [4].

Despite of the above design challenges, most of today's AMS circuits are still fully custom designed by experienced designers. Therefore, design checking techniques becomes a pressing need to assist AMS designers to develop robust AMS circuits. Existing design checking techniques can be divided into two categories, design verification and design for testability (DfT), as illustrated in Fig. 2. The purpose of verification is to ensure that the performance of an AMS design meets its specification under process, voltage and temperature (PVT) variations and different working conditions [5]. On the other hand, DfT techniques aim at embedding testability into the design, by adding auxiliary circuitries for testing purpose [6]. Instead of directly checking AMS

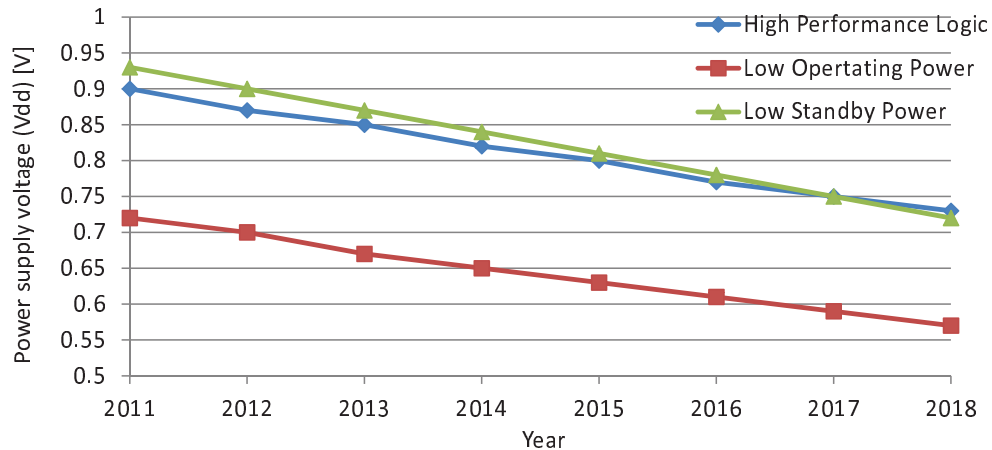


Fig. 1. Power supply voltage of future CMOS technologies predicted by ITRS.

designs, DfT provides the option of checking a chip after it is manufactured, and thus faults can be discovered in product test or in field. More specifically, those DfT techniques for in-field test are also known as in-situ test.

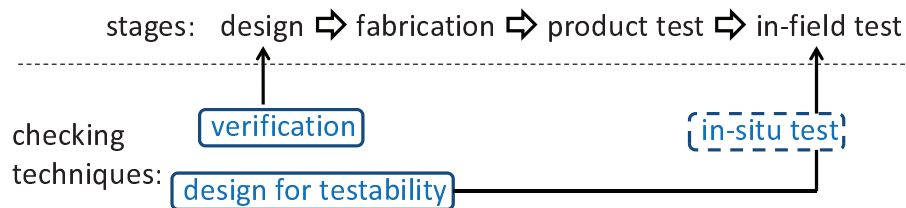


Fig. 2. Product stage vs. checking techniques.

This dissertation focuses on improving the robustness of AMS designs in highly scaled technologies, by developing novel formal verification and in-situ test techniques. Compared with conventional AMS verification that relies more on heuristically chosen simulations, formal verification provides a mathematically strict way of checking the target performance [7]. Moreover, formal verification techniques are more suitable to be implemented as automatic verification tools. On the other hand, though the

idea of in-situ test has been applied for many years, the powerful and low-cost digital processing capabilities of today's CMOS technologies are enabling many new in-situ test schemes in a mixed-signal environment.

The dissertation is composed of 3 subtopics: formal verification of AMS circuits (Chapter 2), high-resolution on-chip jitter measurement (Chapter 3) and in-situ test of all digital phase locked loops (PLLs) and polar transmitters (Chapter 4 and Chapter 5). Their relationships are shown in Fig. 3. Note that Chapter 3 gives a general solution for the jitter testing in AMS circuits, while Chapter 4 and Chapter 5 are applicable to specific types of AMS systems.

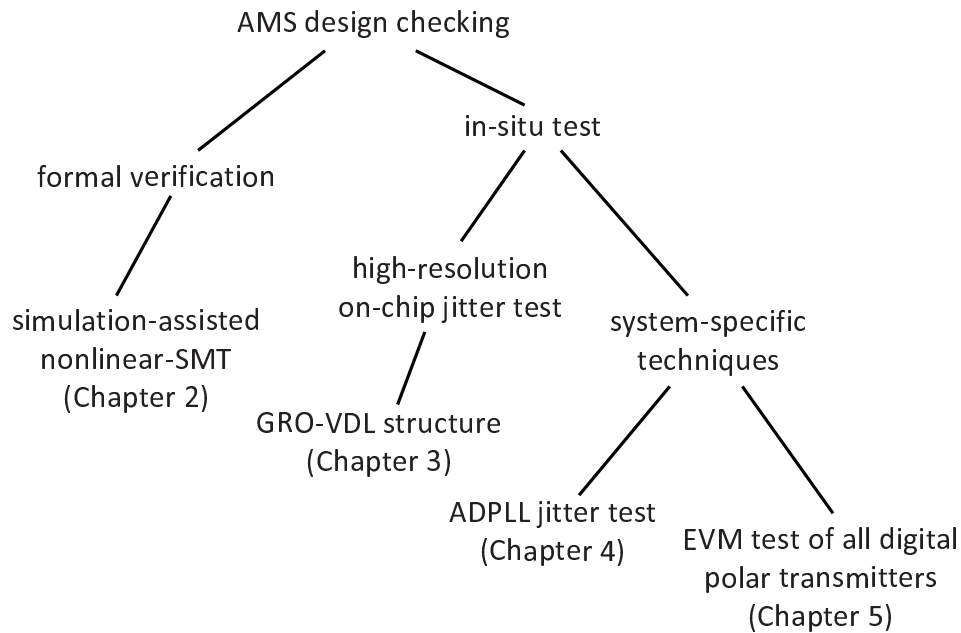


Fig. 3. Organization of the subtopics.

A. Formal verification of AMS circuits

Traditionally, errors in hardware are discovered empirically in the design stage, by verifying them under different situations. The most popular method for verifying an IC design is simulation. The disadvantage of simulation based verification is that it is difficult to obtain total confidence in the correctness of a design of any complexity. For example, the initial states and inputs of an analog circuit are continuous in their values while simulations could only sample discrete points in the continuous space. In contrast, formal verification is an alternative that mathematically proves if a design functions as required. More specifically, formal verification carries out a decision procedure to check whether a mathematical model for the design satisfies some given properties in the specification.

Formal verification of digital systems has found great success in practice [8]. In contrast, AMS circuits operate in continuous or hybrid state spaces and have far more complex analog characteristics and performances. As such, formal verification of complex AMS circuits remains as a significant challenge. Nevertheless, the success of its digital counterpart has made formal analog verification a subject of growing research interest.

This dissertation proposes a methodology that leverages SAT modulo theory (SMT)-based Satisfiability techniques to tackle the challenges arising from the inherent analog and/or hybrid natures of AMS systems. This work is largely motivated by recent advancements on nonlinear SMT (NL-SMT) solvers capable of solving the SAT problem with large Boolean combinations of nonlinear arithmetic constraints involving transcendental functions [9, 10]. The NL-SMT-based technique can be applied to yield conservative check of dynamic design properties. To accelerate the technique, a simulation-assisted SAT approach is also proposed that simultaneously exploit the ef-

efficiency of simulation and the conservativeness of SAT. A powerful Bayesian inference based technique is developed to dynamically tradeoff between the costs of simulation and NL-SMT. This allows intelligent on-the-fly determination of optimal number of simulation runs that gives rise to the minimum total runtime of the simulation-assisted SAT approach. The feasibility and efficacy of the proposed methodology are demonstrated on conservative verification of dynamic properties of a charge-pump PLL.

B. High-resolution on-chip jitter measurement

Timing precision, measured in the form of jitter, is extremely crucial for a broad range of high-speed high-precision digital and analog ICs. Jitter is one of the most important performances for the clock data recovery (CDR) in I/O circuitry as well as the clock generation in high-speed digital signal processing circuitry, where phase locked loops (PLLs) or delay locked loops (DLLs) are employed [11]. As an example, today's on-chip serial-links operate at a data rate of multi-Gb/s [12]. The clock jitter in serial-link transceivers degrades the transmitted and received data margin. It may also cause the received data to fall outside the design boundary. Moreover, from the perspective of RF applications, jitter performance is also a key concern because clock jitter will turn into the phase noise of wireless signals [1]. Hence, high-resolution jitter characterization is an important way to detect performance degradations or even malfunctions.

Traditionally, jitter is measured using external testing equipment. The state-of-the-art time interval analyzers (TIA) provide femto-second resolution. The achievable resolution is limited by the distortion and the noise injected along the on-chip to off-chip signal propagation path. In this regard, low cost on-chip solutions with high

resolution are particularly appealing, because signal distortion/noise can be largely alleviated by measuring the jitter right on the chip. More importantly, without the need for any expensive external equipment, in-situ jitter characterization allows built-in test and monitoring of design performance, and provides the option of self healing and correction in the events of jitter incurred failures.

In the dissertation, a novel structure of GRO-PVDL is proposed for on-chip jitter measurement. The GRO-PVDL is a two-level structure: the first level is a gated ring oscillator (GRO) providing a coarse measurement; and the second level further measures the residue from the first level with a fine resolution. The raw resolution of the GRO is improved through a Vernier-style structure at the second level. With the feature of quantization noise shaping, an even finer effective resolution can be achieved. Implemented with a commercial 90nm CMOS technology, the GRO-PVDL can achieve a sampling frequency of 200MHz and an effective resolution of 0.8ps.

C. In-situ test of all digital PLLs and polar transmitters

While digital testing is aiming at catastrophic and processing/manufacturing errors, the target of AMS testing is the functionality within acceptable upper and lower performance limits. AMS circuits have a nominal behavior and an uncertainty range due to PVT variations. The error or deviation from the nominal behavior must be measured with an extremely high precision to meet the requirements of today's high-resolution applications. The test cost is further raised when AMS circuits under test are part of a complex SoC rather than stand-alone components.

In this dissertation, the reconfigurability of recent all-digital PLL designs is exploited to provide novel in-situ output jitter test and diagnosis abilities under multiple parametric variations of key analog building blocks. Digital signatures are collected

and processed under specifically designed loop filter configurations to facilitate low-cost high-accuracy performance prediction and diagnosis, by systematically analyzing the interaction between the analog blocks and the digital blocks.

As an extension, an in-situ test scheme is proposed to provide online testing for all-digital PLL based polar transmitters. Multiple digital signatures are collected by adding a branch digital filter optimized for the maximum sensitivities to nonidealities, which provides testing results on the fly. The test signatures are processed using simple digital processing to provide an estimate for error vector magnitude (EVM), a key RF performance measure for the transmitter. Additionally, a digital self-calibration scheme is proposed to eliminate the EVM degradation due to large wide-band digitally controlled oscillator (DCO) gain mismatch. It is shown that a proper exploration of digital implementation style is instrumental for facilitating novel low-cost built-in test and calibration solutions for mixed-signal and RF applications.

CHAPTER II

SMT-BASED FORMAL VERIFICATION OF AMS CIRCUITS

As introduced in the first chapter, formal verification techniques for AMS circuits are a subject of growing research interest. In this chapter, a formal verification framework is presented that integrates nonlinear SAT modulo theory (SMT) solvers and Bayesian inference guided simulation exploration, aiming at the verification of AMS transient behaviors.

A. Introduction

The ongoing technology and design trends move towards integration of more functionality on the same chip, leading to the development of mixed-signal SoCs. Coupled with the increasing complexity of analog and mixed-signal (AMS) ICs, these trends have made the efficient verification of AMS circuits a pressing need. Formal verification of digital systems has found great success in practice. In contrast, analog and mixed-signal circuits operate in a continuous state space and have far more complex analog characteristics and performances. As such, verification of complex analog and mixed-signal ICs remains as a significant challenge. The success of its digital counterpart has nevertheless made formal analog verification a subject of growing research interest.

A number of approaches have been proposed for formal verification of analog circuits and a survey can be found from [13]. Among these techniques, theorem-proving based methods such as [14] check the design properties by applying proof rules, equivalence checking compares the outputs of two different models (e.g. SPICE vs. behavioral) for a given set of input conditions [15,16]. There also exist techniques that perform state-space exploration by converting continuous dynamics to approxi-

mated discrete models [17, 18]. State-space exploration can also be accomplished by using a popular class of reachability analysis originated from verification of hybrid systems [7, 19, 20]. These methods overapproximate the reachable state based on a geometrical representation such as polyhedra in the multi-dimensional space. Recently, an elegant reachability analysis technique is specifically developed for phase lock loops (PLLs) [21]. One of the key ideas in the approach is to overapproximate the switching times of the charge pump and perform reachability analysis using linear continuous models with uncertain parameters. In somewhat different directions, the monotonic property of MOSFET devices and numerical computation are used to find all DC solutions of a ring oscillator for verifying start-up conditions [22]. Boolean satisfiability (SAT) based circuit-level analog verification has also been demonstrated [23].

The presented work is largely motivated by recent advancements on automated reasoning of large Boolean combinations of nonlinear arithmetic constraints involving transcendental functions [9, 10]. Different from the SAT engine employed in [23], which can only operate in the Boolean and linear domains, techniques in the latter category are built upon a tight integration of recent Davis-Putnam-Logemann-Loveland (DPLL)-style SAT solving techniques with interval-based arithmetic constraint solving within a SAT modulo theory (SMT) framework. These techniques have the potential to process large constraint systems with Boolean combinations of multiple thousand arithmetic nonlinear constraints over thousands of variables [10]. For convenience, these types of techniques are referred to as *NL-SMT*.

The aforementioned NL-SMT techniques can handle nonlinear device/circuit characteristics, one inherent property of analog operations, making them a potentially appealing choice for analog and mixed-signal (a.k.a hybrid) circuit verification. However, practical limitations of solving capability still exist when such SMT solvers are employed for some of challenging AMS verification tasks.

For dynamic properties of nonlinear circuits, it is envisioned that modeling abstraction is required to render the transient verification (through reachability analysis) practical. Techniques such as [15, 16] may be used to build conservative behavioral models to account for factors such as modeling error and parameter variations for a large AMS circuit. NL-SMT can then be applied to the behavioral models to yield conservative check of dynamic design properties. However, acceleration techniques are still desired. To this end, a *simulation-assisted SAT* approach is proposed that simultaneously exploit the efficiency of simulation and the conservativeness of SAT. Simulation-assisted SAT can dramatically reduce the number of invoked NL-SMT calls, leading to large verification speedups. A powerful Bayesian inference based technique is developed to learn from the simulation history and dynamically trade-off between the costs of simulation and NL-SMT. This allows for intelligent on-the-fly determination of optimal number of simulation runs that gives rise to the minimum total runtime of the simulation-assisted SAT approach. To be able to flexibly model arbitrary nonlinear dynamics and the resulting reachable state space, the reachable state space is tracked using a collection of hyper cubes with adjustable discretization resolutions.

Compared with the verification tool *fSPICE* in [23] that also employs SAT solving techniques, the proposed approach has two obvious advantages. Because *fSPICE* relies on linear SAT solvers, nonlinear models have to be conservatively represented by interval combinations. In order to achieve a given accuracy and at the same time keep the runtime scalable, *fSPICE* has to introduce heuristic techniques of abstraction refinement and non-uniform splitting [23] and repeatedly invokes linear SAT solvers to find one solution. In the simulation-assisted NL-SMT approach, however, to find one solution with a given accuracy, the NL-SMT solver only needs to be invoked once, and the functions of abstraction refinement and non-uniform splitting are handled by

the NL-SMT solver in a more efficient way because its efficiency is optimized in the nonlinear SMT solving algorithms.

Moreover, the proposed approach is especially efficient for verifying transient behaviors. *f*SPICE and other SAT-based hybrid verification technique simply treat the transient verification problem as a huge DC verification problem by unrolling the transient behaviors over the time, which may be very computationally intensive or infeasible. In contrast, the proposed approach provides the option to split the problem into small subproblems such that the scale of the transient verification can be controlled. More importantly, with the assistance of random simulation to explore the reachable space, the efficiency of verification can be largely improved. In other words, we are applying SAT solvers in a more proper way, to check conservativeness rather than to find solutions.

The basic NL-SMT based reachability analysis approach is *general* in the sense that it can be applied to any analog and mixed-signal (hybrid) circuits modeled using many different types of *nonlinear* dynamic models (albeit practical capacity limitations exist). The application of this approach is demonstrated on a very challenging example, lock time verification of a charge-pump PLL. The generality of the approach forces explicit tracking of discrete switching events, avoiding potential overapproximations incurred otherwise. With additional PLL-specific speedup techniques, we demonstrate the successful lock time verification through the use of NL-SMT that explicitly tracks the nonlinear dynamics of the circuit over a large number of time steps and discrete switching events.

B. Hybrid systems

Hybrid system is a concept originally used in control theory. A hybrid system is a dynamic system that exhibits both continuous dynamics (flow) and discrete dynamics (jump) behaviors. A hybrid automaton is a mathematical model for precisely describing hybrid systems [24].

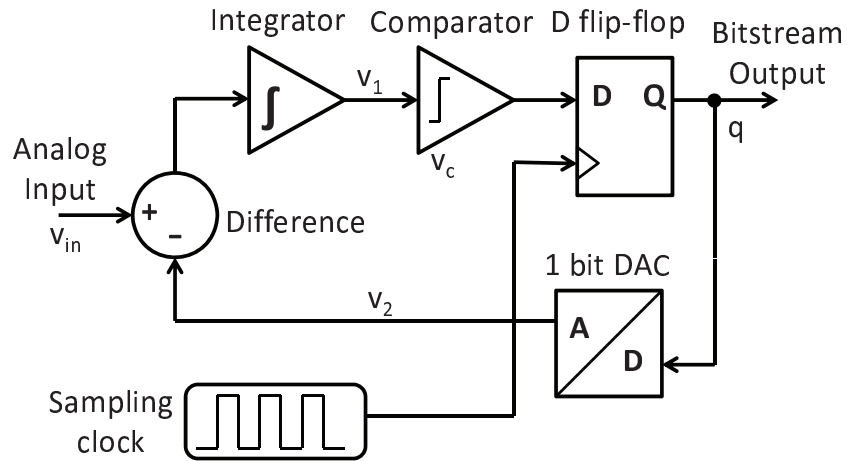
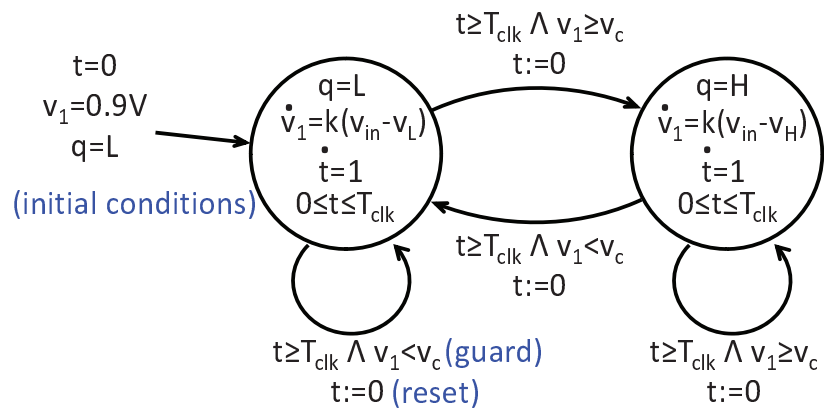
Definition 1. A *hybrid automaton* is a tuple $H = (X, M, J, F, I)$, where:

- $X \subseteq \mathbb{R}^n$ is an ordered finite set of continuous variables;
- M is a finite set of discrete states;
- $F \in M \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ assigns a vector field to each mode, the continuous dynamics in mode m is $\dot{\mathbf{x}} = f_m(\mathbf{x})$;
- $J \in M \times \mathbb{R}^n \rightarrow M \times \mathbb{R}^n$ is the jump relation, a jump is triggered by a *guard* condition and followed by a *reset* action.
- $I \subseteq M \times \mathbb{R}^n$ is the initial condition

More general forms of hybrid automaton also includes inputs, nondeterministic evolutions and even stochastic effects. AMS circuits are apparently within the category of hybrid systems, such as the 1-bit analog-to-digital convertor (ADC) shown in Fig. 4.

If we assume that the input level keeps unchanged at v_{in} , then the ADC's transient behavior can be described by the automaton shown in Fig. 5. Here t is time, $q = L/H$ represents the 2 discrete states of the ADC, v_1 is the integrator output (a continuous variable), k is the integrator gain and v_c is the threshold voltage of the comparator. When $q = L(H)$, the voltage level of the 1-bit DAC output is $v_L(v_H)$.

For $v_L = 0$, $v_H = 1V$, $v_c = 0.5V$ and the sampling clock frequency is 10MHz, the transient trajectories are given in Fig. 6. The dark solid line shows the v_1 trajectory

Fig. 4. 1-bit $\Delta - \Sigma$ ADC.Fig. 5. Hybrid automaton of 1-bit $\Delta - \Sigma$ ADC.

corresponding to its initial value of $0.9V$, while the light solid lines are a bunch of v_1 trajectories when its initial value ranges from $0.8V$ to $1V$.

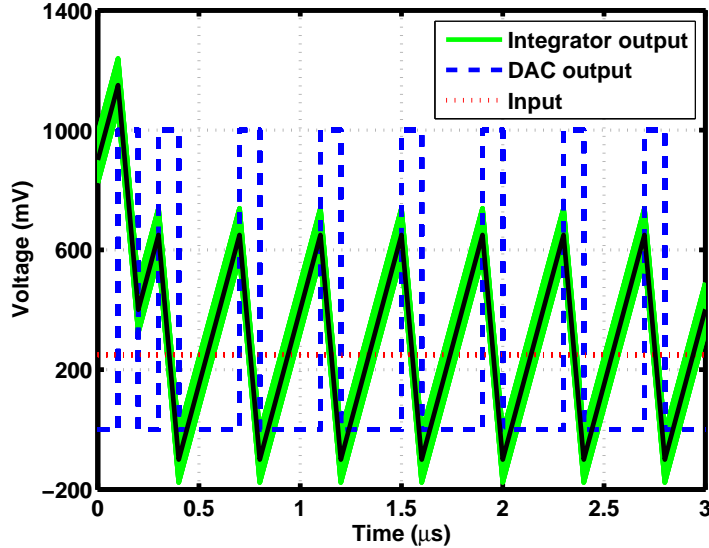


Fig. 6. Transient trajectories of 1-bit $\Delta - \Sigma$ ADC.

Note that in this example the occurrence of discrete transitions is synchronized with the sampling clock with a period of T_{clk} . Generally speaking, both synchronous and asynchronous discrete transitions may exist for AMS circuits, with the former triggered by clock edges and the latter caused by sudden switches of analog signals.

C. NL-SMT based verification

As previously mentioned, the recent NL-SMT solver [10] is able to solve satisfiability problems composed of boolean combinations of multiple arithmetic nonlinear constraints. In this section, an NL-SMT based framework is introduced for verifying the transient specifications of AMS circuits.

1. Formulation of NL-SMT constraints

Transient verification of hybrid systems like AMS circuits can be conducted by performing reachability analysis. *Reachability analysis* is to analyze the reachable space (both discrete and continuous) of the system, when system starts from some uncertain initial state I and/or follows some uncertain dynamics F and J (e.g. for the above ADC the gain the integrator k is uncertain due to process variation). A typical target of the analysis is to ensure the safety of a hybrid system by checking if any trajectory will enter a predefined bad/dangerous region within a given time. In order to perform reachability analysis using NL-SMT solver, hybrid automaton needs to be represented by NL-SMT constraints.

a. Initial space constraints

The entire initial space I is scattered among different discrete modes. In the i_{th} discrete mode \mathbf{m}_i , continuous variables \mathbf{x} are initially constrained within a region defined by $\mathbf{S}_i(\mathbf{x}) \bowtie \mathbf{0}$, where \bowtie stands for $=, <, >, \leq$ or \geq , and \mathbf{S}_i is a vector of functions that can be nonlinear. Therefore, the initial space can be represented by NL-SMT constraints as:

$$\bigvee_i \{ \mathbf{m}(t)|_{t=0} = \mathbf{m}_i \wedge \mathbf{S}_i(\mathbf{x}(t)|_{t=0}) \bowtie \mathbf{0} \}, \quad (2.1)$$

where $\mathbf{m}(t)$ and $\mathbf{x}(t)$ are discrete modes and continuous variable space that are reachable at time t .

Referring back to the previous $\Delta-\Sigma$ ADC, its discrete states are $\mathbf{M} = \{\mathbf{m}_1, \mathbf{m}_2\}$ (\mathbf{m}_1 represents $q = L$ and \mathbf{m}_2 represents $q = H$), and the integrator output voltage

v_1 is the continuous variable. An example of its initial state could be:

$$\begin{aligned} & \{\mathbf{m}(t)|_{t=0} = \mathbf{m}_1 \wedge v_1 \geq 0.1 \wedge v_2 \leq 0.2\} \\ & \vee \{\mathbf{m}(t)|_{t=0} = \mathbf{m}_2 \wedge v_1^2 - 0.3v_1 + 0.02 \leq 0\}, \end{aligned} \quad (2.2)$$

representing an initial space that distributes in both discrete mode \mathbf{m}_1 and discrete mode \mathbf{m}_2 . For \mathbf{m}_1 , the initial continuous space is the integrator output voltage v_1 between 0.1V and 0.2V, while the initial continuous space for \mathbf{m}_2 is constrained by $v_1^2 - 0.3v_1 + 0.02 \leq 0$.

b. Hybrid dynamics constraints

First let us consider *transient simulation* of hybrid systems, where discrete mode $\mathbf{m}(t)$ and continuous variable $\mathbf{x}(t)$ are calculated with a time step of Δt . Δt should be small enough to track both continuous dynamics and discrete dynamics. For simplicity, Δt is fixed in this paper. As illustrated in Fig. 7, for each Δt , hybrid dynamics are separated into 2 steps: first continuous flow and then discrete jump. According to continuous dynamics, \mathbf{x} evolves from $\mathbf{x}(t_0)$ to $\mathbf{x}^*(t_0 + \Delta t)$. If $\mathbf{x}^*(t_0 + \Delta t)$ hits any guard condition, the corresponding reset action will be taken.

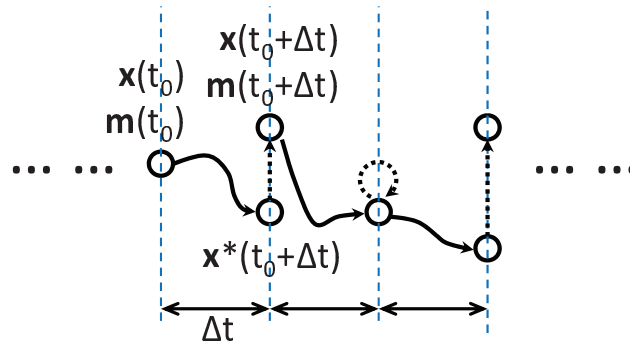


Fig. 7. Step-by-step evolution of hybrid automaton.

The derivatives of continuous variables are determined by the value of \mathbf{x} as well as the current discrete mode \mathbf{m} .

$$\dot{\mathbf{x}}(t) = \mathbf{F}_{\dot{\mathbf{x}}}(\mathbf{m}(t), \mathbf{x}(t)), \quad (2.3)$$

where $\mathbf{F}_{\dot{\mathbf{x}}}$ is a vector of functions that can be nonlinear. Using trapezoid Euler method, we have:

$$\frac{\mathbf{F}_{\dot{\mathbf{x}}}(\mathbf{m}(t_0), \mathbf{x}^*(t_0 + \Delta t)) + \mathbf{F}_{\dot{\mathbf{x}}}(\mathbf{m}(t_0), \mathbf{x}(t_0))}{2} = \frac{\mathbf{x}^*(t_0 + \Delta t) - \mathbf{x}(t_0)}{\Delta t}. \quad (2.4)$$

For simulation, numerical computation will kick in here to solve $\mathbf{x}^*(t_0 + \Delta t)$. In NL-SMT approach, instead, we formulate a set of constraints as:

$$\bigvee_i \left\{ \mathbf{m}(t_0) = \mathbf{m}_i \rightarrow \frac{\mathbf{F}_{\dot{\mathbf{x}}}(\mathbf{m}_i, \mathbf{x}^*(t_0 + \Delta t)) + \mathbf{F}_{\dot{\mathbf{x}}}(\mathbf{m}_i, \mathbf{x}(t_0))}{2} = \frac{\mathbf{x}^*(t_0 + \Delta t) - \mathbf{x}(t_0)}{\Delta t} \right\}, \quad (2.5)$$

where $A \rightarrow B$ is equivalent to $\neg A \vee B$.

For example, if currently the previous ADC is in discrete mode \mathbf{m}_1 (\mathbf{m}_2), meaning its output $q = L(q = H)$, then the continuous flow is described by the derivatives of the continuous variables $\dot{v}_1 = k(v_{in} - v_L)$ ($\dot{v}_1 = k(v_{in} - v_H)$). Therefore the corresponding SMT constraints are:

$$\begin{aligned} \{ \mathbf{m}(t_0) = \mathbf{m}_1 \rightarrow \frac{k(v_{in}(t_0 + \Delta t) - v_L) + k(v_{in}(t_0) - v_L)}{2} = \frac{v_1^*(t_0 + \Delta t) - v_1(t_0)}{\Delta t} \} \\ \vee \{ \mathbf{m}(t)|_{t=0} = \mathbf{m}_2 \rightarrow \frac{k(v_{in}(t_0 + \Delta t) - v_H) + k(v_{in}(t_0) - v_H)}{2} = \frac{v_1^*(t_0 + \Delta t) - v_1(t_0)}{\Delta t} \}. \end{aligned} \quad (2.6)$$

After the continuous flow, there may also be discrete jumps. We note $G_i^{(j)}$ as the j th guard condition for the i th discrete mode \mathbf{m}_i , and $R_i^{(j)}$ as the corresponding reset action for the discrete jump. In fact, $G_i^{(j)}$ is a set of continuous space, and $R_i^{(j)}$ is a mapping from $\mathbf{x}^*(t_0 + \Delta t)$ to $\mathbf{m}(t_0 + \Delta t)$ and $\mathbf{x}(t_0 + \Delta t)$. Particularly, $G_i^{(0)}$ is noted as the condition of no discrete jump in \mathbf{m}_i , and therefore $R_i^{(0)}$ does not change

anything. Then discrete jump can be represented by the following constraints:

$$\bigvee_{i,j:i \geq 1, j \geq 0} \{ \mathbf{m}(t_0) = \mathbf{m}_i \wedge \mathbf{x}^*(t_0 + \Delta t) \in G_i^{(j)} \longrightarrow \\ (\mathbf{m}(t_0 + \Delta t), \mathbf{x}(t_0 + \Delta t)) = R_i^{(j)}(\mathbf{x}^*(t_0 + \Delta t)) \} \quad (2.7)$$

Also, we take the ADC as an example, whose discrete jumps only happen at its sampling clock edges. Therefore its guard condition should include time $t \geq T_{clk}$ for all the discrete jumps, where T_{clk} is the sampling clock period. And if any discrete jump is triggered, time t is reset to 0, indicating the start of a new sampling clock cycle. Meanwhile, the discrete jump is also determined by the integrator output voltage v_1 at the clock edges. If the ADC is currently in discrete mode $\mathbf{m}_1(\mathbf{m}_2)$, a discrete jump is only triggered when $t \geq T_{clk}$ and $v_1 \geq v_c(v_1 < v_c)$, where v_c is the threshold voltage of the comparator. So the constraints for discrete jumps are the disjunctions of the following constraints:

$$\{ \mathbf{m}(t_0) = \mathbf{m}_1 \wedge t^*(t_0 + \Delta t) < T_{clk} \longrightarrow \\ \mathbf{m}(t_0 + \Delta t) = \mathbf{m}_1 \wedge t(t_0 + \Delta t) = t^*(t_0 + \Delta t) \wedge v_1(t_0 + \Delta t) = v_1^*(t_0 + \Delta t) \}, \quad (2.8)$$

$$\{ \mathbf{m}(t_0) = \mathbf{m}_1 \wedge t^*(t_0 + \Delta t) \geq T_{clk} \wedge v_1^*(t_0 + \Delta t) < v_c \longrightarrow \\ \mathbf{m}(t_0 + \Delta t) = \mathbf{m}_1 \wedge t(t_0 + \Delta t) = 0 \wedge v_1(t_0 + \Delta t) = v_1^*(t_0 + \Delta t) \}, \quad (2.9)$$

$$\{ \mathbf{m}(t_0) = \mathbf{m}_1 \wedge t^*(t_0 + \Delta t) \geq T_{clk} \wedge v_1^*(t_0 + \Delta t) \geq v_c \longrightarrow \\ \mathbf{m}(t_0 + \Delta t) = \mathbf{m}_2 \wedge t(t_0 + \Delta t) = 0 \wedge v_1(t_0 + \Delta t) = v_1^*(t_0 + \Delta t) \}, \quad (2.10)$$

$$\{ \mathbf{m}(t_0) = \mathbf{m}_2 \wedge t^*(t_0 + \Delta t) < T_{clk} \longrightarrow \\ \mathbf{m}(t_0 + \Delta t) = \mathbf{m}_2 \wedge t(t_0 + \Delta t) = t^*(t_0 + \Delta t) \wedge v_1(t_0 + \Delta t) = v_1^*(t_0 + \Delta t) \}, \quad (2.11)$$

$$\{ \mathbf{m}(t_0) = \mathbf{m}_2 \wedge t^*(t_0 + \Delta t) \geq T_{clk} \wedge v_1^*(t_0 + \Delta t) \geq v_c \longrightarrow \\ \mathbf{m}(t_0 + \Delta t) = \mathbf{m}_2 \wedge t(t_0 + \Delta t) = 0 \wedge v_1(t_0 + \Delta t) = v_1^*(t_0 + \Delta t) \}, \quad (2.12)$$

$$\{\mathbf{m}(t_0) = \mathbf{m}_2 \wedge t^*(t_0 + \Delta t) \geq T_{clk} \wedge v_1^*(t_0 + \Delta t) < v_c \longrightarrow \mathbf{m}(t_0 + \Delta t) = \mathbf{m}_1 \wedge t(t_0 + \Delta t) = 0 \wedge v_1(t_0 + \Delta t) = v_1^*(t_0 + \Delta t)\}. \quad (2.13)$$

Note that time t is a special continuous variable, because its continuous derivative is always $\dot{t} = 1$, and the corresponding constraint is $t^*(t_0 + \Delta t) = t(t_0) + \Delta t$.

c. Other constraints

The verification target could also be formulated into NL-SMT constraints depending on its form. For typical targets like avoiding bad region, the corresponding NL-SMT constraints are similar to those for initial conditions.

In addition, to include the effect of process variation in verification, device parameters \mathbf{p} are considered as a special kind of continuous variables, and therefore NL-SMT constraints could be formulated for them accordingly. The specialty is that \mathbf{p} is independent of discrete states and continuous variables, and does not change over the time: $\mathbf{p}(t_0 + \Delta t) = \mathbf{p}(t_0)$.

2. Basic NL-SMT approach

For the purpose of transient verification, an NL-SMT based approach is proposed to find all reachable space conservatively.

a. State-space discretization

As described above, reachability analysis can be formulated into NL-SMT constraints. If transient verification is bounded over a time duration t_{max} , then an obvious way to generate the NL-SMT problem is to make conjunction of (1) initial space constraints, (2) verification target constraints and (3) hybrid dynamics constraints which are “unrolled” from time 0 to t_{max} with a step of Δt . Such idea has been applied in

[23] using a SAT solver that accepts Boolean and linear constraints. However, the worst-case cost of solving NL-SMT problems increases exponentially with the problem dimension. As a result, it would be infeasible to run the verification over a large number of steps. A compromised approach is to split a big NL-SMT problem into a series of NL-SMT subproblems, as illustrated in Fig. 8. Each subproblem is limited to a few time steps so that the problem scale can be handled by NL-SMT solver. This strategy reduces the total cost to linear in t_{max} . For simplicity, every subproblem is limited to 1 time step in this work.

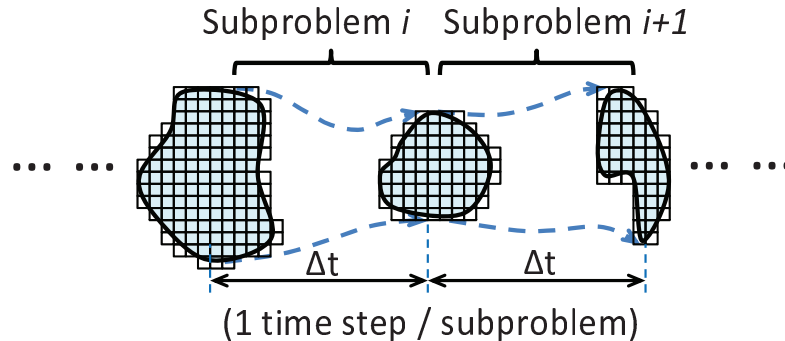


Fig. 8. The split of NL-SMT problem.

The reachable space at the ending of each subproblem needs to be saved as the initial conditions of the subsequent subproblem. To save the reachable space, the entire state-space is discretized into fixed-grid boxes. A box that contains a reachable point is considered as a *reachable box*. The reachable space can be conservatively saved as a set of reachable boxes. Note that over-approximation of reachable space is introduced to ensure conservativeness. The size of boxes could be adjusted on the fly for tradeoff between computational cost and over-approximation. Although other shapes have been shown to be effective for some specific dynamics, e.g. zonotopes are successfully applied for linear dynamics [21], boxes (or high-dimensional cubes)

are still the most flexible choice for general nonlinear dynamics. The condition, that a point $(\mathbf{m}(t), \mathbf{x}(t))$ is in reachable boxes at time t , could be written into NL-SMT constraints as:

$$\bigvee_{i,j} \{\mathbf{m}(t) = \mathbf{m}_i \wedge \mathbf{L}_i^{(j)}(t) \leq \mathbf{x}(t) \leq \mathbf{U}_i^{(j)}(t)\} \quad (2.14)$$

where $\mathbf{L}_i^{(j)}(t)(\mathbf{U}_i^{(j)}(t))$ is the lower(upper) bound of the j th reachable box in the i th discrete mode \mathbf{m}_i at time t .

b. Box mergence

Mergence of boxes, as an auxiliary technique, can help accelerate solving NL-SMT problems. Through box mergence, as illustrated in Fig. 9, the number of conjunction/disjunction clauses in constraints is reduced, and thus solver runtime is saved. In the implementation, a simple greedy algorithm is adopted for box mergence, as shown in Alg. 1. Note that different merging algorithms will lead to different sets of merged boxes.

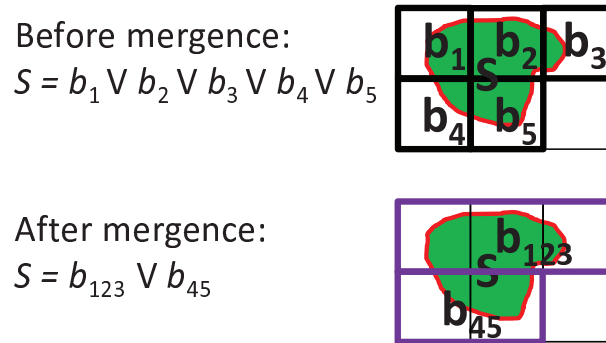


Fig. 9. Box mergence.

The greedy algorithm involves 2 lists of boxes, one storing unmerged boxes (L_{um}) and the other storing merged boxes (L_{m}). Initially, L_{um} stores the set of boxes to be

merged and L_m is empty. Each box b_i in L_{um} is visited one by one, and if b_i can be merged into any boxes b'_j in L_m , then b'_j in L_m will be updated with $b_i \cup b'_j$, the box merged from b_i and b'_j . If not, b_i is inserted into L_m . After all boxes in L_m are visited, the algorithm compares the number of boxes in L_m and in L_{um} . If the numbers are the same, which means that the last run of visit does not result in any mergence and so the algorithm stops. Otherwise, L_{um} dumps all its content and takes all the content of L_m , after which L_m is cleared to be an empty list again. Then another run of visit to the boxes in L_{um} will be performed. This process repeats until the size of L_m equals the size of L_{um} after a run of visit.

c. Basic flow of invoking NL-SMT solver

The goal of each subproblem is to find all the reachable space at its ending time. Starting from an empty set of reachable box B_{rch} , when B_{rch} is returned it should be filled with boxes that conservatively cover the reachable space. This task needs to be accomplished by means of NL-SMT solver.

The iSAT NL-SMT solver [10] employed in this work is based on a tight integration of DPLL algorithm and the interval constraint propagation (ICP) technique. Based on interval arithmetic, interval constraint propagation locates the intervals containing all solutions to the problem constraints. To practically force the solver to terminate, we set a threshold δ corresponds to the discretization resolution (i.e. box size) such that a real variable is no longer considered in the decision process if the interval length of the variable is less than δ . When the solver returns, it either returns one possible solution or UNSAT. Note here that the solver provides a guarantee on unsatisfiability, i.e. an UNSAT result indicates that there is indeed no solution to the problem constraints.

Given such properties of the solver, a flow of invoking NL-SMT solver is given

Algorithm 1 Greedy box mergence

```
while TRUE do  
     $L_m = \emptyset$ ;  
    for each box  $b_i \in L_{um}$  do  
        if a box  $b'_j \in L_{um}$  that can be merged with  $b_i$  then  
             $b'_j = b'_j \cup b_i$ ; //merge  
        else  
            insert  $b_i$  into  $L_m$ ;  
        end if  
    end for  
    if sizeof( $L_m$ ) == sizeof( $L_{um}$ ) then  
        break;  
    else  
         $L_{um} = L_m$ ;  
    end if  
end while  
return  $L_m$ 
```

Algorithm 2 Basic NL-SMT approach

 $B_{rch} = \emptyset;$ **repeat**

formulate constraints for initial space;

formulate constraints for hybrid dynamics;

 formulate constraints for $s_{rch} \notin B_{rch};$

invoke NL-SMT solver;

if a reachable solution s_{rch} is found **then** locate box b that contains $s_{rch};$ $B_{rch} = B_{rch} \cup \{b\};$ **end if****until** NL-SMT solver returns UNSATmerge $B_{rch};$ return $B_{rch};$

in Alg. 2. This algorithm is invoked for every subproblem, or at each time step if 1 subproblem covers 1 time step. At first, the set of reachable box B_{rch} is empty. Then 3 types of the NL-SMT constraints are formulated, including for initial space, for hybrid dynamics, and for $s_{rch} \notin B_{rch}$. The initial space is actually the reachable space of the last subproblem. $s_{rch} \notin B_{rch}$ means that s_{rch} is not inside the reachable space B_{rch} . Note that these 3 types of constraints are implicitly conjunct to each other. With initial space and hybrid dynamics formulated as constraints, the solver will return a solution s_{rch} that is in the reachable space. Then the box that contains s_{rch} is added into B_{rch} . If the solver is invoked repeatedly, more reachable points will be found and B_{rch} will get larger. In each round, most importantly, constraints must also be formulated for $s_{rch} \notin B_{rch}$, which is the negation of Eq.(2.14). This can keep each new solution out the reachable boxes already found. In Fig. 10, for example, supposing boxes 1 and 2 are already found, they would be blocked for next NL-SMT invoke so that new solution could only appear in boxes 3, 4 and 5. Finally, when all the reachable boxes are found, the last invoking solver will return UNSAT, providing a guarantee on conservativeness. Note that for this flow the number of NL-SMT invoking is equal to the total number of reachable boxes plus one, because every invoking returns a new reachable box, except the last invoking that confirms the conservativeness.

D. Simulation-assisted NL-SMT

Though the above NL-SMT based approach is capable of conservatively finding all reachable boxes, the number of NL-SMT invokes is equal to the number of reachable boxes plus one, which can be very large. Considering it is still very costly to invoke NL-SMT solver that frequently, the basic NL-SMT approach is obviously not

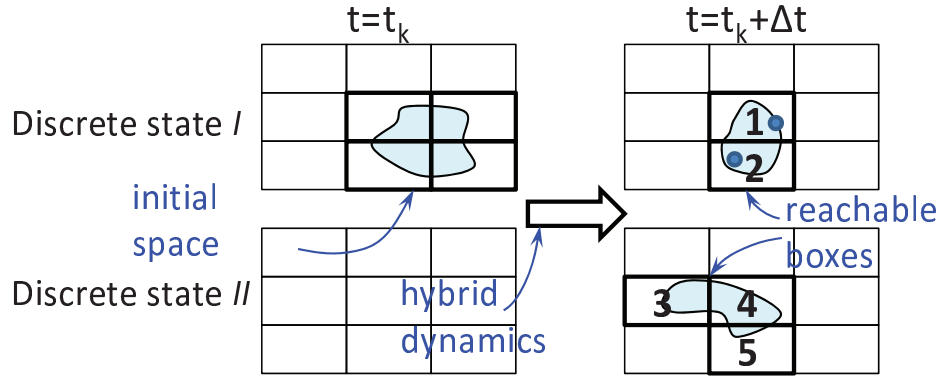


Fig. 10. Search of reachable boxes.

efficient for transient verification. This stimulates us to search additional assistance. In this section, simulation-assisted NL-SMT is proposed that dramatically increases verification speed and keeps conservativeness at the same time.

1. Simulation-assisted NL-SMT flow

A key observation here is that *transient simulation* can find a solution with much lower cost than NL-SMT solver. And if the solution is located in a box that is not visited before, then it is just as valuable as the solution given by NL-SMT solver. Therefore we can quickly explore the reachable state space by means of simulation from samples within the initial space. Note that the sampling strategy could be simply uniformly random. More sophisticated sampling techniques could be adopted to further increase the coverage of simulation-based exploration. On the other hand, no matter how dense or how wise the sampling strategy is, the coverage can hardly be 100% for complex dynamics, i.e. there is no conservativeness guarantee for simulation-based exploration.

Fig. 11 compares the pros and cons of the above 2 approaches, and suggests a simulation-assisted NL-SMT approach. Majority of reachable boxes are found in the

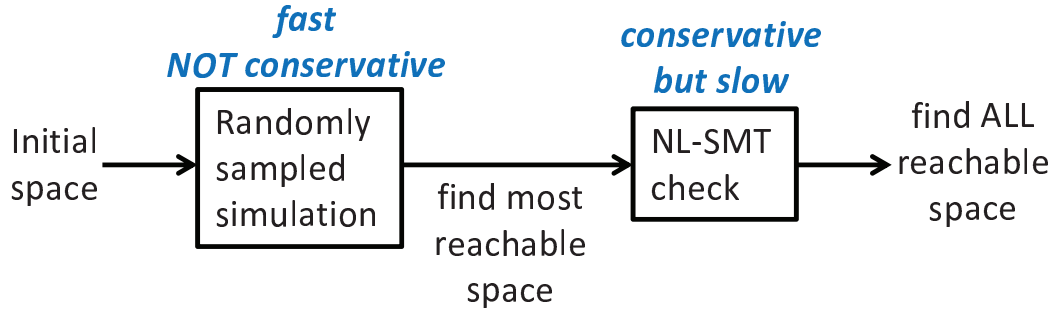


Fig. 11. Flow of simulation-assisted approach.

first stage, through simulations starting from points x_{ini} that are randomly sampled in initial boxes B_{ini} . In the next stage, NL-SMT solver kicks in to find the remaining reachable boxes that are missed in the first stage. In Fig. 12, for example, if random simulations quickly reach points in box b'_1 , b'_3 and b'_4 , then when NL-SMT check begins, B_{rch} already includes box b'_1 , b'_3 and b'_4 . Consequently, only 2 times of invoking NL-SMT solver is needed, the first for finding box b'_2 and the second for confirming conservativeness.

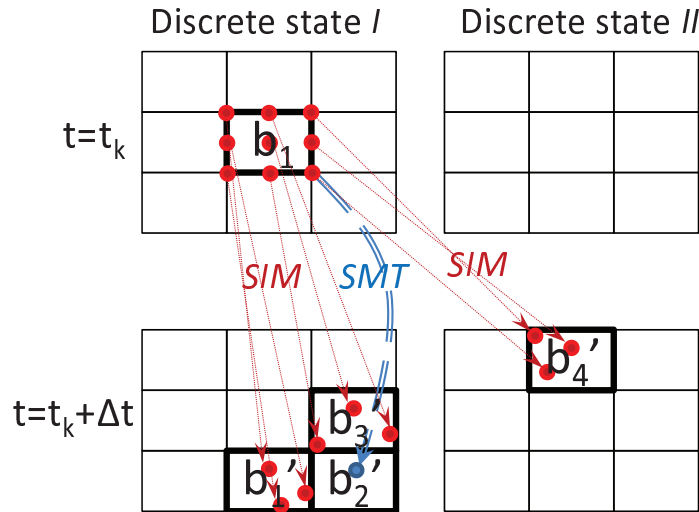


Fig. 12. Simulation exploration and NL-SMT check.

The benefit of simulation assistance is that most reachable spaces can be found through simulations, and therefore the chance of invoking NL-SMT solver can be remarkably decreased. Although simulations might have a lot of “waste”, i.e. simulations from different samples repeatedly visit solutions in the boxes that are already found, yet the cost of simulation is so much smaller than NL-SMT invoking that the total time consumption is still much smaller than basic NL-SMT approach.

The simulation-assisted NL-SMT approach is summarized in Alg.3. Note that simulation-based exploration stops when condition of stopping simulation, namely *Stop*, is true. In practice, the runtime cost of simulation could be so small that *Stop* can be checked for every hundred or even more samples of simulation. Moreover, the determination of the stop condition *Stop* is very important to runtime performance. A systematic way of defining *Stop* is given in next section.

The fundamental principle of simulation assisted NL-SMT verification is to find reachable spaces by simulations quickly, and then rely on NL-SMT solver to ensure the conservativeness. The best-case scenario will be that all the reachable spaces from one starting box can be found by simulation, and NL-SMT solver needs to be invoked only once to confirm the conservativeness. In this sense, the proposed simulation-assisted NL-SMT flow is combining the best of the two worlds: simulation and verification.

2. Stop condition

There is an important question related to the above simulation assisted NL-SMT flow: when should we stop simulations and start NL-SMT based conservativeness check? One intuitive answer is: if we have a larger(smaller) initial space or higher(lower) state-space dimension, then we should run more(less) samples of simulation. This

Algorithm 3 Simulation-assisted NL-SMT approach

 $B_{rch} = \emptyset;$
*/*random simulation exploration*/*
repeat

 randomly sample x_{ini} in B_{ini} ;

 run simulation from x_{ini} and visit x_{rch} ;

 locate box b that contains x_{rch} ;

 $B_{rch} = B_{rch} \cup \{b\};$
until condition of stopping simulation $Stop$ is true

*/*NL-SMT conservativeness check*/*
repeat

formulate constraints for initial space;

formulate constraints for hybrid dynamics;

 formulate constraints for $s_{rch} \notin B_{rch}$;

invoke NL-SMT solver;

if a reachable solution s_{rch} is found **then**

 locate box b that contains s_{rch} ;

 $B_{rch} = B_{rch} \cup \{b\};$
end if
until NL-SMT solver returns UNSAT

 merge B_{rch} ;

 return B_{rch} ;

answer suggests a definition of *Stop*, which can be mathematically expressed as:

$$N_s > N_{B_I} \rho^d \quad (2.15)$$

where N_s is the number of samples of simulation that are already run for the current subproblem or time step, N_{B_I} is the number of boxes in initial space, d is the state-space dimension and ρ is sampling density, a user chosen parameter. This gives a condition of stopping simulation that changes for different time steps, since in reachability analysis N_{B_I} is changing along the time. However, Eq.(2.15) is still called the static *Stop* condition, because it is non-adaptive and hence non-optimal as it does not track the evolution of hybrid dynamics within a subproblem.

To seek a more powerful dynamic scheme for stopping simulation, we have the the following useful observation. It is likely to find new reachable boxes in first several simulation samples while this becomes more difficult in the later phase of simulation sampling as a bulk of reachable boxes have already been found. This prompts us to monitor the numbers of new boxes found in recent samples and stop simulation when only a small number of new boxes have been discovered in the recent history.

We put our intuition in a more rigorous manner by developing the following statistical learning approach. The algorithm for simulation-assisted NL-SMT approach is rewritten as Alg.4. The NL-SMT conservativeness check part of Alg.4 is not detailed since it is the same as that of Alg.3. Note that the *Stop* condition is re-evaluated for every k random samples of simulation. So first of all, we need to estimate the number of new boxes that will be found if k additional samples of simulation are run, namely $N_{new}^{(k)}$, based on the history of simulation results. More specifically, the objective is to calculate $\mathbf{E}[N_{new}^{(k)} | \mathbf{H}]$, the expectation of $N_{new}^{(k)}$ given an observation \mathbf{H} that represents the history of previous samples. Note that \mathbf{H} only covers the previous samples for the current subproblem or time step.

Algorithm 4 Simulation-assisted NL-SMT approach with dynamic stop condition

```

 $B_{rch} = \emptyset;$ 

/*random simulation exploration*/
 $\mathbf{H} = \emptyset;$  //sampling history
repeat
  /*randomly sample  $k$  points*/
  for  $i = 1 \rightarrow k$  do
    randomly sample  $x_{ini}$  in  $B_{ini}$ ;
    run simulation from  $x_{ini}$  and visit  $x_{rch}$ ;
    locate box  $b$  that contains  $x_{rch}$ ;
     $B_{rch} = B_{rch} \cup \{b\};$ 
    update  $\mathbf{H}$  with  $x_{ini}$  and  $b$ ;
  end for
  evaluate  $Stop$  with the sampling history  $\mathbf{H}$ ;
until  $Stop$  is true

/*NL-SMT conservativeness check*/
.....

merge  $B_{rch}$ ;
return  $B_{rch}$ ;

```

Supposing $\mathbf{E}[N_{new}^{(k)}|\mathbf{H}]$ is already obtained, then the *Stop* condition can be derived with $\mathbf{E}[N_{new}^{(k)}|\mathbf{H}]$. If simulation is stopped at this point of time, these unvisited boxes must be covered by running the SAT solver $\mathbf{E}[N_{new}^{(k)}|\mathbf{H}]$ times. Therefore, stopping simulation is only beneficial if the runtime cost of k simulations is more than compensated by that of the SAT runs:

$$k \cdot \tau_{sim} > \mathbf{E}[N_{new}^{(k)}|\mathbf{H}] \cdot \tau_{smt} \quad (2.16)$$

where the left side is the runtime of k samples of simulation, and the right side is the total runtime of finding $\mathbf{E}[N_{new}^{(k)}|\mathbf{H}]$ new boxes by NL-SMT, τ_{sim} and τ_{smt} are the estimated runtime of one simulation run and one invoking of NL-SMT solver, respectively. Hence, a practical stopping condition is as follows:

$$\mathbf{E}[N_{new}^{(k)}|\mathbf{H}] < \kappa, \quad (2.17)$$

where $\kappa = k\tau_{sim}/\tau_{smt}$.

In the flow described above, the key is the link from the sampling history \mathbf{H} to $\mathbf{E}[N_{new}^{(k)}|\mathbf{H}]$, i.e. to calculate the posterior expectation of the number of new boxes to be found in the next k samples of simulation. Actually, when using the term of “posterior expectation”, we are implicitly assuming that it is a statistical problem within the framework of Bayesian inference.

3. Statistical framework

In order to apply the standard flow of Bayesian inference, the process of random sampling and simulation needs to be modeled in a statistical framework. When randomly sampling a point x_{ini} in the initial space as the starting point of the simulation, the simulation result is a point x_{rch} in the state space, representing the state of the hybrid system at the end of the current subproblem or time step. Given all the valid space

is discretized into m boxes b_1, \dots, b_m , x_{rch} will fall in one box (b_{rch}) in the candidate box set $B = \{b_1, \dots, b_m\}$: $x_{rch} \in b_{rch} \in B$. Note that the candidate box set B can be conservatively pruned to be smaller than the entire valid space, according to the system dynamics. Because x_{ini} is randomly sampled, the visited box b_{rch} is actually a random variable whose value can be any element in the candidate box set B . For a fixed initial space and fixed system dynamics, $\mathbf{P}(b_{rch} = b_i)$, the probability that the visited box is b_i , is also fixed as long as the starting point x_{ini} is sampled with a fixed probability distribution in the initial space. For simplicity, x_{ini} is uniformly sampled in the initial space in our implementation. The fixed probability distribution of b_{rch} is noted as:

$$\boldsymbol{\theta} = \{\theta_1, \dots, \theta_m\}, \quad (2.18)$$

where θ_i is the probability of visiting b_i , i.e. $b_{rch} = b_i$, with $\theta_i \geq 0$ and $\sum_{i=1}^m \theta_i = 1$. This statistical view of the sampling and simulation process is illustrated in Fig. 13.

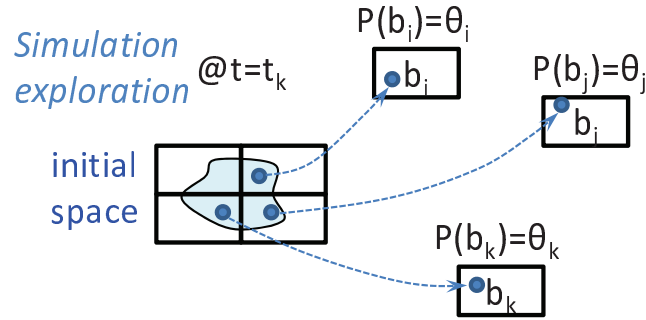


Fig. 13. Statistical view of simulation exploration.

It is also noted that each time the sampling of x_{ini} always follows uniform distribution, so $\boldsymbol{\theta}$, the probability distribution of the visited box b_{rch} , is not only fixed but also independent from each other for multiple samples. Therefore, the sequence of visited boxes \mathbf{b}_{rch} is a sequence of independent and identically distributed (IID) random

variables. From statistical point of view, the sampling and simulation process is analogous to rolling an m -side dice with uneven probability distribution $\boldsymbol{\theta} = \{\theta_1, \dots, \theta_m\}$, where θ_i is the probability of having the i th side on the top.

In our context, the probability distribution $\boldsymbol{\theta}$, that describes the possibilities of a sample of simulation visiting each candidate box, is an unknown parameter and our target is to make Bayesian inference about it from the history of sampling and simulation, so that we can finally compute the posterior expectation $\mathbf{E}[N_{new}^{(k)} | \mathbf{H}]$.

4. Bayesian inference

a. Principle of Bayesian inference

The principle of Bayesian inference [25] is briefly given as follows. If we assume that \mathcal{A} is one explanation for the observed history \mathcal{B} , and \mathcal{C} summarizes the prior assumptions, then Bayes' rule states:

$$\mathbf{P}(\mathcal{A}|\mathcal{B}\mathcal{C}) = \frac{\mathbf{P}(\mathcal{A}|\mathcal{C})\mathbf{P}(\mathcal{B}|\mathcal{A}\mathcal{C})}{\mathbf{P}(\mathcal{B}|\mathcal{C})}. \quad (2.19)$$

Before having the observation \mathcal{B} , only \mathcal{C} is known, but afterwards $\mathcal{B}\mathcal{C}$ (\mathcal{B} and \mathcal{C}) is known. In response to the observation \mathcal{B} , Bayes' rule suggests updating the probability of the explanation \mathcal{A} being true, from $\mathbf{P}(\mathcal{A}|\mathcal{C})$ into $\mathbf{P}(\mathcal{A}|\mathcal{B}\mathcal{C})$. Here $\mathbf{P}(\mathcal{A}|\mathcal{C})$ is called the *prior* probability and $\mathbf{P}(\mathcal{A}|\mathcal{B}\mathcal{C})$ is the *posterior* probability.

Supposing that $\mathcal{A}_1, \mathcal{A}_2, \dots$ are exhaustive and mutually exclusive explanations (exactly one of \mathcal{A}_i is true while the rest false), then the posterior probability of \mathcal{A}_i being true can be given by Bayes' rule:

$$\mathbf{P}(\mathcal{A}_i|\mathcal{B}\mathcal{C}) = \frac{\mathbf{P}(\mathcal{A}_i|\mathcal{C})\mathbf{P}(\mathcal{B}|\mathcal{A}_i\mathcal{C})}{\sum_j \mathbf{P}(\mathcal{A}_j|\mathcal{C})\mathbf{P}(\mathcal{B}|\mathcal{A}_j\mathcal{C})}. \quad (2.20)$$

If the exhaustive and mutually exclusive explanations are no longer countable,

but are infinitely many, then the discrete probability distribution of \mathcal{A} turns into the probability density of \mathbf{a} , where \mathbf{a} is one point in the continuous range of the explanation. And Bayes' rule leads to:

$$\mathbf{p}(\mathbf{a}|\mathcal{BC}) = \frac{\mathbf{p}(\mathbf{a}|\mathcal{C})\mathbf{P}(\mathcal{B}|\mathbf{a}\mathcal{C})}{\int \mathbf{p}(\mathbf{a}|\mathcal{C})\mathbf{P}(\mathcal{B}|\mathbf{a}\mathcal{C})d\mathbf{a}}, \quad (2.21)$$

where $\mathbf{p}(\mathbf{a}|\mathcal{C})$ and $\mathbf{p}(\mathbf{a}|\mathcal{BC})$ are the prior and posterior probability densities of the explanation \mathbf{a} , respectively. For convenience, the prior assumption \mathcal{C} is usually not explicitly written, so Bayesian inference can be summarized by:

$$\mathbf{p}(\mathbf{a}|\mathcal{B}) = \frac{\mathbf{p}(\mathbf{a})\mathbf{P}(\mathcal{B}|\mathbf{a})}{\int \mathbf{p}(\mathbf{a})\mathbf{P}(\mathcal{B}|\mathbf{a})d\mathbf{a}}. \quad (2.22)$$

b. Bayesian inference for $\boldsymbol{\theta}$

Recalling that our target is to make inference about $\boldsymbol{\theta}$, through Eq. (2.22), we can start from an initial guess of $\boldsymbol{\theta}$, and include information from the observed history, to make a new guess of $\boldsymbol{\theta}$. Ideally speaking, if the observed history is infinitely long, then the new guess of $\boldsymbol{\theta}$ will be infinitely close to its true value, no matter how off the initial guess is. Here the initial and new guesses of $\boldsymbol{\theta}$ are the prior and posterior probability densities of the parameter $\boldsymbol{\theta}$, respectively noted as $\mathbf{p}(\boldsymbol{\theta})$ and $\mathbf{p}(\boldsymbol{\theta}|\mathbf{H})$, where \mathbf{H} is the observed history. For the outcome of n samples, the history \mathbf{H} can be defined as:

$$\mathbf{H} = \{h_1, \dots, h_m\}, \quad (2.23)$$

where h_i is the count of visiting box b_i within the n samples and $\sum_{i=1}^m h_i = n$. Note that the order of sampling outcome is not reflected in \mathbf{H} , because the sampling outcome is assumed to be IID. In the end, $\boldsymbol{\theta}$ can be updated by rewriting Eq. (2.22) into:

$$\mathbf{p}(\boldsymbol{\theta}|\mathbf{H}) = \frac{\mathbf{p}(\boldsymbol{\theta})\mathbf{P}(\mathbf{H}|\boldsymbol{\theta})}{\int \mathbf{p}(\boldsymbol{\theta})\mathbf{P}(\mathbf{H}|\boldsymbol{\theta})d\boldsymbol{\theta}}. \quad (2.24)$$

The probability densities/distributions in Eq. (2.24) should be clearly distinguished. The prior and posterior probability densities in the Bayesian inference refer to the probability densities of the parameter $\boldsymbol{\theta}$, and the parameter $\boldsymbol{\theta}$ itself is the probability distribution of a single sampling outcome b_{rch} , i.e. $\mathbf{P}(b_{rch} = b_i|\boldsymbol{\theta}) = \theta_i$.

To conduct Bayesian inference, $\mathbf{P}(\mathbf{H}|\boldsymbol{\theta})$ in Eq. (2.24) should also be known, which means the probability of distribution of the history \mathbf{H} conditionally on $\boldsymbol{\theta}$ (the probability distribution of a single outcome b_{rch}). For the statistical model of the sampling and simulation process, the probability of observing \mathbf{H} from the outcomes of n samples, conditionally on $\boldsymbol{\theta}$, is given by the *multinomial* distribution: $\mathbf{H}|\boldsymbol{\theta} \sim Mn(n, \boldsymbol{\theta})$. More specifically, the multinomial distribution can be given by [25]:

$$\mathbf{P}(\mathbf{H}|\boldsymbol{\theta}) = \begin{cases} \frac{n!}{\prod_{i=1}^m h_i!} \prod_{j=1}^m \theta_j^{h_j}, & \sum_{i=1}^m h_i = n; \\ 0, & \text{otherwise.} \end{cases} \quad (2.25)$$

Theoretically, now the probability density $\boldsymbol{\theta}$ can be updated from any reasonable initial guess with the sampling history, by means of Bayesian inference. However, the computation of Eq. (2.24) is usually too difficult to implement, or has very high cost even it can be implemented. To ease the computation, we choose a specific type of probability densities as the prior probability density of $\boldsymbol{\theta}$, namely *Dirichlet* probability density [26]. For the Bayesian inference of Eq. (2.24) with $\mathbf{P}(\mathbf{H}|\boldsymbol{\theta})$ being multinomial distribution, Dirichlet probability density has the property of conjugacy: if the prior probability of $\boldsymbol{\theta}$ is Dirichlet, then the posterior probability of $\boldsymbol{\theta}$ is also Dirichlet. This way, the mathematical computation can be largely eased.

Supposing the parameter $\boldsymbol{\theta}$ has a Dirichlet probability density, then its probability density is given by

$$\mathbf{p}(\boldsymbol{\theta}) = \frac{\Gamma(\sum_{i=1}^m \alpha_i)}{\prod_{i=1}^m \Gamma(\alpha_i)} \prod_{i=1}^m \theta_i^{\alpha_i - 1}, \quad (2.26)$$

which is also noted as $\boldsymbol{\theta} \sim \text{Diri}(\boldsymbol{\alpha})$, where $\boldsymbol{\alpha} = \{\alpha_1, \dots, \alpha_m\}$ ($\alpha_i > 0$ for any i) is a vector that has the same length as $\boldsymbol{\theta}$, and $\Gamma(\cdot)$ is the gamma function whose definition is

$$\Gamma(c) = \int_0^{\infty} e^{-u} u^{c-1} du. \quad (2.27)$$

Noting that Dirichlet probability density is a function of $\boldsymbol{\alpha}$, so to choose a Dirichlet probability density as the prior probability density of $\boldsymbol{\theta}$ is to choose the value of $\boldsymbol{\alpha}$. For the prior probability density of $\boldsymbol{\theta} \sim \text{Diri}(\boldsymbol{\alpha})$, the corresponding posterior probability density of $\boldsymbol{\theta}$ is $\boldsymbol{\theta}|\mathbf{H} \sim \text{Diri}(\boldsymbol{\alpha}+\mathbf{H})$, which can be derived through Bayesian inference with the sampling history \mathbf{H} [26]. The posterior probability density $\boldsymbol{\theta}|\mathbf{H} \sim \text{Diri}(\boldsymbol{\alpha} + \mathbf{H})$ can be easily obtained by replacing α_i in Eq. (2.26) with $\alpha_i + h_i$:

$$\mathbf{p}(\boldsymbol{\theta}|\mathbf{H}) = \frac{\Gamma(\sum_{i=1}^m \alpha_i + h_i)}{\prod_{i=1}^m \Gamma(\alpha_i + h_i)} \prod_{i=1}^m \theta_i^{\alpha_i + h_i - 1}. \quad (2.28)$$

Note that the parameter $\boldsymbol{\alpha}$ plus the history \mathbf{H} together determines the posterior probability density of $\boldsymbol{\theta}$, so $\boldsymbol{\alpha}$ is called *prior strength*, representing the prior assumptions for $\boldsymbol{\theta}$. Intuitively, the value of \mathbf{H} is larger for longer observation history because there will be more counts of visiting candidate boxes, and thus \mathbf{H} will be more dominant in $\boldsymbol{\alpha} + \mathbf{H}$, causing less influence of the prior strength $\boldsymbol{\alpha}$ on the posterior probability density of $\boldsymbol{\theta}$. This intuition fits the learning mechanism of Bayesian inference.

5. Computation of $\mathbf{E}[N_{new}^{(k)}]|\mathbf{H}$

By conducting Bayesian inference starting from a Dirichlet prior probability density, the posterior probability density of $\boldsymbol{\theta}$ can be obtained. But recall that our final target is to compute the posterior expectation of the number of new boxes that will be visited in the next k sampled (a.k.a. $\mathbf{E}[N_{new}^{(k)}]$) from the probability distribution of

each single sampling outcome (a.k.a. $\boldsymbol{\theta}$). If k is small, then the calculation of $\mathbf{E}[N_{new}^{(k)}]$ is trivial. Taking $k = 1$ as an example, we have:

$$\mathbf{E}[N_{new}^{(1)}] = 1 \cdot \sum_{i:b_i \in B^*} \theta_i + 0 \cdot \sum_{i:b_i \notin B^*} \theta_i = \sum_{i:b_i \in B^*} \theta_i, \quad (2.29)$$

where B^* is the set of unvisited boxes, which can be derived by removing the visited boxes in the candidate box set B . However, the direct calculation of $\mathbf{E}[N_{new}^{(k)}]$ becomes infeasible when k gets big. Fortunately, the following theorem provides a more practical solution.

Theorem 1. Given the set of unvisited boxes B^* , we have:

$$\mathbf{E}[N_{new}^{(k)}] = \sum_{i:b_i \in B^*} V_i^{(k)}, \quad (2.30)$$

where $V_i^{(k)}$ is the probability that box b_i will be visited in the next k samples at least once, and the right side of Eq. (2.30) is the sum of $V_i^{(k)}$ for all the boxes in the unvisited boxes set B^* .

The proof of Thm.1 is provided in Appendix. Thm.1 suggests that $\mathbf{E}[N_{new}^{(k)}]$ can be calculated by means of first calculating $V_i^{(k)}$, which is much easier to compute. Given $\boldsymbol{\theta}$ is the probability distribution of a single sampling outcome, the probability that box b_i will be visited in the next k samples at least once is $1 - (1 - \theta_i)^k$. So both $V_i^{(k)}$ and $\mathbf{E}[N_{new}^{(k)}]$ can be written as the functions of $\boldsymbol{\theta}$:

$$V_i^{(k)}(\boldsymbol{\theta}) = 1 - (1 - \theta_i)^k. \quad (2.31)$$

$$\mathbf{E}[N_{new}^{(k)}](\boldsymbol{\theta}) = \sum_{i:b_i \in B^*} V_i^{(k)}(\boldsymbol{\theta}) = \sum_{i:b_i \in B^*} [1 - (1 - \theta_i)^k]. \quad (2.32)$$

Finally we can compute the posterior expectation $\mathbf{E}[N_{new}^{(k)}|\mathbf{H}]$. Supposing the prior probability density of $\boldsymbol{\theta}$ is $\boldsymbol{\theta} \sim \text{Diri}(\boldsymbol{\alpha})$, and given a sampling history \mathbf{H} , then

the posterior probability density of $\boldsymbol{\theta}$ is $\boldsymbol{\theta}|\mathbf{H} \sim \text{Diri}(\boldsymbol{\alpha} + \mathbf{H})$. So the posterior expectation $\mathbf{E}[N_{new}^{(k)}|\mathbf{H}]$ can be given by:

$$\mathbf{E}[N_{new}^{(k)}|\mathbf{H}] = \int \mathbf{E}[N_{new}^{(k)}](\boldsymbol{\theta})\mathbf{p}(\boldsymbol{\theta}|\mathbf{H})d\boldsymbol{\theta}. \quad (2.33)$$

Substituting Eq. (2.31) and Eq. (2.32) into Eq. (2.33), we get:

$$\begin{aligned} \mathbf{E}[N_{new}^{(k)}|\mathbf{H}] &= \int \sum_{i:b_i \in B^*} [1 - (1 - \theta_i)^k] \mathbf{p}(\boldsymbol{\theta}|\mathbf{H}) d\boldsymbol{\theta} \\ &= \sum_{i:b_i \in B^*} (\int \mathbf{p}(\boldsymbol{\theta}|\mathbf{H}) d\boldsymbol{\theta} - \int (1 - \theta_i)^k \mathbf{p}(\boldsymbol{\theta}|\mathbf{H}) d\boldsymbol{\theta}) \end{aligned} \quad (2.34)$$

Considering $\int \mathbf{p}(\boldsymbol{\theta}|\mathbf{H}) d\boldsymbol{\theta} = 1$ since $\boldsymbol{\theta}|\mathbf{H}$ is a probability density, Eq. (2.34) can be simplified as:

$$\mathbf{E}[N_{new}^{(k)}|\mathbf{H}] = \sum_{i:b_i \in B^*} (1 - \int (1 - \theta_i)^k \mathbf{p}(\boldsymbol{\theta}|\mathbf{H}) d\boldsymbol{\theta}), \quad (2.35)$$

where

$$\int (1 - \theta_i)^k \mathbf{p}(\boldsymbol{\theta}|\mathbf{H}) d\boldsymbol{\theta} = \int_0^1 \int_0^1 \cdots \int_0^1 (1 - \theta_i)^k \mathbf{p}(\boldsymbol{\theta}|\mathbf{H}) d\theta_1 d\theta_2 \cdots d\theta_m. \quad (2.36)$$

Substituting the expression of $\mathbf{p}(\boldsymbol{\theta}|\mathbf{H})$ in Eq. (2.28) into Eq. (2.35), and after a lengthy but trivial derivation, Eq. (2.35) can be transformed into:

$$\mathbf{E}[N_{new}^{(k)}|\mathbf{H}] = \sum_{i:b_i \in B^*} \left(1 - \frac{\prod_{v=\sum_{j \neq i} \alpha_j + \sum_j h_j + k - 1}^v}{\prod_{v=\sum_j \alpha_j + \sum_j h_j + k - 1}^v} \right), \quad (2.37)$$

where $\mathbf{E}[N_{new}^{(k)}|\mathbf{H}]$ is actually a function of $\boldsymbol{\alpha}$, \mathbf{H} and B^* , because the posterior expectation of the number of new boxes that will be visited in the next k samples is determined by (1) the prior assumption about the probability distribution of the outcome of a single sample, (2) the history of sampling outcomes and (3) the set of unvisited boxes.

Based on Eq. (2.37), to further ease the computation, we choose $\boldsymbol{\alpha} = \{\alpha_0, \alpha_0, \dots\}$

with all its element the same. And considering $\sum_j h_j = n$ (n is the total number of previous samples), $\mathbf{E}[N_{new}^{(k)}|\mathbf{H}]$ can be given by:

$$\mathbf{E}[N_{new}^{(k)}|\mathbf{H}] = m_{B^*} \left[1 - \prod_{i=1}^k \frac{(m-1)\alpha_0 + n + i - 1}{m\alpha_0 + n + i - 1} \right], \quad (2.38)$$

where m_{B^*} is the number of unvisited boxes and m is the number of all the candidate boxes. Note that for a long sampling history (a large n), $\mathbf{E}[N_{new}^{(k)}|\mathbf{H}]$ is proportional to k : $\mathbf{E}[N_{new}^{(k)}|\mathbf{H}] \approx k\mathbf{E}[N_{new}^{(1)}|\mathbf{H}]$, as is illustrated in Fig. 14.

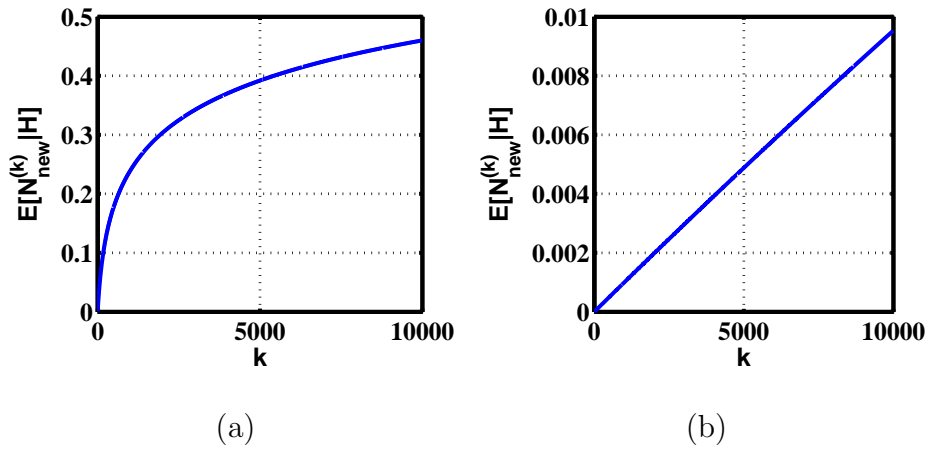


Fig. 14. $\mathbf{E}[N_{new}^{(k)}|\mathbf{H}]$ vs k ($m_{B^*}=100$, $m=1,000$, $\alpha_0=0.001$): (a) $n=100$ (b) $n=100,000$.

E. PLL lock time verification

In this section, the proposed NL-SMT verification methodology is applied to the verification of PLL lock time.

1. Charge pump PLL

The charge pump based PLL studied in this work is shown in Fig. 15. The voltage-controlled oscillator (VCO) output is fed back through a $1/N$ frequency divider. The phase/frequency difference between *ref* and *div* is detected by a phase frequency detector (PFD), whose output controls the current of charge pumps (CPs). Through negative feedback, the PLL output frequency should be finally locked to around N times the reference frequency.

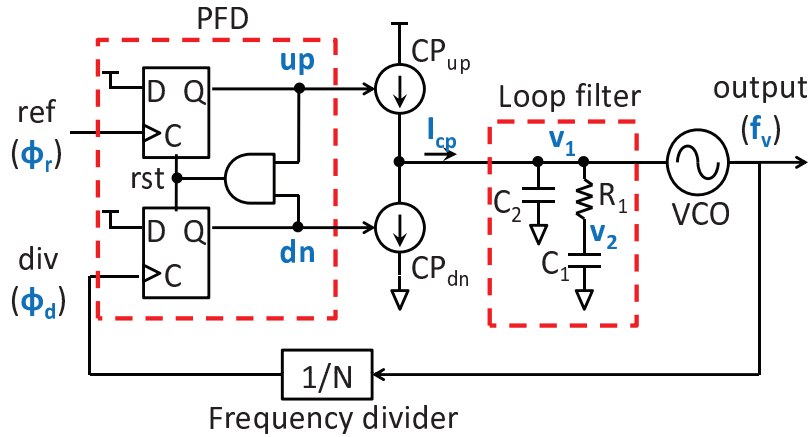


Fig. 15. Block diagram of charge pump based PLL.

A typical implementation of the PFD is composed of 2 D flip-flops (DFFs). If *ref* (*div*) is taking the lead, *ref* (*div*) first gives a rising edge. Consequently, DFF1(DFF2) output *up* (*dn*) becomes 'H', and further turns on the upper(down) CP to pump i_{cp} into(out of) the loop filter. The CP is turned off when *div* (*ref*) catches up by also giving a rising edge to reset both DFFs output to be 'L'. The corresponding timing diagram is shown in Fig. 16. In practice, the duration when both *up* and *dn* are 'H' is so short that such situation can be safely neglected. The discrete transition of PFD can be described by the hybrid automaton in Fig. 17, where ϕ_r and ϕ_d are the phase

of reference clock and the divided clock, respectively.

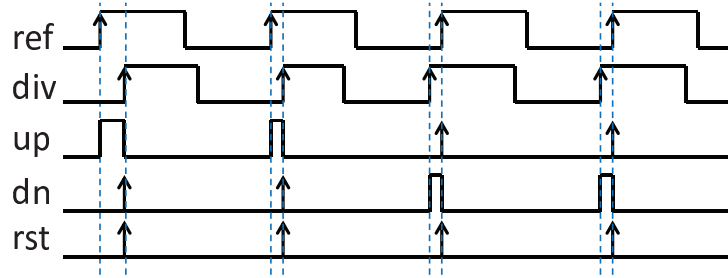


Fig. 16. Timing diagram of PFD.

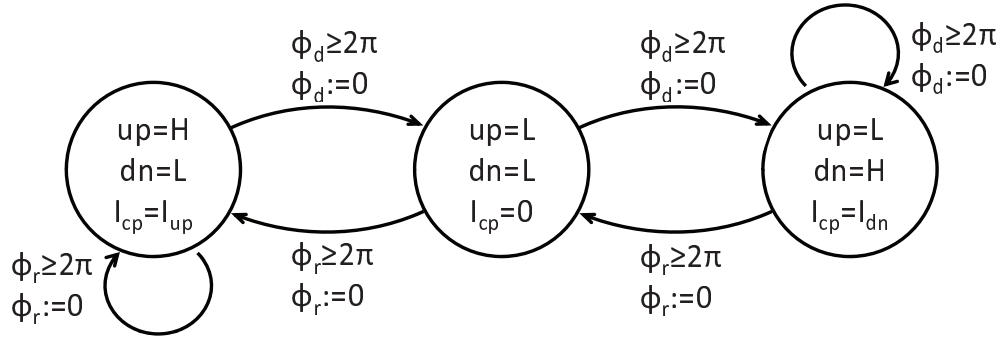


Fig. 17. PFD hybrid automaton.

The loop filter is a linear RC network including 2 internal node voltages v_1 and v_2 . v_1 is also the control signal of VCO. Nonlinearity of the VCO control curve is modeled as a polynomial. For simplicity, a 2nd-order polynomial is used:

$$f_v(v_1) = c_f^{(2)}v_1^2 + c_f^{(1)}v_1 + c_f^{(0)} \quad (2.39)$$

where f_v is VCO frequency, $c_f^{(0)}$, $c_f^{(1)}$, $c_f^{(2)}$ are polynomial coefficients.

Therefore, the continuous dynamics of PLL are:

$$\begin{bmatrix} \dot{v}_1 \\ \dot{v}_2 \end{bmatrix} = \begin{bmatrix} \frac{-1}{R_1 C_2} & \frac{1}{R_1 C_2} \\ \frac{1}{R_1 C_1} & \frac{-1}{R_1 C_1} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{C_2} \\ 0 \end{bmatrix} I_{cp} \quad (2.40)$$

$$\begin{bmatrix} \dot{\phi}_d \\ \dot{\phi}_r \end{bmatrix} = 2\pi \begin{bmatrix} f_v(v_1)/N \\ f_r \end{bmatrix} \quad (2.41)$$

where I_{cp} is the charge pump current, f_r is the reference frequency.

Table I lists the valid value/range of the discrete states and the continuous variables. Nevertheless, in the implementation, the continuous space has only 3 dimensions (v_1, v_2, ϕ_r) for simplicity, because ϕ_r has no uncertainty and independent of other continuous variables.

Table I. Valid locations of state variables

state/variable	valid value/range
v_1, v_2	$[0, 1V]$
ϕ_d, ϕ_r	$[0, 2\pi)$
{up, dn}	{H,L} {L,L} {L,H}

2. SMT constraints for lock time

The proposed NL-SMT based verification flow is applied to verify a key performance of PLL, lock time. PLL is considered as locked, if the phase difference between *div* and *ref* remains within a small interval $[\Delta\underline{\phi}_L, \Delta\overline{\phi}_L]$ for at least time t_{min} (or at least k_{min} successive time steps). A *lock time specification* requires PLL to get locked in less than time T_{max} after a change in division ratio or a phase/frequency perturbation. A specific set of SMT constraints should be formulated to check the lock time specification.

A set of auxiliary variables are introduced: k_{suc} (integer), $Lock$ (boolean), $Pass$ (boolean). k_{suc} serves as a counter that records the number of successive time steps that phase difference remains in the small interval, and its initial value $k_{suc}(0)$ should be set to 0. If the current phase difference is in $[\Delta\underline{\phi}_L, \Delta\overline{\phi}_L]$, then k_{suc} increments by one, otherwise, k_{suc} is reset of 0. Once k_{suc} reaches is k_{min} , $Lock$ is set to *true*, indicating PLL is currently locked. If $Lock$ is *true* and the current time t has not reached T_{max} , then $Pass$ is set to *true* indicating the lock time specification is successfully verified. The initial values of $Lock$ and $Pass$ are both *false*. Obviously, it is meaningless to continue reachability analysis after T_{max} . Therefore, the reachability analysis finishes either with “pass” when $Pass(t) = true$, or with “fail” for $t > T_{max}$.

To update these variables when reachability analysis moves forward, the following SMT constraints are added as part of dynamic constraints:

$$\begin{aligned} & [\mathbf{mod}_{[-\pi, \pi]}(\phi_d(t_0 + \Delta t) - \phi_r(t_0 + \Delta t)) \geq \Delta\underline{\phi}_L \\ & \wedge \mathbf{mod}_{[-\pi, \pi]}(\phi_d(t_0 + \Delta t) - \phi_r(t_0 + \Delta t)) \leq \Delta\overline{\phi}_L] \\ & \longrightarrow k_{suc}(t_0 + \Delta t) = k_{suc}(t_0) + 1, \end{aligned} \quad (2.42)$$

$$\begin{aligned} & [\mathbf{mod}_{[-\pi, \pi]}(\phi_d(t_0 + \Delta t) - \phi_r(t_0 + \Delta t)) < \Delta\underline{\phi}_L \\ & \vee \mathbf{mod}_{[-\pi, \pi]}(\phi_d(t_0 + \Delta t) - \phi_r(t_0 + \Delta t)) > \Delta\overline{\phi}_L] \\ & \longrightarrow k_{suc}(t_0 + \Delta t) = 0, \end{aligned} \quad (2.43)$$

$$k_{suc}(t_0 + \Delta t) \geq k_{min} \iff Lock(t_0 + \Delta t), \quad (2.44)$$

$$Lock(t_0 + \Delta t) \wedge (t_0 + \Delta t < T_{max}) \iff Pass(t_0 + \Delta t), \quad (2.45)$$

where $\mathbf{mod}_{[-\pi, \pi]}$ is modulo function with its output ranges from $-\pi$ to π , $A \leftrightarrow B$ stands for $(A \rightarrow B) \wedge (B \rightarrow A)$, and $A \rightarrow B$ is equivalent to $\neg A \vee B$.

3. Fast forwarding

Although the hybrid automaton of charge pump PLL has only 3 discrete states and 4 continuous variables, its reachability analysis is not trivial for existing verification techniques [21]. Because most reachability analysis techniques focus on improving the efficiency of handling continuous dynamics, while for lock time verification, hundreds and thousands of discrete switches will occur before charge pump PLL gets locked, suggesting a large number of splitting the reachable space. The situation gets even more challenging when nonlinear dynamics are considered. To ease the computational complexity, a fast forwarding technique is exploited particularly for the reachability analysis of charge pump PLL.

Notice that continuous variables can be analytically expressed between any two discrete switches. For example, the analytical forms of v_1 and v_2 can be obtained by solving loop filter dynamics under a fixed I_{cp} :

$$\begin{aligned}
 v_1(t) = & \frac{I_{cp}(t-t_0)+C_2v_1(t_0)+C_1v_2(t_0)}{C_1+C_2} \\
 & + \frac{C_1(v_1(t_0)-v_2(t_0))}{C_1+C_2} e^{(\frac{-1}{R_1C_1} + \frac{-1}{R_1C_2})(t-t_0)} \\
 & - \frac{I_{cp}R_1C_1^2}{(C_1+C_2)^2} (e^{(\frac{-1}{R_1C_1} + \frac{-1}{R_1C_2})(t-t_0)} - 1),
 \end{aligned} \tag{2.46}$$

$$\begin{aligned}
 v_1(t) = & \frac{I_{cp}(t-t_0)+C_2v_1(t_0)+C_1v_2(t_0)}{C_1+C_2} \\
 & - \frac{C_2(v_1(t_0)-v_2(t_0))}{C_1+C_2} e^{(\frac{-1}{R_1C_1} + \frac{-1}{R_1C_2})(t-t_0)} \\
 & + \frac{I_{cp}R_1C_1C_2}{(C_1+C_2)^2} (e^{(\frac{-1}{R_1C_1} + \frac{-1}{R_1C_2})(t-t_0)} - 1),
 \end{aligned} \tag{2.47}$$

where t_0 is the starting time of the current continuous behavior. The analytical form of ϕ_d can also be derived using symbolic analysis engine:

$$\begin{aligned}
 \phi_d(t) = F_{\phi_d}(& t, v_1(t_0), v_2(t_0), \phi_d(t_0), \\
 & I_{cp}, R_1, C_1, C_2, c_f^{(0)}, c_f^{(1)}, c_f^{(2)}),
 \end{aligned} \tag{2.48}$$

which is too long to be listed here.

Based on this observation, instead of moving forward with a small time step, fast forwarding can be applied over one reference period T_{ref} , as illustrated in Fig. 18. Each fast forwarding starts from right after a discrete switch due to ϕ_r crossing 2π (named as ϕ_r switch) and stops when the next ϕ_r switch is finished, covering a duration of T_{ref} . Discrete switches due to ϕ_d crossing 2π (named as ϕ_d switches) are scattered between 2 neighboring ϕ_r switches. Since analytical forms of the continuous variables do not exist across discrete switch, symbolic analysis is only applied between discrete switches. Several segments of symbolic analysis might be carried out during a fast forwarding, depending on the number of ϕ_d switches. The ending values of continuous variables in one segment become the starting values for the next segment.

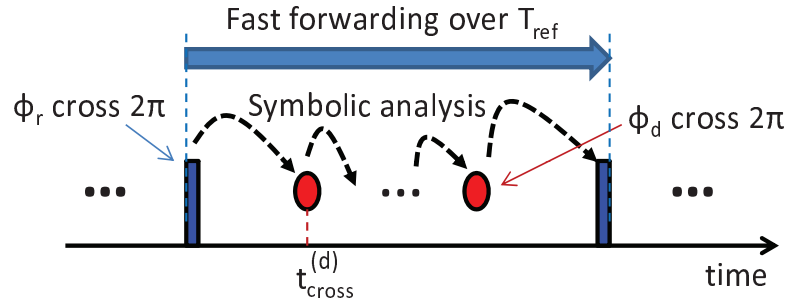


Fig. 18. Timing diagram of fast forwarding.

While ϕ_r switches are synchronous, ϕ_d switches are asynchronous. The number and moments of asynchronous ϕ_d switches need to be explored on-the-fly. The exploration flow is shown in Fig. 19. The period starts right after the last switch of ϕ_r , and the initial time is saved as t_i and updated t_0 with t_i . At the beginning of the loop, it is checked if there are any ϕ_d switches before the end of the current period. This is done by substituting $t = t_i + T_{ref}$ into Eq. (2.48) and checking if $\phi_d(t)$ is larger than 2π . $\phi_d(t) < 2\pi$ means no ϕ_d switches will happen in the rest of this period. In this case, the flow will jump out of the loop and fast forward to $t_i + T_{ref}$,

using Eq. (2.46)-(2.48), and finally a ϕ_r switch is applied. On the other hand, if the checking result is $\phi_d(t) \geq 2\pi$, then the flow will find out the moment of the ϕ_d switch by solving t in Eq. (2.48) with $\phi_d(t) = 2\pi$ and the solution is saved as $t_{cross}^{(d)}$. Note that this solving process will be implicitly handled by the NL-SMT solver. Then the flow will fast forward to $t_{cross}^{(d)}$ and apply a ϕ_d switch. After updating t_0 with $t_{cross}^{(d)}$, the loop is restarted from the beginning. Note that in this flow, the value of I_{cp} is always updated according to the current values of up and dn .

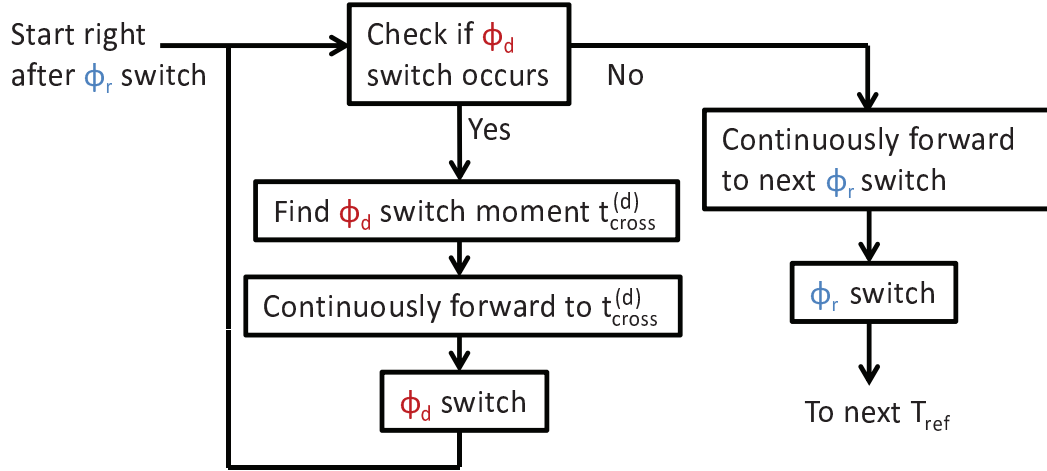


Fig. 19. Flow of fast forwarding.

The above flow should be converted into SMT constraints, so that the dynamics of fast forwarding can fit into the proposed NL-SMT-based verification framework. A methodology of the conversion is given in Fig. 20. First, a *loop* statement is unrolled into a tree of *if-else* statements. Given an upper bound \bar{R} of rounds within the loop, the if-else tree can be cut down to a length of \bar{R} . In our case, \bar{R} should be no less than the number of ϕ_d switches. In the proposed simulation-assisted NL-SMT approach, the value of \bar{R} can be estimated from simulation results. To keep conservativeness, a guard band could be added. Next, “if A then B , else C ” statement is transformed

into constraint $(A \rightarrow B) \vee (\neg A \rightarrow C)$. This transformation is applied to all the “if A then B , else C ” statements, and the resulted constraints are conjuncted together. This way the SMT constraints for the fast forwarding dynamics are generated.

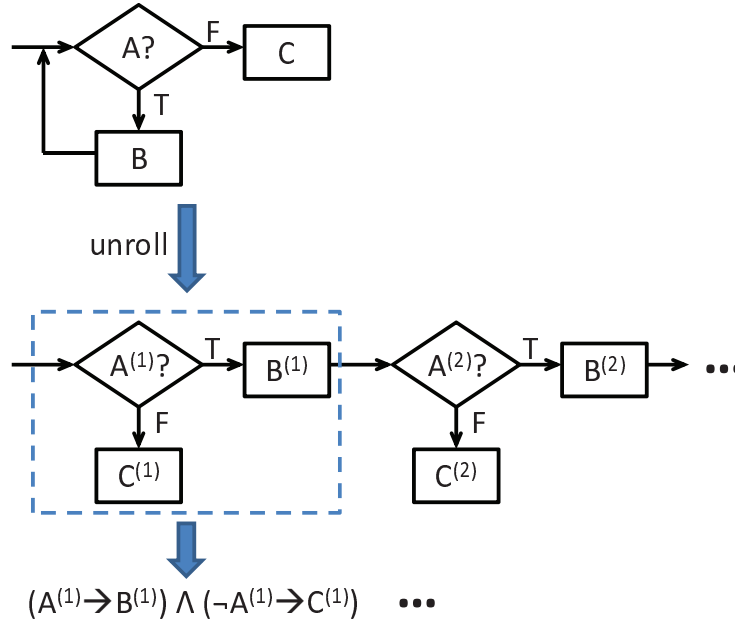


Fig. 20. From flow to constraints.

F. Experimental results

The simulation-assisted NL-SMT based reachability analysis is applied to check the *frequency hopping* of PLL: to change the value of feedback frequency division N and then verify PLL lock time. Given the reference clock frequency is 10MHz, if N is changed from 36 to 100, the VCO frequency is expected to start from 360MHz and finally get locked to 1GHz. The initial condition for v_1 and v_2 is set to $v_1 \in [0.7, 0.71]$, $v_2 \in [0.6, 0.61]$ to model their uncertainty. All the following experimental results are obtained using a 4-core Intel CPU Q9450 processor running at 2.66 GHz

with 8 GB memory.

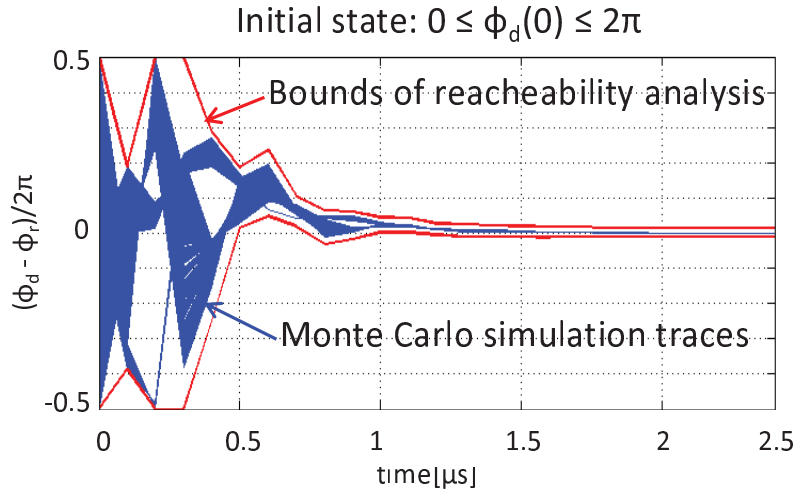


Fig. 21. Reachable space of ϕ_d computed by the proposed reachability analysis.

Starting from $\phi_d \in [0, 2\pi)$, Fig.21 shows the reachable space of ϕ_d obtained by reachability analysis, and as a reference the results of a 1000-sample Monte Carlo simulation that starts randomly from the initial state space. If the lock condition is set to $|\phi_d - \phi_r| \leq 0.01 \times 2\pi$ lasting for at least 5 reference clock cycles, then PLL is verified to get locked before $2.1\mu s$.

To demonstrate the effectiveness of Bayesian inference technique, as a reference, we first run the reachability analysis with the static *Stop* condition defined in Eq.(2.15). Fig. 22 lists the total runtime contribution from simulation and NL-SMT as well as the number of NL-SMT solver invoking, versus different sampling densities ρ . It can be seen the minimum runtime is around 1000 seconds. On the other hand, the runtime of reachability analysis with the dynamic *Stop* condition is 873 seconds, indicating its effectiveness in online learning and reduction of verification runtime. Also note that for the Bayesian-based case, the number of invoking NL-SMT solver

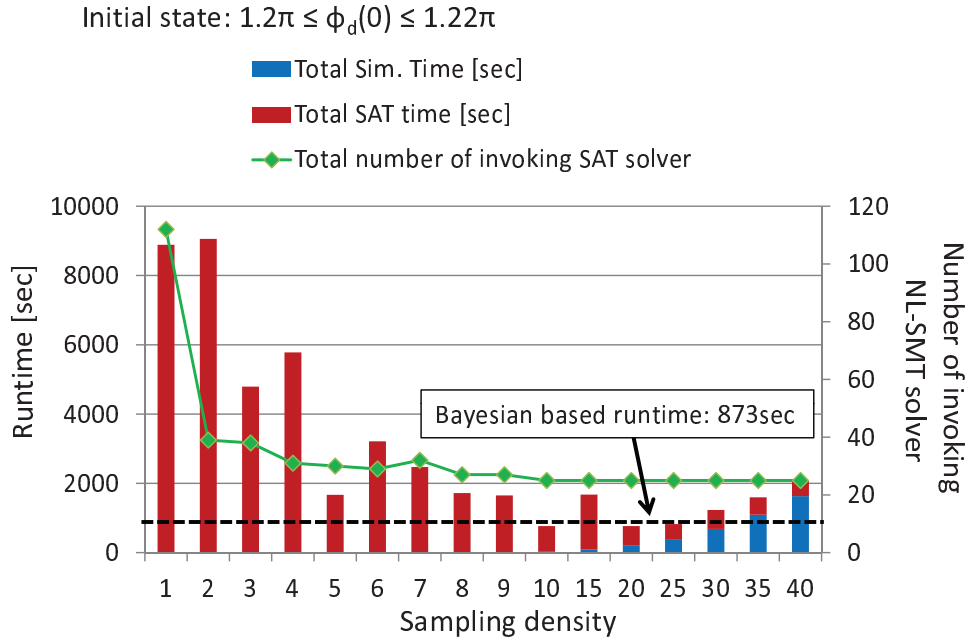


Fig. 22. Runtime vs sampling density.

is 26. Considering the verification time is 25 reference cycles, most reachable space is discovered by the Bayesian-based random simulation, and only one box is explored by the NL-SMT solver. For the dynamic *Stop* condition based case, the *Stop* condition is checked for every 1,000 samples of simulations ($k = 1,000$), and Eq. (2.38) is used for computing $\mathbf{E}[N_{new}^{(k)}|\mathbf{H}]$. Though can be updated on the fly, the runtime of one simulation run τ_{sim} and the runtime of invoking NL-SMT solver once τ_{smt} are estimated as fixed values for simplicity. According to the average runtime of 10,000 random samples of simulation and the average runtime of 100 times of invoking NL-SMT solver for the PLL case, the ratio between τ_{sim} and τ_{smt} is set to be $\kappa = 1.4 \times 10^{-7}$.

Fig. 23 compares the numbers of samples of simulation per time step between dynamic and static *Stop* conditions over the time. It can be seen although at first the numbers of samples of simulation are the same for the two *Stop* conditions. As time goes on, PLL starts to get locked and all trajectories start to converge to a confined

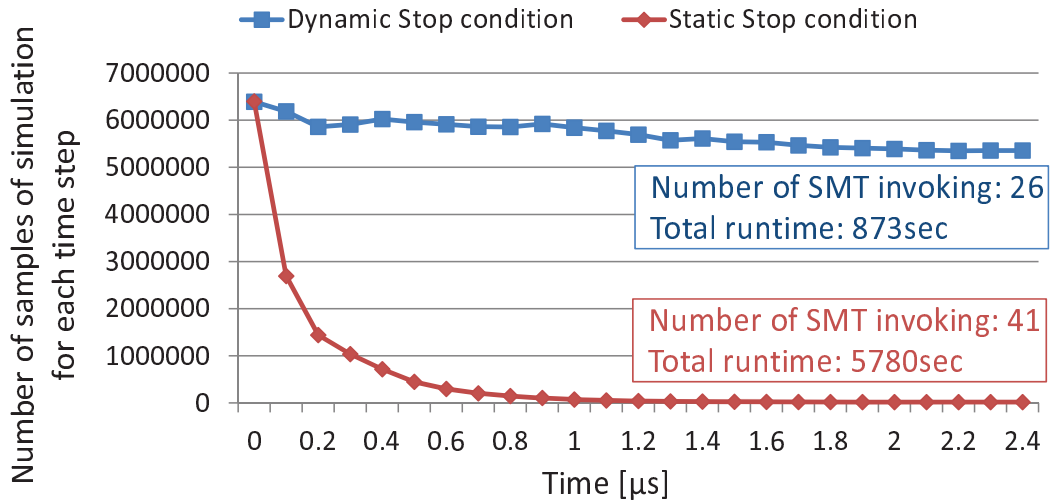


Fig. 23. The numbers of samples of simulations for dynamic and static *Stop* conditions.

region of the state space. Corresponding to the shrinking reachable space, the static *Stop* condition blindly chooses to use less number of samples of simulation and to stop simulation earlier. In contrast, the number of samples of simulation only slightly decreases for the dynamic *Stop* condition, based on Bayesian inference that learns from the sampling history. Finally, the total numbers of SMT invoking are 26 for the dynamic *Stop* condition, versus 41 for the static *Stop* condition. And correspondingly the dynamic *Stop* condition leads to a total runtime (including simulation and SMT) of 873 seconds, while due to the high runtime cost of SMT solving the total runtime is 5,780 seconds for the static *Stop* condition, which demonstrates the intelligence of the dynamic *Stop* condition for achieving the minimum total runtime.

To find out the influence of k , the number of samples of simulation between two dynamic *Stop* condition checks, Fig.24 compares the numbers of samples of simulation per time step over the time for different values of k , from 1 to 10^5 . The corresponding total runtime ranges from 864 seconds to 891 seconds. Although a smaller k means the dynamic *Stop* condition should be checked more often, Fig.24 shows the variation

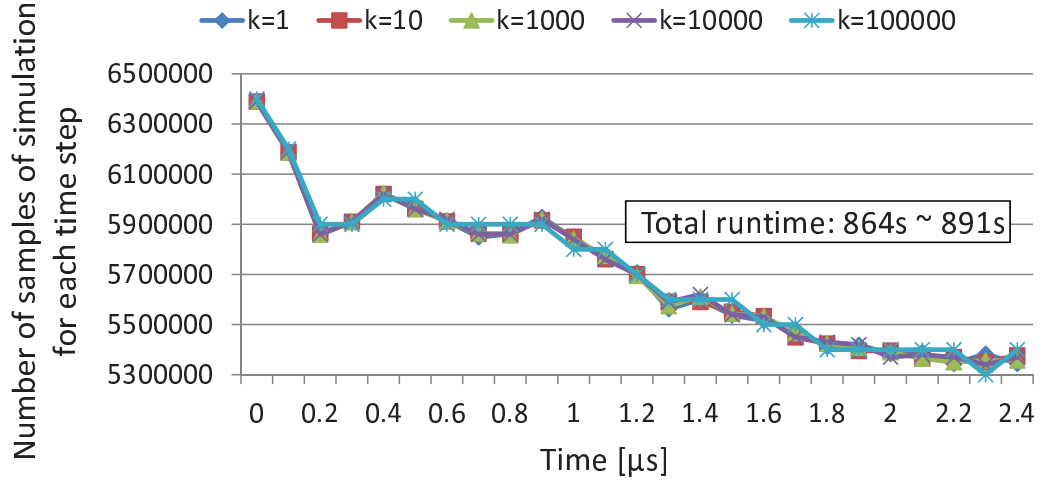


Fig. 24. The numbers of samples of simulations for dynamic *Stop* condition with different k .

of k has little impact on the number of samples of simulation and the total runtime, meaning the dynamic *Stop* condition is insensitive to k . Referring back to Eq. (2.16) that mathematically expresses the dynamic *Stop* condition, its insensitivity to k suggests that given the same sampling history \mathbf{H} , the posterior expected number of new boxes that will be discovered in the next k samples of simulation ($\mathbf{E}[N_{new}^{(k)}|\mathbf{H}]$) is proportional to k . This proportional relationship fits our earlier observation that $\mathbf{E}[N_{new}^{(k)}|\mathbf{H}] \approx k\mathbf{E}[N_{new}^{(1)}|\mathbf{H}]$ for a long sampling history. On the other hand, if k is so large that it is comparable to the number of samples of simulation per time step, then the stop moment of simulation may be delayed unnecessarily, leading to wasteful simulation. In Fig. 24, the largest $k = 10^5$ is still much smaller than the number of samples of simulation per time step, ranging from 5.3×10^6 to 6.4×10^6 , and thus little negative impact can be observed. Therefore, as a general strategy of choosing k , a smaller k is preferred as long as the computation of Eq. (2.38) is much less costly than k samples of simulations, meaning it does no harm to check the dynamic *Stop* conditions as often as possible if the checking cost is affordable.

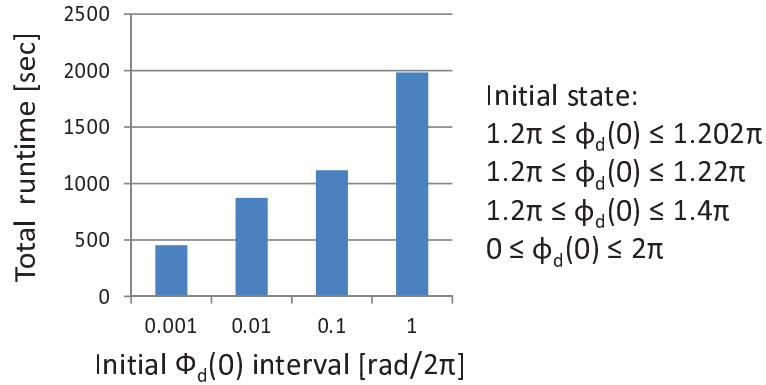


Fig. 25. Runtime vs initial space (with simulation assistance and Bayesian inference).

Fig. 25 shows the runtime scalability with respect to the size of initial state space, with the Bayesian inference based simulation stopping scheme. Initial state space increases with the initial uncertainty of the divided clock phase ϕ_d . Note that the reachability analysis is performed for $2.5\mu\text{s}$ (25 reference clock cycles) and PLL design parameters are $C_1=2.5\text{pF}$, $C_2=0.6\text{pF}$ and $R_1=160\text{kohm}$. In contrast to the runtime results in Fig.25, the runtime of reachability analysis using basic NL-SMT flow (without simulation assistance) is more than 5 hours even for the smallest initial state space.

G. Summary

A nonlinear SMT based approach is presented for verification of dynamic properties of nonlinear AMS designs. Towards enabling practical NL-SMT based AMS verification, simulation-assisted SAT and Bayesian inference are leveraged to accelerate the verification. The feasibility and efficacy of the proposed methodology are demonstrated on conservative verification of the lock time of a charge pump based PLL.

H. Appendix

For clarity, Thm.1 is rewritten here with all the conditions clearly stated. Given a set of boxes $B = \{b_1, \dots, b_M\}$, we randomly visit a box b_{rch} in B . Now we will repeat the random visit for k times, and each time the probability distribution is a vector $\boldsymbol{\theta} = \{\theta_1, \dots, \theta_M\}$, where θ_i is the probability of visiting box b_i . Also each visit is independent from other visits. Therefore, the sequence of visited boxes \mathbf{b}_{rch} is a sequence of IID random variables. And also given that B^* is a subset of B , then the theorem states that:

$$\mathbf{E}[N_{B^*}^{(k)}] = \sum_{i:b_i \in B^*} V_i^{(k)}, \quad (2.49)$$

where $\mathbf{E}[N_{B^*}^{(k)}]$ is the expected number of boxes in B^* that will be visited in k visits, i.e. the expected number of boxes in \mathbf{b}_{rch} , without counting the repeated appearance of previously visited boxes; and $V_i^{(k)}$ is the probability that box b_i is visited in k visits at least once, so $\sum_{i:b_i \in B^*} V_i^{(k)}$ means the sum of $V_i^{(k)}$ for all the boxes b_i in B^* .

Proof. The theorem can be proven by induction.

★ For $k = 1$, we have:

$$\mathbf{E}[N_{B^*}^{(1)}] = \sum_i \theta_i N_{B^*}^{(1)}(i), \quad (2.50)$$

where $N_{B^*}^{(1)}(i)$ is the number of visited boxes that are in B^* , if the first and the only visited box is b_i . Obviously, $N_{B^*}^{(1)}(i)$ is either 1 if $b_i \in B^*$, or 0 if $b_i \notin B^*$:

$$N_{B^*}^{(1)}(i) = \begin{cases} 1, & b_i \in B^*; \\ 0, & b_i \notin B^*. \end{cases} \quad (2.51)$$

If we rewrite Eq.(2.50) as:

$$\mathbf{E}[N_{B^*}^{(1)}] = \sum_{i:b_i \in B^*} \theta_i N_{B^*}^{(1)}(i) + \sum_{i:b_i \notin B^*} \theta_i N_{B^*}^{(1)}(i), \quad (2.52)$$

then by substituting Eq.(2.51) into it, we have:

$$\mathbf{E}[N_{B^*}^{(1)}] = \sum_{i:b_i \in B^*} \theta_i \cdot 1 + \sum_{i:b_i \notin B^*} \theta_i \cdot 0 = \sum_{i:b_i \in B^*} \theta_i. \quad (2.53)$$

Also considering $V_i^{(1)}$ is the same as the probability that box b_i is hit by 1 visit, which is θ_i , we can rewrite Eq.(2.53) into:

$$\mathbf{E}[N_{B^*}^{(1)}] = \sum_{i:b_i \in B^*} V_i^{(1)}, \quad (2.54)$$

which states that the theorem is proven for $k = 1$.

★ Now we need to prove that: if for k visits the hypothesis $\mathbf{E}[N_{B^*}^{(k)}] = \sum_{i:b_i \in B^*} V_i^{(k)}$ stands true, then for $k + 1$ visits, $\mathbf{E}[N_{B^*}^{(k+1)}] = \sum_{i:b_i \in B^*} V_i^{(k+1)}$ is also true.

First, we split $\mathbf{E}[N_{B^*}^{(k+1)}]$ into:

$$\mathbf{E}[N_{B^*}^{(k+1)}] = \mathbf{E}[N_{B^*}^{(k)}] + \mathbf{E}[\Delta N_{B^*}^{(k+1)}], \quad (2.55)$$

where $\Delta N_{B^*}^{(k+1)} = N_{B^*}^{(k+1)} - N_{B^*}^{(k)}$ is the increment of the number of visited boxes that are in B^* due to the $(k + 1)$ th visit, recalling that $N_{B^*}^{(k)}$ and $N_{B^*}^{(k+1)}$ are the numbers of visited boxes that are in B^* by k visits and $k + 1$ visits, respectively. Substituting the hypothesis for k visits into Eq.(2.55), we have:

$$\mathbf{E}[N_{B^*}^{(k+1)}] = \sum_{i:b_i \in B^*} V_i^{(k)} + \mathbf{E}[\Delta N_{B^*}^{(k+1)}]. \quad (2.56)$$

The 2nd term of the right side of Eq.(2.56) can be expressed as:

$$\mathbf{E}[\Delta N_{B^*}^{(k+1)}] = \sum_{i_1, \dots, i_{k+1}} \theta_{i_1} \cdots \theta_{i_{k+1}} \Delta N_{B^*}^{(k+1)}(i_1, \dots, i_{k+1}), \quad (2.57)$$

where $\Delta N_{B^*}^{(k+1)}(i_1, \dots, i_{k+1})$ is the increment of the number of visited boxes that are in B^* due to the $(k+1)$ th visit, if the $k+1$ visits hit $b_{i_1}, \dots, b_{i_{k+1}}$ in order. Obviously, $\Delta N_{B^*}^{(k+1)}(i_1, \dots, i_{k+1})$ can only be 0 or 1. Depending on the result of $b_{i_1}, \dots, b_{i_{k+1}}$, however, there are three possible situations. The first is that $b_{i_{k+1}}$ is in B^* and $b_{i_{k+1}}$ is not hit by the previous k visits. In this case, $\Delta N_{B^*}^{(k+1)}(i_1, \dots, i_{k+1}) = 1$. The second situation is that $b_{i_{k+1}}$ is in B^* but $b_{i_{k+1}}$ is already visited in the previous k visits. In this case, $\Delta N_{B^*}^{(k+1)}(i_1, \dots, i_{k+1}) = 0$. The last situation is that $b_{i_{k+1}}$ is not in B^* at all, and thus $\Delta N_{B^*}^{(k+1)}(i_1, \dots, i_{k+1}) = 0$. These three situations are summarized by:

$$\Delta N_{B^*}^{(k+1)}(i_1, \dots, i_{k+1}) = \begin{cases} 1, & b_{i_{k+1}} \in B^*, i_{k+1} \notin \{i_1, \dots, i_k\}; \\ 0, & b_{i_{k+1}} \in B^*, i_{k+1} \in \{i_1, \dots, i_k\}; \\ 0, & b_{i_{k+1}} \notin B^*. \end{cases} \quad (2.58)$$

Corresponding to the three situations, Eq.(2.57) can be rewritten as:

$$\begin{aligned} & \mathbf{E}[\Delta N_{B^*}^{(k+1)}] \\ &= \sum_{i_{k+1}} \theta_{i_{k+1}} \sum_{i_1, \dots, i_k} \theta_{i_1} \cdots \theta_{i_k} \Delta N_{B^*}^{(k+1)}(i_1, \dots, i_{k+1}) \\ &= \sum_{i_{k+1}: b_{i_{k+1}} \in B^*} \theta_{i_{k+1}} \sum_{i_1, \dots, i_k} \theta_{i_1} \cdots \theta_{i_k} \Delta N_{B^*}^{(k+1)}(i_1, \dots, i_{k+1}) \\ & \quad + \sum_{i_{k+1}: b_{i_{k+1}} \notin B^*} \theta_{i_{k+1}} \sum_{i_1, \dots, i_k} \theta_{i_1} \cdots \theta_{i_k} \Delta N_{B^*}^{(k+1)}(i_1, \dots, i_{k+1}) \\ &= \sum_{i_{k+1}: b_{i_{k+1}} \in B^*} \theta_{i_{k+1}} \sum_{i_1, \dots, i_k: \{i_1, \dots, i_k\} \not\ni i_{k+1}} \theta_{i_1} \cdots \theta_{i_k} \Delta N_{B^*}^{(k+1)}(i_1, \dots, i_{k+1}) \\ & \quad + \sum_{i_{k+1}: b_{i_{k+1}} \in B^*} \theta_{i_{k+1}} \sum_{i_1, \dots, i_k: \{i_1, \dots, i_k\} \ni i_{k+1}} \theta_{i_1} \cdots \theta_{i_k} \Delta N_{B^*}^{(k+1)}(i_1, \dots, i_{k+1}) \\ & \quad + \sum_{i_{k+1}: b_{i_{k+1}} \notin B^*} \theta_{i_{k+1}} \sum_{i_1, \dots, i_k} \theta_{i_1} \cdots \theta_{i_k} \Delta N_{B^*}^{(k+1)}(i_1, \dots, i_{k+1}), \end{aligned} \quad (2.59)$$

then by substituting Eq.(2.58) into it, we have:

$$\mathbf{E}[\Delta N_{B^*}^{(k+1)}] = \sum_{i_{k+1}: b_{i_{k+1}} \in B^*} \theta_{i_{k+1}} \sum_{i_1, \dots, i_k: \{i_1, \dots, i_k\} \not\ni i_{k+1}} \theta_{i_1} \cdots \theta_{i_k}. \quad (2.60)$$

Also note that $\sum_{i_1, \dots, i_k: \{i_1, \dots, i_k\} \not\ni i_{k+1}} \theta_{i_1} \cdots \theta_{i_k}$ is the probability that box $b_{i_{k+1}}$ is not

visited by the previous k visits. Eq.(2.60) can be changed into:

$$\begin{aligned}\mathbf{E}[\Delta N_{B^*}^{(k+1)}] &= \sum_{i_{k+1}:b_{i_{k+1}} \in B^*} \theta_{i_{k+1}} \mathbf{P}(i_{k+1} \notin \{i_1, \dots, i_k\}) \\ &= \sum_{i_{k+1}:b_{i_{k+1}} \in B^*} \theta_{i_{k+1}} (1 - \mathbf{P}(i_{k+1} \in \{i_1, \dots, i_k\})),\end{aligned}\quad (2.61)$$

where $\mathbf{P}(\mathbf{A})$ is the probability of event \mathbf{A} , and it is obvious that the sum of the probabilities of $i_{k+1} \in \{i_1, \dots, i_k\}$ and $i_{k+1} \notin \{i_1, \dots, i_k\}$ should be 1. Because $\mathbf{P}(i_{k+1} \in \{i_1, \dots, i_k\})$ is the probability that box $b_{i_{k+1}}$ is visited in the previous k visits at least once, which is $V_{i_{k+1}}^{(k)}$, so Eq.(2.61) can be further changed into:

$$\mathbf{E}[\Delta N_{B^*}^{(k+1)}] = \sum_{i_{k+1}:b_{i_{k+1}} \in B^*} \theta_{i_{k+1}} (1 - V_{i_{k+1}}^{(k)}). \quad (2.62)$$

Substituting the above expression of $\mathbf{E}[\Delta N_{B^*}^{(k+1)}]$ back into Eq.(2.56), we can get:

$$\mathbf{E}[N_{B^*}^{(k+1)}] = \sum_{i:b_i \in B^*} [V_i^{(k)} + \theta_i(1 - V_i^{(k)})]. \quad (2.63)$$

On the other hand, $V_i^{(k+1)}$, the probability that box b_i is visited in the $k+1$ visits at least once, can be expressed by conditional probabilities as:

$$\begin{aligned}V_i^{(k+1)} &= \mathbf{P}(b_{i_{k+1}} = b_i | \{b_{i_1}, \dots, b_{i_k}\} \not\ni b_i) \cdot \mathbf{P}(\{b_{i_1}, \dots, b_{i_k}\} \not\ni b_i) \\ &\quad + \mathbf{P}(b_{i_{k+1}} \in B | \{b_{i_1}, \dots, b_{i_k}\} \ni b_i) \cdot \mathbf{P}(\{b_{i_1}, \dots, b_{i_k}\} \ni b_i),\end{aligned}\quad (2.64)$$

which is based on the observation that, the event that box b_i is hit by the $k+1$ visits at least once means either (1) $b_{i_{k+1}}$ (the box that is hit by the $(k+1)$ th visit) is b_i if the previous k visits has not hit b_i ; or (2) $b_{i_{k+1}}$ can be any box in B if the previous k visits has already hit b_i . Since the sequence of visited boxes is IID, we have:

$$\mathbf{P}(b_{i_{k+1}} = b_i | \{b_{i_1}, \dots, b_{i_k}\} \not\ni b_i) = \mathbf{P}(b_{i_{k+1}} = b_i) = \theta_i \quad (2.65)$$

and

$$\mathbf{P}(b_{i_{k+1}} \in B | \{b_{i_1}, \dots, b_{i_k}\} \ni b_i) = \mathbf{P}(b_{i_{k+1}} \in B) = 1, \quad (2.66)$$

Substituting them back into Eq.(2.64), also considering the probabilities that the previous k visits has already hit b_i and that the previous k visits has not hit b_i are $V_i^{(k)}$ and $1 - V_i^{(k)}$, respectively, we have:

$$V_i^{(k+1)} = V_i^{(k)} + \theta_i(1 - V_i^{(k)}). \quad (2.67)$$

Finally, by substituting Eq.(2.67) back into Eq.(2.63), we have:

$$\mathbf{E}[N_{B^*}^{(k+1)}] = \sum_{i:b_i \in B^*} V_i^{(k+1)}, \quad (2.68)$$

and therefore the theorem is proven for $k + 1$ visits.

□

CHAPTER III

HIGH-RESOLUTION ON-CHIP JITTER MEASUREMENT

In-situ test takes a different approach from verification to enhance the design robustness, by integrating testability into hardware designs. This chapter focuses on the in-situ test of jitter, which is needed by today's high-speed high-precision digital and analog ICs. Note that the proposed in-situ testing structure can be applied to characterize different types of jitter-related analog performance. A preliminary work of the proposed structure is published in [27].

A. Introduction

Timing precision, measured in the form of jitter, is extremely crucial for a broad range of high-speed high-precision digital and analog ICs. Jitter is one of the most important performances for the clock data recovery (CDR) in I/O circuitry as well as the clock generation in high-speed digital signal processing circuitry, where phase locked loops (PLLs) or delay locked loops (DLLs) are employed [11]. As an example, today's on-chip serial-links operate at a data rate of multi-Gb/s [12]. The clock jitter in serial-link transceivers degrades the transmitted and received data margin. It may also cause the received data to fall outside the design boundary. Moreover, from the perspective of RF applications, jitter performance is also a key concern because clock jitter will turn into the phase noise of wireless signals [1]. Hence, high-resolution jitter characterization is essential.

Traditionally, jitter is measured using external testing equipment. State-of-the-art time interval analyzers (TIA) provide femto-second accuracy. However, not only are they expensive, but also their achievable resolution is limited by the noise injected along the on-chip to off-chip signal propagation path. In this regard, low cost high-

resolution on-chip solutions are particularly appealing. More importantly, in-situ jitter characterization allows online monitoring of design performance, and further provides the possibility of self diagnosis and healing in the events of jitter incurred failures.

Nevertheless, providing low-cost high-resolution on-chip jitter characterization remains a significant challenge to date. In [28] [29], an analog based approach is developed to convert the jittered time duration into the voltage of a capacitor, and an analog-to-digital convertor (ADC) is used to convert the voltage into digital code. However, due to its analog operation, this approach is highly sensitive to process variation and vulnerable to noise interference like power supply noise. In contrast, delay line based technique directly converts the jittered time duration into digital code, by means of a line of delay cells and D flip-flops (DFFs) [30]. Its time resolution is determined by the delay of a single delay cell. To further improve the resolution, Vernier delay lines (VDL) techniques employs two delay lines, whose cell delays are slightly different from each other and the delay difference is its time resolution [31]. The drawback of VDL is its small measuring range. A similar technique, Vernier ring oscillator (VRO) employs two ring oscillators, whose oscillation periods are slightly different from each other [32] [33]. Compared with VDL, VRO provides larger measuring range, but its sampling frequency is much lower. Moreover, a sophisticated calibration scheme is usually required for Vernier-style techniques to compensate the mismatch between cell delays. Recently, gated ring oscillator (GRO) has been proposed to achieve an effective resolution higher than a single delay cell, through over-sampling and quantization noise shaping [34]. Merging the principles of VRO and GRO, [35] developed a technique named Vernier GRO which involves two Vernier-style GROs. And the already reduced quantization noise of the VRO is further shaped by the GRO operation. However, its highest sampling frequency is

largely limited by the VRO sampling frequency, which makes it not suitable for high speed applications.

In this work, a novel built-in jitter measurement technique is presented that well fits the need for the jitter measurement of high-speed signals. The proposed structure is composed of a GRO, a delay line and a digital signal processing (DSP) unit. The GRO can provide a coarse measurement of the input time duration, and the resolution is equal to the stage delay of the GRO. With the assistance of the delay line, the time residue which is not converted in the coarse measurement will be measured with a finer resolution. The cell delay of the delay line is slightly different from the stage delay of the GRO. The combination of the delay line and the GRO forms a Vernier-style structure, which converts the time residue in the coarse measurement into digital code with a resolution much finer than the GRO stage delay. On top on that, the GRO's benefit of quantization noise shaping is inherited to the fine measurement result, and therefore an even higher effective resolution can be achieved. Finally, the DSP unit is responsible of decoding the outputs of the GRO and the delay line into digital codes, as well as converting the fine code into the fractional part of the coarse code, where the fine resolution is implicitly calibrated with respect to the coarse resolution on the fly. Moreover, pipelined structures are applied in the DSP unit to enable a high sampling frequency.

Note that the proposed technique is different from the Vernier GRO in [35]. Because the fine measurement is only applied for the time residue from the coarse level in the two-level measurement structure, the proposed technique can achieve a sampling frequency much higher than Vernier GRO, which makes it a better candidate for high speed applications. Moreover, the proposed structure has a high tolerance for the typical delay mismatch of 90nm technologies, which is demonstrated by behavioral model simulations.

This chapter proceeds as follows. Sec.B introduces the proposed architecture of built-in jitter measurement technique. Sec.C describes the circuit implementation. In Sec.D, the experimental results for the proposed technique are presented using a commercial 90nm CMOS process, and the influence of delay mismatch is analyzed. The final section draws a brief summary.

B. Proposed structure

The block diagram of the proposed on-chip jitter measurement scheme is shown in Fig. 26. Its target is to convert the input time into digital code. The input time is given as the time difference between the rising edges of *START* and *STOP*.

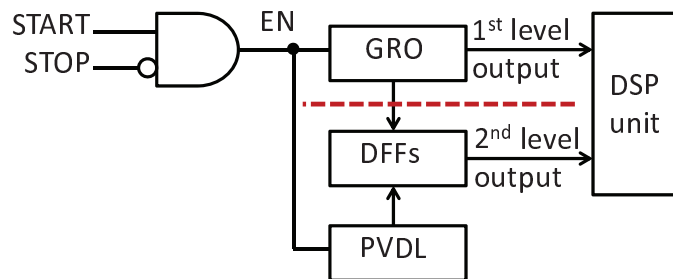


Fig. 26. Block diagram of GRO-PVDL structure.

The proposed structure is named as GRO-PVDL, because it has two levels: the first level is a GRO that provides coarse measurement; the second level provides fine measurement by a VDL-style structure. Unlike standard VDL which requires two delay lines, only one delay line is needed at the second level and the GRO at the first level serves as the other delay line. So the delay line at the second level is called partial Vernier delay line (PVDL). This section first introduces VDL and GRO, and then the principle of the proposed GRO-PVDL is explained.

1. Vernier delay line

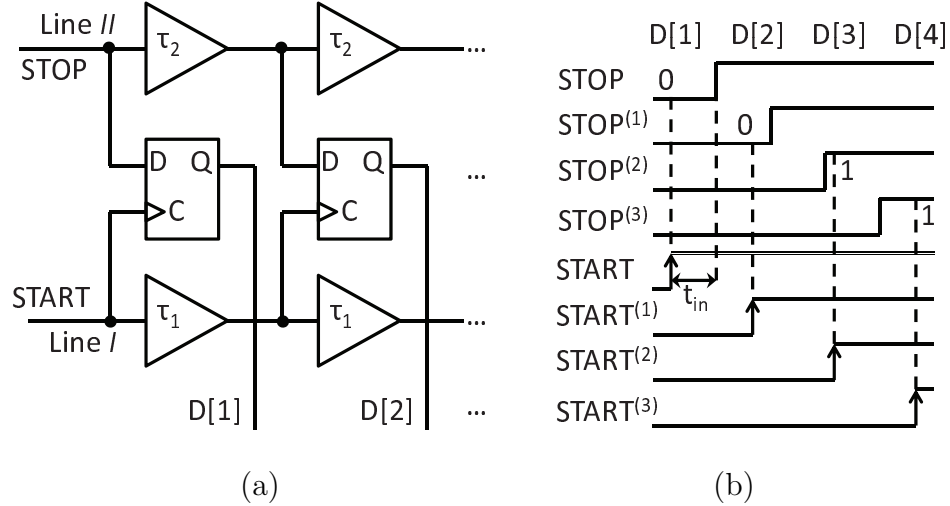


Fig. 27. VDL: (a) structure, (b) timing diagram.

*START(*i*): START delayed by $i \cdot \tau_1$, STOP(*i*): STOP delayed by $i \cdot \tau_2$

Fig. 27(a) illustrates a typical VDL that is composed of two tapped delay lines, whose cell delays are τ_1 and τ_2 . Note that τ_1 is slightly larger than τ_2 : $\tau_1 - \tau_2 = \tau_\Delta$. The taps of line *I* are connected to the *clock* inputs of a series of DFFs. The *data* inputs of the DFFs come from the corresponding taps of line *II*. The input time duration is given as the time interval (t_{in}) between the rising edges of *START* and *STOP* signals that the inputs of line *I* and line *II*, respectively.

Fig. 27(b) illustrates an example of VDL timing diagram. The output of DFFs, digital code *D*, has the pattern of 0..01..1: 0's followed by 1's. With the increase of t_{in} , the number of 0's in *D* will increase proportionally. Therefore, the time duration can be digitized through decoding *D* (counting the number of 0's in *D*).

An equivalent *D* value can also be obtained by the one-line structure in Fig. 28(a). Note that it is just an imaginary design for convenient explanation, since a single cell

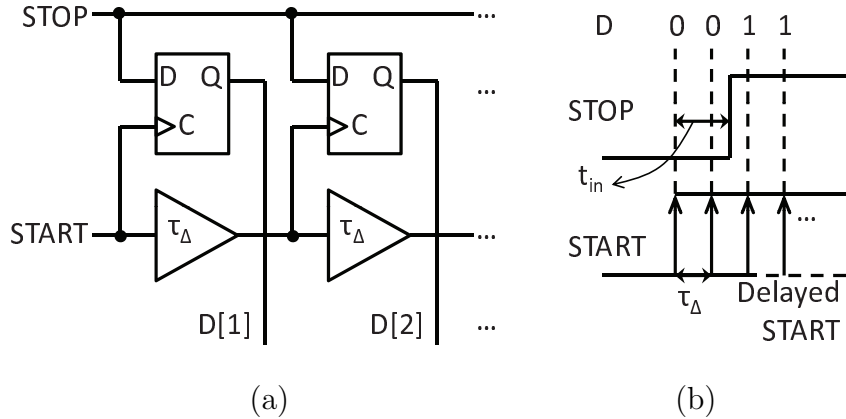


Fig. 28. Equivalence of VDL: (a) structure, (b) timing diagram.

with the delay of τ_Δ is usually too small to implement. From its corresponding timing diagram in Fig. 28(b), we can directly see the proportional relationship between the number of 0's in *D* and t_{in} , and that the resolution is τ_Δ . Since the resolution of VDL is the difference between two cell delays, it is not limited to the smallest delay of a single cell.

2. Gated ring oscillator

GRO structure was first proposed in [34] as part of a time-to-digital converter (TDC) for all digital phase-locked loops (ADPLLs). In the two-level structure proposed in this chapter, GRO serves at the first level to provide the coarse measurement. Its principle is illustrated in Fig. 29. The time difference between the rising edges of periodic signals *START* and *STOP* is converted into a periodic pulse signal *EN*, with jitter on its pulse width. When *EN* is high, GRO is oscillating like a normal ring oscillator, triggering the subsequent counters that capture the signal transitions at the GRO taps G_A , G_B and G_C . As *EN* transits from high to low, GRO stops oscillating suddenly, with the voltages at each tap frozen. Then GRO continues oscillating from its frozen state when *EN* becomes high again. This way, the counters

are actually counting the number of GRO stage delays within the pulse width of EN . So the resolution of GRO equals its stage delay τ_G , which is the delay of a single inverter. Also note that the clock to sample ADD is generated by delaying $STOP$. The delay d_{CK} is inserted to ensure that ADD is sampled while satisfying the setup time constraint of the registers, which means that the GRO has been frozen and the counters and the adder have finished their operations.

Because GRO phase does not change when EN is low, we can make an equivalent timing diagram by removing the frozen intervals and linking the oscillating intervals together, as shown in Fig. 30. For convenient explanation, we assume that ADD changes instantaneously with the GRO phase. At the beginning time $t[i]$ and the ending time $t[i + 1]$ of the i th EN pulse width $t_{EN}[i]$, the values of ADD are $ADD[i]$ and $ADD[i + 1]$, respectively. Then the coarse measurement of $t_{EN}[i]$ is given by:

$$DIFF[i] = ADD[i + 1] - ADD[i], \quad (3.1)$$

with the unit of τ_G .

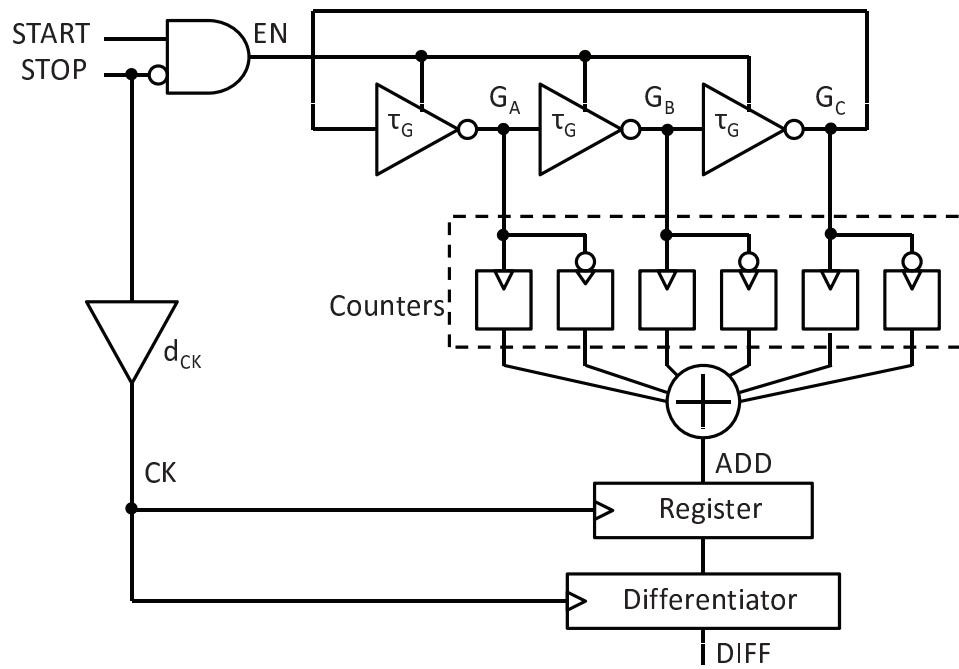
The quantization errors for $ADD[i]$ and $ADD[i + 1]$ are:

$$q[i] = t[i] - t_{ADD}[i], \quad (3.2)$$

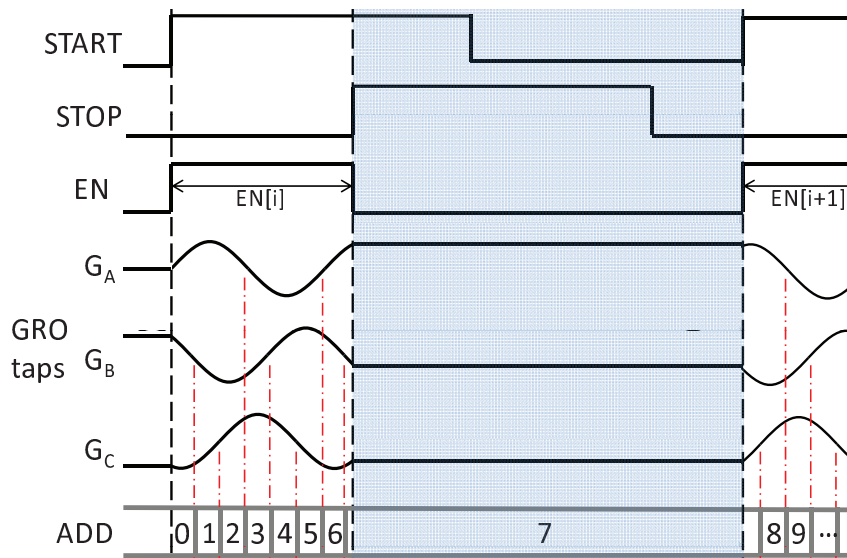
$$q[i + 1] = t[i + 1] - t_{ADD}[i + 1], \quad (3.3)$$

where $t_{ADD}[i]$ ($t_{ADD}[i + 1]$) is the time when ADD transits from $ADD[i]$ ($ADD[i + 1]$) to $ADD[i] + 1$ ($ADD[i + 1] + 1$), and $q[i]$ ($q[i + 1]$) uniformly ranges within $[0, \tau_G)$. Combining Eq. (3.1)-(3.3), and also considering:

$$t_{ADD}[i + 1] - t_{ADD}[i] = (ADD[i + 1] - ADD[i]) \cdot \tau_G, \quad (3.4)$$



(a)



(b)

Fig. 29. GRO: (a) structure, (b) timing diagram.

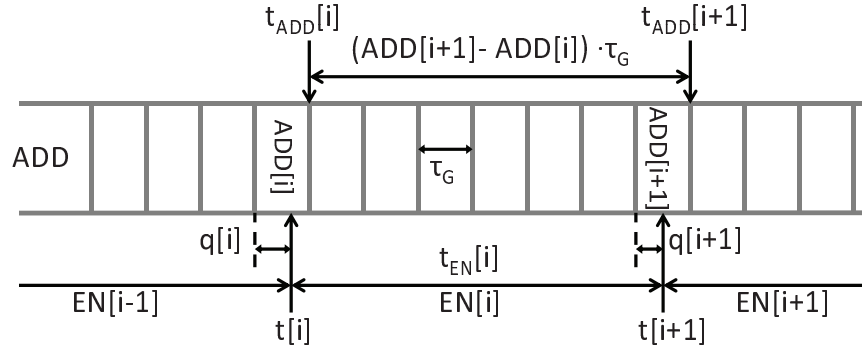


Fig. 30. Equivalent timing diagram of GRO.

we have:

$$t_{EN}[i] = t[i+1] - t[i] = DIFF[i] \cdot \tau_G + (q[i+1] - q[i]), \quad (3.5)$$

where the first term on the right side is the coarse measurement times the unit τ_G , and the second term is the overall quantization error of the coarse measurement.

As explained above, thanks to the phase freezing during the disable state, the quantization error of GRO is $q_{sn}[i] = q[i+1] - q[i]$, instead of $q[i]$. For general cases, $q[i]$ can be treated as a white noise [34]. Note that $q_{sn}[i]$ can be obtained by filtering $q[i]$ with a first-order high-pass filter. Therefore, compared with the white noise shape of q , the frequency spectrum of q_{ns} is shaped to be small at lower frequencies and large at higher frequencies, as illustrated in Fig. 31. The benefit of quantization noise shaping lies at the low frequencies where q_{ns} is smaller than q . If the sampling frequency is much higher than the bandwidth of the measured signal, then by safely low-pass filtering the measurement result, the high frequency power of q_{ns} will be dropped and therefore an effective resolution finer than τ_G can be achieved. To avoid confusion, τ_G will be called GRO's *raw resolution*. The principle of GRO quantization noise shaping is similar to that of $\Delta\Sigma$ ADC [36].

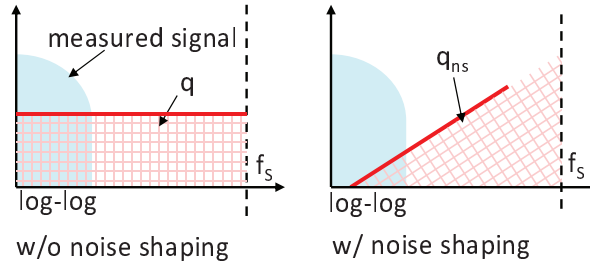


Fig. 31. Frequency spectra of without/with quantization noise shaping.

3. The proposed GRO-PVDL structure

To achieve higher resolution for jitter measurement, a novel GRO-PVDL structure is proposed that improves GRO's raw resolution as well as keeps the feature of the first order quantization noise shaping. The GRO-PVDL structure includes a GRO, a delay line, a group of DFFs and a subsequent DSP unit. The first level is a standard GRO. At the second level, the PVDL cell delay τ_P is designed to be slightly longer than the GRO stage delay τ_G : $\tau_P - \tau_G = \tau_\Delta$, so that the PVDL and the GRO together compose a VDL-style structure.

The input to the PVDL is also from the pulse signal EN . Each tap of the PVDL triggers the *clock* input of a DFF. The data inputs to all the DFFs come from G_{xor} , the XOR of all GRO taps. As shown in Fig. 32, G_{xor} is switching its polarity every GRO stage delay τ_G , when the GRO is shifting its phase. Note that here we temporarily assume that G_{xor} is changing instantaneously with the GRO phase. Referring back to the GRO time diagram in Fig. 30, the GRO has a phase residue of $\tau_G - q[i]$ ($q[i]$ is the quantization error for $ADD[i]$). Therefore, τ_G will switch its polarity at $\tau_G - q[i]$ after the rising edge of the i th EN pulse $EN[i]$. Thanks to the VDL-style structure at the second level, $\tau_G - q[i]$ can be further digitized with a resolution of τ_Δ .

As shown in Fig. 33, the EN rising edge is delayed by the PVDL with its single cell delay τ_P , to trigger the *clock* inputs of the DFFs one by one. On the other hand,

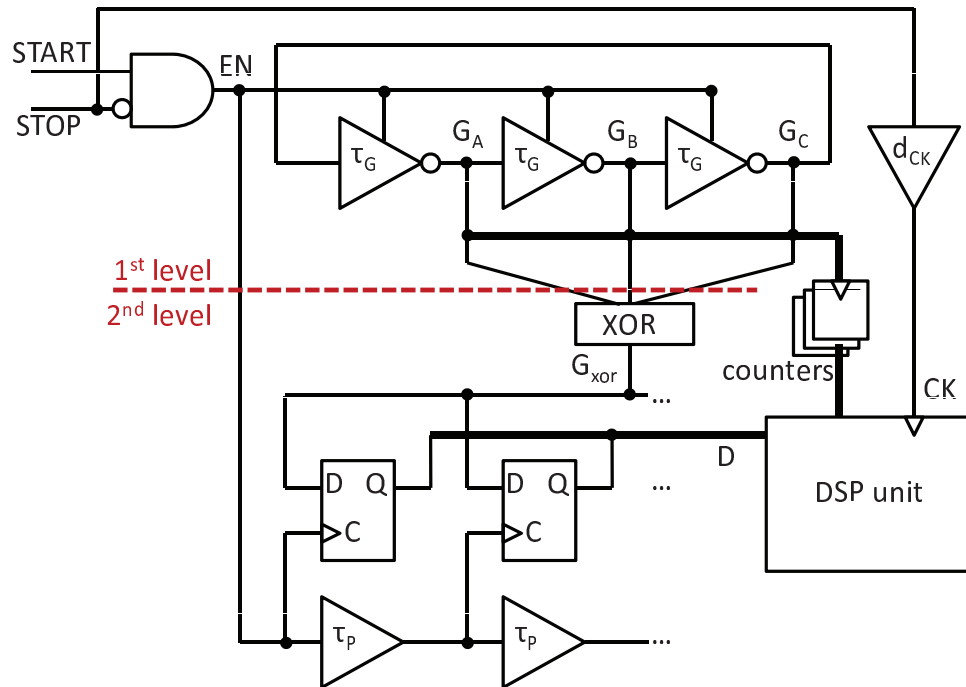


Fig. 32. The proposed GRO-PVDL structure.

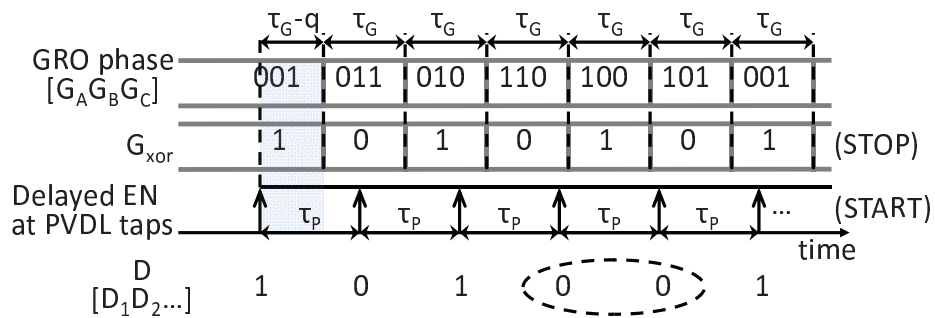


Fig. 33. The timing diagram of GRO-PVDL.

G_{xor} will switch every τ_G . The first G_{xor} switch is later than the EN rising edge by $\tau_G - q$. According to the principle of the standard VDL in Fig. 27, $\tau_G - q$ can be digitized by decoding the outputs of the DFFs D . Note that the pattern of D is 1010... or 0101..., and the position of the first double 1's or 0's indicates the time duration of $\tau_G - q$. Because the resolution of the second level is τ_Δ , we have:

$$\tau_G - q[i] = C_v[i] \cdot \tau_\Delta + q_f[i], \quad (3.6)$$

where C_v is the output of the VDL-style structure that is decoded from D , and q_f is the fine quantization error ranging within $[0, \tau_\Delta)$ which is also a white noise. Similarly, for the next measurement, we have:

$$\tau_G - q[i + 1] = C_v[i + 1] \cdot \tau_\Delta + q_f[i + 1]. \quad (3.7)$$

Through the differentiation of the fine measurement, i.e. subtracting Eq. (3.7) from Eq. (3.6), we have:

$$q[i + 1] - q[i] = DIFF_f[i] \cdot \tau_\Delta + (q_f[i] - q_f[i + 1]), \quad (3.8)$$

where $DIFF_f[i] = C_v[i] - C_v[i + 1]$ is the fine measurement at the second level. It is seen that the quantization error of the coarse measurement is further converted into digital code $DIFF_f$ with a finer resolution τ_Δ . Substituting Eq. (3.8) into Eq. (3.5), we can write:

$$t_{EN}[i] = DIFF_c[i] \cdot \tau_G + DIFF_f[i] \cdot \tau_\Delta + (q_f[i + 1] - q_f[i]), \quad (3.9)$$

where $DIFF_c$ represents the coarse measurement (the replacement of $DIFF$ in Eq. (3.5) for clarity), $DIFF_f$ represents the fine measurement, and the last term on the right side is the overall quantization error.

Eq. (3.9) tells us that not only the residue of the coarse measurement is digitized

with a finer resolution, but the feature of the first order quantization noise shaping is also kept for the fine measurement. Similar to GRO, by low-pass filtering the measurement result, an effective resolution finer than the raw resolution can be achieved. Last but not least, the calibration between the fine resolution and the coarse resolution is required for combining the fine and coarse measurements to provide an overall measurement. This calibration is implemented in the DSP unit, and will be introduced in the next section.

C. Circuit implementation

The proposed GRO-PVDL architecture is implemented using a commercial 90nm CMOS technology. All the circuits except of the DPS unit are designed with analog design flow. Because the analog properties of the GRO, the PVDL and the DFFs have a significant influence on measurement accuracy. By contrast, the DSP unit is designed with digital design flow. In this section, the circuit implementation of GRO-PVDL is presented, including practical issues and circuit optimizations.

1. GRO

To save hardware cost, the GRO at the first level is implemented with three stages, and the stage delay is designed to be 25ps. A critical issue of the GRO design is *gating phase shift*, which is also called gating skew in [37]. As mentioned earlier, because the GRO perfectly freezes its phase while disabled, the first-order quantization noise shaping can be achieved. In practical implementation, however, the GRO phase would shift due to the charge redistribution within “floating” delay cells.

To illustrate the issue of GRO gating phase shift, we take an inverter-based delay cell in Fig. 34(a) as an example. Fig. 34(b) is a simplified model of the gated

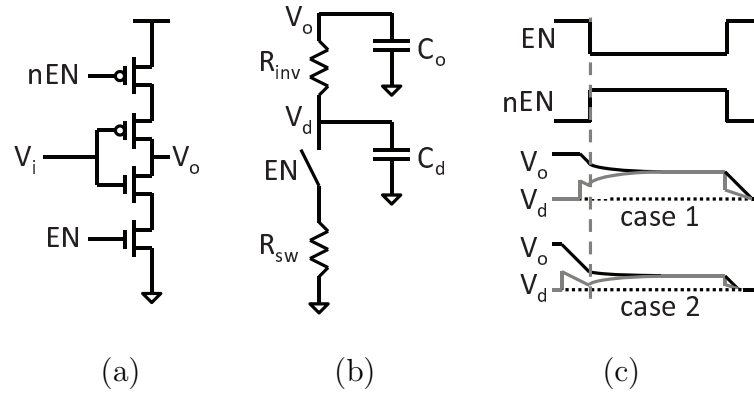


Fig. 34. GRO phase shift: (a) gated inverter, (b) simplified model, (c) waveform.

inverter, which is valid when its output voltage V_o is falling and the PMOS transistors are already in cutoff region. The waveforms in Fig. 34(c) show that once EN turns low, V_o will drop due to the charge redistribution between C_o and C_p . And from this level V_o will continue falling, when EN turns back to high. Due to the drop of V_o , the GRO phase is not fully frozen when EN is low, instead an extra phase shift is introduced. Moreover, the amount of gating phase shift is determined by the GRO phase when EN switches from high to low, as the comparison of case 1 and case 2 in Fig. 34(c).

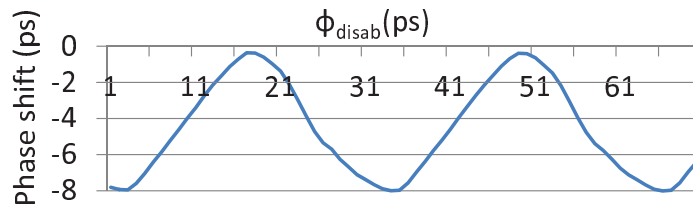


Fig. 35. Simulated phase shift *vs* φ_{disab} of a three-stage inverter-based GRO.

Fig. 35 shows the simulated gating phase shift as a function of the GRO phase at the disabling moment (φ_{disab}) for a three-stage inverter-based GRO. The gating phase shift is given as absolute time, and it varies within a range of around 8ps. Given φ_{disab}

is random for general cases, the variation range of gating phase shift will finally turn into noise in the measurement result. Also note that the DC value of the gating phase shift is usually not critical for jitter measurement, because the measurement result can be easily calibrated by adding a DC offset. Therefore it is desired to minimize the range of gating phase shift. To lower it, one possible solution is to increase the capacitive loading of each stage by inserting dummy transistors, so that the effect of charge redistribution can be minimized. But this will increase the GRO stage delay.

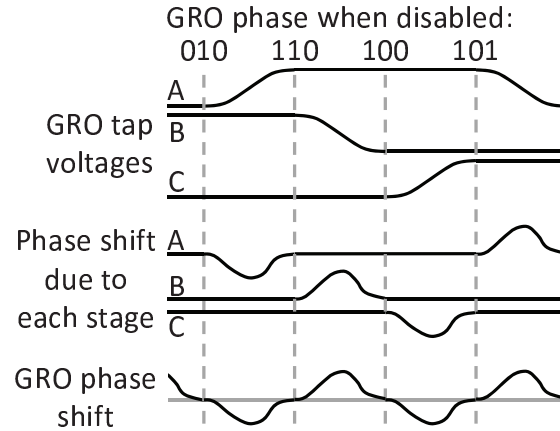


Fig. 36. Gating phase shift of three-stage inverter-based GRO.

Another solution comes from the observation that the gating phase shifts of the three-stage inverter-based GRO are complementary to each other for rising V_o and falling V_o , as is illustrated in Fig. 36. The gating phase shift is minimum when V_o is rising and maximum when V_o is falling. So it is possible to cancel the minimum and maximum phase shifts with each other using *differential* delay cells instead of single-ended inverters, because differential structure has two output voltages V_o and nV_o , and the rising/falling of V_o is always accompanied with the falling/rising of nV_o .

In the implementation, the differential delay cell with cross-coupled inverters (CCI) [38] is chosen to build up the GRO. As is shown in Fig. 37, it is composed of

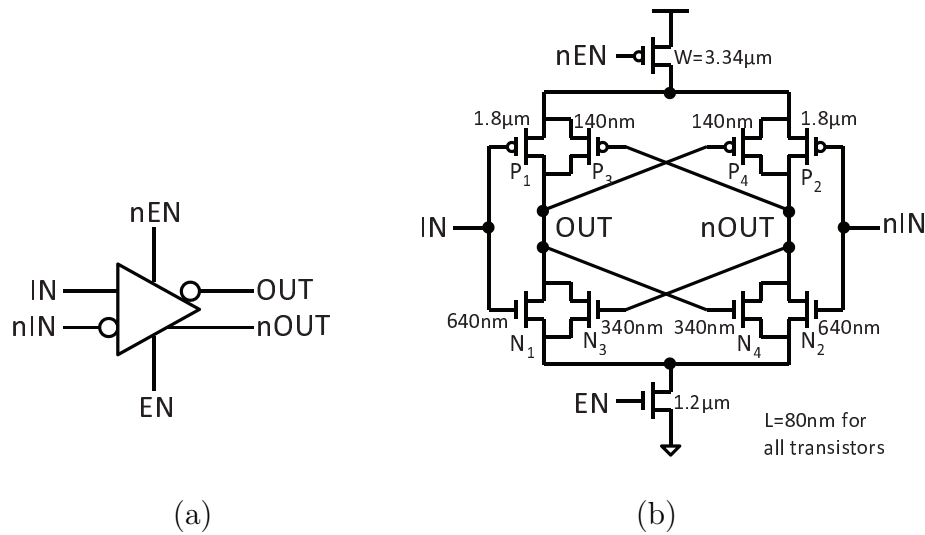


Fig. 37. Gated delay cell with CCI: (a) symbol, (b) schematic.

two large inverters (P_1/N_1 and P_2/N_2), whose outputs are coupled with each other through two small inverters (P_3/N_3 and P_4/N_4). And two transistors are inserted to enable oscillation gating, one PMOS above and one NMOS beneath. Thanks to the cross coupling, the gating phase shifts due to the charge redistribution of the two larger inverters will be mostly canceled with each other. Fig. 38 shows the simulated range of gating phase shift of the three-stage CCI-cell-based GRO. Note that the phase shift range is only around one.5ps, which is much lower than that of the inverter-based GRO.

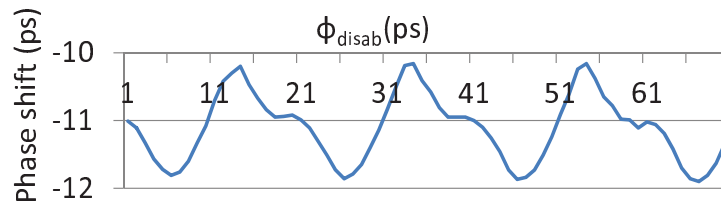


Fig. 38. Simulated phase shift *vs* ϕ_{disab} of a three-stage CCI-cell-based GRO.

Apart from reducing the gating phase shift, the CCI delay cell has other merits

that make it a good candidate to build up the GRO. The differential structure can achieve higher power supply rejection, as well as to provide rail-to-rail output voltage swing [39]. Moreover, the delay cell with CCI has higher immunity to rising/falling delay mismatch, and thus the GRO will have a more even tap delay.

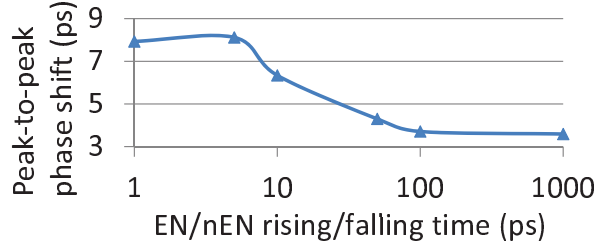


Fig. 39. Simulated phase shift *vs* EN/nEN rising/falling time of a three-stage inverter-based GRO.

Finally, the range of gating phase shift can be further reduced by increasing the rising/falling time of EN/nEN . Fig. 39 demonstrates this phenomenon for the three-stage inverter-based GRO. An intuitive explanation is that the GRO is “weakly” oscillating when EN/nEN is rising/falling, and as a result the GRO phase when disabled φ_{disab} is more like a continuous variable than a fixed value. So the shifted phases are effectively averaged over the range of φ_{disab} . So in our implementation, by adding dummies to increase the load capacitance of EN/nEN , the rising/falling time of EN/nEN is enlarged to be 80ps (more than three GRO stage delays). And simulated peak-to-peak phase shift is pushed to lower than 1ps.

Referring back to Fig. 30, the noise due to gating phase shift, similar to the quantization noise, is added to both $t[i]$ and $t[i + 1]$ because it is injected at the beginning of each EN pulse. Therefore, the noise due to gating phase shift is also shaped like the quantization noise.

2. Counters

To count every stage delay of the GRO, two counters are needed for each tap, one to count rising edges and one to count falling edges. One challenge of the counter design is its high input frequency. For the three-stage GRO with 25ps stage delay, the counters are required operate at an input frequency of $f_o = \frac{1}{2N\tau_G} = 6.67\text{GHz}$. Asynchronous counters are therefore adopted to handle such high frequency, whose structure is shown in Fig. 40. Compared with synchronous counters, asynchronous counters can operate at a much higher input frequency, and the highest operating frequency will not decrease for larger bit width [40]. Simulation shows that an asynchronous counter made of standard cell DFFs is already able to handle over 8GHz input. For asynchronous counters using true single phase clocked (TSPC) DFFs [41], an operating frequency as high as 20GHz can be achieved in simulation. And the measuring range of the input time duration is $2^k \cdot 6\tau_G$, where $\tau_G=25\text{ps}$ and k is the bit width of the counters. In the implementation, the counter bit width is designed to be 4, which allows a measuring range of 2.4ns. Also note that the asynchronous counter will automatically switch its output from all 1's to all 0's when overflow occurs, and a flag signal OF is output to indicate the occurrence of overflow.

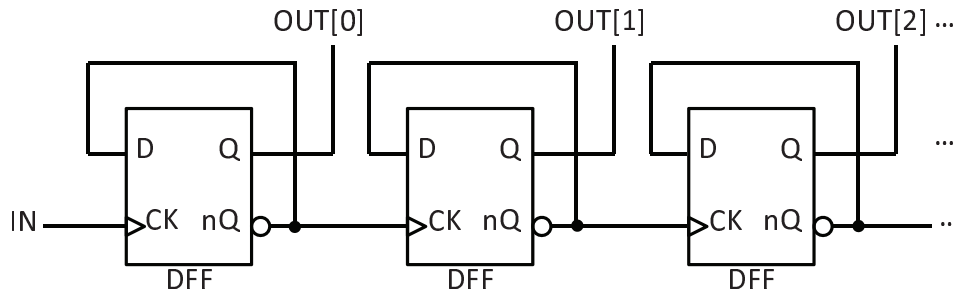


Fig. 40. Asynchronous counter.

For the three-stage GRO, six of such asynchronous counters would be needed. To reduce the hardware cost, however, the implementation only requires one asynchronous counter, using a phase tracking technique [37]. Fig. 41 shows the counter implementation, where a single-ended design is shown for simplicity, while the real circuits are implemented with differential structures. Instead of six counters, only one counter is used to count the number of the rising edge of the GRO tap G_C . For the three-stage GRO, G_C has a rising edge for every 6 GRO stage delays, i.e. a GRO cycle. Therefore, the counter output N_{cycle} is multiplied by 6 to represent the number of GRO tap transitions.

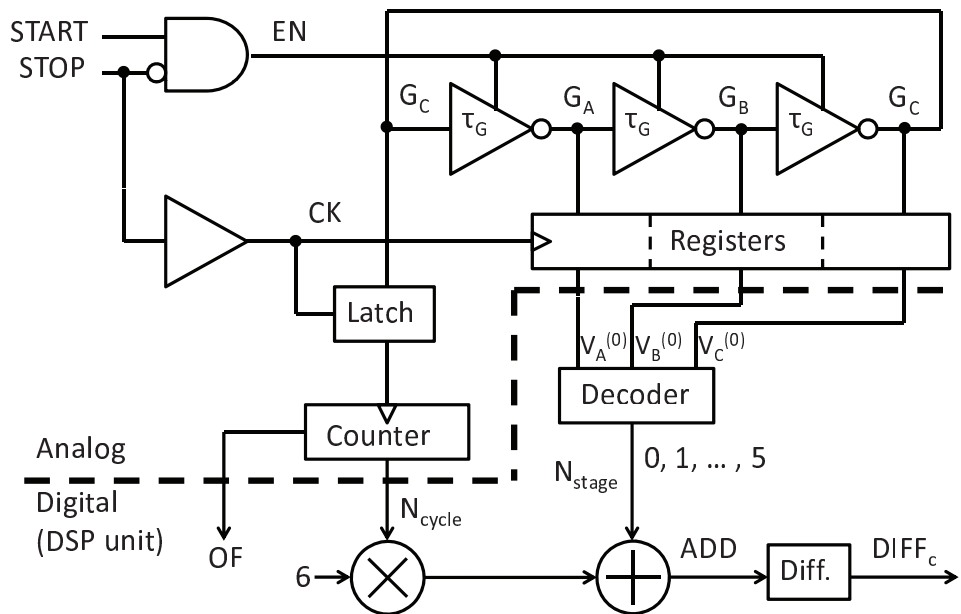


Fig. 41. Phase tracking based counting structure (single-ended version).

On the other hand, the GRO phase can be translated into the number of GRO tap transition smaller than 6. Between any two successive rising edges of G_C , the GRO phase shifts in a fixed pattern with a length of 6, as illustrated in Fig. 42. So by chronologically decoding the GRO phases into 0, 1, ..., 5, the number of GRO tap

transitions by modulo 6 can be obtained:

$$N_{\text{stage}} = F_{ch}(V_A^{(0)}V_B^{(0)}V_C^{(0)}), \quad (3.10)$$

where F_{ch} is the function of chronological decoding, and $V_A^{(0)}V_B^{(0)}V_C^{(0)}$ is the GRO phase. Note that $V_A^{(0)}V_B^{(0)}V_C^{(0)}$ is sampled when it is frozen, and the phase decoder is implemented in the DSP unit. Finally, the number of total tap transitions ADD is given by $6N_{\text{cycle}} + N_{\text{stage}}$.

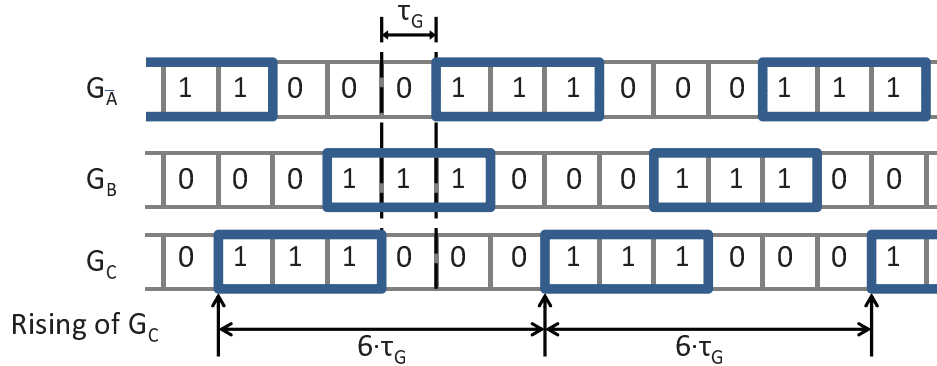


Fig. 42. Phase transition of three-stage GRO.

In spite of saving hardware cost, the above counting scheme has an issue that would result in over/under-counting. The timing mismatch between N_{stage} and N_{cycle} might occur when the GRO phase is frozen around the rising edge of G_C . As is shown in Fig. 43, if N_{stage} has not been switched by the rising edge of G_C but N_{cycle} has, then ADD is over-counted by 5; if N_{stage} has been switched by the rising edge of G_C but N_{cycle} has not, then ADD is under-counted by 5. A latch sharing technique can be applied to solve the issue [37]. As shown in Fig. 44, the register connected to G_C is separated into two latches, and the first latch is shared with the counter. This way, the counter input and the decoder input are guaranteed to be synchronized to each other.

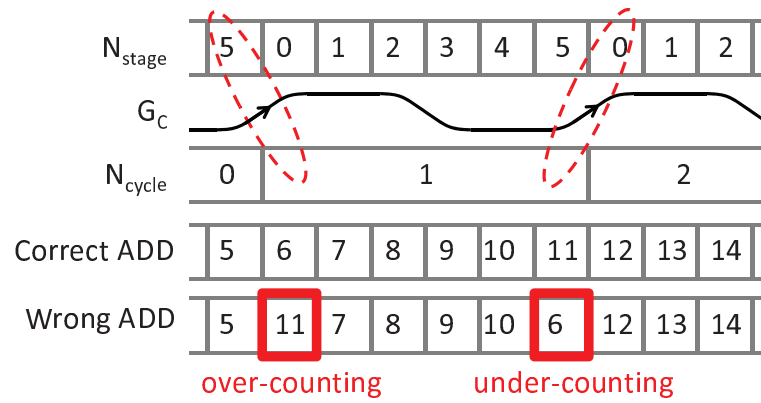


Fig. 43. Over/under-counting due to counter/decoder input mismatch.

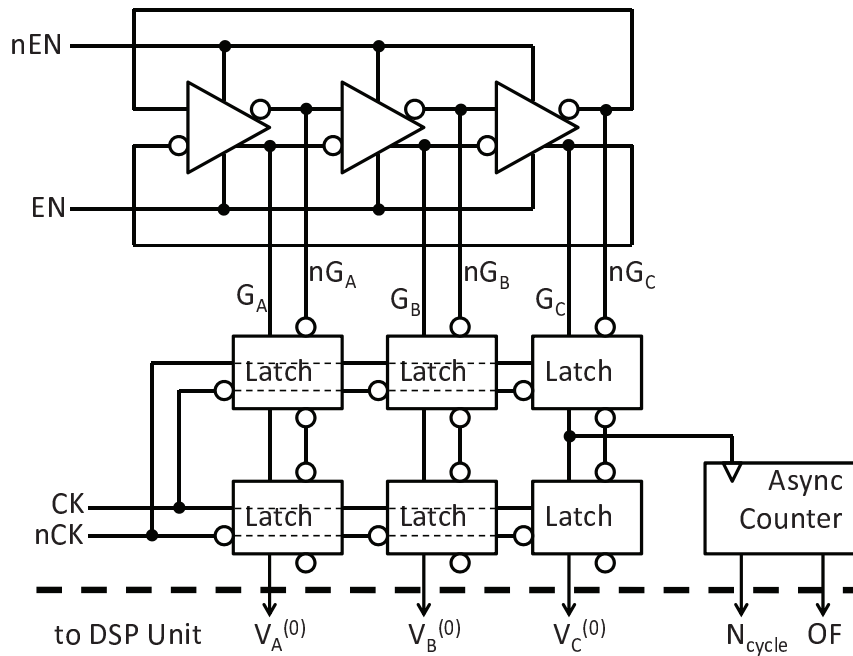


Fig. 44. GRO counting structure with latch sharing.

Another issue of the counter is the *glitch* at its input, which will also cause over-counting. When the counter input is frozen at close to some middle level between high and low, glitch might occur due to noise, as illustrated in Fig. 45. The latch between G_C and the counter can largely reduce the chance of glitch. Moreover, a pair of small cross-coupling inverters are inserted at the differential output of the latch, in order to force the output to a logic level and ensure it will not be overturned by noise.

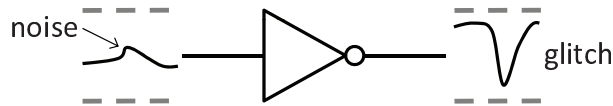


Fig. 45. Glitch illustration.

3. PVDL

The PVDL is composed of a line of differential delay cells with CCI, and its schematic is shown in Fig. 46. And the cell delay τ_P is design to be 30ps by proper dummy transistor loading. Considering the GRO stage delay $\tau_G=25ps$, a raw resolution of $\tau_\Delta=5ps$ can be achieved with the proposed GRO-PVDL structure. As mentioned before, the residue of the coarse measurement ranges from 0 to τ_G . Consequently, at least $\tau_G/\tau_\Delta=5$ delay cells are needed in order to cover the range of the coarse residue. To ensure safety under process variation, the PVDL has a length of 10 delay cells in the implementation.

Note that the delay cell of the PVDL is similar to that of the GRO, but with no gating transistors. Compared with single-ended delay cells, the CCI-based delay cells not only have higher immunity to power supply noise, but also have smaller delay mismatch due to process variation. Because the delay variation between the two large

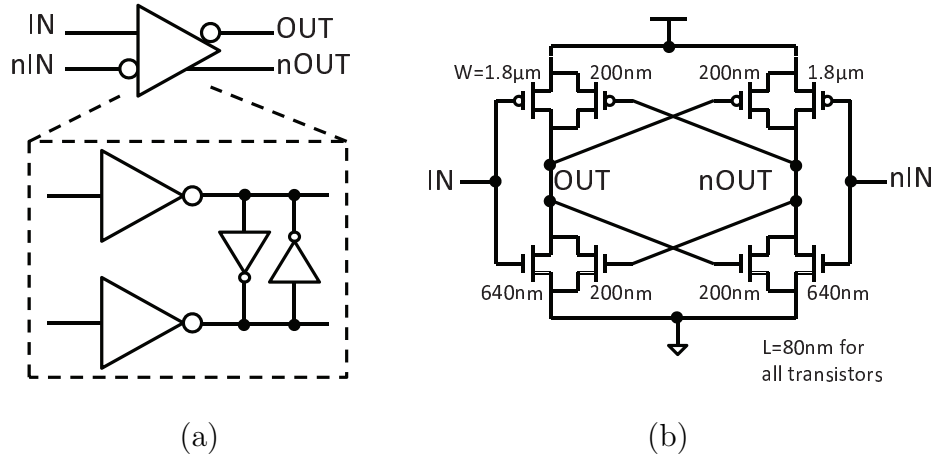


Fig. 46. Differential delay cell: (a) symbol and gate-level schematic (b) schematic.

inverters in Fig. 46(a) will be canceled with each other. If we simply assume the delay of CCI-based delay cell is the arithmetic average of the delays of its two inverters, then the delay variance of the entire cell is half of that of a single inverter.

4. DFFs

A differential structure is adopted for the DFF design. As shown in Fig. 47, the differential DFFs structure is composed of two differential latches with CCI. Such structure was first proposed in [42] to achieve minimum power as well as propagation delay. Note that the clock input is also differential such that it could be triggered by the differential output of the PVDL. The DFF structure has very small setup/hold time (less than 0.3ps in simulation). And thanks to the cross coupling inverters, the outputs will be quickly forced to high/low level, even when the setup/hold time condition is not met. Therefore it is suitable for the application as a lead/lag detector.

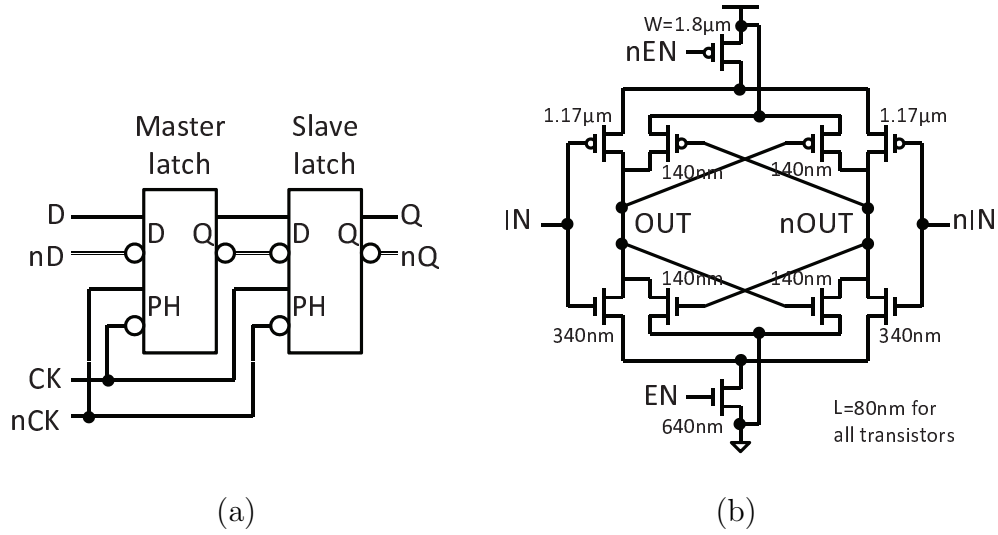


Fig. 47. (a) Gate-level schematic of differential DFFs (b) schematic of differential latch.

5. GRO-PVDL

Referring back to Fig. 32, within the proposed GRO-PVDL structure, the DFFs are supposed to sample the XOR of the GRO taps. Since the state delay of the GRO is close to minimum, the XOR gate has to operate at an input switching frequency of $1/\tau_G=40\text{GHz}$. However, it is impossible to implement an XOR gate operating at such high frequency. Our solution is to directly sample the GRO taps and then XOR the sampled values of the DFFs in the subsequent DSP unit. As shown in Fig. 48, each PVDL tap triggers the clock inputs of three DFFs, whose data inputs are connected to G_A, G_B, G_C . The outputs of the DFFs are $V_A^{(1)}V_B^{(1)}V_C^{(1)}, V_A^{(2)}V_B^{(2)}V_C^{(2)}, \dots$ And the outputs of the VDL-style structure $D^{(1)}D^{(2)}\dots$ are generated in the DSP unit, by $D^{(i)} = V_A^{(i)} \oplus V_B^{(i)} \oplus V_C^{(i)}$.

An issue of the GRO-PVDL structure is the uneven stage delays of the GRO at the moments of enabling and disabling, which is caused by the charge redistribution inside the delay cells. The unevenness of stage delays is more prominent when the

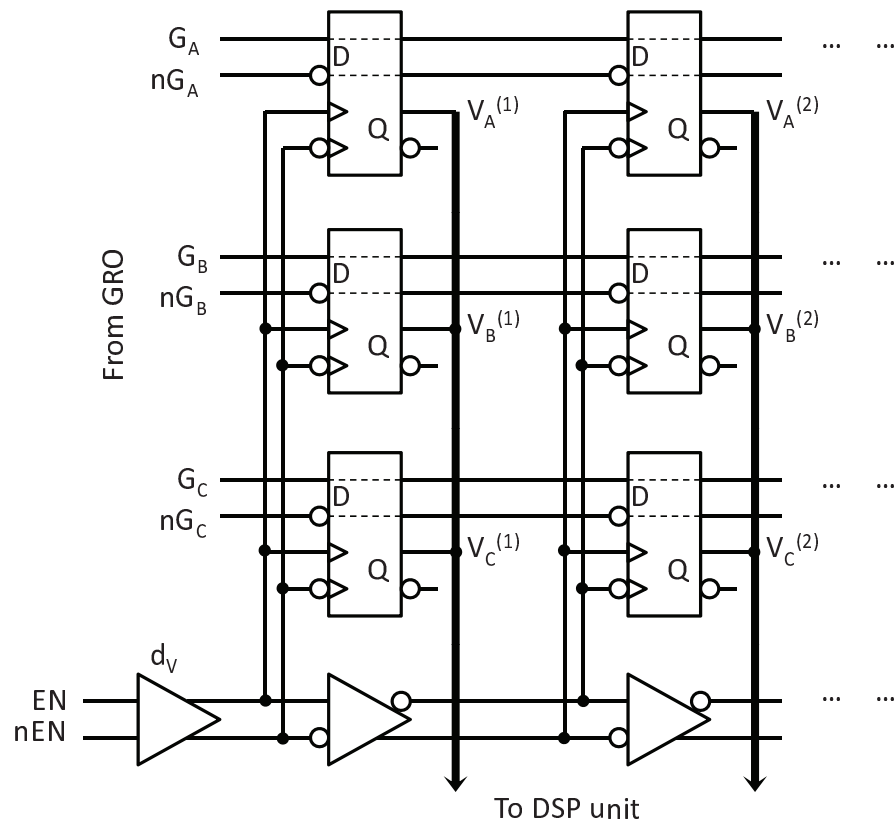


Fig. 48. Schematic of PVDL and DFFs.

rising/falling time of EN/nEN is intentionally increased to achieve smaller variation of the gating phase shift, as illustrated in Fig. 49. This may lead to inaccuracy or even malfunction of the fine VDL-style measurement.

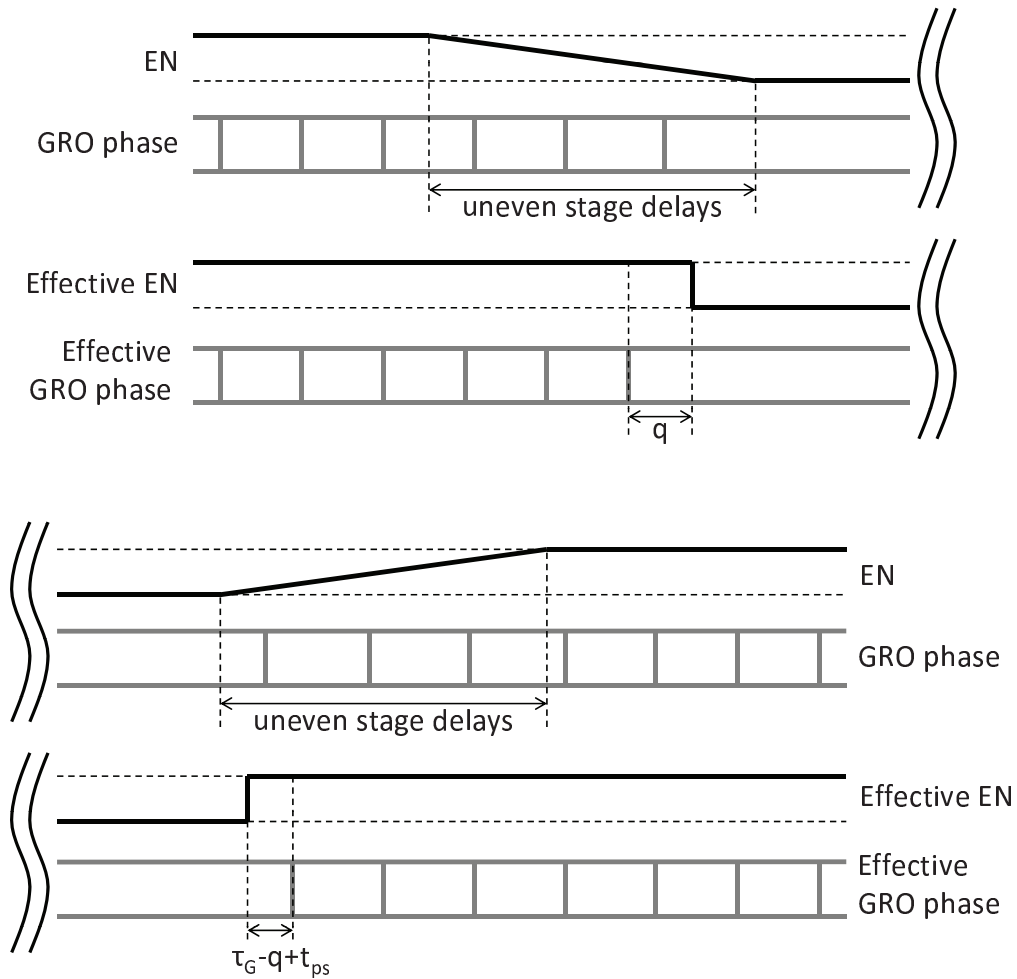


Fig. 49. Uneven stage delays of the GRO, and the effective timing diagram.

The issue of uneven stage delays can be solved by inserting a delay of d_V at the PVDL input, as in Fig. 48. d_V should be large enough to ensure the sampling clocks from the PVDL can avoid the period of uneven stage delays, i.e. the sampling clock from the first PVDL tap rises after the GRO has entered the period when its stage

delays are even.

For the convenient explanation of the solution details, we first draw an effective timing diagram for the GRO enabling and disabling process, as in Fig. 49. With the effect of uneven stage delays reflected by the gating phase shift t_{ps} , conceptually we consider an effective EN that is steep and the effective GRO phase that has even stage delays. As is stated previously in the GRO design that the variance of the gating phase shift can be reduced to small enough, so here t_{ps} is treated as invariant. Compared with Fig. 33, it is seen that $\tau_G - q + t_{ps}$, rather than $\tau_G - q$, needs to be measured at the second level of the GRO-PVDL.

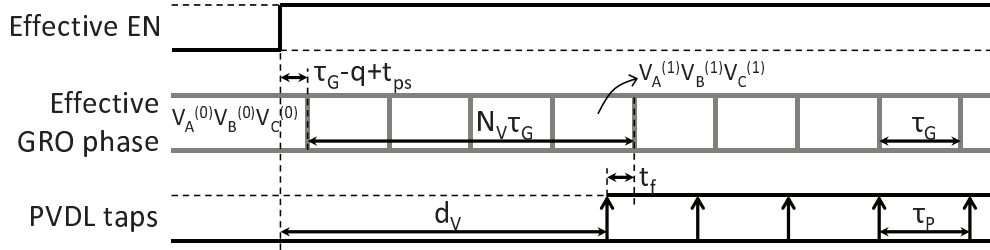


Fig. 50. Timing diagram of the GRO-PVDL with d_V .

Due to the uneven stage delays, however, $\tau_G - q + t_{ps}$ cannot be measured directly. Thanks to the inserted delay d_V , the time between the rising edge of the first PVDL tap and the next GRO phase switch, t_f , can be correctly measured with the fine resolution τ_Δ by the VDL-style structure:

$$t_f[i] = C_V[i] \cdot \tau_\Delta + q_f[i], \quad (3.11)$$

where C_V is output of the VDL-style structure. On the other hand, from the timing diagram in Fig. 50, the relationship between $\tau_G - q + t_{ps}$ and t_f is:

$$(\tau_G - q[i] + t_{ps}) + N_V[i] \cdot \tau_G = d_V + t_f[i], \quad (3.12)$$

where N_V is the number of the GRO phase switches during the inserted delay time d_V . From Eq. (3.11) and Eq. (3.12), we can derive the expression of q :

$$q[i] = (N_V[i] + 1) \cdot \tau_G - C_V[i] \cdot \tau_\Delta - q_f[i] - d_V + t_{ps}. \quad (3.13)$$

Substituting Eq. (3.13) into Eq. (3.5), we finally have:

$$t_{EN}[i] = (DIFF_c[i] + \Delta N_V[i]) \cdot \tau_G + DIFF_f[i] \cdot \tau_\Delta + (q_f[i] - q_f[i + 1]), \quad (3.14)$$

where $\Delta N_V[i] = N_V[i + 1] - N_V[i]$. Compared with the measurement result for the ideal GRO-PVDL (Eq. (3.9)), the coarse measurement has an extra term ΔN_V . N_V can be obtained by decoding the GRO phases $V_A^{(0)}V_B^{(0)}V_C^{(0)}$ and $V_A^{(1)}V_B^{(1)}V_C^{(1)}$ like in the counting structure using Eq. (3.10). Here $V_A^{(0)}V_B^{(0)}V_C^{(0)}$ is the GRO phase when frozen, and $V_A^{(1)}V_B^{(1)}V_C^{(1)}$ is the GRO phase sampled by the first clock from the PVDL. Because ΔN_V is part of the fine measurement but it has the unit of τ_G , and thus it can be treated as the carry from the fine measurement to the coarse measurement. Also note that fixed offsets like d_V and t_{ps} are canceled in the expression of t_{EN} , which means the exact value of the inserted delay d_V does not influence the final measurement, as long as d_V is large enough for the fine measurement to avoid uneven GRO stage delays.

6. DSP unit

The DSP unit is responsible of translating the “raw” digital information generated by the GRO-PVDL structure into an output digital code that is the measurement of the input time duration. As is shown in Fig. 51, there are three modules: the coarse code generator, the VDL decoder and the fine code generator. Note that an extra low-pass filter is needed to remove the high-frequency quantization noise in the output M_{oa} . Since a standard low-pass filter can be implemented with low hardware

cost, its design is not detailed here. The final output is the overall measurement M_{oa} that has 15 bits, with 5 bits as the integer part and 10 bits as the fractional part. And one LSB of the integer part represents the coarse resolution τ_G .

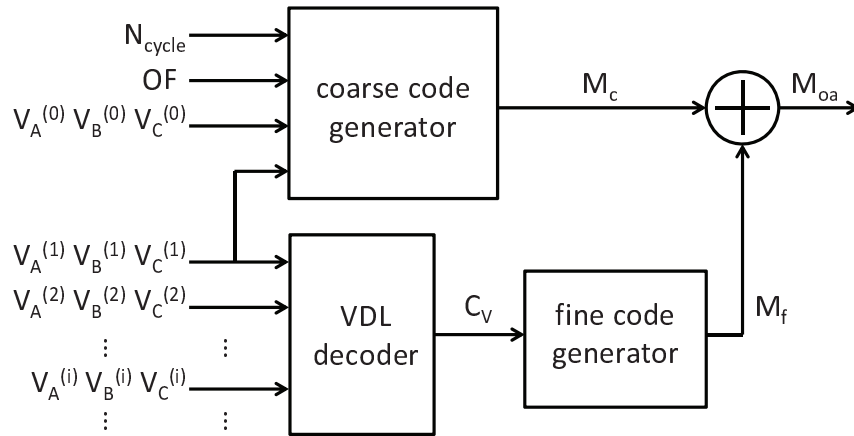


Fig. 51. Block diagram of the DSP unit.

The ever scaling CMOS technology is providing more digital resources with lower hardware cost. This allows us to achieve high speed as the first design priority, in order to help increase the sampling frequency of measurement. Therefore, pipelined structures are adopted to achieve a higher data throughput. The entire DSP unit is composed of synchronous sequential logic triggered by a single clock source whose frequency is the same as the sampling frequency of the GRO-PVDL, and the largest pipeline latency is 10 clock cycles. Using 90nm CMOS technology, the circuits is successfully synthesized given the clock frequency of 500MHz, with the smallest timing slack of 0.13ns.

a. Coarse code generator

The block diagram of the coarse code generator is shown in Fig. 52. For convenient explanation, it is divided into three parts.

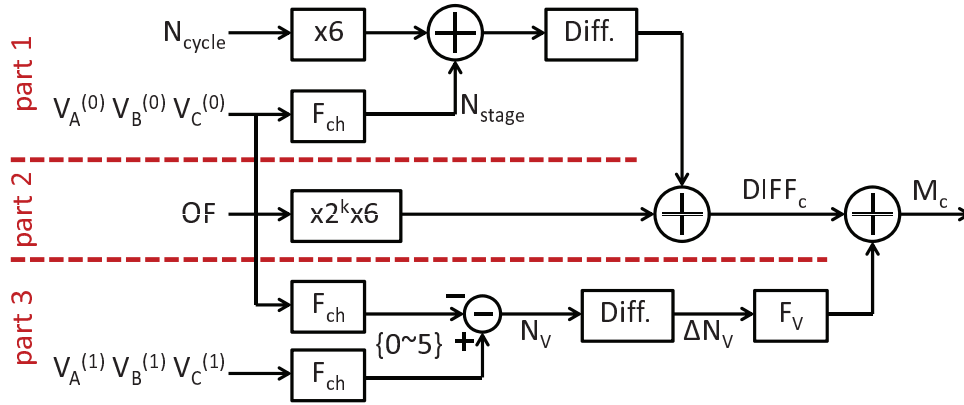


Fig. 52. Implementation of coarse code generator.

The coarse measurement $DIFF_c$ is the digitization of EN pulse width t_{EN} with a resolution of GRO stage delay τ_G . Referring back to the GRO principle, $DIFF_c$ is the differentiation of ADD which is the number of the total GRO tap transitions during t_{EN} . As introduced for the counter design, ADD is given by $6N_{cycle} + N_{stage}$, where N_{cycle} is the output of the counter, and N_{stage} is decoded from the GRO phase $V_A^{(0)}V_B^{(0)}V_C^{(0)}$ sampled when the GRO is disabled ($N_{stage} = F_{ch}(V_A^{(0)}V_B^{(0)}V_C^{(0)})$). The first part of the coarse code generator implements the above function.

Nevertheless, $DIFF_c$ should also include the overflow of the asynchronous counter. When overflow happens, the counter output N_{cycle} automatically switches from all 1's to all 0's, and at the same time the overflow flag OF becomes 1 (OF is 0 when there is no overflow). Therefore, the second part of the coarse code generator includes the effect of overflow by adding $OF \times 2^k \times 6$ into $DIFF_c$, where k is the bit width of the asynchronous counter and $k=4$ in the implementation.

The third part of the coarse code generator deals with ΔN_V , the carry from the fine measurement to the coarse measurement. Referring back to the delay insertion technique that solves the issue of the uneven GRO stage delays, ΔN_V is the differentiation of N_V , and N_V is the number of the GRO phase switches during the inserted delay time d_V , and $N_V = F_{\text{ch}}(V_A^{(1)}V_B^{(1)}V_C^{(1)}) - F_{\text{ch}}(V_A^{(0)}V_B^{(0)}V_C^{(0)})$, where $V_A^{(0)}V_B^{(0)}V_C^{(0)}$ is the GRO phase when frozen, and $V_A^{(1)}V_B^{(1)}V_C^{(1)}$ is the GRO phase sampled by the first clock from the PVDL. Therefore, ΔN_V can be given by:

$$\Delta N_V = \text{Diff}\{F_{\text{ch}}(V_A^{(1)}V_B^{(1)}V_C^{(1)}) - F_{\text{ch}}(V_A^{(0)}V_B^{(0)}V_C^{(0)})\}, \quad (3.15)$$

where $\text{Diff}\{\}$ means differentiation. Since the inserted delay d_V is fixed, N_V that represents the digitization of d_V should only fluctuate between $\{\lfloor \frac{d_V}{\tau_G} \rfloor, \lceil \frac{d_V}{\tau_G} \rceil\}$ (the floor and the ceiling of $\frac{d_V}{\tau_G}$). Consequently, ΔN_V ranges within $\{-1, 0, 1\}$. However, due to the periodicity of the GRO phase, F_{ch} can only provide a range of $\{0, 1, \dots, 5\}$, which causes $\Delta N_V \in \{-6, -5, \dots, 5, 6\}$. To solve this contradiction, another mapping F_V is applied on ΔN_V :

$$F_V(\Delta N_V) = \begin{cases} \Delta N_V - 6, & \Delta N_V \geq 2 \\ \Delta N_V + 6, & \Delta N_V \leq -2 \\ \Delta N_V, & \text{otherwise} \end{cases}, \quad (3.16)$$

Finally, the coarse measurement M_c is combined by DIFF_c and $F_V(\Delta N_V)$.

b. VDL decoder

The VDL decoder generates the output of the VDL-style structure at the second level of the GRO-PVDL. The input of the module are $V_A^{(1)}V_B^{(1)}V_C^{(1)}, V_A^{(2)}V_B^{(2)}V_C^{(2)}, \dots$, where $V_A^{(i)}V_B^{(i)}V_C^{(i)}$ represents the GRO phase sampled by the i th clock generated by the PVDL. As shown in Fig. 53, $V_A^{(i)}V_B^{(i)}V_C^{(i)}$ is first compressed into a 1-bit width signal

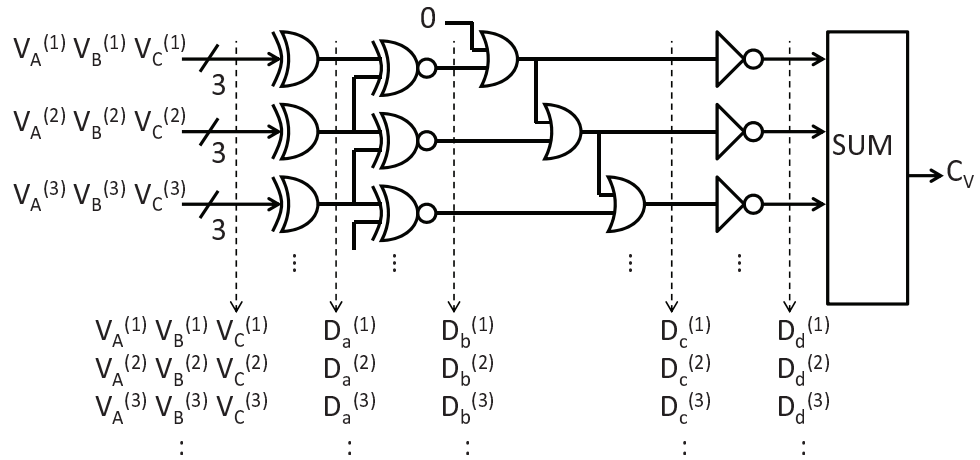
$D_a^{(i)}$ through an XOR gate. Referring back to the principle of the GRO-PVDL, the VDL output should represent the position of the first double 0's or 1's in $D_a^{(1)} D_a^{(2)} \dots$. Through the combinational logic in Fig. 53(a), $D_a^{(1)} D_a^{(2)} \dots$ can be finally generated, which is a sequence of 1's followed by 0's, and the boundary between 1's or 0's is at the same position as the first double 0's or 1's in $D_a^{(1)} D_a^{(2)} \dots$. Therefore, this position can be obtained by adding the bits of $D_a^{(1)} D_a^{(2)} \dots$. In order to achieve a higher operating frequency, the combinational logic of the VDL decoder in Fig. 53(a) is divided into 8 smaller combinational logics, and implemented using a pipeline structure with a propagation latency of 8 clock cycles.

A good feature of the proposed VDL decoder is bubble suppression. In practice, due to noise and delay mismatch, there may be more than one sequence of double 0's or 1's in $D_a^{(1)} D_a^{(2)} \dots$, a.k.a. bubbles. Fortunately, the proposed logics are capable of removing all the bubbles. Therefore $D_a^{(1)} D_a^{(2)} \dots$ is always bubble-free. An example of bubble suppression is demonstrated in Fig. 53(b).

c. Fine code generator

From the VDL output C_V , the fine measurement $DIFF_f$ can be generated simply by $DIFF_f[i] = C_V[i] - C_V[i + 1]$. But considering the fine measurement and the coarse measurement have different resolutions (τ_Δ and τ_G respectively), the question is how to combine them together to get an overall measurement M_{oa} . To solve it, the fine measurement should change its unit to the same as the coarse resolution. Let M_f be the fine measurement with the unit of τ_G , then $M_g \cdot \tau_G = DIFF_f \cdot \tau_\Delta$. So we have $M_f = \frac{\tau_\Delta}{\tau_G} \cdot DIFF_f$. For example, if τ_Δ and τ_G are 5ps and 25ps, exactly as designed, then $M_f = \frac{1}{5} DIFF_f$. Unfortunately, the ratio between τ_Δ and τ_G is unknown due to process variation, and thus needs to be calibrated.

In the proposed fine code generator, the VDL output C_V is first transformed into



(a)

	$V_A^{(i)}$	$V_B^{(i)}$	$V_C^{(i)}$	$D_a^{(i)}$	$D_b^{(i)}$	$D_c^{(i)}$	$D_d^{(i)}$	C_v
	0	0	1	1	0	0	1	3
	0	1	1	0	0	0	1	
1 st double	0	1	0	1	0	0	1	
0s/1s	1	1	0	0	1	1	0	
	1	0	1	0	0	1	0	
bubble	0	1	1	0	1	1	0	
	1	1	0	0	:	1	0	
	:	:	:	:	:	:	:	

(b)

Fig. 53. VDL decoder: (a) implementation (b) bubble suppression.

another code $C_V^{(c)}$ whose unit is τ_G . Then $C_V^{(c)}$ is differentiated to generate M_f whose unit is also τ_G . Since now the fine measurement M_f and the coarse measurement M_c have the same unit of τ_G , we have:

$$t_{\text{EN}} = (M_c + M_f) \cdot \tau_G + q_f^{(\text{sn})}, \quad (3.17)$$

where $q_f^{(\text{sn})}$ is the shaped quantization noise.

To transform C_V into $C_V^{(c)}$, we start from the observation that $\max(C_V) \cdot \tau_\Delta = \tau_G$, where $\max(C_V)$ is the maximum value of C_V . This is because the VDL-style structure digitized the residue of the coarse measurement which is within $[0, \tau_G)$. Also note $\text{avg}(C_V) = \max(C_V)/2$, where $\text{avg}(C_V)$ is the average value of C_V , if the residue is uniformly distributed. In practice, the use of $\text{avg}(C_V)$ is preferred, because $\text{avg}(C_V)$ is more statistically stable than $\max(C_V)$. Therefore, $\frac{\tau_\Delta}{\tau_G}$ can be estimated by calculating $\text{avg}(C_V)$:

$$\frac{\tau_\Delta}{\tau_G} = \frac{1}{2\text{avg}(C_V)}. \quad (3.18)$$

Because the unit of $C_V^{(c)}$ is τ_G , we have:

$$C_V^{(c)} \cdot \tau_G = C_V \cdot \tau_\Delta. \quad (3.19)$$

Substituting Eq. (3.19) into Eq. (3.18), finally we can get:

$$C_V^{(c)} = \frac{C_V}{2\text{avg}(C_V)}. \quad (3.20)$$

The block diagram of this operation is shown in Fig. 54. The averager is implemented using an infinite impulse response (IIR) filter: $y = \frac{\alpha x}{1 + (\alpha - 1)z^{-1}}$, where $\alpha = 2^{-8}$. The divider is implemented in pipelined structure to increase the operating frequency.

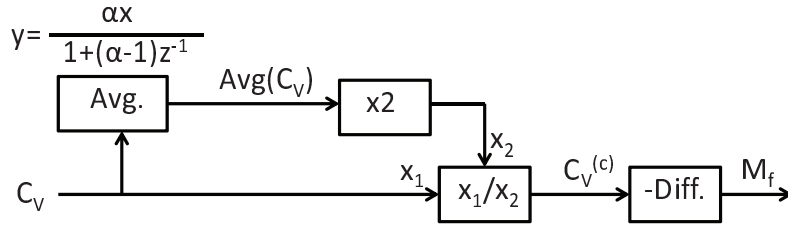


Fig. 54. Implementation of fine code calibration.

D. Experimental results

The proposed GRO-PVDL structure is implemented using a commercial 90nm CMOS technology. The GRO, the PVDL and the DFFs are designed with analog flow and their layouts are drawn manually. The DSP unit is designed with digital design flow, and synthesized with the constraint of the 200MHz clock frequency. And timing check is passed for the digital part with back annotations extracted from the automatically generated layout, with the smallest timing slack of 2.97ns. The layout of the entire system takes an area of 0.013mm², as shown in Fig. 55. At 1.2V power supply and the sampling frequency of 200MHz, the power consumption is 2.05mW for the 400ps *EN* pulse width (0.92mW for the digital part and 1.13mW for the analog part).

Post-layout simulation is run for the proposed structure. Note that the simulation is a mixed-signal simulation: SPICE simulation for the analog-designed parts with R and C extracted from the layout, and Verilog simulation for the digital part. The inputs in the simulation are two pulse signals of 200MHz, one as *START* input and the other as *STOP* input. So the sampling frequency of the measurement is also 200MHz. Besides the resistance and capacitance extracted from the layout, the SPICE simulation for the analog part will include the effects of (1) the across-chip process variation, with the process variation models included in the foundry provided process design kit (PDK); (2) the thermal and flicker noise of transistors, using the

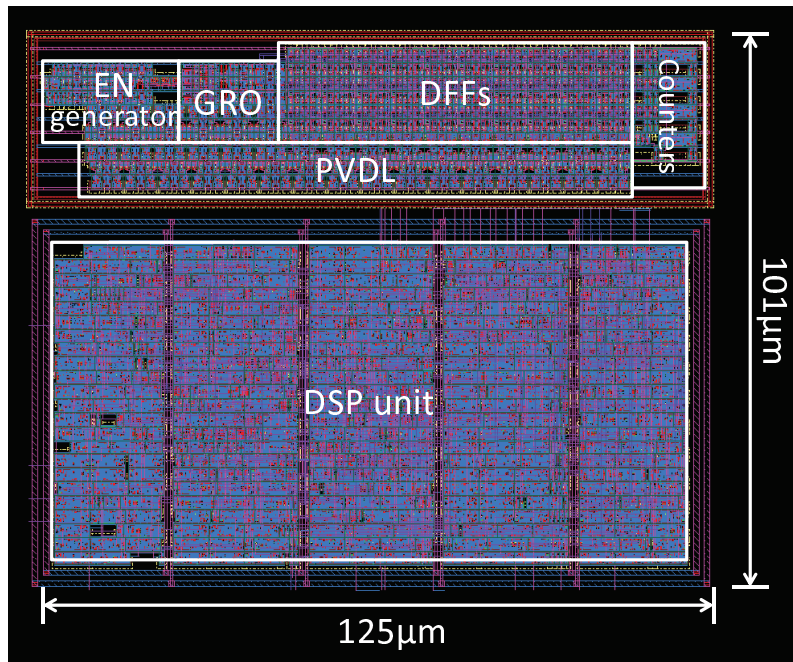


Fig. 55. Layout of the entire GRO-PVDL structure.

transient noise models included in the design PDK; (3) a manually injected power supply noise, using a Verilog-AMS module that models a white Gaussian noise whose standard deviation is 0.012V, i.e. 1% of the power supply voltage.

To find out the effective resolution, a single tone jitter input is first applied. The time difference between the rising edges of the two input signals is sinusoidal with a frequency of 500kHz and a peak-to-peak amplitude of 0.5ps, in addition to a DC level of 400ps. Fig. 56 shows the corresponding measurement result in both frequency and time domains. Note the DC offset is removed from the power spectral density (PSD) for clear observation of the quantization noise, and the PSD is generated from 16,384 samples of measurement, using Welch's averaged modified periodogram method of spectral estimation [43] with Hanning window. As a reference, the ideal PSD of the shaped quantization noise without the delay mismatch is given by the dash line, which

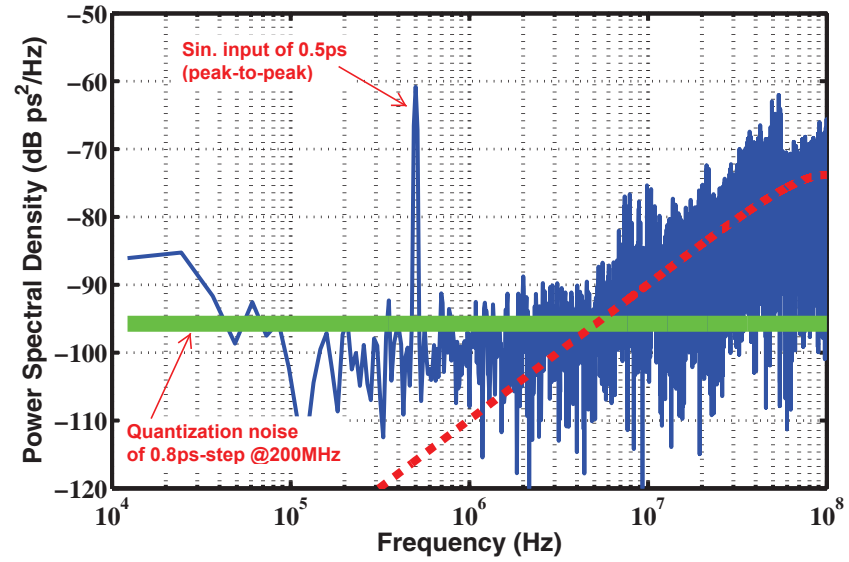
can be theoretically derived by [34]:

$$S_{\text{ideal}}(f) = \frac{1}{f_s} \frac{\tau_{\Delta}^2}{12} |1 - e^{-2\pi j \frac{f}{f_s}}|^2, \quad (3.21)$$

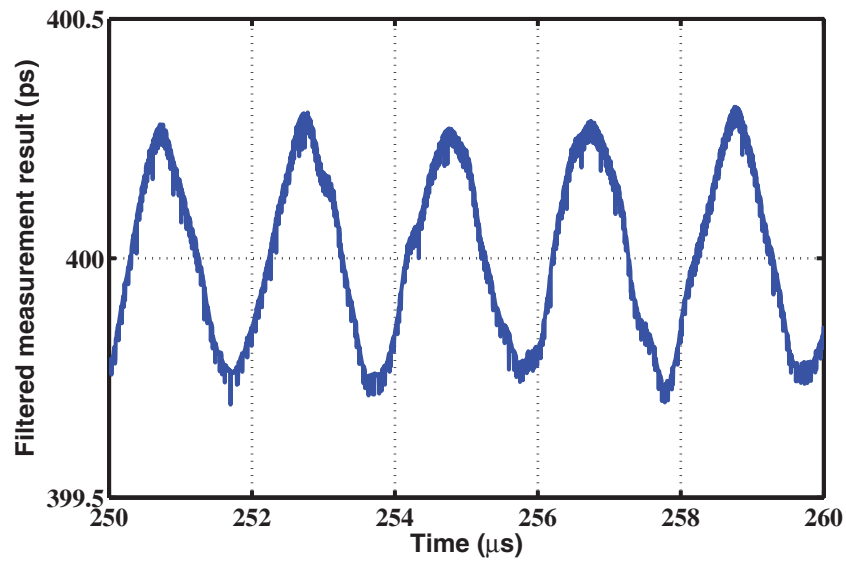
where $f_s=200\text{MHz}$ is the sampling frequency, and τ_{Δ} is the fine raw resolution, i.e. the difference of the average GRO cell delay and the average PVDL cell delay ($\tau_{\Delta} \approx 5\text{ps}$). As illustrated in Fig. 56(a), most of the quantization noise is pushed towards high frequencies (the Nyquist frequency =100MHz is half of the sampling frequency). At frequencies between 50kHz and 5MHz, the noise is comparable to the ideal quantization noise (without noise shaping) that is produced by a classical quantizer with 0.8ps steps and a sampling frequency of 200MHz, which is represented by the straight thick line. Therefore, the proposed on-chip jitter measurement can achieve an effective resolution of 0.8ps. Alternatively, if a classical quantizer with 5MHz sampling frequency is used as reference, then its quantization step should be reduced to around 130fs to achieve the equivalent noise level.

To see the noise contribution from different noise sources, a post-layout simulation with the same input jitter is also run as a reference, but without the device noise of transistors and the power supply noise. Comparing Fig. 56(a) with Fig. 57, we can see that among all the measurement noise, the noise at the lower frequencies is dominated by the flicker noise while the noise at the higher frequencies is dominated by the shaped quantization noise, and the frequency range that is most sensitive to the input jitter is from 50kHz to 5MHz, where the thermal noise and power supply noise is dominating.

A random jitter input is also applied to emulate the jitter of high speed signals. The jitter under measurement is composed of a random jitter and a 1MHz sinusoidal jitter. The random jitter is generated by filtering a white Gaussian noise using a low-pass filter (second-order IIR filter with cutoff frequency at 2MHz). And the simulated



(a)



(b)

Fig. 56. GRO-PVDL measurement for 0.5ps_{pp} sin. input: (a) PSD (b) transient view (after low-pass filtering).

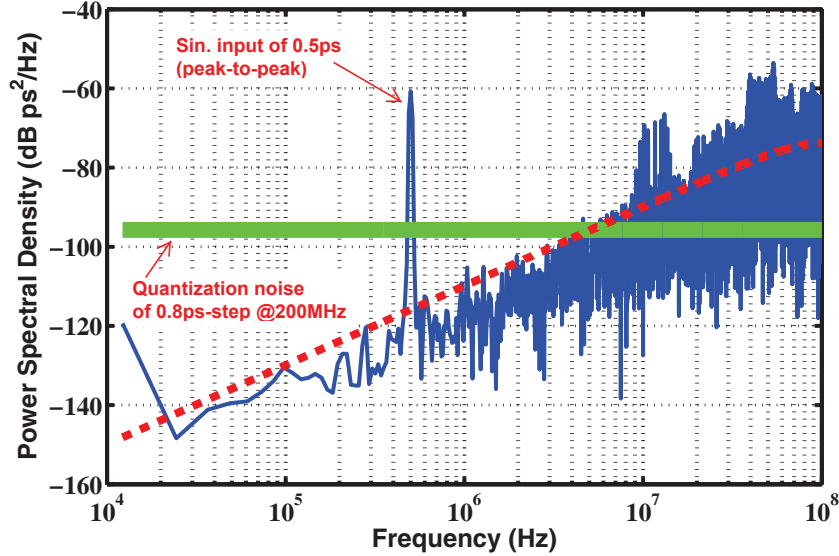


Fig. 57. PSD of the measurement for 0.5ps_{pp} sin. input (transistor noise and power supply noise are NOT included in the simulation).

measurement results are generated by low-pass filtering the GRO-PVDL output. In Fig. 58, the histogram of the measurement result within 50 μ s is compared with that of the input jitter. Given the input jitter is 7.31ps (RMS) and the measured result through simulation is 7.34ps (RMS) (the DC level is removed when calculating the jitter RMS), a relative error of 0.41% is obtained.

1. Delay mismatch analysis

Due to process variation, the cell delays of the GRO and the PVDL vary for different chips and different cells on one chip. For the proposed high-resolution jitter measurement technique, the effect of process variation has to be considered.

Chip-to-chip variations cause the average cell delays to deviate from the designed values (25ps for the GRO and 30ps for the PVDL). The deviations of the average delays $\overline{\tau_{\Delta}}$ and $\overline{\tau_{G}}$ can be calibrated. As introduced for the DSP unit, the fine code

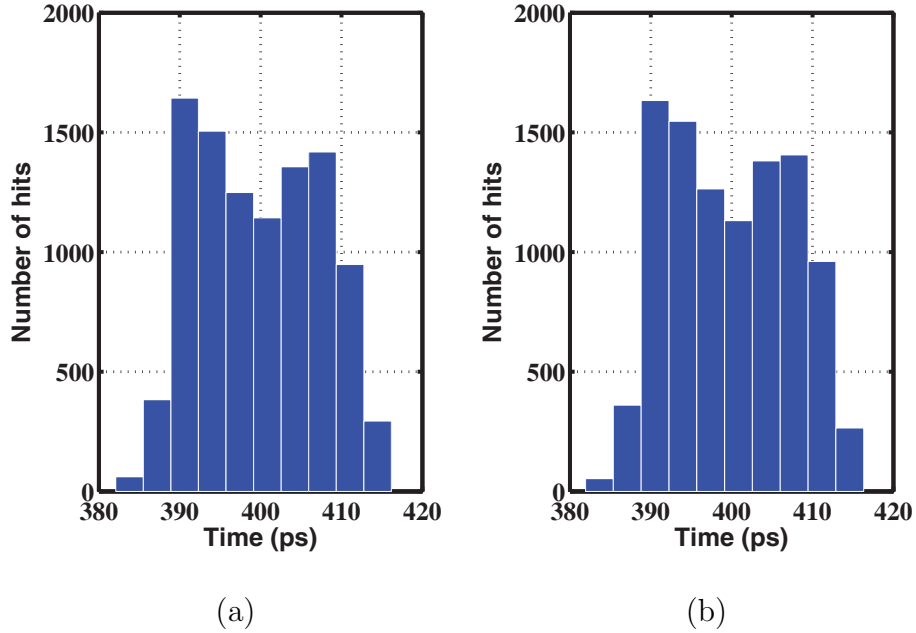


Fig. 58. The histograms of (a) input jitter (b) measurement result (after low-pass filtering).

generator implicitly calibrates the ratio between $\overline{\tau_{\Delta}}$ and $\overline{\tau_{G}}$ on the fly, and requires no external assistance or reconfiguration. As for the absolute value of $\overline{\tau_{G}}$, it can be calibrated off-line, by measuring a periodic signal whose pulse width is already known. According to [44], the calibration accuracy can be raised to an acceptable level by increasing the calibration time and averaging the calibration result.

On the other hand, across-chip variations cause the mismatch between the delays of the same type of cells, as is illustrated in Fig. 59. Sophisticated calibration schemes [45] can be applied, in order to calibrate the delay mismatch of the delay cells of the GRO/PVDL. However, such calibration schemes are usually very costly. Fortunately, the following analysis will show that the proposed GRO-PVDL structure has tolerance to the typical delay mismatch so that the accuracy of the measurement will not be remarkably degraded.

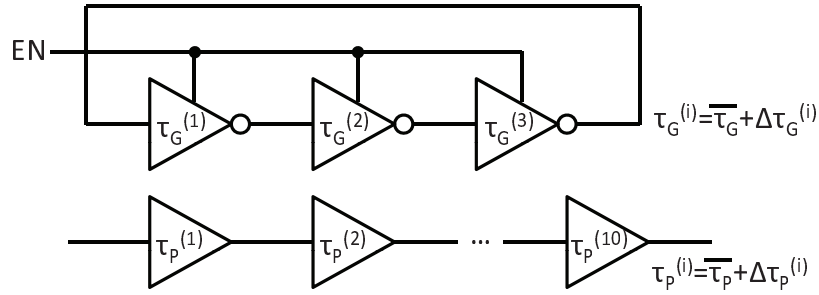


Fig. 59. Delay mismatch of the GRO and the PVDL.

The cell delay mismatches of the GRO and the PVDL lead to the unevenness and uncertainty of the resolution, and finally result in noise in the measurement. Given the discreteness and high nonlinearity of the system, it is difficult to analyze the influence of the delay mismatch analytically. Alternatively, simulations based on the behavioral model of the GRO-PVDL structure are carried out to analyze the degradation of the measurement accuracy due to the delay mismatch.

The behavioral model is built up using a Matlab program, in which the delay mismatch is modeled as normal distributions. And to focus on the influence of the delay mismatch, other noise sources are not modeled. As shown in Fig. 59, $\tau_G^{(i)}$ ($\tau_P^{(i)}$) is the delay of the i th cell of the GRO (the PVDL), and its deviation from the average cell delay is $\Delta\tau_G^{(i)}$ ($\Delta\tau_P^{(i)}$). In the following simulations, the average delay is set to be $\overline{\tau_G} = 25ps$ ($\overline{\tau_P} = 30ps$) for the GRO (the PVDL), while the mismatch $\Delta\tau_G^{(i)}$ ($\Delta\tau_P^{(i)}$) is generated with a normal distribution whose mean is 0, and standard deviation is $k_{\text{mis}}\overline{\tau_G}$ ($k_{\text{mis}}\overline{\tau_P}$). Note that the standard deviation is proportional to the cell delay with a ratio of k_{mis} , which represents the level of delay mismatch. Through SPICE-level Monte Carlo simulation with the nominal process variation given in the PDK, we can find the typical value of k_{mis} is 3.0%. Therefore, the PSD of the measurement is obtained using the behavioral simulation with $k_{\text{mis}}=3.0\%$, as in Fig. 60. In order to explore the tolerance of the proposed GRO-PVDL structure

to the delay mismatch, behavioral simulations are also run for mismatch levels larger than the nominal value: $k_{\text{mis}}=5\%$ and 10% , and the corresponding PSDs are shown in Fig. 60, too. In the simulation, the sampling frequency of the measurement is 200MHz. The time difference between the rising edges of the two input signals has a DC value of 400ps, plus a sinusoidal signal with a frequency of 500kHz and a peak-to-peak amplitude of 0.5ps. For generating each PSD, 65,536 samples of measurement are used.

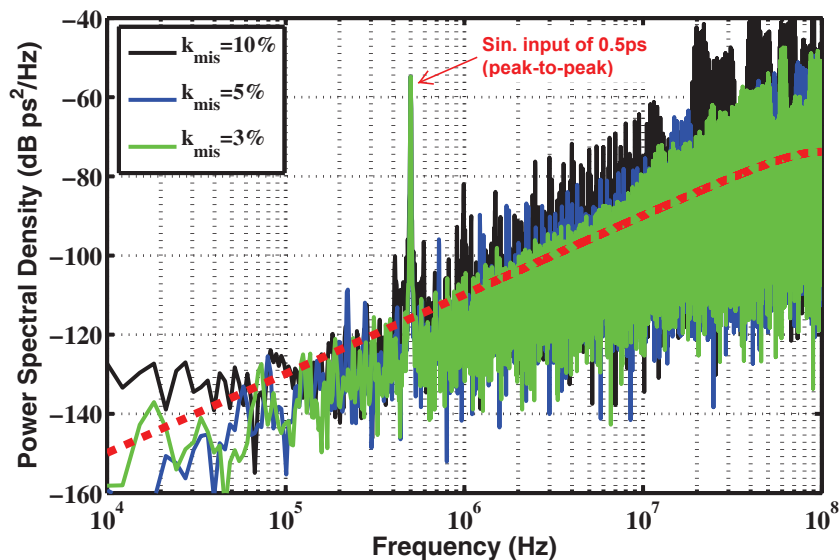


Fig. 60. Simulated PSD with different delay mismatches ($k_{\text{mis}} = 3\%$, 5% , 10%).

Compared with the ideal level of the shaped quantization noise, the noise exacerbation due to delay mismatch lies mostly in high frequencies. This is because the position of the delay cell that is hit at the beginning and ending of each measurement is randomly distributed on the GRO and the PVDL, and thus the resulted mismatch errors are canceled between different measurements, especially for a long measurement time. Considering the high frequency noise will be filtered by the subsequent

low-pass filter, the degradation of the effective resolution is very limited. The effective resolutions corresponding to different levels of delay mismatch can be obtained by comparing the noise at frequencies lower than 5MHz with the ideal quantization noise produced by a classical quantizer with a sampling frequency of 200MHz, as listed in Table II.

Table II. Effective resolutions for different k_{mis} .

k_{mis}	Effective resolution @200MHz (ps)
3% (nominal)	0.7
5%	1.0
10%	1.5

2. Specification comparison

Table III compares the specifications of this work with those in earlier studies. Here we focus on the comparison with two previous techniques that also utilize the quantization noise shaping through the GRO principle: the multi-path GRO in [34] and the Vernier GRO in [35].

[34] proposes a multi-path structure that improves the raw resolution of GRO from 30-35ps to 6ps using a 47-stage GRO connected by multiple paths. A multi-path GRO has tens of delay stages, and each stage has multiple inputs and one output. The signal paths that connect the GRO taps need to be carefully designed, otherwise the complex path connection may easily lead to the malfunction of the multi-path GRO, such as oscillating at a wrong frequency due to the domination of small oscillation loops inside the GRO. And the large number of delay stages in multi-path GROs also increases the hardware overhead. As listed in Table III, the multi-path GRO in [34]

Table III. Comparison of Specifications.

Ref.	[34]	[35]	This work
Process (nm)	130	90	90
Raw resol. (ps)	6	5.8	5
Effective resol. (ps)	1@50MHz	3.2@25MHz	0.8@200MHz
Sampling freq. (MHz)	50	25	200
Area (mm ²)	0.04	0.027	0.013
Power (mW)	2.2 (1.5V)	3.6 (1.2V)	2.05 (1.2V)
Technique	multi-path GRO	Vernier GRO	GRO-PVDL

takes an area of 0.04mm², while the GRO-PVDL in this work only takes 0.013mm². After performing an ideal scaling (130/90)² to include the process difference (130nm for [34] vs 90nm for this work), the GRO-PVDL still takes 32% less area than the multi-path GRO.

On the other hand, the Vernier GRO achieves a raw resolution of 5.8ps by means by two Vernier-style GROs. But its drawback is that the measurement requires a time much longer than the *EN* pulse width, which largely limits its highest sampling frequency. Considering the advantage of noise shaping can only be achieved with large over-sampling rate, the limited sampling frequency of the Vernier GRO (25MHz) does not allow much room for the improvement of the effective resolution. Therefore, the improvement from the raw resolution 5.8ps to the effective resolution 3.2ps is only 45%. In contrast, the proposed GRO-PVDL in our work can achieve a sampling frequency of 200MHz, and thus the raw resolution of 5ps is improved by 84% to achieve an effective resolution of 0.8ps.

Compared with these two previous structures, the GRO-PVDL structure not only has lower hardware overhead, but also has a higher sampling frequency. Moreover,

the digital circuits in the GRO-PVDL take a larger proportion in the total hardware, which improves the overall robustness of the system.

E. Summary

A novel structure of GRO-PVDL is proposed for the purpose of on-chip jitter measurement of high-speed signals. The structure is composed of two level: the first level is the GRO providing the coarse measurement; and the second level further measures the residue from the first level with the fine resolution. The GRO-PVDL structure improves the raw resolution of the GRO through the Vernier-style structure at the second level that reuses the GRO on the first level in addition to a PVDL. At the same time, the GRO feature of quantization noise shaping is also preserved by the GRO-PVDL, and thus an even finer effective resolution can be achieved. The proposed structure also includes a pipeline DSP unit with online calibration between the fine resolution and the coarse resolution. Besides, the proposed GRO-PVDL is shown to be highly tolerable to the delay mismatch, from the analysis based on the behavioral model. Implemented with a commercial 90nm CMOS technology, the GRO-PVDL can achieve a sampling frequency of 200MHz and an effective resolution of 0.8ps.

CHAPTER IV

IN-SITU TEST OF ALL DIGITAL PLLS

Unlike the in-situ jitter measurement technique proposed in Chapter III, this chapter introduces an in-situ test scheme that is specifically designed for a specific types of AMS circuits: all digital PLLs (ADPLLs). The proposed in-situ test scheme is based on the loop reconfiguration of ADPLLs, which takes advantage of the close interaction between the key analog building blocks and the digital loop filter. The work in this chapter is also published in [46].

A. Introduction

Ensuring analog/mixed-signal design robustness and providing low-cost built-in test solutions remain as a significant challenge due to the complex analog nature of circuit operation [47] [48] [49]. The performance improvement of digital transistors via scaling has stimulated wide interest in digitally intensive analog implementations [50] [51]. This has not only provided appealing new design tradeoffs, but also motivated us to exploit such implementation style for novel analog built-in self test (BIST) solutions.

A digital-like BIST approach is proposed to the test and diagnosis of the output jitter, a key complex RF analog performance, of recent all-digital PLL(ADPLL) designs [51], whose block diagram is shown in Fig. 61, where the phase of the input (ϕ_R , the reference phase) and the phase of the output (ϕ_V , the variable phase) are normalized by their own periods. N is the frequency control word which defines the frequency ratio between the output and the input, and it could be a fractional number. An accumulator generates the integral part of ϕ_V and a time-to-digital converter (TDC) provides its fractional part. The phase error between ϕ_V and $N \cdot \phi_R$ is filtered by a loop filter and adjusts a digitally-controlled oscillator (DCO) in a negative

feedback manner such that $\phi_V \approx N \cdot \phi_R$.

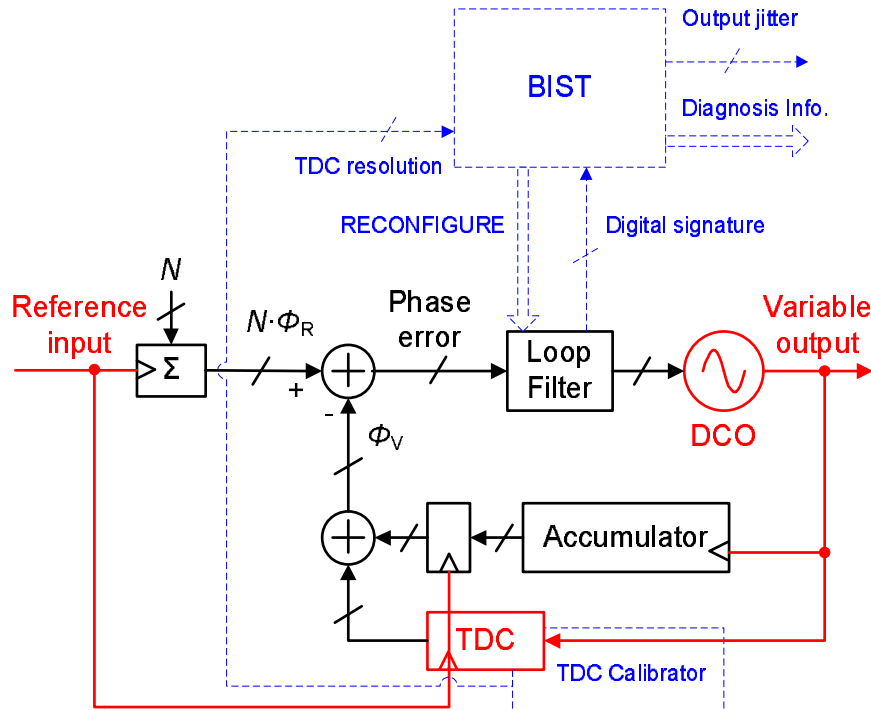


Fig. 61. All-digital PLL block diagram including BIST.

Since the digital processing and control blocks are implemented in robust digital logic, we target the jitter performance degradation introduced by parametric variations of key analog blocks including the TDC and the DCO, as well as the reference jitter. The prediction of the output jitter is based upon processing low-frequency phase error signals, the test signatures, in digital form. Unlike prior work [1] that also utilizes digital signatures for jitter testing, the novel employment of loop filter reconfiguration and on-chip TDC calibrator makes the BIST scheme proposed in this chapter possible to provide reliable diagnosis and test under multiple analog performance perturbations. The digital-like design implementation has enabled easy reconfiguration and led to the low cost of the proposed approach.

In the proposed BIST scheme, multiple digital signatures are extracted for observing the “syndromes” under different loop filter configurations. By means of the transfer function analysis, the mapping from the signatures to the output jitter is precalculated and stored in the BIST scheme. Moreover, for the purpose of diagnosis, the signatures can also be mapped to the levels of different noise sources, with the assistance of the TDC calibrator. The hardware overhead of the BIST is mainly from the digital signal processing on the signatures and additional filters, which is relatively small compared to the whole ADPLL system, and could be further reduced through the reuse of on-chip processor.

B. Principle of jitter estimation and diagnosis

In this section, the noise models used in this work are presented and the signal analysis that leads to the proposed BIST and diagnosis.

1. Noise model

The three noise sources in the ADPLL, the reference clock jitter, the TDC quantization noise and the DCO phase noise, are mathematically modeled in the frequency domain, to help analyze the output jitter.

The reference clock is usually generated by a crystal oscillator, and thus provides a single-tone spectrum with little spectral spread. Since the reference phase noise has a relatively flat spectrum, it can be treated as constant from dc to half of the sampling frequency, the reference frequency f_{REF} , whose power spectral density (PSD) is:

$$\Phi_{\text{REF}}(\Delta f) = L_{\text{R}}, \quad (4.1)$$

where Δf is the frequency offset ranging from $-f_{\text{REF}}/2$ to $f_{\text{REF}}/2$, and L_{R} is a

constant describing the noise level of the reference phase noise.

Although in the realistic situation its low frequency components have higher slopes, their bandwidth is so small that the corresponding frequency drifts hardly show up in the concerned time, such as one GSM burst: $577 \mu\text{s}$ or one WCDMA slot: $667 \mu\text{s}$.

The second noise source is the TDC quantization noise due to its time resolution. Similar to the quantization noise of analog-to-digital converter (ADC), the TDC quantization noise can be modeled as an additive random variable with uniform distribution and white noise spectral characteristic. Its effective time jitter J_{TDC} (RMS) can be expressed as [52]:

$$J_{\text{TDC}} = \Delta t_{\text{res}} / \sqrt{12}, \quad (4.2)$$

where Δt_{res} is the time resolution of the TDC. And its PSD normalized to the DCO phase is:

$$\begin{aligned} \Phi_{\text{TDC}}(\Delta f) &= L_{\text{TDC}} \\ &= (2\pi J_{\text{TDC}} f_{\text{DCO}})^2 / (f_{\text{REF}}) \\ &= 4\pi^2 N^2 J_{\text{TDC}}^2 f_{\text{REF}}, \end{aligned} \quad (4.3)$$

where L_{TDC} is the constant noise level for Δf from $-f_{\text{REF}}/2$ to $f_{\text{REF}}/2$, and f_{DCO} is the DCO frequency.

The assumption of uniform distribution means that the TDC generates different quantization levels with equal probabilities, which is true except for some special situations: e.g. an integral N that results in a bang-bang phase detection.

Besides the noise from the reference clock and the TDC, the DCO is another major noise source. The phase noise spectrum of an oscillator can be divided into three segments [53], as is shown in Fig. 62. The $1/\Delta f^2$ segment is called the wander noise, generally referred to as the thermal noise and caused by the white-noise fluctuation

of the oscillating frequency. The DCO quantization noise and the DCO power supply noise will also cause the wander noise [54] [55]. The $1/\Delta f^3$ segment at lower offset frequencies is called the flicker noise, and the flat segment is the thermal electronic noise due to external sources, such as an output buffer.

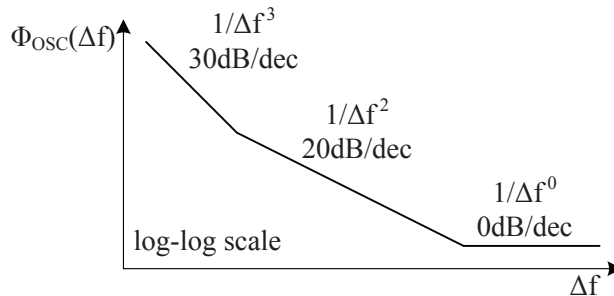


Fig. 62. The phase noise spectrum of a typical oscillator.

Given the flicker noise and the wander noise are the dominant noise mechanisms of the DCO within the frequency range concerned, the PSD of the DCO phase noise can be modeled as:

$$\Phi_{\text{DCO}}(\Delta f) = L_{\text{F}}/\Delta f^3 + L_{\text{W}}/\Delta f^2, \quad (4.4)$$

where L_{F} and L_{W} are the noise levels of the flicker noise and the wander noise, respectively.

2. Transfer function analysis

When the ADPLL is locked in the tracking mode, linear frequency-domain transfer functions are applicable under the small signal assumption. The s -domain model of an ADPLL system is shown in Fig. 63, where $\phi_{\text{n,REF}}$ is the phase noise from the reference input, $\phi_{\text{n,TDC}}$ is from the TDC noise and $\phi_{\text{n,DCO}}$ is the phase noise of the DCO. In the proposed BIST scheme, the loop filter is separated into two cascaded filters, LF1 and

LF2. ϕ_E and $\hat{\phi}_E$ are the potential signatures. The DCO gain calibration gives \hat{K}_{DCO} as an estimate of the DCO gain K_{DCO} . The coefficient r indicates the calibration error of the TDC resolution.

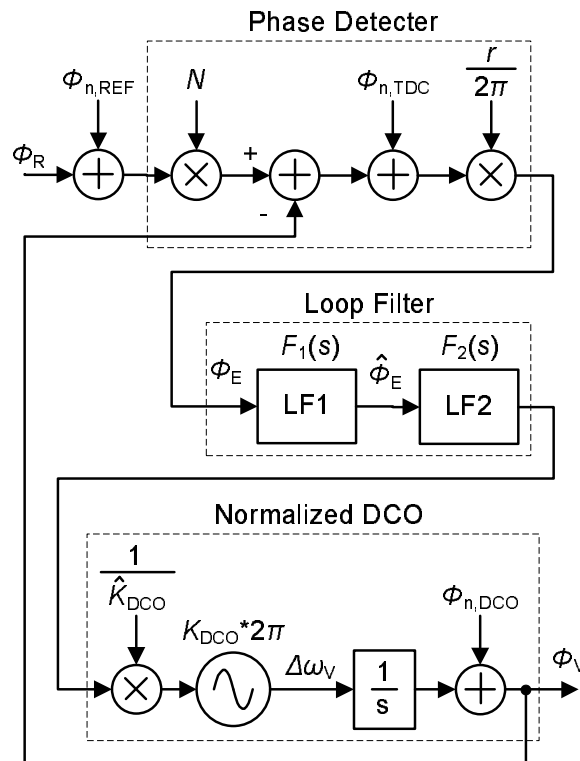


Fig. 63. s -domain model of ADPLL including noise sources.

The open-loop transfer function is defined as:

$$H_{\text{ol}}(s) = \frac{1}{s} F_1(s) F_2(s) r K_{\text{DCO}} / \hat{K}_{\text{DCO}} \quad (4.5)$$

where $F_1(s)$ and $F_2(s)$ are the transfer functions of LF1 and LF2. The z -domain transfer functions of the digital filters can be converted to s -domain by substituting z with e^{s/f_R} . $K_{\text{DCO}}/\hat{K}_{\text{DCO}} \approx 1$ and $r \approx 1$ are assumed because of the DCO gain calibration.

Table IV. Transfer functions from noise sources to output phase noise and digital signatures

from to	Reference	TDC	DCO
$\phi_{n,O}$	$H_{R2O}(s) = \frac{NH_{ol}(s)}{1 + H_{ol}(s)}$	$H_{T2O}(s) = \frac{H_{ol}(s)}{1 + H_{ol}(s)}$	$H_{D2O}(s) = \frac{1}{1 + H_{ol}(s)}$
ϕ_E	$H_{R2P1}(s) = \frac{N}{1 + H_{ol}(s)}$	$H_{T2P1}(s) = \frac{1}{1 + H_{ol}(s)}$	$H_{D2P1}(s) = \frac{1}{1 + H_{ol}(s)}$
$\hat{\phi}_E$	$H_{R2P2}(s) = \frac{NF_1(s)}{1 + H_{ol}(s)}$	$H_{T2P2}(s) = \frac{F_1(s)}{1 + H_{ol}(s)}$	$H_{D2P2}(s) = \frac{F_1(s)}{1 + H_{ol}(s)}$

The closed-loop transfer functions from the noise sources to the output phase noise $\phi_{n,O}$ and the potential signatures are listed in Table IV. For typical noise levels and loop settings, the TDC noise has the least influence among the three noise sources, and the noise components at frequencies higher than $f_R/2$ is tens of dBs smaller and their spectrum aliasing is therefore neglectable.

Given Table IV, the single-sided PSD of the output phase noise can be expressed as:

$$\begin{aligned}
 S_{\phi_{n,O}}(f) = & \Phi_{\text{REF}}(f) |H_{R2O}(2\pi jf)|^2 \\
 & + \Phi_{\text{TDC}}(f) |H_{T2O}(2\pi jf)|^2 \\
 & + \Phi_{\text{DCO}}(f) |H_{D2O}(2\pi jf)|^2
 \end{aligned} \tag{4.6}$$

Referring back to Eq. (5.5), (5.4), (5.3), Eq. (4.6) can be transformed as:

$$S_{\phi_{n,O}}(f) = (NL_R + L_T)|H_{T2O}(2\pi jf)|^2 + L_W|H_{D2O}(2\pi jf)|^2/f^2 + L_F|H_{D2O}(2\pi jf)|^2/f^3 \quad (4.7)$$

According to the Parseval theorem [52], the power of the output phase noise is $\overline{\phi_{n,O}^2} = 2 \int_0^{f_R/2} S_{\phi_O}(f)df$. Because signals are sampled at the reference frequency f_R in digital blocks, the integration range is from 0 to $f_R/2$ to meet Nyquist theorem. The power of the output phase noise in the normal working mode can be written as:

$$\overline{\phi_{n,O}^2} = C_{T2O}(NL_R + L_T) + C_{W2O}L_W + C_{F2O}L_F, \quad (4.8)$$

where:

$$C_{T2O} = 2 \int_0^{f_R/2} |H_{T2O}(2\pi jf)|^2 df \quad (4.9)$$

$$C_{W2O} = 2 \int_0^{f_R/2} |H_{D2O}(2\pi jf)|^2/f^2 df \quad (4.10)$$

$$C_{F2O} = 2 \int_0^{f_R/2} |H_{D2O}(2\pi jf)|^2/f^3 df \quad (4.11)$$

It can be seen from Eq. (4.8) that the power of the output phase noise is a linear combination of the noise levels of the noise sources.

In the proposed BIST scheme, three signatures under different configurations are collected and processed. Similar to the derivation of $\overline{\phi_{n,O}^2}$, the power of each signature can also be approximated with a linear combination of the noise levels of the noise sources. Therefore, we can have:

$$\begin{bmatrix} \overline{\hat{\phi}_{SIG}^{(1)2}} \\ \overline{\hat{\phi}_{SIG}^{(2)2}} \\ \overline{\hat{\phi}_{SIG}^{(3)2}} \end{bmatrix} = \begin{bmatrix} C_T^{(1)} C_W^{(1)} C_F^{(1)} \\ C_T^{(2)} C_W^{(2)} C_F^{(2)} \\ C_T^{(3)} C_W^{(3)} C_F^{(3)} \end{bmatrix} \begin{bmatrix} NL_R + L_T \\ L_W \\ L_F \end{bmatrix}, \quad (4.12)$$

where $\overline{\hat{\phi}_{\text{SIG}}^{(i)2}}$ is the average power of the selected signature for the i th configuration, and $C_{\text{T}}^{(i)}$, $C_{\text{W}}^{(i)}$, $C_{\text{F}}^{(i)}$ are the corresponding coefficients. So the noise levels can be given by:

$$\begin{bmatrix} NL_{\text{R}} + L_{\text{T}} \\ L_{\text{W}} \\ L_{\text{F}} \end{bmatrix} = \begin{bmatrix} C_{\text{T}}^{(1)} C_{\text{W}}^{(1)} C_{\text{F}}^{(1)} \\ C_{\text{T}}^{(2)} C_{\text{W}}^{(2)} C_{\text{F}}^{(2)} \\ C_{\text{T}}^{(3)} C_{\text{W}}^{(3)} C_{\text{F}}^{(3)} \end{bmatrix}^{-1} \begin{bmatrix} \overline{\hat{\phi}_{\text{SIG}}^{(1)2}} \\ \overline{\hat{\phi}_{\text{SIG}}^{(2)2}} \\ \overline{\hat{\phi}_{\text{SIG}}^{(3)2}} \end{bmatrix} \quad (4.13)$$

Substituting Eq. (4.13) back into Eq. (4.8), and the power of phase noise at the output can be given by:

$$\overline{\phi_{\text{n,O}}^2} = \begin{bmatrix} C_{\text{T2O}} \\ C_{\text{W2O}} \\ C_{\text{F2O}} \end{bmatrix}^T \begin{bmatrix} C_{\text{T}}^{(1)} C_{\text{W}}^{(1)} C_{\text{F}}^{(1)} \\ C_{\text{T}}^{(2)} C_{\text{W}}^{(2)} C_{\text{F}}^{(2)} \\ C_{\text{T}}^{(3)} C_{\text{W}}^{(3)} C_{\text{F}}^{(3)} \end{bmatrix}^{-1} \begin{bmatrix} \overline{\hat{\phi}_{\text{SIG}}^{(1)2}} \\ \overline{\hat{\phi}_{\text{SIG}}^{(2)2}} \\ \overline{\hat{\phi}_{\text{SIG}}^{(3)2}} \end{bmatrix} \quad (4.14)$$

In fact, Eq. (4.14) gives the mapping from the signatures to the output jitter, and Eq. (4.13) gives out the mapping from the signatures to the noise-related parameters of the analog blocks. Note that the coefficients in Eq. (4.13), (4.14) are determined by transfer functions; C_{T2O} , C_{W2O} , C_{F2O} are calculated from the transfer functions for the normal working configuration, while $C_{\text{T}}^{(i)}$, $C_{\text{W}}^{(i)}$, $C_{\text{F}}^{(i)}$ need to be calculated from the transfer functions for the three BIST configurations. It is important to note that these transfer functions are fully determined by digital logic that is assumed to be robust and hence independent of analog block variations. This implies that all the information required by the proposed scheme, i.e. the coefficient matrices in Eq. (4.13) and Eq. (4.14), can be precomputed and stored on-chip in the form of constants. Moreover, the noise level of TDC can be directly calculated by Eq. (4.2) and Eq. (5.4), given the TDC resolution provided by the TDC calibrator.

C. BIST scheme

The block diagram of the proposed BIST is shown in Fig. 64. In the BIST mode, the TDC calibrator reconfigures the TDC delay chain into a ring oscillator to provide on-line calibration of the resolution. The loop filter is separated into two cascaded filters and provides two internal signals ϕ_E and $\hat{\phi}_E$ to be potential signatures. The loop filter characteristics can be altered in three pre-stored reconfigurations. Reconfiguration exposes the parametric fluctuation of the TDC, the DCO and the reference signal to digital test signatures with varying sensitivities so as to provide sufficient information for test and diagnosis. Each pre-store configuration is selected by setting *config_num* and forcing the loop to settle. The reconfiguration controller also designates the digital signature by *sig_sel*. The estimation mapper receives the designated signature, processes it and stores the processing result. After the ADPLL has run under the three configurations, the three signatures are all collected by the estimation mapper. Together with the TDC resolution provided by the TDC calibrator, the estimated output jitter, TDC resolution, noise performances of the DCO and reference that are causing the output jitter level (i.e. diagnosis) are outputted. Noting that the loop filter is configured as two cascaded filters only at BIST mode, it could be configured to any other forms at normal working state.

1. Reconfigurable loop filters

As mentioned before, the loop filter is separated into two cascaded filters, LF1 and LF2. LF2 is a first-order IIR for building a type-II loop. LF1 provides an option for higher-order loop by a cascade of single-pole IIR filters, which is unconditionally stable. Any of the IIR filters could be bypassed to adjust the loop order. The z -

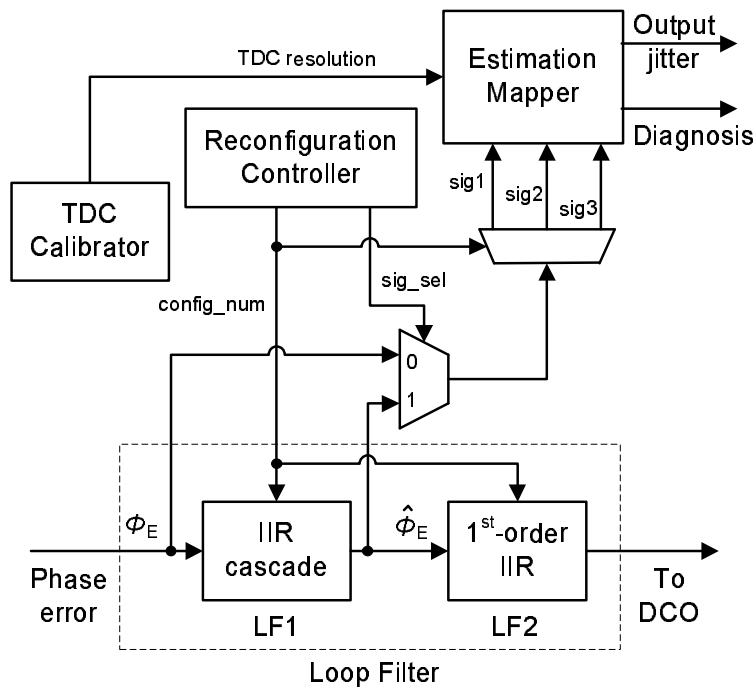


Fig. 64. BIST block diagram.

domain transfer functions of LF1 and LF2 are:

$$F_1(z) = \prod_{i=0}^3 \frac{\lambda_i}{z - (1 - \lambda_i)} \quad (4.15)$$

$$F_2(z) = \alpha + \frac{\varrho}{z - 1} \quad (4.16)$$

where λ_i , α and ϱ are filter coefficients. LF1 and LF2 can be easily reconfigured by changing their coefficients. These coefficients are set to integer powers of two, such that the multiplications can be easily implemented by bit-shifters.

The purpose of loop filter reconfiguration is to distinguish the contribution of each noise source to different signatures as much as possible. In order to optimize the loop configurations for BIST purpose, the sensitivity of the signatures to the objective

performance should be maximized. The sensitivity function can be defined as:

$$M = \left| \frac{\overline{\partial \hat{\phi}_{\text{SIG}}^{(1)2}}}{\partial \phi_{\text{n,O}}^2} \right| + \left| \frac{\overline{\partial \hat{\phi}_{\text{SIG}}^{(2)2}}}{\partial \phi_{\text{n,O}}^2} \right| + \left| \frac{\overline{\partial \hat{\phi}_{\text{SIG}}^{(3)2}}}{\partial \phi_{\text{n,O}}^2} \right|, \quad (4.17)$$

which can be calculated from Eq. (4.14).

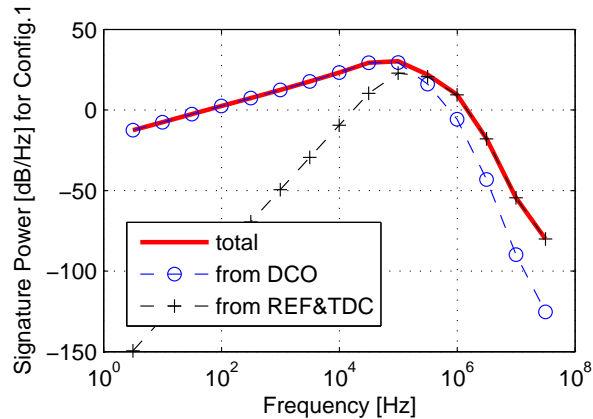
An example of configuration setup is given in Table V. The reference frequency is 26MHz and $N = 96.15$. Under such configuration setup and assuming the noise sources have typical noise levels, the power spectra of the three signatures are drawn in Fig. 65. For Config. 1, contributions from the noise sources to the signature power are well balanced. The signature power mainly comes from the DCO noise for Config. 2, while for Config. 3 the noise from the reference input and the TDC dominates the signature power.

Table V. An example of configuration setup. (LF1 is bypassed for Config. 3)

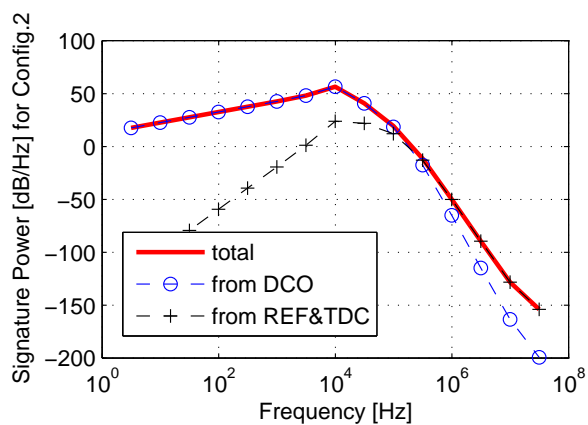
Config.	λ_1	λ_2	λ_3	λ_4	α	ρ	Sig.
1	2^{-3}	2^{-3}	2^{-3}	2^{-4}	2^{-7}	2^{-15}	$\hat{\phi}_{\text{E}}$
2	2^{-6}	2^{-6}	2^{-6}	2^{-7}	2^{-10}	2^{-20}	$\hat{\phi}_{\text{E}}$
3	—	—	—	—	2^{-4}	2^{-10}	ϕ_{E}

2. TDC calibrator

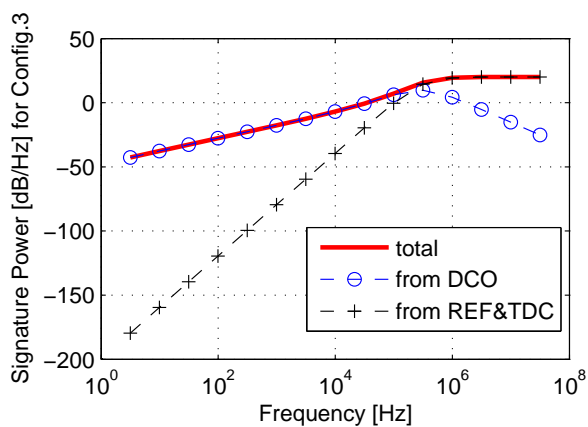
The TDC resolution is subject to change with process variations, so it needs to be calibrated for each individual chip. In order to achieve that, the TDC delay cells can be reconfigured into a ring oscillator by inserting inverters to connect the head and the tail of the TDC delay line. And the total number of the inverters in the ring oscillator is odd to ensure proper oscillation. The oscillating frequency is calculated by counting the number of its clock cycles in certain reference clocks, as shown in



(a) Config. 1 in Table V.



(b) Config. 2 in Table V.



(c) Config. 3 in Table V.

Fig. 65. The composition of the signature power spectral density.

Fig. 66.

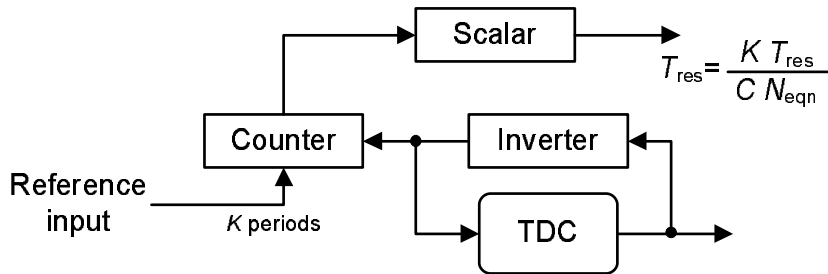


Fig. 66. TDC resolution calibration.

Supposing there are N_{eqn} equivalent inverters in the configured oscillation loop, and for K reference clock periods we count C oscillation cycles, then the TDC resolution T_{res} (i.e. the average inverter delay) can be calculated as:

$$T_{\text{res}} = \frac{K \cdot T_{\text{R}}}{C \cdot N_{\text{eqn}}}, \quad (4.18)$$

where T_{R} is the reference clock period.

3. Hardware overhead

The hardware overhead of the proposed BIST scheme is mainly from the digital signal processing on the signatures and additional filters. Table VI compares the areas of the BIST scheme and the ADPLL system. It can be seen that the BIST area is 9.3% of the ADPLL area. Moreover, [51] has provided a system-on-chip solution including both the ADPLL and an on-chip digital signal processor (DSP). Thus, the cost of signature processing can be further saved by reusing the on-chip DSP, and the area ratio of the BIST to the ADPLL can be reduced to 1.3%.

Table VI. Hardware overhead

		area [μm^2]
ADPLL	digital	70000
	TDC	140000
	DCO	165000
BIST	signature processing	30000
	additional filters	5000

D. Simulation results

In the section, the proposed BIST scheme is evaluated by a behavioral modeling and simulation environment, and Monte Carlo analysis is carried out to show the accuracy of the BIST results.

1. Setup of simulation environment

The simulation environment is based on a standard event-driven simulator, Verilog. The whole ADPLL system, including digital logic and control, and analog circuits like TDC and DCO, is integrated in a Verilog-based simulation environment. The RTL description of Verilog for digital circuits is a behavioral model that can simulate the digital circuits free from error. On the other hand, the behavioral models of the analog circuits and input signals need to be built up carefully to include the factors that will influence the noise performance of the ADPLL.

The signals at the interfaces of the IIR filters have a width of 23 bits, 8 as the integral part and 15 as the fractional part. Most digital parts are synchronized by the reference clock. The accumulator is working at the frequency of the output clock. A $\Sigma\Delta$ dithering working at the 1/4 of the output clock frequency for removing spurs

in the PSD of output phase noise is also modeled.

Time-domain phase noises are simulated in the models of both the DCO and the reference input. The flat segment is modeled as a white Gaussian noise. The wander noise is modeled as an accumulative jitter, which is the integration of a white Gaussian noise. The flicker noise is modeled by a weighted sum of low-pass filters through time-domain filtering of white noise [56]. Effect of the TDC nonlinearity is included in the TDC model, with its differential nonlinearity (DNL) and integral nonlinearity (INL) both lower than $0.7LSB$. It is assumed that the TDC resolution has a random error with 5% due to the accuracy of the TDC calibrator. The center frequency of reference clock is set to $10MHz$ and the feedback division ratio N is set to 240.1875 so that output clock is about $2.4GHz$.

2. Monte Carlo analysis

Based on the described simulation environment, Monte Carlo analysis is carried out to evaluate the BIST scheme. In the analysis, the variational parameters are the phase noise levels of the DCO and the reference input, the TDC resolution and its DNL and INL. For each variational parameter, the variance is set to $3\sigma = 10\%$ for a 90nm CMOS technology [57].

2,000 Monte-Carlo simulation samples are generated by conducting the event-driven simulation for the ADPLL system. The RMS jitter of the output clock during $10ms$ is estimated by the proposed BIST method. In the BIST mode, the ADPLL runs $10ms$ for each configuration.

In order to evaluate the accuracy of test and diagnosis results, the output jitter, the phase noise level of the reference input and the DCO are also directly measured in Monte Carlo analysis. For the output jitter, the distributions of direct measurement and BIST estimation are compared in Fig. 67. According to the pass/fail line

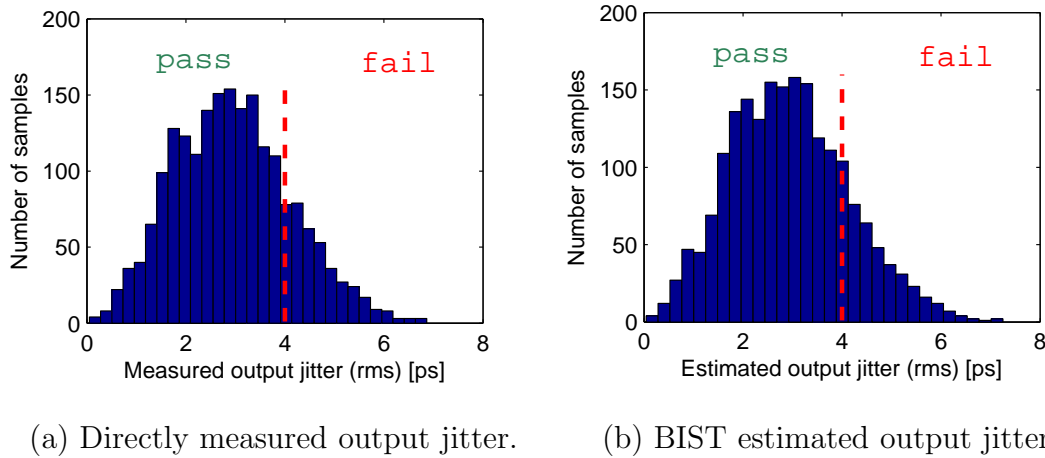


Fig. 67. BIST estimation VS. directly measurement.

(jitter=4ps) in the figures, the defect escape rate is 1.5% and the yield loss rate is 2%. Fig. 68 and Fig. 69 show the accuracy of output phase noise estimation. As can be seen, the overall relative error is roughly 5%. The highest relative error of 20% occurs when output jitter is smaller than 1ps. The average relative error comes to its minimal point when output jitter is around 5ps. This estimation accuracy is not good enough for marginal production test. However, this BIST scheme could effectively detect large deviations of the analog modules from their nominal values, as well as diagnose the noise sources in the ADPLL.

For each test case, the BIST scheme proposed in [1], where no TDC resolution and loop filter reconfiguration is employed, is also simulated. Fig. 70 compares the results of the proposed BIST scheme and the previous one. For the previous BIST scheme, the estimation error increases in proportion to the contribution of the reference noise to the output, while for the proposed BIST scheme, the error always keeps at a low level. This is because the proposed scheme is able to separate the influences of the reference noise and the DCO noise.

The diagnosis results are presented in Fig. 71, versus the percentage contribu-

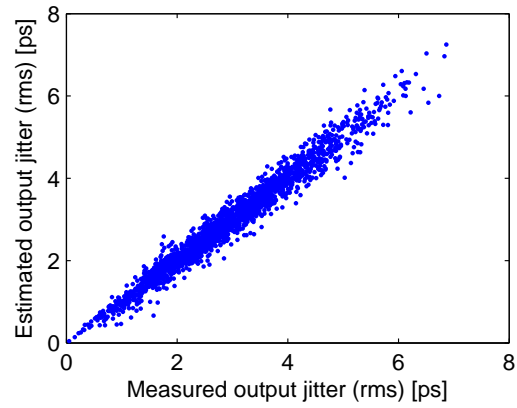


Fig. 68. Estimated jitter compares with measured jitter

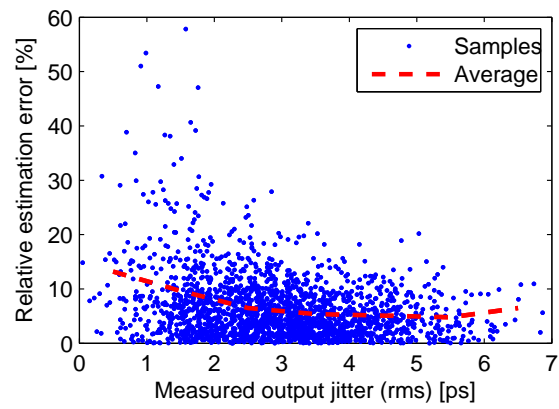


Fig. 69. Relative estimation error. Error is averaged in each 1ps interval of the output jitter.

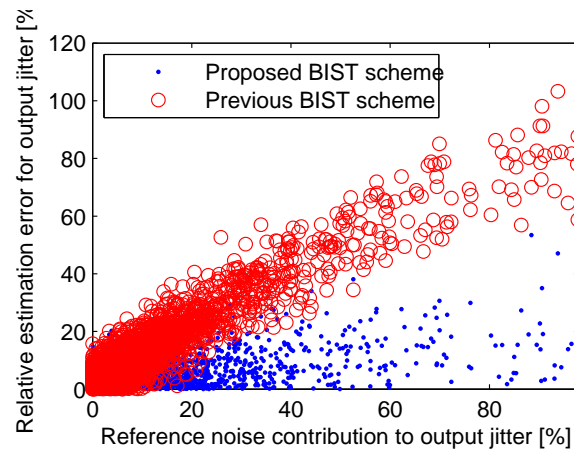


Fig. 70. The proposed BIST scheme VS. the BIST scheme in [1].

tion of the corresponding noise source to the output jitter. For the diagnosis of the reference noise and the TDC noise, the relative estimation error is below 5%. The DCO noise diagnosis only has a high relative error when its contribution to the output jitter is low, and fortunately in this case the phase noise of the DCO is noncritical to the output jitter performance. The relative error drops to about 10% for large DCO contributions.

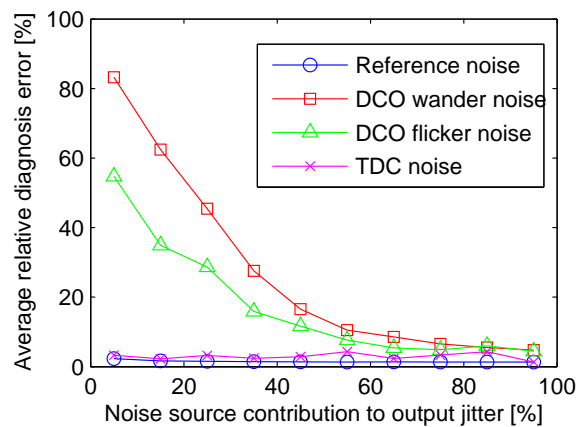


Fig. 71. Average error of the diagnosis of the four main noise sources.

E. Summary

A BIST approach is proposed, targeting complex RF jitter performance of ADPLLs. Digital signatures are collected and processed under specifically designed loop filter configuration and a signature-to-performance mapping is derived based on simplified noise models and transfer function analysis. Monte Carlo analysis is carried out within a behavioral modeling and simulation environment to evaluate the accuracy of the BIST results. The overall relative error for the output jitter estimation is roughly 5%.

CHAPTER V

IN-SITU TEST AND CALIBRATION OF
ALL DIGITAL POLAR TRANSMITTERS

This chapter extends the in-situ test scheme in Chapter IV to measure the error vector magnitude performance of all digital polar transmitters. But unlike the BIST scheme in Chapter IV, this test scheme can provide measurements on-the-fly. The in-situ calibration of a key analog block, digitally-controlled oscillator is also implemented. The work in this chapter is also published in [58].

A. Introduction

With the continuing technology scaling, the performance improvement of digital transistors has stimulated wide interest in digital intensive analog implementation [59] [50]. However, low-cost self-adaptation and built-in self-test (BIST) solutions for analog/mixed-signal designs, especially those for RF wireless applications, remain as a significant challenge due to the complex analog nature of circuit operation [48] [1]. In this chapter, the interaction between the analog and digital domains is exploited for a recent digital polar transmitter architecture, to provide a novel BIST solution aiming at its key performance measure for modulation quality, error vector magnitude (EVM).

For mobile communication with high data rates, the polar transmission [60] can solve the contradiction between the spectral efficiency of modulation schemes and the power efficiency of power amplifiers (PA). An all-digital polar transmitter (ADPT) architecture is proposed [59], as shown in Fig. 72. The coordinate rotation digital computer (CORDIC) transforms the baseband data streams, I and Q , to their polar coordinates. In the digital-to-RF-amplitude converter (DRAC), the amplitude

modulation (AM) is realized by the digitally controlled PA (DPA). In the phase modulation (PM) path, the frequency deviation Δf modulates the frequency of the digitally-controlled oscillator (DCO) through an all-digital phase-locked-loop (ADPLL).

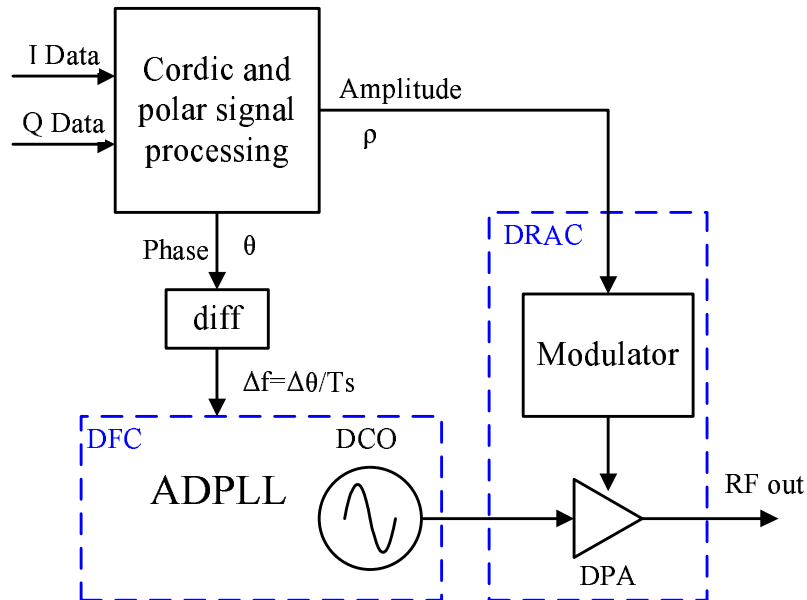


Fig. 72. Diagram of an all-digital polar RF modulator.

For polar transmission, there are three noise sources [61]: the AM path, the PM path and the delay mismatch between the AM and PM paths. For the ADPT, the third source is a minor issue because the delay matching is guaranteed by the control clock cycle of digital circuits [59]. In the AM and PM paths, the nonlinear mapping from digital control words to analog outputs will cause distortion in modulation. Therefore, previous works have aimed at calibrating/compensating these nonlinearities. For the AM path, [62] utilizes the on-chip receiver to conduct the adaptive digital linearization of the DPA, whereas in [63] this is achieved by coupling the RF signal to the reference clock of the ADPLL. For the PM path, [64] proposes a least-mean

square based gain calibration technique of the DCO. Given the nonlinear distortion can be mostly eliminated, random noises from analog blocks (like thermal noise, shot noise, flicker noise, etc.) will become the main noise sources.

Focusing on the PM path, i.e. the ADPLL, an RF BIST scheme is proposed to estimate the modulation quality degradation due to the random noises from the analog blocks in the PM path and the reference clock jitter. The proposed BIST scheme directly aims at the EVM performance, and can provide accurate EVM estimate under multiple parametric process, voltage, and temperature (PVT) variations.

While the bulk of the transmitter, namely digital signal processing and control blocks, is implemented using robust digital logic, the main sources of performance degradations are key analog blocks and their parametric variations, and the jitter of the reference clock. The proposed RF BIST scheme specifically targets the DCO phase noise, finite resolution of the time-to-digital converter (TDC) in addition to the reference clock jitter. By introducing an optimized digital filter, we collect multiple realtime low-frequency phase error signals in the digital form as test signatures. We conduct in-depth noise analysis to elucidate the correspondence between the selected digital test signatures and the EVM. Such correspondence makes it possible to adopt simple digital processing and look-up tables (LUT) to accurately predicate the complex EVM performance of the RF transmitter.

The proposed BIST scheme is based on the linear system analysis. To ensure the linear operation of the ADPLL, the nonlinearity of the DCO that is caused by the DCO gain mismatch needs to be calibrated. The DCO gain calibration in [64] targets at a narrow band application, EDGE (200kHz). For the WCDMA application (5MHz), however, a much wider frequency tuning range will lead to significant PM path distortion due to the DCO gain mismatch. Therefore, a wide-band DCO self-calibration is also proposed in this chapter.

B. ADPLL

In this section, the ADPLL architecture as well as the principle of two-point modulation are introduced. And the requirements of the WCDMA polar transmission on the ADPLL are discussed.

1. Architecture

The ADPLL architecture is shown in Fig. 73. The phase detection is accomplished by a TDC and a digital phase accumulator, where the phase of reference clock (FREF) is multiplied by the frequency control word (FCW) and then compared to the phase of the DCO signal (FDCO). The purely digital signal, phase error (PHE), is filtered by the loop filter, and the filtered DCO tuning word (DTW) is used to tune the FDCO. When the loop is stable, the DCO frequency f_{DCO} can be expressed as:

$$f_{\text{DCO}} \approx N \times f_{\text{REF}}, \quad (5.1)$$

where f_{REF} is the frequency of the FREF and N is the value of the FCW and can be a fractional number.

2. Two-point modulation

Two-point modulation is a common technique for polar transmitters [61]. Its objective is to obtain a quick response at the oscillator frequency without changing the lock-in state of the loop, by synchronously modulating both the frequency divider ratio and the oscillator frequency control signal.

For the ADPLL, two-point modulation is realized by changing the FCW and the DTW at the same time. In order to quantitatively control the DTW offset to match the FCW offset, the DCO gain needs to be normalized. The direct DCO modulation

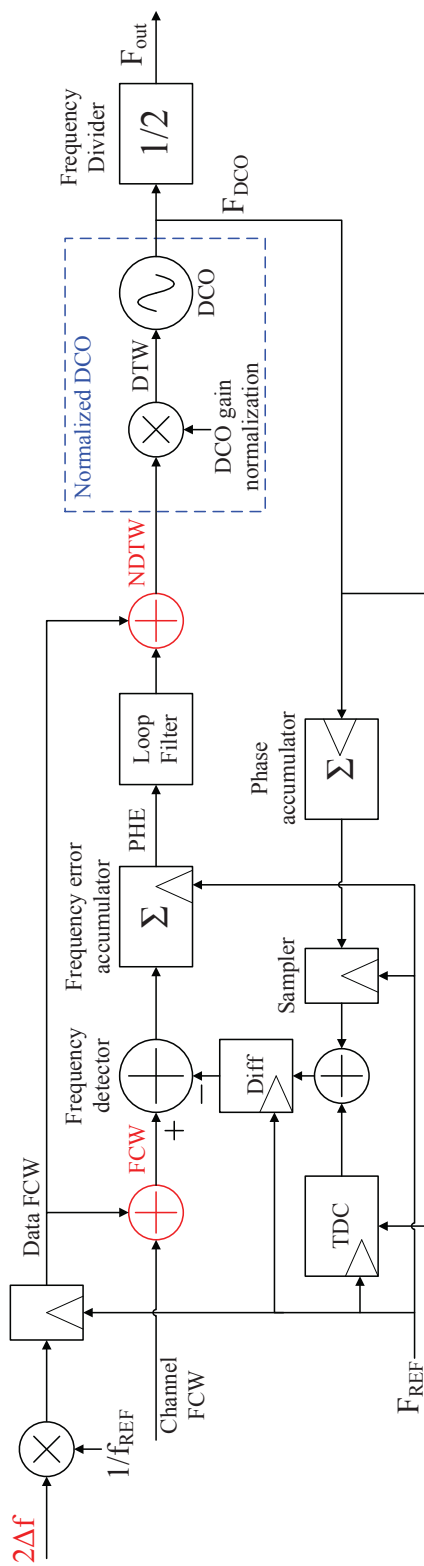


Fig. 73. The ADPLL architecture with two-point modulation.

is achieved through adding the FCW offset to the normalized DTW (NDTW).

3. Requirements of WCDMA

Eq. (5.1) shows that the DCO frequency is controlled by the FCW. Here we focus on the range of the FCW, which is the summation of the channel FCW that is set by the channel frequency and the data FCW that is fed by the modulating frequency deviation Δf .

For a discrete-time system like the ADPLL, the modulation will be distorted due to spectrum aliasing if the changing rate of the phase signal θ is over half of the sampling frequency. In the ADPLL, the reference clock is also used as the sampling clock, which is usually tens of megahertz. For the rest of the chapter, a reference frequency of 26MHz is assumed. On the other hand, although the I/Q signals of WCDMA are bandlimited (5MHz), the CORDIC transforms them to θ with an unlimited bandwidth by performing *arctan* on the ratio of Q to I . This undesired bandwidth growth can be alleviated using a time-domain signal processing method in [65]. Fig. 74 shows the frequency deviation of a typical WCDMA modulation after the bandwidth reduction. Using this technique, the changing rate of θ , i.e. Δf , will be within the range of $\pm 13\text{MHz}$. Though the cost of EVM degradation is inevitable, such degradation is carefully controlled to a minimum level.

C. RF BIST for EVM

In this section, a BIST scheme is proposed that builds up an accurate mapping from the digital signatures, PHE and PHE1, to the EVM degradation due to such noises. PHE is the phase detection output. PHE1 is produced by filtering PHE with an additional branch filter. The signatures have different sensitivities to different noise

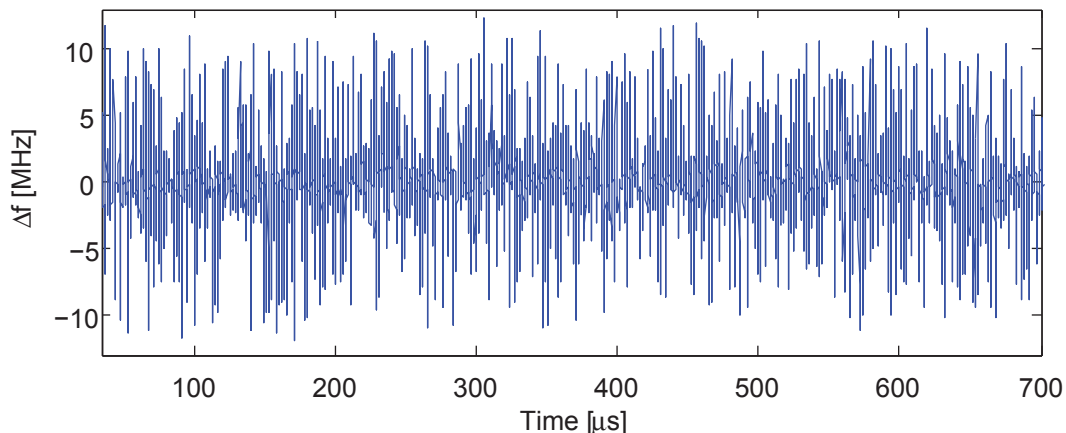


Fig. 74. The frequency deviation of a typical WCDMA modulation in one WCDMA slot ($667\mu\text{s}$) with the bandwidth reduction technique.

sources, which are pre-calculated from the filter configuration and stored in the BIST scheme. The branch filter is optimized to distinguish the contributions of different noise sources to the signatures from each other. Note that the branch filter does not influence the normal functioning of the ADPLL, which enables online testing.

1. z -domain model

Given the assumption that the DCO gain can be calibrated, the ADPLL can be treated as a linear system. The influence of the noise sources in the ADPLL on the EVM degradation can be analyzed based on the z domain model in Fig. 75. The open-loop transfer function is defined as:

$$H_{\text{OL}}(z) = r \cdot H_{\text{LF}}(z)/(z - 1), \quad (5.2)$$

where $H_{\text{LF}}(z)$ is the transfer function of the loop filter. A common first-order loop filter ($H_{\text{LF}}(z) = \alpha + \rho/(z - 1)$) is used in the following discussion. The $r = K_{\text{DCO}}/\hat{K}_{\text{DCO}}$ factor, where \hat{K}_{DCO} is an estimate of the DCO gain, K_{DCO} . Because of the DCO gain calibration that will be introduced later, $r = 1$ is safely assumed. z -domain expres-

sions can be transformed to the frequency domain by substituting $z = e^{2\pi jf/f_{\text{REF}}}$, where f_{REF} is the sampling rate.

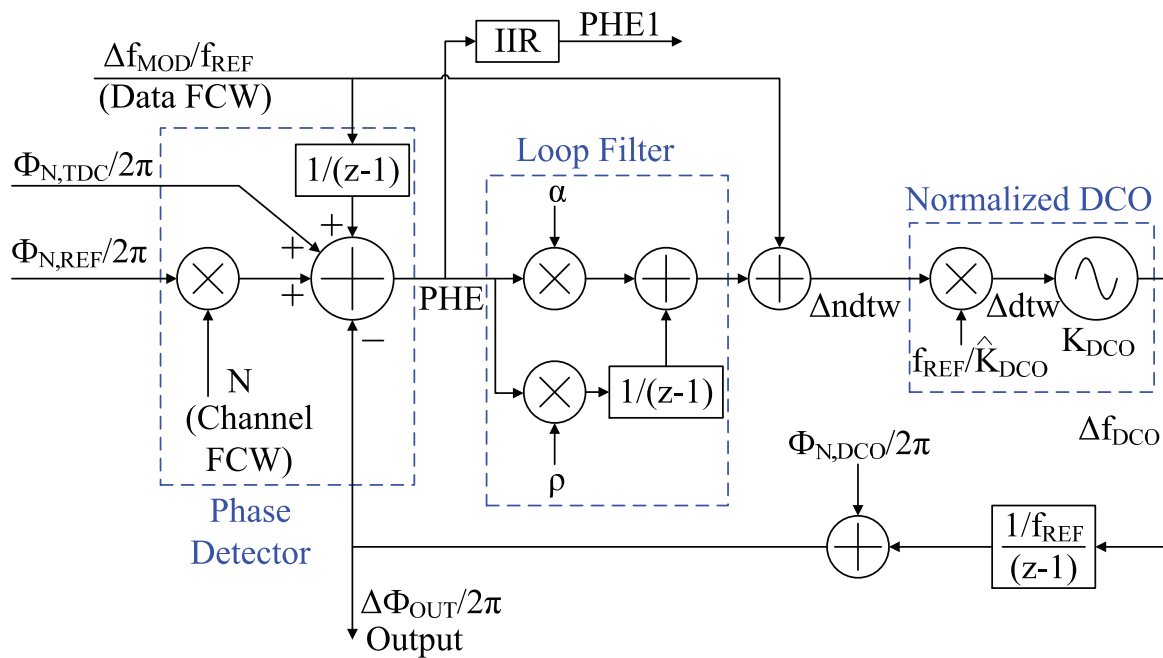


Fig. 75. The z -domain model of the ADPLL including noise sources.

Again, because of the assumption that the DCO gain can be calibrated, the PM path distortion will be mostly contributed by the reference clock jitter, the TDC quantization noise and the DCO phase noise, rather than the DCO gain mismatch. The three sources of noise are included in the z domain model of the ADPLL: $\phi_{N,\text{REF}}$ from the reference clock, $\phi_{N,\text{DCO}}$ from the DCO and $\phi_{N,\text{TDC}}$ from the TDC. The closed-loop transfer functions from the noise sources to the output phase noise and the digital signatures are listed in Table VII, where $H_{\text{IIR}}(z)$ is the transfer function of the branch low-pass IIR filter.

Table VII. Transfer functions from noise sources to output phase noise and digital signatures

to \ from	Ref.(R)	TDC(T)	DCO(D)
Output(O)	$H_{R2O}(z) = \frac{NH_{OL}(z)}{1 + H_{OL}(z)}$	$H_{T2O}(z) = \frac{H_{OL}(z)}{1 + H_{OL}(z)}$	$H_{D2O}(z) = \frac{1}{1 + H_{OL}(z)}$
PHE(P)	$H_{R2P}(z) = \frac{N}{1 + H_{OL}(z)}$	$H_{T2P}(z) = \frac{1}{1 + H_{OL}(z)}$	$H_{D2P}(z) = \frac{1}{1 + H_{OL}(z)}$
PHE1(P1)	$H_{R2P1}(z) = \frac{NH_{IIR}(z)}{1 + H_{OL}(z)}$	$H_{T2P1}(z) = \frac{H_{IIR}(z)}{1 + H_{OL}(z)}$	$H_{D2P1}(z) = \frac{H_{IIR}(z)}{1 + H_{OL}(z)}$

2. Noise analysis

The three noise sources are mathematically modeled in the frequency domain to help analyze the relationship between PHE and the output phase noise.

The reference clock is usually generated by a crystal oscillator, and thus provides a single-tone spectrum with little spectral spread. The relatively flat spectrum can be simplified as constant, whose power spectral density (PSD) is:

$$\Phi_{\text{REF}}(\Delta f) = L_{\text{REF}}, \quad (5.3)$$

where Δf is the frequency offset (from $-f_{\text{REF}}/2$ to $f_{\text{REF}}/2$). L_{REF} is a constant describing the noise level of the reference phase noise.

The second noise source is from the TDC due to its time resolution. Similar to the quantization noise of analog-to-digital converters, the TDC quantization noise

can be modeled as an additive random variable with uniform distribution and white noise spectral characteristic:

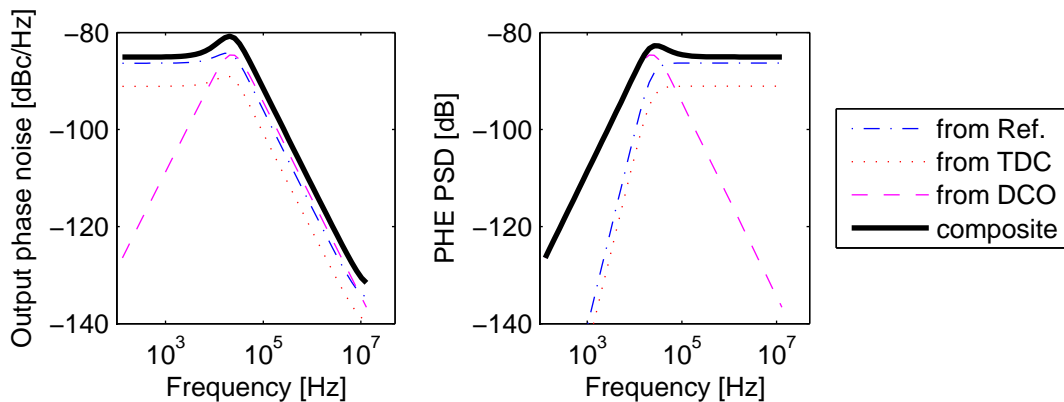
$$\Phi_{\text{TDC}}(\Delta f) = L_{\text{TDC}}, \quad (5.4)$$

where L_{TDC} is the constant noise level.

Apart from the reference clock and the TDC, the DCO is another major noise source. The wander noise is the dominant noise mechanism of the DCO within the frequency range concerned. It is generally caused by the white-noise fluctuation of the oscillating frequency. Besides, the DCO quantization noise and the DCO power supply noise will also cause the wander noise [54] [55]. The PSD of the DCO wander noise can be modeled as:

$$\Phi_{\text{DCO}}(\Delta f) = L_{\text{DCO}}/\Delta f^2, \quad (5.5)$$

where L_{DCO} is the noise level of the wander noise.



(a) The output phase noise.

(b) The PSD of PHE.

Fig. 76. The composition of the noise.

The frequency-domain noise models can be instantiated according to typical noise levels. Together with the transfer functions in Table VII, the spectra of the output

Table VIII. The comparison of the noise contributions (low frequency range: 100Hz-100kHz; high frequency range: 100kHz-13MHz).

		ref.	TDC	DCO
Output phase noise	@all freq.	42.87%	14.29%	42.83%
	@low freq.	44.33%	14.78%	40.89%
	@high freq.	35.60%	11.87%	52.53%
PHE	@all freq.	74.67%	24.89%	0.44%
	@low freq.	46.75%	15.58%	37.67%
	@high freq.	74.93%	24.98%	0.09%

phase noise and PHE are shown in Fig. 76, under the loop settings of $f_{\text{REF}}=26\text{MHz}$, $N=150.7$, $\alpha = 2^{-7}$ and $\rho = 2^{-15}$. It is noted that the DCO noise contribution to PHE decreases dramatically as frequency increasing. Table VIII shows that within the high frequency range, only 0.09% of PHE power comes from the DCO phase noise. Such huge difference between the noise contributions makes it possible to distinguish the DCO phase noise from the other two noises, and therefore enables an accurate estimation of the output phase noise.

3. BIST principle

The relationship between the phase noise of the ADPLL output and the EVM performance is depicted in Fig. 77. Assuming the error vector \vec{v}_e is much shorter than the ideal vector \vec{v}_i , and there is no noise in the amplitude, we can write:

$$|\vec{v}_e| \approx |\vec{v}_i| \cdot \sin \phi_{\text{OUT}} \approx |\vec{v}_i| \cdot \phi_{\text{OUT}}, \quad (5.6)$$

where ϕ_{OUT} is the output phase error. Also considering the EVM definition:

$$\text{EVM}_{\text{RMS}} = \left[\frac{|\vec{v}_e|}{|\vec{v}_i|} \right]_{\text{RMS}}, \quad (5.7)$$

we have:

$$\text{EVM}_{\text{RMS}} \approx \phi_{\text{OUT,RMS}}, \quad (5.8)$$

which means that the EVM (RMS) equals to the phase noise (RMS) of the ADPLL output, if only the PM path distortion is considered.

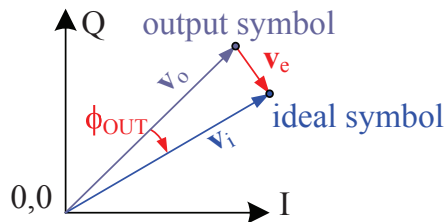


Fig. 77. The relationship between the EVM and the phase noise.

The RMS value of the phase noise can also be related to its spectrum through the Parseval theorem:

$$\phi_{\text{OUT,RMS}}^2 = \int_{-f_{\text{REF}}/2}^{f_{\text{REF}}/2} \Phi_{\text{OUT}}(f) df, \quad (5.9)$$

where $\Phi_{\text{OUT}}(f)$ is the PSD of the output phase noise, and can be further expressed by the noise levels:

$$\Phi_{\text{OUT}}(f) = L_{\text{REF}} |H_{\text{R2O}}(f)|^2 + L_{\text{TDC}} |H_{\text{T2O}}(f)|^2 + L_{\text{DCO}} |H_{\text{D2O}}(f)|^2 / f^2, \quad (5.10)$$

in which the transfer functions are defined in Table VII. Noticing $H_{\text{R2O}} = N \cdot H_{\text{T2O}}$ for any frequency, a new noise level can be defined to treat the reference noise and

the TDC noise as a whole:

$$L_{R\&T} = NL_{REF} + L_{TDC}. \quad (5.11)$$

Combining Eq. (5.8) to Eq. (5.10) together, we can write:

$$EVM_{RMS}^2 \approx L_{R\&T}C_{T2O} + L_{DCO}C_{D2O}, \quad (5.12)$$

where C_{T2O} and C_{D2O} are power transfer coefficients, i.e. the power transfer coefficient from position X to position Y is defined by the integration of $H_{X2Y}(f)$ or $H_{X2Y}(f)/f^2$:

$$C_{X2Y} = \begin{cases} \int_{-f_{REF}/2}^{f_{REF}/2} |H_{X2Y}(f)|^2/f^2 df & \text{if } X \text{ refers to DCO} \\ \int_{-f_{REF}/2}^{f_{REF}/2} |H_{X2Y}(f)|^2 df & \text{else.} \end{cases} \quad (5.13)$$

All possible instances of H_{X2Y} are listed in Table VII.

In Eq. (5.12), the power transfer coefficients can be pre-calculated according to the loop settings, while the noise levels are unknown because of the PVT variations. The proposed BIST scheme provides a way to calculate these noise levels by processing PHE and PHE1.

Similar to Eq. (5.12), the power difference between PHE and PHE1 can be written as:

$$PHE_{RMS}^2 - PHE1_{RMS}^2 = L_{R\&T}(C_{T2P} - C_{T2P1}) + L_{DCO}(C_{D2P} - C_{D2P1}). \quad (5.14)$$

As mentioned before, the high frequency components of PHE are dominated by the reference and TDC noise contributions. So in Eq. (5.14), the second term at the right side is much smaller than the first term, and thus Eq. (5.14) can be approximated as:

$$PHE_{RMS}^2 - PHE1_{RMS}^2 \approx L_{R\&T}(C_{T2P} - C_{T2P1}), \quad (5.15)$$

which can be used to calculate $L_{R\&T}$. And L_{DCO} can also be calculated considering

$$\text{PHE1}_{\text{RMS}}^2 = L_{R\&T}C_{T2P1} + L_{DCO}C_{D2P1}. \quad (5.16)$$

Given Eq. (5.12) to Eq. (5.16), the EVM performance can finally be estimated by:

$$\begin{aligned} \text{EVM}_{\text{RMS}}^2 \approx & \text{PHE}_{\text{RMS}}^2 \left(\frac{C_{T2O}C_{D2P1} - C_{T2P1}C_{D2O}}{C_{T2P}C_{D2P1} - C_{T2P1}C_{D2P1}} \right) \\ & + \text{PHE1}_{\text{RMS}}^2 \left(\frac{C_{T2P}C_{D2O} - C_{T2O}C_{D2P1}}{C_{T2P}C_{D2P1} - C_{T2P1}C_{D2P1}} \right). \end{aligned} \quad (5.17)$$

4. BIST scheme

The block diagram of the proposed BIST scheme is shown in Fig. 78. The average powers of both PHE and PHE1 are calculated within one WCDMA time slot(667 μ s). And then $\text{PHE}_{\text{RMS}}^2$ and $\text{PHE1}_{\text{RMS}}^2$ are linearly combined to generate the EVM estimation:

$$\text{EVM}_{\text{RMS,estimate}}^2 = \text{PHE}_{\text{RMS}}^2 K_{\text{PHE}} + \text{PHE1}_{\text{RMS}}^2 K_{\text{PHE1}}, \quad (5.18)$$

where K_{PHE} and K_{PHE1} are the coefficients of the linear combination. Compared to Eq. (5.17), they can be pre-calculated by:

$$\begin{aligned} K_{\text{PHE}} &= \frac{C_{T2O}C_{D2P1} - C_{T2P1}C_{D2O}}{C_{T2P}C_{D2P1} - C_{T2P1}C_{D2P1}}, \\ K_{\text{PHE1}} &= \frac{C_{T2P}C_{D2O} - C_{T2O}C_{D2P1}}{C_{T2P}C_{D2P1} - C_{T2P1}C_{D2P1}}. \end{aligned} \quad (5.19)$$

Similarly, for the sake of diagnosis, the noise estimations can be generated by linearly combining $\text{PHE}_{\text{RMS}}^2$ and $\text{PHE1}_{\text{RMS}}^2$, and the corresponding coefficients can be pre-calculated, too.

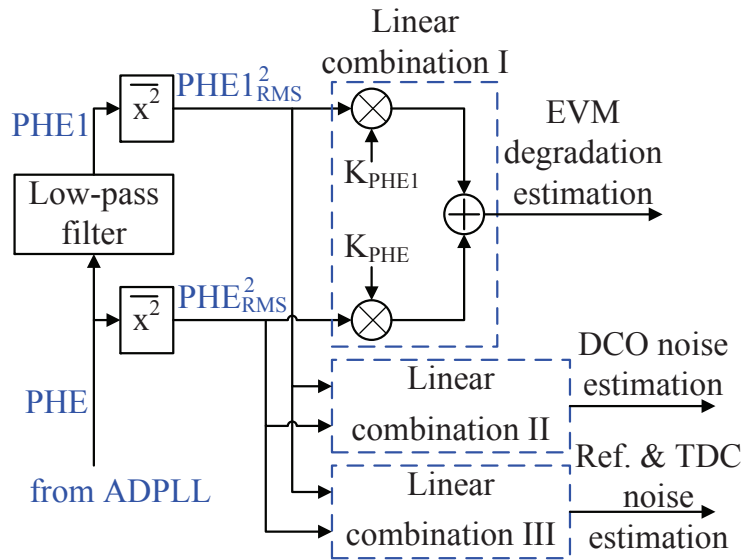


Fig. 78. BIST block diagram.

5. The optimization of the branch filter

When optimizing the structure of the additional branch filter, 3 objectives are considered. The first is to minimize the systematic error ε_{sys} introduced by the approximation of Eq. (5.15). The second is to minimize the influence of the random fluctuation of the signature observation due to the randomness of noises. This objective can be expressed as to minimize the EVM sensitivity to the digital signatures, which is defined as:

$$S = \left| \frac{\partial \text{EVM}_{\text{RMS}}^2}{\partial \text{PHE}_{\text{RMS}}^2} \right| + \left| \frac{\partial \text{EVM}_{\text{RMS}}^2}{\partial \text{PHE1}_{\text{RMS}}^2} \right| \quad (5.20)$$

$$\approx |K_{\text{PHE}}| + |K_{\text{PHE1}}|.$$

Last but not least, the hardware cost of the additional branch filter should also be minimized.

The low-pass filter is implemented by a cascade of single-pole IIR filters, whose

Table IX. The systematic errors and the EVM sensitivities to the digital signatures.

	$\lambda = 2^{-1}$	$\lambda = 2^{-2}$	$\lambda = 2^{-3}$
$N_{\text{IIR}} = 1$	$\varepsilon_{\text{sys}} = 0.20\%$ $S = 1.9953$	$\varepsilon_{\text{sys}} = 0.17\%$ $S = 1.3494$	$\varepsilon_{\text{sys}} = 0.16\%$ $S = 1.1928$
$N_{\text{IIR}} = 2$	$\varepsilon_{\text{sys}} = 0.14\%$ $S = 1.4572$	$\varepsilon_{\text{sys}} = 0.12\%$ $S = 1.1864$	$\varepsilon_{\text{sys}} = 0.10\%$ $S = 1.1494$
$N_{\text{IIR}} = 3$	$\varepsilon_{\text{sys}} = 0.12\%$ $S = 1.3215$	$\varepsilon_{\text{sys}} = 0.11\%$ $S = 1.1546$	$\varepsilon_{\text{sys}} = 0.09\%$ $S = 1.1559$

z -domain transfer function has the following form:

$$H_{\text{IIR}}(z) = \left(\frac{\lambda}{z - (1 - \lambda)} \right)^{N_{\text{IIR}}}, \quad (5.21)$$

where λ is usually a negative integer power of two for the ease of hardware implementation, and N_{IIR} is the number of the cascading filters. This structure will ensure that the low-pass filter has a unit gain at DC in this scheme, so that only the high frequency power of PHE is filtered. The filter bandwidth is proportional to λ , N_{IIR} can be increased when a larger rolloff is needed.

Table IX lists ε_{sys} and S corresponding to different groups of λ and N_{IIR} . It is obtained by the transfer function analysis under typical noise levels. Based on the 3 objectives of the filter optimization, the low-pass filter with $\lambda = 2^{-3}$ and $N_{\text{IIR}} = 3$ should be one of the optimized choices.

D. DCO gain calibration

The above BIST scheme is based on the assumption that noises are propagated in linearized systems, which requires the incorporation with the wide-band DCO non-

linearity calibration.

1. DCO gain mismatch

The DCO for polar WCDMA transmitters in 90nm CMOS is shown in Fig. 79 [66]. The DCO core is composed of a cross coupled gm core, an LC tank, a current source and a 2nd harmonic trap. The output frequency is tuned by switching on/off the MOS capacitors according to the DTW. Note that the proposed DCO gain calibration can also be applied to other DCO designs.

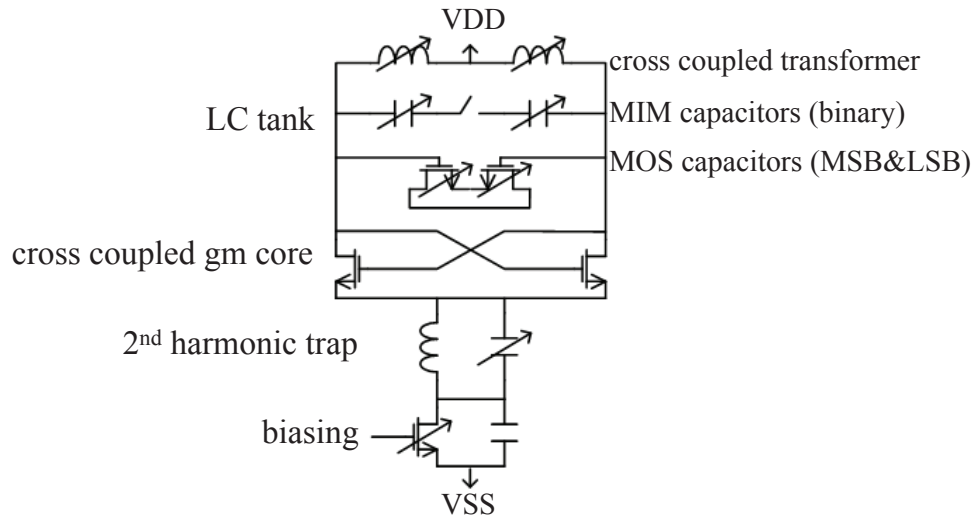


Fig. 79. DCO core with LC tank and biasing network in [12].

The DCO gain K_{DCO} is defined as:

$$K_{\text{DCO}} = \Delta f_{\text{DCO}} / \Delta \text{DTW}, \quad (5.22)$$

where ΔDTW is the increment of the DTW, and Δf_{DCO} is the corresponding fre-

Table X. EVM degradation due to DCO gain mismatch (PM path only, no random noise).

	EVM
no DCO gain mismatch	2.04%
the DCO gain mismatch with 1σ cap mismatch	3.51%
the DCO gain mismatch with 3σ cap mismatch	6.44%

quency offset. K_{DCO} can also be expressed as [67]:

$$K_{\text{DCO}} = -\frac{1}{4\pi C\sqrt{LC}} \cdot \frac{\Delta C}{\Delta\text{DTW}}. \quad (5.23)$$

where L is the inductance and C is the total capacitance parallel to the inductor, ΔC is the change of the total capacitance due to ΔDTW . Eq. (5.23) suggests the cap mismatch will lead to the variation of the DCO gain.

The DCO gain mismatch will cause frequency errors in the feed forward path in the two-point modulation scheme. Since this frequency error directly modulates the DCO frequency, it will result in considerable modulation distortion. According to the cap mismatch information from a commercial 90nm technology and the foundry provided design kit, Table X lists the EVM degradations due to different DCO cap mismatch deviations, σ is the typical deviation of cap mismatch (around 5%). Although no random noise from analog blocks is considered, there is still some EVM degradation even with no DCO gain mismatch because of the bandwidth reduction technique.

2. DCO gain calibration

Given the DCO mismatch is pushing the EVM performance to the pass/fail margin (typically around 5% for the PM path), a DCO gain calibration scheme is proposed

for WCDMA. Its objective is to compensate the DCO gain mismatch caused by the cap mismatch. The calibration scheme has two modes: (i) DCO mismatch detection and (ii) DCO gain compensation. Mode(i) functions at the system power-on reset or during the time interval between transmission windows, whereas Mode(ii) takes effect throughout the modulation process.

The entire structure of the calibration is shown in Fig. 80. If *mode_sel* is set to be 0, then Mode (i) is chosen, when the sampled DCO gains are detected and stored into a lookup table (LUT), as in Fig. 81(a). If *mode_sel* is set to be 1, then Mode (ii) is chosen, when the contents of the LUT are linearly interpolated to compensate the DCO gain, as in Fig. 81(b).

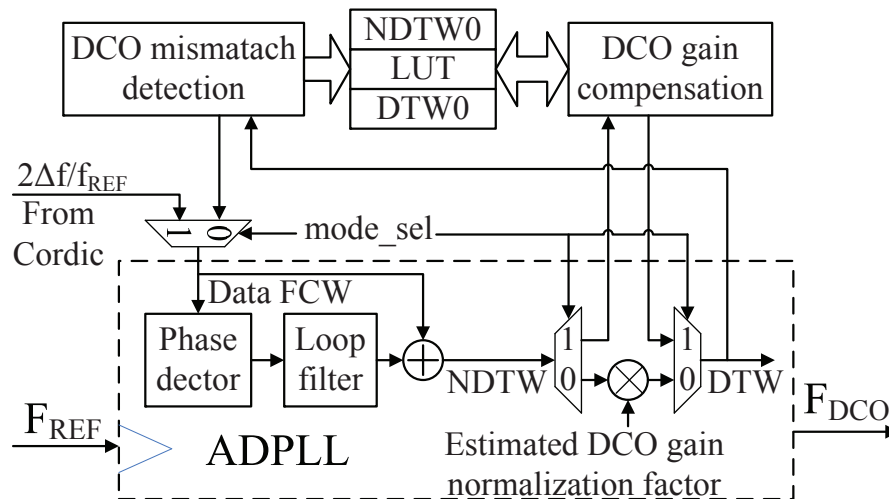


Fig. 80. The block diagram of the calibration scheme.

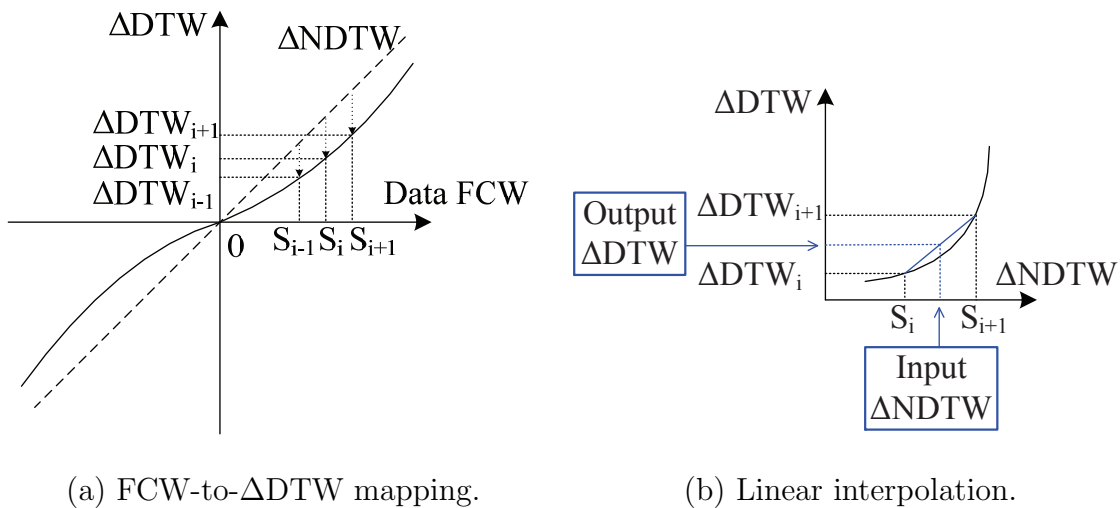


Fig. 81. The relationship between Δ FCW/ Δ NDTW and Δ DTW.

E. Simulation results

1. Simulation platform

The SPICE simulation of the entire ADPLL-based transmitter will take days for $1\mu\text{s}$ of simulated time, and therefore is not practical [68]. An event-driven simulation platform using VHDL is proposed in [68] to analyze the ADPLL noise performance, which is validated by real chip measurements. This approach is adopted in this chapter, but using Verilog and System Verilog.

The simulation platform is composed of the test environment (Env.) using System Verilog and the design under test (DUT) using Verilog, as shown in Fig. 82. According to the WCDMA Release 5 standard, 8-PSK modulation scheme is adopted. In the stimulus module, there are the random transmission data generation, the subsequent CORDIC, and the calibration control. The synthesizable Verilog describes the RTL implementation of the DUT, including the digital circuits of the ADPLL and the additional calibration and BIST circuits. The noise models, though as part of the DUT, are realized by System Verilog, and their noise levels can be configured

by the configuration module. The ADPLL output is imported to the score board, where its EVM is measured and compared with the EVM estimation from the BIST.

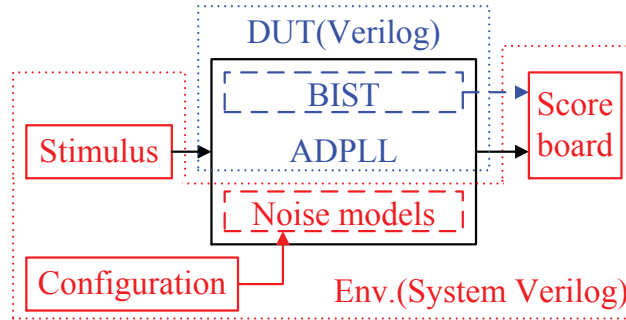


Fig. 82. The event-driven simulation platform for the ADPLL.

The digital circuits are simulated at RTL level, e.g. the digital signals use 6 bits for the integral part and 15 bits for the fractional part, such that the quantization effect is modeled. The noise models can closely depict the real performance of the analog circuits, e.g. both the reference jitter and the DCO phase noise are modeled by three segments [68]. Also, the TDC nonlinearity is included in the TDC model, with its differential nonlinearity and integral nonlinearity both lower than 0.7 LSB [69]. Both the noise levels and the TDC resolution can be configured through the Env.

2. Simulation results for DCO gain calibration

The proposed DCO gain calibration are run for two DCO instances obtained by a commercial 90nm technology and the foundry provided design kit: one with the 1σ cap mismatch and the other with the 3σ cap mismatch. Here σ is the typical deviation of cap mismatch for the technology, which is around 5%. The constellation graphs for the 3σ case are shown in Fig. 83.

The output EVM versus the sample step (the gap between two adjacent samples

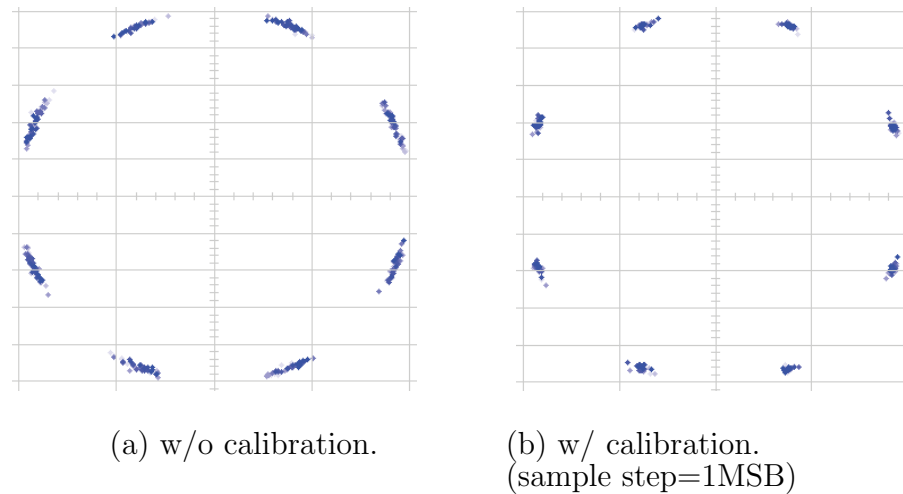


Fig. 83. The constellation graphs for the worst mismatch case.

of the DCO gain) is shown in Fig. 84(a), with the noise models configured to the typical noise levels, whereas Fig. 84(b) is the results at twice the typical noise levels. And the EVM results with no DCO gain mismatch are also shown for reference. It is clearly seen that the denser distribution of samples leads to better DCO gain curve fitting when detecting the DCO mismatch, and therefore results in better EVM performance. Taking the sample step of 1MSB, the proposed calibration can improve the EVM performance to the reference level, which means the EVM degradation mainly result from the noise sources like the reference noise, the TDC noise and the DCO phase noise, rather than the DCO mismatch. This observation is in accordance with the assumption in the BIST principle analysis.

3. Simulation results for EVM BIST

Monte Carlo analysis is carried out to evaluate the proposed BIST scheme. 2,000 Monte-Carlo simulation samples are generated by conducting the event-driven simulation of the ADPLL-based polar transmitter. For each sample the configuration control in the Env. generate random noise levels of the noise models. The variance

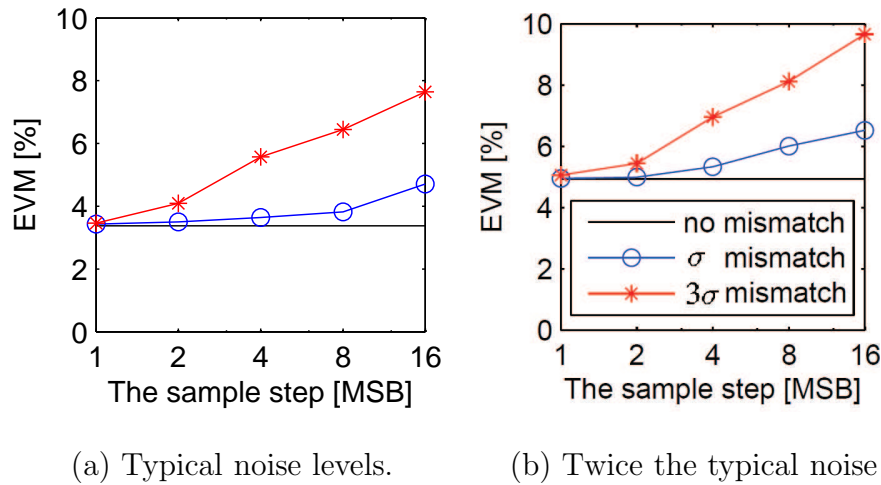


Fig. 84. The output EVM versus the sample step.

of each noise level is set to $3\sigma = 10\%$ of the mean value.

The EVM of the ADPLL output clock during one WCDMA time slot is estimated by the proposed BIST method. Fig. 85 compares the measured EVM and the estimation by the BIST. As can be seen, the BIST can provide an accurate EVM estimation (with errors smaller than 3% for 95% samples) when working together with the DCO gain calibration. This verifies our earlier assumption that the PM path distortion caused by the DCO gain mismatch is neglectable after DCO gain calibration. On the other hand, the estimation error is large without the DCO gain calibration. This is because the DCO mismatch contributes to a significant part of EVM degradation that cannot be handled by the BIST scheme. Therefore, the EVM BIST is only effective when the DCO gain calibration is taking effects. Besides, when the EVM is lower than 3%, the EVM estimation tends to be smaller than the measured value. The reason is that the proposed BIST cannot detect the EVM degradation caused by the bandwidth reduction technique mentioned before. But this will not be a problem, since such cases are far away from the pass/fail margin.

A pass/fail test is also carried out, as shown in Fig. 86. The 3GPP standard

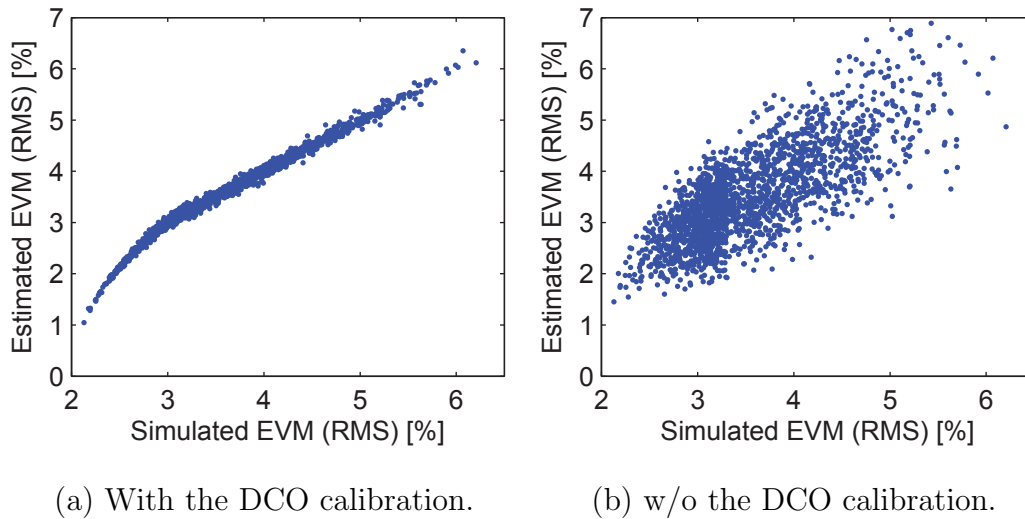


Fig. 85. Estimated EVM vs. simulated EVM.

requires the EVM not to exceed 17.5% for a WCDMA transmitter. Since only the PM path distortion is considered here, we set the pass/fail line as $\text{EVM}_{\text{RMS}} = 5\%$. If the pass/fail test is used for production test, then the defect escape rate is 0.35%, and the yield loss rate is 0.75%.

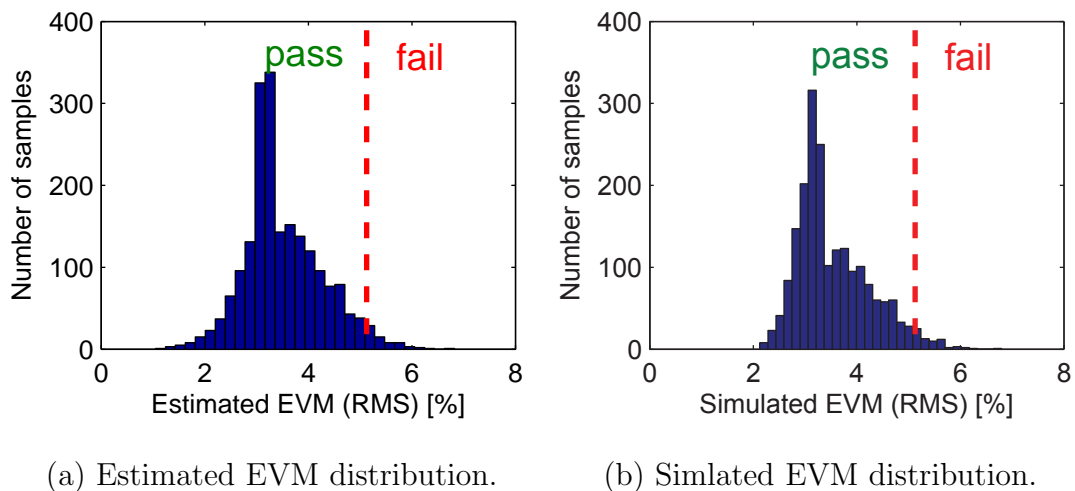


Fig. 86. Pass/fail test.

Table XI. Hardware overhead estimation using 90nm CMOS technology

		area [μm^2]
ADPLL	digital	70000
	TDC	140000
	DCO	165000
Calibration	digital processing	5400
	LUT	820
BIST	digital processing	27300
	IIR filter	560

F. Implementation issues

In Table XI, the areas are estimated for the ADPLL (from existing layouts) and the proposed self-calibration and BIST (from synthesized results), using a commercial 90nm technology. The hardware overhead is 9.1% of the ADPLL area, which is acceptable. Most of these costs can be further saved in a system-on-chip solution proposed in [59]. Both the ARM7 MPU and the C54 DPS are running at the clock rate of 104MHz, four times faster than f_{REF} of the ADPLL. This implies their processing speed is not a restriction to their reuse for digital processing. For the sake of the DCO gain calibration, around 640 samples are needed to cover the FCW tuning range of WCMDA, with the sample step of 1MSB. Supposing the LUT takes 2 bytes per sample, it will take 10kb memory totally, which is relatively small compared to 2.5Mb on-chip SRAM. If the digital processing and the LUT can realized by reusing the on-chip resource, the only area overhead will be from the additional branch IIR filter, which is less than 1% of the ADPLL area.

G. Summary

A self-calibration is proposed to compensate the DCO gain mismatch as well as a BIST targeting the EVM performance for an ADPLL-based WCDMA polar transmitter, which are validated by an event-driven simulation platform. The digital-like implementation has led to the low cost of the proposed approaches. The proposed BIST scheme is focused on the EVM degradation from the PM path. Together with the self-testing of the AM path, it is possible to provide a complete solution for the BIST of EVM in the future.

CHAPTER VI

CONCLUSIONS AND FUTURE DIRECTIONS

A. Conclusions

This dissertation emphasizes on developing novel verification and test techniques for improving the robustness of AMS designs in highly scaled CMOS technologies. A formal verification framework is proposed that incorporates nonlinear SMT solving techniques and simulation exploration, with a Bayesian inference based approach to balance the costs of simulation and SMT solving. The feasibility and efficacy of the proposed methodology are demonstrated on the verification of lock time specification of a charge-pump PLL. On the other hand, in-situ test techniques are proposed for AMS designs for the error detection after fabrication. First, a novel two-level structure of GRO-PVDL is proposed to measure the jitter performance for high-speed high-resolution applications on chip. Taking advantage of quantization noise shaping, an effective resolution of 0.8ps is achieved using 90nm CMOS technology. Second, the reconfigurability of recent ADPLL designs is exploited to provide novel in-situ output jitter test and diagnosis abilities under multiple parametric variations of key analog building blocks. As an extension, an in-situ test scheme is proposed to provide online testing for ADPLL based polar transmitters.

B. Future directions

First, the verification of AMS designs, especially those with complex nonlinear dynamics, may take a middle way between formal verification and conventional simulation-based verification. Although formal verification techniques have found great success in digital designs and linear analog designs, the huge cost of formally verifying nonlin-

ear properties is against their feasibility for nonlinear AMS designs. Therefore, it is desired to combine formal checking techniques and simulations, to achieve both high coverage and efficiency. It is also anticipated that statistical framework might be the “glue” in such combination. Because statistical methods can potentially provide a rigorous view of uncertain circuit properties and a statically defined coverage for the verification that is composed of formal methods and simulations.

Second, in-situ test designs may become an essential part of future AMS designs. Analog designs now have increasing digital content that tests or calibrates the variable analog performance in highly scaled technologies. The future development of in-situ test techniques may emphasize on two directions. The first is to develop innovative techniques that transform analog properties into digital signals to achieve both high observability and low interference to analog circuits. The second is to increase the testability of analog circuits by exploiting the interaction between the analog and digital circuits in a mix-signal environment.

Last but not least, an even higher level picture in the future is the combination of verification and DfT techniques. Considering that verification and DfT techniques are both aiming at error detection, tradeoff between verification and DfT techniques can be leveraged to minimize the overall implementation cost as well as maximize the overall error coverage. For example, we can tradeoff between the runtime cost of verification and the hardware cost of DfT. If due to runtime limitation, 100% error coverage is too costly to achieve in verification, then some error checks can be intentionally skipped in verification. At the same time, the accompanied in-situ test will emphasize on the errors that are not covered by the verification, given such errors can be detected directly on the manufactured chips with an affordable extra hardware cost.

REFERENCES

- [1] R. B. Staszewski, I. Bashir, and O. Eliezer, “Rf built-in self test of a wireless transmitter,” *IEEE Transaction on Circuits System II, Express Briefs*, vol. 54, no. 2, pp. 186–190, February 2007.
- [2] S. Nassif, “Process variability at the 65nm node and beyond,” in *Proceedings of IEEE Custom Integrated Circuits Conference*, September 2008, pp. 1–8.
- [3] “The international technology roadmap for semiconductors,” <http://public.itrs.net/>, accessed in April 2012.
- [4] B.C. Lim, J. Kim, and M.A. Horowitz, “An efficient test vector generation for checking analog/mixed-signal functional models,” in *Proceedings of the 47th Design Automation Conference*, June 2010, pp. 767–772.
- [5] H. Chang and K. Kundert, “Verification of complex analog and rf ic designs,” *Proceedings of the IEEE*, vol. 95, no. 3, pp. 622–639, March 2007.
- [6] M. Jarwala and S.J. Tsai, “A framework for design for testability of mixed analog/digital circuits,” in *Proceedings of the IEEE Custom Integrated Circuits Conference*, May 1991, pp. 13–15.
- [7] S. Gupta, B.H. Krogh, and R.A. Rutenbar, “Towards formal verification of analog designs,” in *Proceedings of International Conference on Computer Aided Design*, November 2004, pp. 210–217.
- [8] W.K. Lam, *Hardware Design Verification: Simulation and Formal Method-Based Approaches (Prentice Hall Modern Semiconductor Design Series)*, Prentice Hall PTR, Upper Saddle River, 2005.

- [9] M. Franzle, C. Herde, T. Teige, S. Ratschan, and T. Schubert, “Efficient solving of large non-linear arithmetic constraint systems with complex boolean structure,” *Journal on Satisfiability, Boolean Modeling, and Computation*, vol. 1, pp. 209–236, 2007.
- [10] “ISAT: Tight integration of satisfiability & constraint solving,” <http://isat.gforge.avacs.org/>.
- [11] M. Wang, Z. Wen, L. Chen, Y. Zhang, and Z. Zhang, “A novel bist approach for testing dlls of soc,” in *IEEE Circuits and Systems International Conference on Testing and Diagnosis*, April 2009, pp. 1–4.
- [12] J. F. Bulzacchelli and et al, “A 10-Gb/s 5-tap DFE/4-tap FFE transceiver in 90-nm CMOS technology,” *IEEE Journal of Solid-State Circuits*, vol. 41, no. 12, pp. 2885–2900, December 2006.
- [13] M. H. Zaki, S. Tahar, and G. Bois, “Formal verification of analog and mixed signal designs: A survey,” *Microelectronics Journal*, vol. 39, pp. 1395–1404, 2008.
- [14] W. Denman, B. Akbarpour, S. Tahar, M.H. Zaki, and L.C. Paulson, “Formal verification of analog designs using metitarski,” in *Formal Methods in Computer-Aided Design*, November 2009, pp. 93–100.
- [15] S. Steinhorst and L. Hedrich, “Advanced methods for equivalence checking of analog circuits with strong nonlinearities,” *Formal Methods in System Design*, vol. 36, pp. 131–147, 2010.
- [16] A. Singh and P. Li, “On behavioral model equivalence checking for large analog/mixed signal systems,” in *IEEE/ACM ICCAD*, November 2010, pp. 55–61.

- [17] W. Hartong, L. Hedrich, and E. Barke, “Model checking algorithms for analog verification,” in *Proceedings of the 39th Design Automation Conference*, 2002, pp. 542–547.
- [18] S. Little, D. Walter, K. Jones, C. J. Myers, and A. Sen, “Analog/mixed-signal circuit verification using models generated from simulation traces,” *International Journal of Foundations of Computer Science*, vol. 21, no. 2, pp. 191–210, 2010.
- [19] G. Frehse, B.H. Krogh, and R.A. Rutenbar, “Verifying analog oscillator circuits using forward/backward abstraction refinement,” in *Proceedings of Design, Automation and Test in Europe*, March 2006, vol. 1.
- [20] B.I. Silva and B.H. Krogh, “Formal verification of hybrid systems using checkmate: a case study,” in *American Control Conference*, 2000, vol. 3, pp. 1679–1683 vol.3.
- [21] M. Althoff, A. Rajhans, B. H. Krogh, S. Yaldiz, X. Li, and L. Pileggi, “Formal verification of phase-locked loops using reachability analysis and continuization,” in *Proceedings of International Conference on Computer Aided Design*, November 2011.
- [22] Mark R. Greenstreet and Suwen Yang, “Verifying start-up conditions for a ring oscillator,” in *Proceedings of the Great Lakes Symposium on VLSI*, February 2008, pp. 201–206.
- [23] S.K. Tiwary, A. Gupta, J.R. Phillips, C. Pinello, and R. Zlatanovici, “First steps towards sat-based formal analog verification,” in *Proceedings of International Conference on Computer Aided Design*, November 2009, pp. 1–8.
- [24] T.A. Henzinger, “The theory of hybrid automata,” in *Proceedings of the*

- Eleventh Annual IEEE Symposium on Logic in Computer Science*, July 1996, pp. 278–292.
- [25] A. Gelman, J. Carlin, H. Stern, and D. Rubin, *Bayesian Data Analysis*, Chapman & Hall, London, 1995.
- [26] J.-M. Bernard, “An introduction to the imprecise dirichlet model for multinomial data,” *International Journal of Approximate Reasoning*, vol. 39, no. 2-3, pp. 123–150, 2005.
- [27] L. Yin, Y. Kim, and P. Li, “High effective-resolution built-in jitter characterization with quantization noise shaping,” in *Proceedings of the 48th Design Automation Conference*, June 2011, pp. 765–770.
- [28] M. Ishida, K. Ichiyama, et al., “On-chip circuit for measuring data jitter in the time or frequency domain,” in *IEEE Radio Frequency Integrated Circuits Symposium*, June 2007, pp. 347–350.
- [29] M. Hsiao, J. Huang, and T. Chang, “A built-in parametric timing measurement unit,” *IEEE Design and Test of Computers*, vol. 21, no. 4, pp. 322–330, July 2004.
- [30] R. B. Staszewski, S. Vemulapalli, et al., “1.3 v 20ps time-to-digital converter for frequency synthesis in 90-nm CMOS,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 53, no. 3, pp. 220–224, March 2006.
- [31] A. H. Chan and G. W. Roberts, “A jitter characterization system using a component-invariant Vernier delay line,” *IEEE Transactions on Very Large Scale Integration Systems*, vol. 12, no. 1, pp. 79–95, January 2004.

- [32] A.H. Chan and G.W. Roberts, “A synthesizable, fast and high-resolution timing measurement device using a component-invariant vernier delay line,” in *Proceedings of International Test Conference*, 2001, pp. 858–867.
- [33] S. Sattler and et al, “PLL built-in self-test jitter measurement integration into 0.18 μ m cmos technology,” in *Proceedings of Testmethods and Reliability of Circuits and Systems Workshop*, 2001.
- [34] M. Z. Straayer and M. H. Perrott, “A multi-path gated ring oscillator TDC with first-order noise shaping,” *IEEE Journal of Solid-State Circuits*, vol. 44, no. 4, pp. 1089–1098, April 2009.
- [35] Ping Lu, P. Andreani, and A. Liscidini, “A 90nm cmos gated-ring-oscillator-based vernier time-to-digital converter for dpll,” in *ESSCIRC, 2011 Proceedings of the*, September 2011, pp. 459–462.
- [36] J. C. Candy and G. C. Temes, *Oversampling Methods for A/D D/A Conversion, Oversampling Delta-Sigma Converters*, IEEE Press, New Jersey, 1992.
- [37] M. Z. Straayer, *Noise Shaping Techniques for Analog and Time to Digital Converters using Voltage Controlled Oscillators*, MIT, Cambridge, MA, 2008.
- [38] I. Hwang, C. Kim, and S. Kang, “A cmos self-regulating VCO with low supply sensitivity,” *IEEE Journal of Solid-State Circuits*, vol. 39, no. 1, pp. 42–48, January 2004.
- [39] T. Kwasniewski, M. Abou-Seido, et al., “Inductorless oscillator design for personal communication devices a 1.2 μ m CMOS process case study,” in *Proceedings of IEEE Custom Integrated Circuits Conference*, May 1995, pp. 327–330.

- [40] K. Hwang and L. Kim, “An area efficient asynchronous gated ring oscillator TDC with minimum GRO stages,” in *Proceedings of IEEE International Symposium on Circuits and Systems*, May 2010, pp. 3973–3976.
- [41] I. Hwang, C. Kim, and S. Kang, “A true single-phase-clock dynamic CMOS circuit technique,” *IEEE Journal of Solid-State Circuits*, vol. 22, no. 5, pp. 899–901, October 1987.
- [42] J. Yuan and C. Svensson, “New TSPC latches and flipflops minimizing delay and power,” in *Digest of Technical Papers of Symposium on VLSI Circuits*, June 1996, pp. 160–161.
- [43] P. Welch, “The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms,” *IEEE Transactions on Audio and Electroacoustics*, vol. 15, no. 2, pp. 70–73, 1967.
- [44] O. Petre and H. G. Kerkhoff, “On-chip tap-delay measurements for a digital delay-line used in high-speed inter-chip data communications,” in *Proceedings of Asian Test Symposium*, November 2002, pp. 122–127.
- [45] T. Hashimoto, H. Yamazaki, A. Muramatsu, T. Sato, and A. Inoue, “Time-to-digital converter with vernier delay mismatch compensation for high resolution on-die clock jitter measurement,” in *IEEE Symposium on VLSI Circuits*, 2008, pp. 166–167.
- [46] L. Yin and P. Li, “Exploiting reconfigurability for low-cost in-situ test and monitoring of digital pll,” in *Proceedings of the 47th Design Automation Conference*, June 2010, pp. 929–934.
- [47] B. Provost and E. Sanchez-Sinencio, “On-chip ramp generators for mixed-signal

- bist and adc self-test,” *IEEE Journal of Solid-State Circuits*, vol. 38, pp. 263–273, February 2003.
- [48] S. R. Das, J. Zakizadeh, S. Biswas, M. H. Assaf, A. R. Nayak, E. M. Petriu, W. B. Jone, and M. Sahinoglu, “Testing analog and mixed-signal circuits with built-in hardware - a new approach,” *IEEE Transaction on Instrumentation and Measurement*, vol. 56, no. 3, pp. 840–855, February 2007.
- [49] B. Dufort and G. W. Roberts, “On-chip analog signal generation for mixed-signal built-in self-test,” *IEEE Journal of Solid-State Circuits*, vol. 34, pp. 318–330, March 1999.
- [50] B. Murmann and B. E. Boser, “A 12 b 75 ms/s pipelined adc using open-loop residue amplification,” *Digest of Technical Papers of IEEE International Solid-State Circuits Conference*, pp. 328–329, February 2003.
- [51] R. Staszewski and P. Balsara, *All-Digital Frequency Synthesizer in Deepsubmicron CMOS*, Wiley-Interscience, Hoboken, 2008.
- [52] W. M. Siebert, “Circuits, signals, and systems,” 1986.
- [53] T. H. Lee and A. Hajimiri, “Oscillator phase noise: A tutorial,” *IEEE Journal of Solid-State Circuits*, vol. 35, pp. 326–336, March 2000.
- [54] R. B. Staszewski, C. M. Hung, N. Barton, M. C. Lee, and D. Leipold, “A digital controlled oscillator in a 90nm digital cmos process for mobile phones,” *IEEE Journal of Solid-State Circuits*, vol. 40, pp. 2203–2211, November 2005.
- [55] F. Herzel and B. Razavi, “A study of oscillator jitter due to supply and substrate noise,” *IEEE Transaction on Circuits System II*, vol. 46, pp. 56–62, January 1999.

- [56] J. Kasdin, “Discrete simulation of colored noise and stochastic processes and $1/f^\alpha$ power law noise generation,” *Proceedings of the IEEE*, vol. 83, no. 5, pp. 802–827, May 1995.
- [57] Y. Xu, K. Hsiung, X. Li, I. Vausieda, S. Boyd, and L. Pileggi, “Opera: Optimization with ellipsoidal uncertainty for robust analog ic design,” *Proceedings of IEEE/ACM Design Automation Conference*, pp. 632–637, June 2005.
- [58] L. Yin and P. Li, “Rf bist for adpll-based polar transmitters with wide-band dco gain calibration,” in *Proceedings of the 12th International Symposium on Quality Electronic Design*, March 2011, pp. 1–8.
- [59] R.B. Staszewski, J.L. Wallberg, S. Rezeq, C.M. Hung, O.E. Eliezer, S.K. Vemulapalli, C. Fernando, K. Maggio, R. Staszewski, N. Barton, et al., “All-digital pll and transmitter for mobile phones,” *IEEE Journal of Solid-State Circuits*, vol. 40, no. 12, pp. 2469–2482, 2005.
- [60] L.R. Kahn, “Single-sideband transmission by envelope elimination and restoration,” *Proceedings of the IRE*, vol. 40, no. 7, pp. 803–806, 1952.
- [61] J. Groe, “Polar transmitters for wireless communications,” *IEEE Communications Magazine*, vol. 45, no. 9, pp. 58–63, 2007.
- [62] K. Waheed and S.N. Ba, “Adaptive digital linearization of a drp based edge transmitter for cellular handsets,” in *50th Midwest Symposium on Circuits and Systems*, 2007, pp. 706–709.
- [63] RB Staszewski, D. Leipold, O. Eliezer, M. Entezari, K. Muhammad, I. Bashir, C.M. Hung, J. Wallberg, R. Staszewski, P. Cruise, et al., “A 24mm² quad-band single-chip gsm radio with transmitter calibration in 90nm digital cmos,” in

Digest of Technical Papers of IEEE International of Solid-State Circuits Conference, 2008, pp. 208–607.

- [64] R.B. Staszewski, J. Wallberg, C.M. Hung, G. Feygin, M. Entezari, and D. Leipold, “Lms-based calibration of an rf digitally controlled oscillator for mobile phones,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 53, no. 3, pp. 225–229, 2006.
- [65] J. Zhuang, K. Waheed, and R.B. Staszewski, “A technique to reduce phase/frequency modulation bandwidth in a polar rf transmitter,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 8, pp. 2196–2207, 2010.
- [66] S. Akhtar, M. Ipek, J. Lin, RB Staszewski, and P. Litmanen, “Quad band digitally controlled oscillator for wcdma transmitter in 90nm cmos,” in *IEEE Custom Integrated Circuits Conference*, 2006, pp. 129–132.
- [67] K. Waheed and R.B. Staszewski, “Characterization of deep-submicron varactor mismatches in a digitally controlled oscillator,” in *Proceedings of the IEEE 2005 Custom Integrated Circuits Conference*, 2005, pp. 605–608.
- [68] R.B. Staszewski, C. Fernando, and P.T. Balsara, “Event-driven simulation and modeling of an rf oscillator,” in *Proceedings of the 2004 International Symposium on Circuits and Systems*, 2004, vol. 4, pp. IV–641.
- [69] R.B. Staszewski, S. Vemulapalli, P. Vallur, J. Wallberg, and P.T. Balsara, “1.3 v 20 ps time-to-digital converter for frequency synthesis in 90-nm cmos,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 53, no. 3, pp. 220–224, 2006.