INVITED PAPER

Boolean Computation Using Self-Sustaining Nonlinear Oscillators

By illustrating the inherent noise-immunity advantages of phase-encoded logic over traditional level-based logic, the author shows how Boolean computation using a wide variety of natural and engineered oscillators becomes possible.

By Jaijeet Roychowdhury, *Fellow IEEE*

ABSTRACT | Self-sustaining nonlinear oscillators of practically any type can function as latches and registers if Boolean logic states are represented physically as the phase of oscillatory signals. Combinational operations on such phase-encoded logic signals can be implemented using arithmetic negation and addition followed by amplitude limiting. With these, general-purpose Boolean computation using a wide variety of natural and engineered oscillators becomes potentially possible. Such phase-encoded logic shows promise for energy-efficient computing. It also has inherent noise immunity advantages over level-based logic.

KEYWORDS | Green computing; oscillators; phase encoding

I. INTRODUCTION

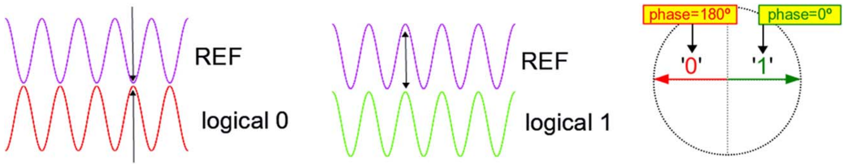
Self-sustaining oscillators abound in nature and in engineered systems; examples include mechanical clocks [1], electronic ring [2]–[4] and LC oscillators [5], spin-torque oscillators [6]–[10], lasers [11]–[13], microelectromechanical systems/nanoelectromechanical systems (MEMS/NEMS)-based oscillators [14], [15], the heart's neuronal pacemakers [16], engineered molecular oscillators such as the repressilator [17], etc. The defining characteristic of a self-sustaining oscillator is that it generates sustained "motion" without requiring any stimulus of a similar nature, i.e., it produces an output that changes with time indefinitely, usually in a periodic or quasi-periodic [18] fashion, in the absence of any input that changes with time. If left undisturbed, most practical self-sustaining oscillators become periodic with time and settle to a single amplitude of oscillation. For the latter property[1] to hold, the oscillator must be nonlinear, i.e., it must be a self-sustaining nonlinear oscillator (SSNO). SSNOs exhibit interesting dynamical properties, for example, synchronization [19]–[21] and pattern formation [22]–[25] can result when they are coupled together. Biological phenomena such as the synchronized flashing of fireflies [26], circadian rhythms [27], [28], and epilepsy [29] result from the interaction of SSNOs, while coupled systems of SSNOs have been shown to have image processing capabilities [24], [30] and have been proposed for associative memories [31], [32].

In this paper, we focus on the use of SSNOs for general-purpose Boolean computation. We first review recent work that establishes that SSNOs can serve as substrates for Boolean latches and flip-flops. By exploiting a phenomenon known as subharmonic injection locking (SHIL), almost any SSNO can store logical states stably if logic is encoded in phase. This result implies that almost any oscillator, from any physical domain, can potentially be used for Boolean computation. Examples include complementary metal–oxide–semiconductor (CMOS) ring oscillators, spin-torque nano-oscillators, synthetic biological oscillators, MEMS/NEMS-based oscillators, nanolasers, and even mechanical clocks.

Manuscript received October 16, 2014; revised August 17, 2015; accepted September 3, 2015. Date of publication October 15, 2015; date of current version October 26, 2015. This work was supported by the U.S. National Science Foundation (NSF) under Grant CCF-1111733 (PHLOGON). Support from NSF Grant EEC-1227020 (NEEDS) was instrumental in enabling the Berkeley MAPP infrastructure, within which design tools for SSNO-based phase logic are being developed.
The author is with the Electrical Engineering and Computer Science Department, University of California, Berkeley, CA 94720 USA (e-mail: jr@berkeley.edu).

Digital Object Identifier: 10.1109/JPROC.2015.2483061

[1]Known technically as asymptotic orbital stability [18].



(a) (b)

Fig. 1. *Encoding Boolean logic using the relative phase of oscillatory signals. (a) Time domain. (b) Phasor.*

Using SSNOs for phase-encoded Boolean logic brings up interesting possibilities. Since logic values are encoded in phase, or time shift, switching between them does not, in principle, involve energy expenditure. We demonstrate this using a high-Q (energy-efficient) oscillator design that consumes essentially no energy to switch quickly (in half an oscillation cycle) between phase-logic states. We also outline how phase logic can have generic noise immunity advantages over level-based encoding of logic.

Phase-encoded logic was first proposed in the 1950s by Goto [33], [34] and von Neumann [35], [36], who showed that if the phase of a signal (relative to another signal, the reference) is used to encode Boolean logic states, combinational operations can be implemented using arithmetic addition and negation. Moreover, they devised a circuit that served as a phase-logic latch, i.e., it could store a Boolean logic state encoded in phase.[2] In the early 1960s, the Japanese constructed phase-logic computers (dubbed parametrons [37]–[40]) that enjoyed brief success on account of their compactness and reliability compared to the vacuum-tube-based machines that were the mainstay of computing at the time. However, phase-based computers were soon overshadowed by level-based ones employing microscopic semiconductor devices within integrated circuits. The difficulty of miniaturizing and integrating components in Goto/von Neumann's phase-logic latches contributed to their demise. Although subsequently, Goto and colleagues showed that Josephson-junction devices could be used for phase logic [41], [42], these require extremely low temperatures for operation, hence are not practical in most applications.

With CMOS miniaturization facing fundamental energy and noise barriers today, there has been an ongoing search for alternative computational paradigms [43], [44]. In this context, the facts that phase-encoded logic allows essentially zero-energy bit flips, and is capable of resisting noise better than level-encoded logic, provide considerable motivation for reexamining it as a candidate technology for the post-CMOS era. Also, that any SSNO can potentially serve as a latch removes an important limitation that prior phase-based logic schemes have faced, e.g., many types of nanoscale SSNOs become candidates for phase latches.[3]

The remainder of this paper is organized as follows. In Section II, the concept of encoding logic in phase is outlined and it is shown how SSNOs can be made to serve as phase-logic latches. An example of a state machine using phase logic is also provided. In Section III, energy consumption and speed in SSNO-based phase logic are explored. The superior noise immunity properties of phase-encoded logic are outlined in Section IV.

II. PHASE-LOGIC LATCHES USING SSNOs

Fig. 1 illustrates the use of relative phases to represent Boolean (binary) logic states.[4] A periodic signal, denoted REF in the figure, serves as a reference with respect to which the phases of other signals are measured. As shown in Fig. 1(a), we choose the opposite phase to represent logical 0, and the same phase to represent logical 1. Any other choice where the two logic levels are maximally separated in phase (i.e., by 180°) would be equally valid. Implicit in this scheme is the assumption that all signals encoding logic using phase are at the same frequency as REF and are phase locked to it. The two phase-encoded Boolean logic states can also be depicted as phasors [46], as shown in Fig. 1(b). In the following, we use "1" and "0" to represent the phase-encoded Boolean states shown in Fig. 1.

A. SHIL Makes SSNOs Phase Bistable

Fig. 2 indicates how an SSNO can be set up as a phase-logic latch, i.e., if left undisturbed, it will output either a "1" or a "0" (and no other phase) indefinitely in phase synchrony with a provided REF signal.[5] We assume that a periodic REF signal with frequency f_{REF}, as shown in Figs. 1 and 2, is available. We also require another signal

[2] This circuit was not, however, an SSNO; it relied on a sinusoidal (AC) parametric pump (power source) to achieve bistability in phase.

[3] In this paper, we use CMOS ring and high-Q LC oscillators for illustration, but other nanoscale SSNOs such as spin-torque oscillators, NEMS-based oscillators, synthetic biological oscillators, etc., could also serve as substrates for SSNO-based phase logic.

[4] Ternary and multistate logic values can also be encoded in phase; indeed, SSNOs can serve as multistate latches [45]. We focus on the binary case in this paper.

[5] A mathematical proof of this fact for a generic SSNO is available in [45].

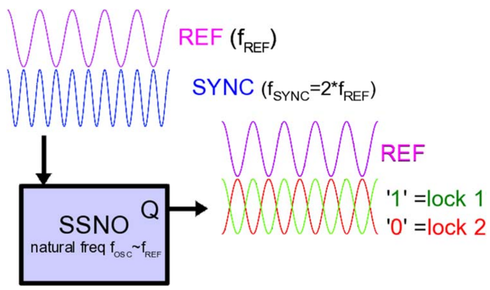


Fig. 2. *SSNO serving as a bistable phase latch.*

SYNC with frequency exactly twice that of REF, i.e., $f_{SYNC} = 2f_{REF}$. SYNC is phase-synchronized to REF, as illustrated in Fig. 2. In practice, SYNC can be derived from REF by frequency doubling [47], or REF from SYNC by frequency division [48].

The SSNO being used as a phase-logic latch needs to have a natural frequency near that of REF, i.e., $f_{OSC} \simeq f_{REF}$, or $f_{OSC} \simeq f_{SYNC}/2$. Fig. 3(a) illustrates SYNC, juxtaposed against the oscillator's output at its natural frequency. Since the oscillator's natural frequency is only approximately half that of SYNC, the two signals are not necessarily phase synchronized, as indicated by the drift between the two signals.

The key to devising a phase latch is to inject the SYNC signal into the oscillator, as shown in Fig. 2. With SYNC injection and under the right conditions [45], subharmonic injection locking occurs: the oscillator "forgets" its natural frequency f_{OSC}, adopts a frequency of exactly $f_{SYNC}/2$, and becomes phase synchronized with SYNC in one of two possible phases that are 180° apart, as depicted in Fig. 3(b) by the signals marked "0" and "1." In other words, when SYNC is injected, the oscillator becomes bistable in phase at exactly half the frequency of SYNC and in phase lock with it. That there must be two stable phase-lock states is intuitive because SYNC can "see" no difference between the two lock states [see Fig. 3(b)]; i.e., if the "0" lock state exists, symmetry dictates that the "1" lock state must also exist.[6] Since the oscillator's output is phase locked to SYNC, it is also phase locked to REF (since SYNC and REF are phase locked by design). The frequency of the oscillator under SHIL becomes identical to that of REF; the key to using the oscillator's two SHIL states for phase logic is that they can be distinguished using REF.

Oscilloscope measurements of bistable SHIL in a CMOS ring oscillator [49] are shown in Fig. 4. The SYNC and REF waveforms shown were generated by a programmable function generator to be in phase lock, with REF at exactly half the frequency of SYNC. It can be seen that the oscillator's output is at the same frequency as REF.

[6]A rigorous proof of SHIL and its bistability can be found in [45].

In Fig. 4(a), observe that the peaks of REF are roughly halfway between the peaks of the oscillator's output, whereas in Fig. 4(b), the peaks of REF and the oscillator's output are almost aligned. These are the two SHIL states.[7]

Using combinational operations, SSNOs featuring bistable SHIL can be turned into *D* latches [50]. We first review how combinational operations can be implemented using phase logic.

B. Combinational Logic in Phase

It is well known that certain sets of basic logical operations, when composed, suffice to implement any combinational logic function. Such sets are called logically or functionally complete [51]. For example, the Boolean function sets {AND, NOT}, {OR, NOT}, {NAND}, and {NOR} are all logically complete. When logic is encoded in phase as in Fig. 1, it is advantageous to use the logically complete set {NOT, MAJ} [35], [36], where NOT is the standard Boolean inversion operation and MAJ is the three-input majority operation, returning whichever Boolean value occurs more than once among its three inputs.[8] For example, MAJ$(0, 0, 1)$ returns 0; MAJ$(1, 0, 1)$ returns 1.

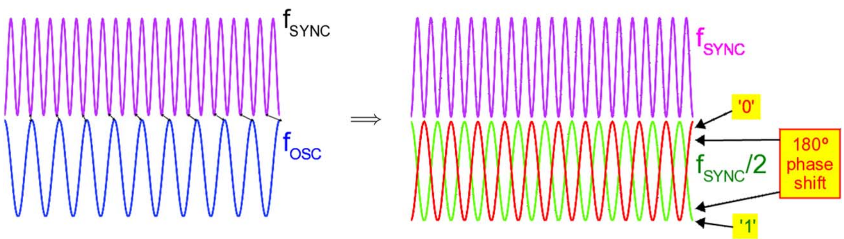
The reason {NOT, MAJ} is interesting for phase-encoded logic is that both functions can be implemented using elementary arithmetic operations. NOT can be implemented simply by arithmetic negation, as is apparent from Fig. 1; it can also be performed in other implementation-specific ways (see Section II-C). MAJ(A, B, C), where A, B, and C are all phase-encoded logic signals taking values in {"0," "1"}, can be implemented by (essentially) adding A, B, and C arithmetically. This is easy to appreciate graphically using the phasor representation for phase logic [Fig. 1(b)], as illustrated using the two examples in Fig. 5. Since "0" and "1" are represented by equal and opposite phasors, adding "0," "0," and "1" leads to the "1" being cancelled by one of the "0"s, leaving "0," which is identical to MAJ("0," "0," "1"). Adding "1," "1," and "1" results in a phasor with three times the amplitude of "1," but with the same phase; if the amplitude is normalized after addition (i.e., via amplitude limiting, easily achieved in certain implementations), the result is "1," which is the same as MAJ("1," "1," "1"). Arithmetic addition with amplitude limiting can be confirmed to be identical to MAJ for all other input combinations.

C. Setting and Resetting SSNO SHIL Logic States; Phase-Based *D* Latches

To exploit SSNO bistability under SHIL (Section II-A) for general purpose computation, it is necessary to control the SSNO's SHIL state. The basic mechanism by which this

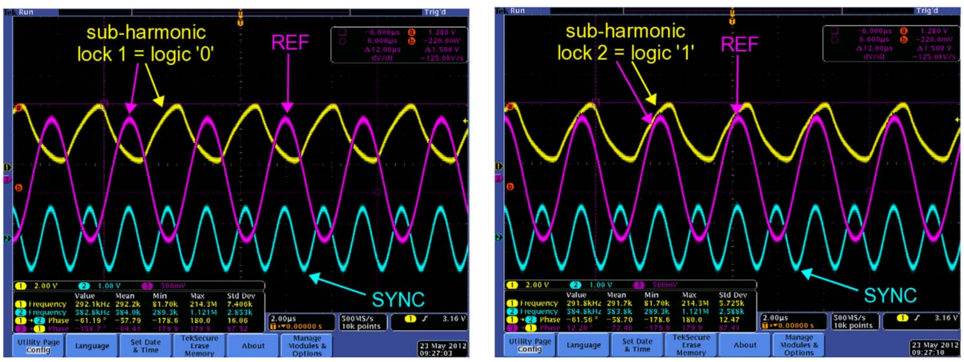
[7]Which state the oscillator locks to depends on initial conditions, transients, noise, etc., during circuit startup. Section II-C describes circuits and techniques for setting and manipulating the state.

[8]That {NOT, MAJ} is logically complete becomes apparent when we note that AND$(A, B) =$ MAJ$(0, A, B)$; or that OR$(A, B) =$ MAJ$(1, A, B)$.



(a) (b)

Fig. 3. ***Subharmonic injection locking in an SSNO stores phase-logic states. (a) Before SHIL. (b) After SHIL.***



(a) (b)

Fig. 4. ***Oscilloscope traces showing bistable SHIL in a CMOS ring-oscillator with SYNC injection. (a) Logic state "0." (b) Logic state "1."***

can be achieved is simple, as illustrated in Fig. 6(a): a phase-encoded logic signal A is injected into the SSNO momentarily, e.g., by closing the switch briefly. It can be shown [49] that under the right circumstances, the SSNO will adopt the logic state of A and retain it after A is no longer injected. Injecting the phase-encoded logic signal A (which is at the frequency of REF) removes SHIL bistability under SYNC injection and sets the oscillator's phase close to that of A [49, Fig. 4][9]; when A is removed, bistability is restored and the oscillator adjusts its phase smoothly to the nearest SHIL stable lock state, i.e., that of A.

Fig. 6(b) shows a CMOS ring SSNO with SYNC and A injections; the two current injections are at the same node in this case, though they can be incorporated in a variety of alternative ways. The dynamics of setting and resetting the SSNO's SHIL state can be seen in the transient simulation plots in Fig. 7. The first cycle of the ring oscillator's output shows startup transients in the absence of SYNC injection. SYNC injection starts at $t \sim 17.5$ ps (see the waveform labeled SYNC). The oscillator responds within about two cycles by changing its frequency to $f_{\mathrm{SYNC}}/2$ and settling to an arbitrary phase-logic state; in this case "1," indicated by the oscillator's stage 2 (red) output's peaks being almost aligned with REF's troughs. At $t \sim 40$ ps, about one cycle of $A =$ "0" is injected momentarily (see the label $A =$ "0" `injected`); the oscillator's waveforms change significantly in response. By about $t \sim 70$ ps, the oscillator settles to the other bistable SHIL state, i.e., "0," as seen by the fact that the trough of REF is no longer aligned with the oscillator's stage 2 (red) output's peaks, but is instead roughly halfway between the peaks. The SHIL state is then switched back to "1" by momentarily injecting $A =$ "1" at $t \sim 80$ ps; the oscillator responds by switching back to phase-logic state "1" by $t \sim 110$ ps, with the stage 2 output's peak aligned again with the trough of REF.

The basic ring oscillator phase latch topology of Fig. 6(b) can be easily adapted [49] into a gated D latch (D latch with Enable) [50] with the help of the

[9]This happens because "simple" (i.e., fundamental harmonic) injection locking [52]–[54], in which the oscillator becomes phase locked to A with exactly one stable state, overrides SHIL.

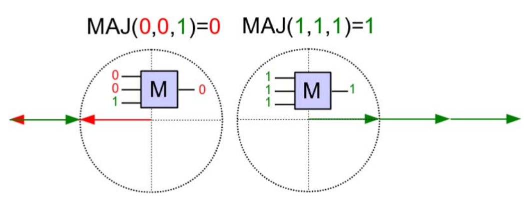
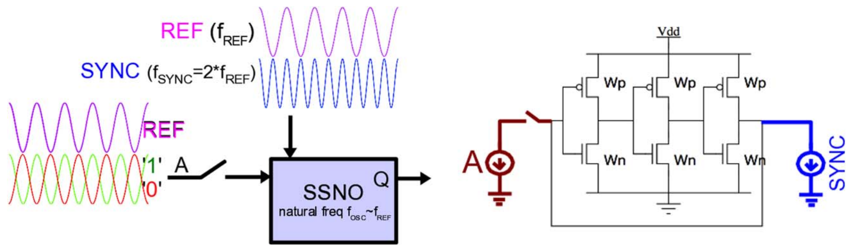


Fig. 5. ***Examples illustrating*** $\mathrm{MAJ}(A, B, C)$ ***in phase logic.***



(a) (b)

Fig. 6. *Controlling the lock state of an SSNO under SHIL. (a) Generic scheme. (b) CMOS ring oscillator example.*

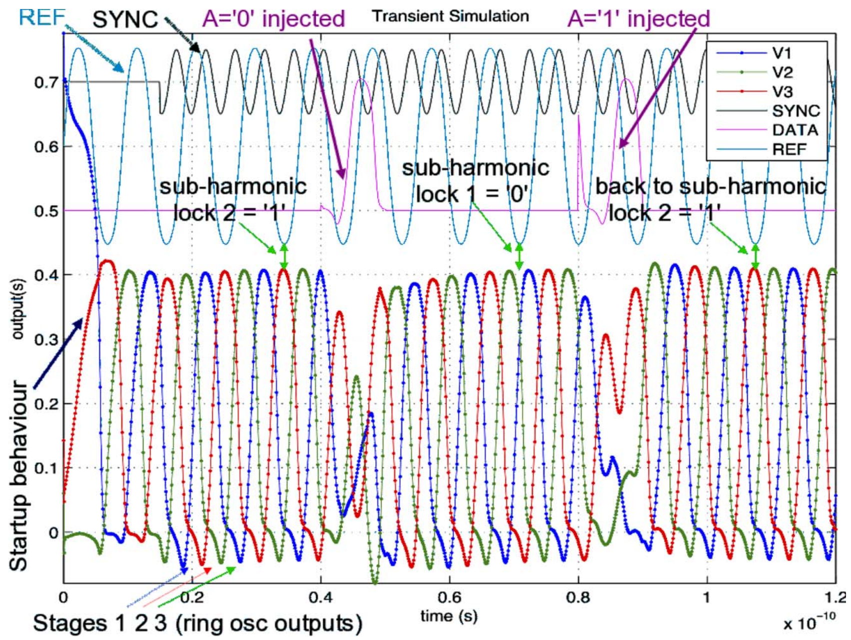


Fig. 7. *Transient simulation of the circuit in Fig. 6(b).*

combinational primitives {NOT, MAJ}, as shown in Fig. 8. The chain of three inverters represents the CMOS SSNO of Fig. 6(b) with the SYNC injection included, but without the input *A*; direct feedback from the last inverter to the first is broken and a MAJ gate introduced, as shown. All logic inputs/outputs (I/Os) (D, EN, and Q) are phase encoded. The inverter driven by the *EN* (Enable) input, representing logical inversion (using phase encoding), can be implemented simply as a standard CMOS inverter. When $EN =$ "1," D is fed to two inputs of the majority gate in the ring oscillator loop, resulting in the ring oscillator's feedback loop being broken and Q being set to D. When $EN =$ "0," D is ignored and complementary logic values are fed to two inputs of the majority gate in the ring oscillator loop, which sets Q to the output of the third inverter in the ring oscillator (which is the third input to the majority gate), thereby completing the ring oscillator's feedback loop, restoring bistability and retaining the previously set state.

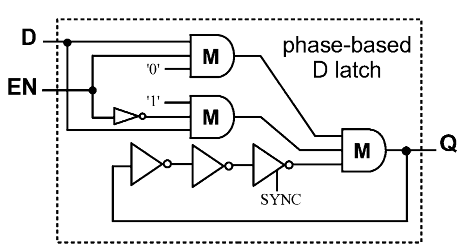


Fig. 8. *Phase-based D latch with enable using a CMOS ring oscillator [49]. The gates marked M are three-input majority gates.*

D. State Machines Using SSNO-Based Phase Logic

With D latches for storage and combinational logic using {NOT, MAJ}, we have the basic components for a von Neumann computer [55] in SSNO-based phase-encoded

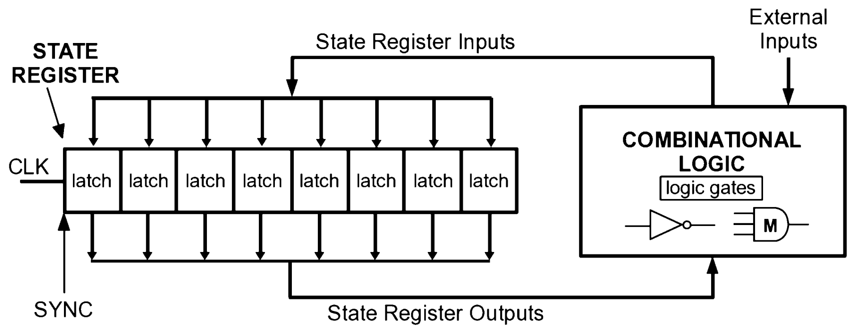


Fig. 9. *Structure of a state machine using SSNO-based phase latches and {NOT, MAJ}-based combinational logic.*

logic. One of the most important units of a computer is the finite state machine (FSM), used for, e.g., the control unit [55], [50] of a stored program computer. The general structure of a state machine, adapted to the phase-logic context, is shown in Fig. 9. All signals are phase encoded, including the CLK signal which alternates between phases "0" and "1," holding each for a few cycles of REF. It is also easy to devise D latches where CLK (ENable) is level based, while the logic signals remain encoded in phase.

Fig. 10(b) shows an example of a simple Mealy FSM that utilizes a full adder for the combinational logic and a single bit for the state, all in phase logic [49]. The latch is constructed using two of the D latches shown in Fig. 8, arranged in a master–slave [50] configuration to prevent races. The two inputs to the state machine, a and b, are inputs to the full adder. The carry-out (cout) bit of the full adder is the input to the latch; the output of the latch feeds back as the carry-in (cin) input of the full adder. This arrangement implies that the "0" $\rightarrow$ "1" state transition can only occur if $a = b =$ "1," and the "1" $\rightarrow$ "0" transition if $a = b =$ "0." The complete state transition diagram of the FSM is shown in Fig. 10(a).

III. ENERGY EFFICIENCY AND SWITCHING SPEED OF PHASE-ENCODED LOGIC

Having outlined the fundamental design and operational principles of SSNO-based phase logic, we now explore two fundamental questions: how much energy does it take to flip a bit in phase-encoded logic, and how quickly can a bit be flipped?

A. Energy Dissipation and Amplitude/Phase Change Rates in High-Q Oscillators

For reference, the minimum energy expended by level-based logic (for which a single inverter serves as an exemplar) in flipping a bit from 0 to 1 and back again to 0 is CV_{DD}^2,[10] averaging $(1/2)CV_{\mathrm{DD}}^2$ per bit flip. Just to maintain oscillation, a minimum energy of $3CV_{\mathrm{DD}}^2$ is dissipated per cycle by the three-stage ring oscillator of the previous section, hence it is not a compelling candidate for energy-efficient computation. Although dissipation can be lowered using small supply voltages,[11] it is typically at the cost of decreased oscillation frequency and logic switching speed.

However, high-Q LC oscillators[12] (e.g., [56] and [57]) are inherently energy efficient, dissipating only about $1/Q$ of the energy stored in the LC tank[13] per cycle, where Q is the quality factor of the oscillator. LC oscillators are also capable of very high-frequency oscillation, e.g., a 300-GHz LC oscillator has been reported [58]. These characteristics make high-Q LC oscillators interesting candidates for exploring how energy efficient, and how fast, phase-encoded logic can be.

Using a proof-of-concept circuit, we show that it is possible to make high-Q LC oscillators suitable for phase logic by subjecting them to SHIL, and to flip their phase-logic states in just half a cycle with no energy consumption (beyond the small amount of energy needed per cycle to maintain oscillation). Phase logic can therefore be about Q times more energy efficient than level-based flipping, without compromising switching speed.[14] With Q factors of $10^2 - 10^6$ readily achievable today, great energy savings over level-based logic can potentially result.

Being able to flip a bit (i.e., disturb one normal oscillation pattern and settle to another) within a single cycle of a high-Q oscillator may appear counterintuitive, since amplitude changes in high-Q oscillators are very slow. The reason is that amplitude changes necessarily involve energy dissipation or accumulation in the LC tank. This energy can be removed or supplied only in small installments per cycle in high-Q oscillators, translating to

[10] Where C is the capacitive load at each inverter and V_{DD} is the supply voltage.

[11] Ring oscillators operating at 100 mV, using standard CMOS technologies, have been reported [3].

[12] More generally, high-quality harmonic oscillators in any physical domain.

[13] $(1/2)CV_{\mathrm{osc}}^2$, where V_{osc} is the peak amplitude of oscillation.

[14] Indeed, SSNO-based phase logic could possibly operate at far higher speeds than level-based logic is currently capable of, depending on the oscillator's frequency.

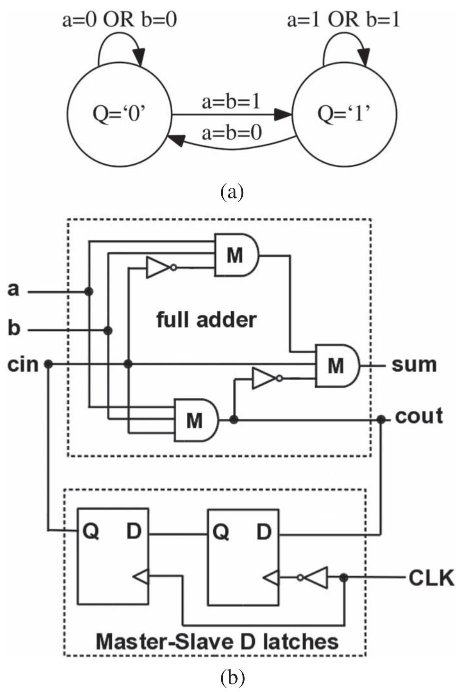


Fig. 10. ***SSNO-based 1-b state machine example [49]. (a) State transition graph. (b) Implementation with phase D latches and {NOT, MAJ} gates.***

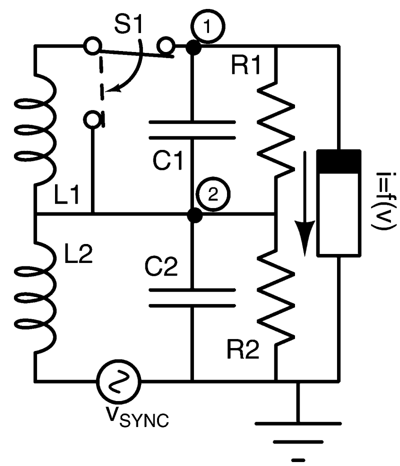


Fig. 11. ***High-Q LC oscillator-based phase-logic latch circuit.***

slow amplitude transients, with time constants of the order of Q cycles of oscillation.

However, flipping a phase-encoded logic bit involves only time shifting or delaying oscillatory waveforms. There appears to be no fundamental physical principle dictating a minimum energy needed to achieve a time shift, therefore, in principle, phase-encoded bit flipping would seem achievable with no energy consumption at all. With no need to supply or remove energy, the speed at which time shifts can be made would seem limited only by the time constants of the oscillatory dynamics of the LC tank. Since the LC tank changes phase by 360° as a matter of course during each cycle of oscillation, it should be possible to shift phase by 180° (i.e., to the other stable SHIL phase-lock state) in half a cycle.[15] Our experiments below confirm this reasoning and provide proof of the concept that zero energy bit flips can be achieved in half a cycle of a high-Q LC oscillator.

B. High-Q LC Oscillator-Based Phase-Logic Latch

Fig. 11 depicts the schematic of a high-Q LC oscillator that serves as a phase-logic latch. The circuit is based on the standard parallel-*RLC* tank and nonlinear resistor topology [59]. The oscillator's main tank is the upper one, consisting of L_1, C_1, and R_1; it is tuned to a natural frequency f_{OSC}, set close to $f_{\mathrm{REF}} = f_{\mathrm{SYNC}}/2$. The single-pole double-throw (SPDT) switch S_1, normally kept closed in the position shown, is used for phase-logic bit flipping, as described below.

To facilitate subharmonic injection locking, a second tank consisting of L_2, C_2, and R_2 is tuned to f_{SYNC} and placed in series with the main tank. The negative resistance nonlinearity is connected across both tanks, as shown. The SYNC signal is injected as a voltage source in series with L_2. The second tank magnifies the effect of SYNC on the nonlinear resistor by a (typically large) factor of $R_2/2\pi f_{\mathrm{SYNC}} L_2$, thereby sensitizing the oscillator to SHIL from the SYNC signal.

The nonlinearity needs a negative differential resistance region to power the circuit and enable self-oscillation. We have used the current–voltage characteristic

$$i = f(v) \triangleq k_1 \tanh(k_2 v) + g_{\mathrm{SHIL}}(v) \tag{1}$$

where

$$g_{\mathrm{SHIL}}(v) \triangleq \begin{cases} k_3^2 (v+A)^2, & \text{if } v < -A \\ 0, & \text{if } -A \le v \le A \\ k_3^2 (v-A)^2, & \text{if } v > A. \end{cases} \tag{2}$$

The $\tanh(\cdot)$ term in (1) provides the negative differential resistance needed for oscillation [59]. The $g_{\mathrm{SHIL}}(v)$ term facilitates second subharmonic injection locking by introducing asymmetry in $f(v)$ for input amplitudes larger than A. Such asymmetry helps second-harmonic components of the voltage input to $f(v)$ affect the phase of the fundamental component of its current output. Describing function-based feedback analysis [59] can be used to show that this feature is important for susceptibility to SHIL.

[15]That the slowness limitation of amplitude changes in high-Q LC oscillators does not apply to their phase/time shifting characteristics appears not to be widely appreciated.

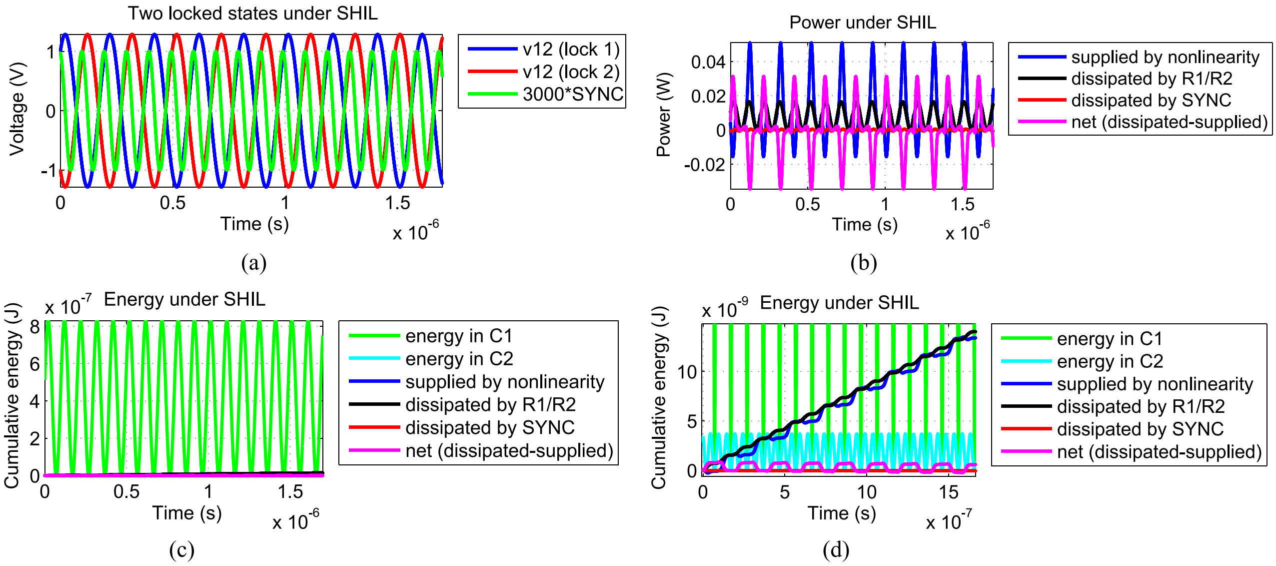


Fig. 12. ***Voltages, power, and energy consumption of LC oscillator under SHIL. (a) Main tank voltage (with SYNC overlaid). (b) Instantaneous power. (c) Cumulative energy. (d) Cumulative energy (detail).***

For natural oscillation to occur (in the absence of any injection at $v_{\rm SYNC}$), it is necessary for

$$\frac{1}{R_2} > -k_1 k_2 > \frac{1}{R_1} \tag{3}$$

i.e., the maximum negative differential resistance of $f(v)$ needs to overcome the loss due to R_1, but not the loss due to R_2; the latter condition prevents fundamental-mode natural oscillation at $f_{\rm SYNC}$. The parameter A in (2) is set at or around the amplitude of natural oscillation (i.e., in the absence of SYNC injection).

The simulations below use the following values of circuit parameters:

$$\begin{aligned} &L_1 = 1\ \text{nH}, \quad C_1 = 1\ \mu\text{F}, \quad R_1 = 100\ \Omega, \quad L_2 = \frac{L_1}{2} \\ &C_2 = \frac{C_2}{2}, \quad R_2 = 90\ \Omega, \quad k_1 = \frac{1}{30}, \quad k_2 = \frac{0.0102}{k_1} \\ &k_3 = 40 k_1 k_2, \quad A = 0.9. \end{aligned} \tag{4}$$

The switch S_1 was modeled with on resistance 0 Ω and off resistance 10 kΩ. With these parameters, $f_{\rm OSC} \simeq (1/2\pi\sqrt{L_1 C_1}) \sim 5.03292$ MHz. $f_{\rm REF}$ was taken to be 5.0328 MHz, with $f_{\rm SYNC} = 2 f_{\rm REF}$. The SYNC injection was

$$v_{\rm SYNC}(t) = 10^{-3} k_1 \cos(2\pi f_{\rm SYNC}\, t). \tag{5}$$

Fig. 12(a) shows the voltage of the main tank of the oscillator under SHIL.[16] The two locks, representing logic levels "0" and "1," can be seen to be exactly 180° out of phase, as predicted by theory [45], [49].

The instantaneous power of the various components of the circuit are shown in Fig. 12(b). Power is supplied to the circuit by the nonlinearity $i = f(v)$, and dissipated primarily by the tank losses R_1 and R_2. The SYNC injection signal can also dissipate or supply power, while the switch S_1 dissipates power, but these amounts are negligible in our setup. Fig. 12(c) and (d) depicts cumulative energies (i.e., integrated power) supplied/dissipated by the components; also overlaid are the instantaneous energies of the tank capacitors C_1 and C_2, the peak values of which represent the total energy stored in each tank. The peak value for C_1 indicates that the energy of the main tank is about 0.829 μJ.

As expected in periodic lock, the energy supplied by the nonlinearity during each cycle exactly compensates the energy dissipated (primarily by the tank losses); the net energy trace in Fig. 12(d) periodically crosses zero, implying that no energy is being gained or lost by the tanks. The energy supplied to (and dissipated by) the oscillator over each cycle is seen to be about 1.685 nJ, implying an effective Q factor[17] of about 492 in subharmonically injection locked operation.

[16]All results are from simulation using MAPP [60], [61]. Harmonic balance (HB) [62], [63] was used to find the two locked steady states; transient simulations were initialized with the HB solutions.

[17]Because of the nonlinear resistor, the Q of a self-sustaining oscillator is typically lower—by about 6× in this case—than the ideal Q factor of the linear tank alone [64].

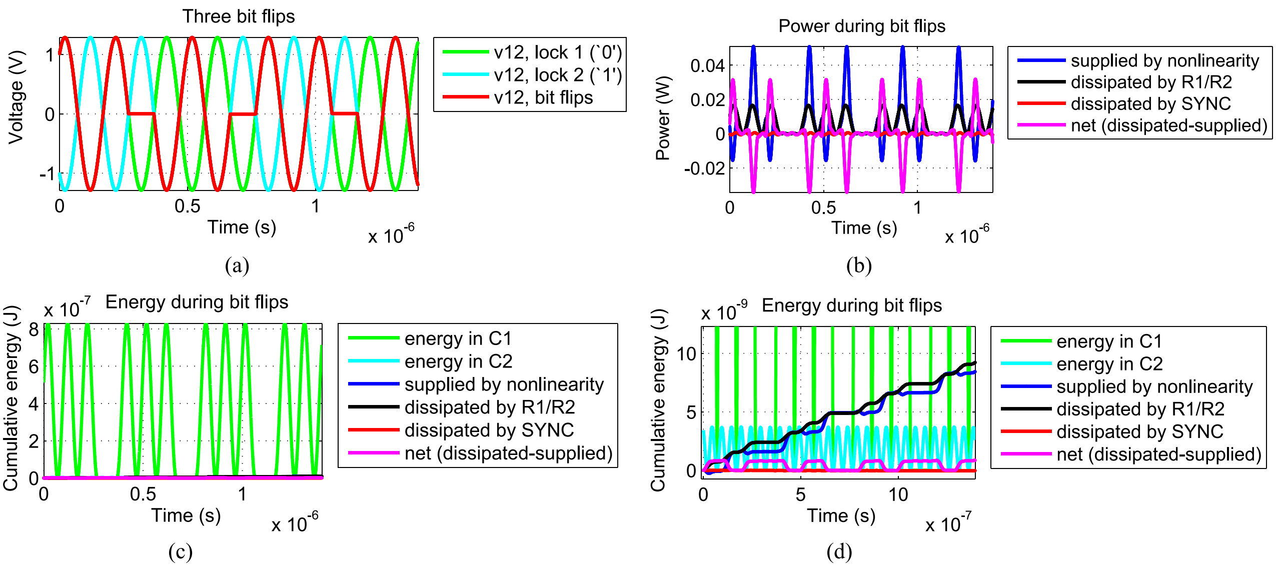


Fig. 13. ***Voltages, power, and energy consumption of LC oscillator undergoing bit flips. (a) Main tank voltage overlaid on "0"/"1" waveforms. (b) Instantaneous power. (c) Cumulative energy. (d) Cumulative energy (detail).***

C. Speed and Energy During Bit Flips

The SPDT switch S_1 in Fig. 11 can be used to flip the oscillator's lock between the two SHIL states shown in Fig. 12(a). If S_1 is flipped to short the inductor L_1 when the voltage across it is zero, the main tank's dynamics are frozen in time until S_1 is flipped back. Flipping the switch for half an oscillation cycle delays the tank just enough to move the oscillator from one lock state to the other.

The simulation results in Fig. 13 illustrate this technique of achieving phase-logic bit flips.[18] S_1 is flipped for half a cycle three times (starting around 0.27, 0.67, and 1.06 μs), leading to three bit flips. Fig. 13(a) shows the voltage across the main tank overlaid on the two lock states of Fig. 12(a), illustrating how well the bit flips from each state to the other. Power waveforms are shown in Fig. 13(b), while cumulative energies are shown in Fig. 13(c) and (d). Energy consumption during bit flipping is small, since the oscillator is essentially stopped when the bit is being flipped. Similar to Fig. 12(d), the net energy graph in Fig. 13(d) crosses zero after bit flipping, indicating that no energy is being gained or lost by the tanks. This shows that the energy benefits due to the high Q of the oscillator are reaped even as bits are flipped at high speed (in half an oscillation cycle).

IV. NOISE IMMUNITY OF PHASE-ENCODED LOGIC

Phase-encoded logic also offers intrinsic noise immunity advantages over level-based logic. The underlying reason for this noise immunity is easy to appreciate graphically.

[18]It is also possible to use other techniques, such as voltage or current injections, to flip the oscillator's state.

Fig. 14 depicts the impact of small and large noise if logic is encoded as levels. For comparison with the phase-encoded case below, a diagram similar to Fig. 1(b) is used to represent the logical states 0 and 1, but these simply represent levels (with no phase); a positive level represents 1 and a negative level (of equal amplitude) represents 0, with the halfway point between them the bit error threshold. In Fig. 14(a), the impact of adding fixed-amplitude "small" noise (i.e., the noise is less than the distance to the threshold, i.e., the "signal") is shown. This random noise adds to, or subtracts from, the signal with equal probability. In either case, the resulting signal remains positive since the noise is small, hence there is no bit error. But if the fixed noise is larger in value than the signal, as shown in Fig. 14(b), this situation changes. When the noise adds to the signal, there is no bit error, but when it subtracts, there is always a bit error, since the result becomes negative, crossing the bit error threshold. Hence, when the noise is larger than the signal, level-based logic encoding suffers a 50% probability of error, i.e., the bit becomes perfectly random, losing all information.

The situation when logic is encoded in phase is depicted in Fig. 15. Here, the signal values "0" and "1" are phasors, exactly as in Fig. 1(b); the noise added is also a phasor at the same frequency. In this case, the bit error thresholds are the vertical phasors at $\pm 90^\circ$, i.e., the phase halfway between the "0" and "1" states. Fig. 15(a) shows the case when the noise amplitude is less than the signal's. Because the noise is random, its phase is uniformly distributed in $[0^\circ, 360^\circ]$, as shown. The worst case phase error caused by the additive noise, denoted $\Delta\theta$, is less than 90° in absolute value; hence, there is no bit error. For "small" noise, therefore, phase encoding and level encoding are identical from a bit error perspective.

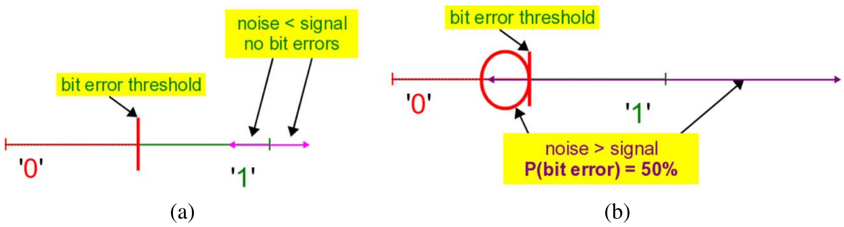


Fig. 14. ***Level-based logic encoding: bit error rates for (a) small noise amplitude and (b) large noise amplitude.***

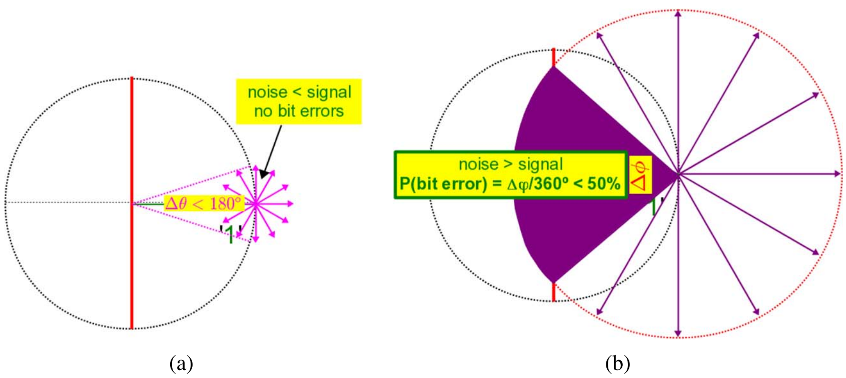


Fig. 15. ***Phase-based logic encoding: bit error probabilities for (a) small noise amplitude and (b) large noise amplitude.***

When the noise amplitude is "large" (i.e., greater than the signal's), the situation in the case of phase encoding differs markedly from that for level encoding, as shown in Fig. 15(b). The shaded region depicts the range of noise phases ($\Delta\phi$) that lead to a bit error. Importantly, $\Delta\phi$ is always less than 180°, implying a bit error probability of less than 50% even when the noise amplitude is greater than that of the signal. Indeed, for noise amplitudes that are only slightly greater than the signal's, the bit error probability is very small, in stark contrast with the level-based case. Phase-based encoding approaches a 50% bit error probability only as the noise amplitude tends to infinity.

These noise characteristics of phase encoding are well known in communication theory [65]; indeed, the above reasoning is essentially identical to that establishing the superior noise performance of binary phase-shift keying (BPSK) over binary amplitude-shift keying (BASK). Phase-based logic encoding simply leverages this fact to improve noise immunity at the physical implementation level of Boolean computing.

V. CONCLUSION

Recent developments in phase-encoded logic have made it a promising alternative computational scheme to explore for today's nanoscale integration era. That almost any SSNO can serve as a phase-logic latch implies that many new substrates for phase-based logic can potentially be exploited. The facts that energy-efficient oscillators serving as phase-logic latches are capable of switching very quickly in an energy-neutral manner, and that phase encoding brings inherent noise immunity benefits, indicate that it is a promising new direction for energy-efficient and robust computing.

Further exploration at the device, circuit, and architectural levels is needed to translate conceptual promise into engineering practice. Nanoscale devices that are naturally oscillatory, such as spin-torque devices [6], [8], [66], [67],[19] seem an obvious first choice for such exploration. Although phase-encoded logic is subject to interconnect delay just as standard level-based logic is, the periodic effect of delays on phase may offer novel opportunities. Similarly, the fact that oscillators for logic need not be designed for delivering significant output power can enable novel designs that best optimize power and speed in the presence of noise and interference, leveraging phase logic's intrinsic low energy and noise resilience properties. Adapting EHF oscillators at hundreds of gigahertz for phase logic may enable computation an order of magnitude faster than possible today. It is even conceivable that optical oscillators (lasers, with fundamental frequencies of ~200 THz) can be adapted for phase logic, leading to computers that operate in the hundreds of terahetz. Another intriguing long-term

[19]However, spin-torque oscillator technology (reliable fabrication, patterning, circuit design, etc.) is currently far behind CMOS.

possibility is designing biological oscillators for phase-based computation and to embed them in cells to enable intelligent molecular-level medicine. Because its concepts can be leveraged by many technologies (especially nanoscale ones) across different physical domains, it seems likely that SSNO-based logic will make a significant impact on next-generation computing. ■

Acknowledgment

The author would like to thank T. Theis for discussions that motivated the energy and speed explorations in Section III. T. Wang designed and demonstrated ring-oscillator-based phase-logic circuits and design tools [49], [60], [68], and provided valuable insights into Q factors of nonlinear oscillators.

REFERENCES

[1] C. Huygens, "Observations of injection locking between grandfather clocks," in *Horologium Oscillatorium*. Paris, France: Apud F. Muget, 1672.

[2] M. D. Feuer *et al.*, "High-speed low-voltage ring oscillators based on selectively doped heterojunction transistors," *IEEE Electron Device Lett.*, vol. EDL-4, no. 9, pp. 306–307, Sep. 1983.

[3] M. J. Deen, M. H. Kazemeini, and S. Naseh, "Performance characteristics of an ultra-low power VCO," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2003, pp. 697–700.

[4] S. Farzeen, G. Ren, and C. Chen, "An ultra-low power ring oscillator for passive UHF RFID transponders," in *Proc. IEEE Int. Midwest Symp. Circuits Syst.*, Aug. 2010, pp. 558–561.

[5] P. Horowitz and W. Hill, *The Art of Electronics*. Cambridge, U.K.: Cambridge Univ. Press, 1995.

[6] S. Kaka *et al.*, "Mutual phase-locking of microwave spin torque nano-oscillators," *Nature*, vol. 437, pp. 389–392, Sep. 2005.

[7] Q. Mistral *et al.*, "Current-driven microwave oscillations in current perpendicular-to-plane spin-valve nanopillars," *Appl. Phys. Lett.*, vol. 88, no. 19, 2006, Art. no. 192507.

[8] D. Houssameddine *et al.*, "Spin-torque oscillator using a perpendicular polarizer and a planar free layer," *Nature Mater.*, vol. 6, pp. 447–453, 2007.

[9] T. Devolder *et al.*, "Spin transfer oscillators emitting microwave in zero applied magnetic field," *J. Appl. Phys.*, vol. 101, 2007, Art. no. 063916.

[10] D. Houssameddine *et al.*, "Spin transfer induced coherent microwave emission with large power from nanoscale MgO tunnel junctions," *Appl. Phys. Lett.*, vol. 93, 2008, Art. no. 022505.

[11] A. E. Siegman, *Lasers*. South Orange, NJ, USA: University Science Books, 1986.

[12] S. Kobayashi and T. Kimura, "Injection locking characteristics of an AlGaAs semiconductor laser," *IEEE J. Quantum Electron.*, vol. 16, no. 9, pp. 915–917, Sep. 1980.

[13] R. G. Hunsperger, "Integrated optics: Theory and technology," in *Advanced Texts in Physics*. New York, NY, USA: Springer-Verlag, 2009.

[14] C. T.-C. Nguyen, "Vibrating RF MEMS for next generation wireless applications," in *Proc. IEEE Custom Integr. Circuits Conf.*, May 2004, pp. 257–264.

[15] X. L. Feng, C. J. White, A. Hajimiri, and M. L. Roukes, "A self-sustaining ultrahigh-frequency nanoelectromechanical oscillator," *Nature Nanotechnol.*, vol. 3, no. 6, pp. 342–346, 2008.

[16] D. C. Sigg, P. A. Iaizzo, Y. F. Xiao, and B. He, *Cardiac Electrophysiology Methods and Models*. Berlin, Germany: Springer-Verlag, 2010.

[17] M. B. Elowitz and S. Leibler, "A synthetic oscillatory network of transcriptional regulators," *Nature*, vol. 403, no. 6767, pp. 335–338, Jan. 2000.

[18] M. Farkas, *Periodic Motions*. New York, NY, USA: Springer-Verlag, 1994.

[19] S. Strogatz, *Sync: The Emerging Science of Spontaneous Order*. New York, NY, USA: Theia, Mar. 2003.

[20] S. H. Strogatz, "From Kuromoto to Crawford: Exploring the onset of synchronization in populations of coupled oscillators," *Physica D, Nonlinear Phenomena*, vol. 143, no. 1–4, pp. 1–20, 2000.

[21] S. H. Strogatz and I. Stewart, "Coupled oscillators and biological synchronization," *Sci. Amer.*, vol. 269, no. 6, pp. 102–109, Dec. 1993.

[22] B. P. Belousov, "A periodic reaction and its mechanism," in *Oscillations and Travelling Waves in Chemical Systems*. New York, NY, USA: Wiley, 1985.

[23] A. N. Zaikin and A. M. Zhabotinsky, "Concentration wave propagation in two-dimensional liquid-phase self-oscillating system," *Nature*, vol. 225, pp. 535–537, 1970.

[24] X. Lai and J. Roychowdhury, "Fast simulation of large networks of nanotechnological and biochemical oscillators for investigating self-organization phenomena," in *Proc. IEEE Asia South Pacific Conf. Design Autom.*, Jan. 2006, pp. 273–278.

[25] P. Bhansali, S. Srivastava, X. Lai, and J. Roychowdhury, "Comprehensive procedure for fast and accurate coupled oscillator network simulation," in *Proc. Int. Conf. Comput.-Aided Design*, Nov. 2008, pp. 815–820.

[26] J. Buck and E. Buck, "Synchronous fireflies," *Sci. Am.*, vol. 234, no. 5, pp. 74–79, 82–85, May 1976.

[27] S. Bernard, D. Gonze, B. Cajavec, H. Herzel, and A. Kramer, "Synchronization-induced rhythmicity of circadian oscillators in the suprachiasmatic nucleus," *PLoS Comput. Biol.*, vol. 3, no. 4, Apr. 2007, DOI: 10.1371/journal.pcbi.0030068.

[28] J. C. Leloup and A. Goldbeter, "Toward a detailed computational model for the mammalian circadian clock," *Proc. Nat. Acad. Sci.*, vol. 100, no. 12, pp. 7051–7056, Jun. 2003.

[29] M. A. Kramer, H. E. Kirsch, and A. J. Szeri, "Pathological pattern formation and cortical propagation of epileptic seizures," *J. Roy. Soc. Lond.*, vol. 2, pp. 113–127, 2005.

[30] T. Yang, R. A. Kiehl, and L. O. Chua, "Image processing in tunneling phase logic cellular nonlinear networks," *Chaos in Circuits and Systems*. Singapore: World Scientific, 2002, pp. 577–591.

[31] T. Nishikawa, Y.-C. Lai, and F. C. Hoppensteadt, "Capacity of oscillatory associative-memory networks with error-free retrieval," *Phys. Rev. Lett.*, vol. 92, no. 10, pp. 273–279, Mar. 2004.

[32] T. Roska *et al.*, "An associative memory with oscillatory CNN arrays using spin torque oscillator cells and spin-wave interactions architecture and end-to-end simulator," in *Proc. 13th Int. Workshop Cellular Nanoscale Netw. Appl.*, Aug. 2012, DOI: 10.1109/CNNA.2012.6331463.

[33] E. Goto, "New Parametron circuit element using nonlinear reactance," *KDD Kenyku Shiryo*, Oct. 1954.

[34] E. Goto, "On the application of parametrically excited nonlinear resonators," *J. Elec. Commun. Engrs. Japan*, vol. 38, pp. 770–775, 1955.

[35] J. von Neumann, "Non-linear capacitance or inductance switching, amplifying and memory devices," U.S. Patent 2,815,488, Dec. 3, 1957.

[36] R. L. Wigington, "A new concept in computing," *Proc. Inst. Radio Eng.*, vol. 47, pp. 516–523, Apr. 1959.

[37] S. Oshima, "Introduction to Parametron," *Denshi Kogyo*, vol. 4, no. 11, p. 4, Dec. 1955.

[38] S. Muroga, "Elementary principle of Parametron and its application to digital computers," *Datamation*, vol. 4, no. 5, pp. 31–34, Sep./Oct. 1958.

[39] Parametron. [Online]. Available: http://www.thocp.net/hardware/parametron.htm

[40] WikipediaParametron. [Online]. Available: http://en.wikipedia.org/wiki/Parametron

[41] E. Goto, *Dc Flux Parametron: A New Approach to Josephson Junction Logic*. Singapore: World Scientific, 1986.

[42] W. Hoe and E. Goto, *Quantum Flux Parametron: A Single Quantum Flux Superconducting Logic Device*. Singapore: World Scientific, 1991.

[43] T. N. Theis and P. M. Solomon, "In quest of the next switch: Prospects for greatly reduced power dissipation in a successor to the silicon field-effect transistor," *Proc. IEEE*, vol. 98, no. 12, pp. 2005–2014, Dec. 2010.

[44] N. Shanbhag *et al.*, "The search for alternative computational paradigms," *IEEE Design Test Comput.*, vol. 25, no. 4, pp. 334–343, Jul./Aug. 2008.

[45] A. Neogy and J. Roychowdhury, "Analysis and design of sub-harmonically injection locked oscillators," in *Proc. IEEE Design Autom. Test Eur. Conf.*, Mar. 2012, pp. 1209–1214.

[46] R. A. DeCarlo and P.-M. Lin, *Linear Circuit Analysis: Time Domain, Phasor, Laplace Transform Approaches*. Upper Saddle River, NJ, USA: Prentice-Hall, 1995.

[47] D. Ozis, N. M. Neihart, and D. J. Allstot, "Differential VCO and passive frequency doubler in 0.18/spl mu/m CMOS for 24 GHz applications," in *Proc. IEEE Radio Frequency Integr. Circuits Symp.*, Jun. 2006, DOI: 10.1109/RFIC.2006.1651084.

[48] B. Razavi, K. F. Lee, and R.-H. Yan, "A 13.4-GHz CMOS frequency divider," in *Dig. Tech. Papers IEEE Int. Solid-State Circuits Conf.*, Feb. 1994, pp. 176–177.

[49] T. Wang and J. Roychowdhury, "PHLOGON: PHase-based LOGic using oscillatory nanosystems," *Unconventional Computation and Natural Computation*, vol. 8553. Berlin, Germany: Springer-Verlag, 2014, pp. 353–366.

[50] A. P. Malvino, *Digital Computer Electronics*. Noida, India: Tata McGraw-Hill, 1995.

[51] W. Wernick, "Complete sets of logical functions," *Trans. Amer. Math. Soc.*, vol. 51, no. 1, pp. 117–132, 1942.

[52] R. Adler, "A study of locking phenomena in oscillators," *Proc. IRE*, vol. 34, pp. 351–357, Jun. 1946.

[53] X. Lai and J. Roychowdhury, "Capturing injection locking via nonlinear phase domain macromodels," *IEEE Trans. Microw. Theory Tech.*, vol. 52, no. 9, pp. 2251–2261, Sep. 2004.

[54] P. Bhansali and J. Roychowdhury, "Gen-Adler: The generalized Adler's equation for injection locking analysis in oscillators," in *Proc. IEEE Asia South Pacific Design Autom. Conf.*, Jan. 2009, pp. 522–227.

[55] J. von Neumann, "First Draft of a Report on the EDVAC," Univ. Pennsylvania, Tech Rep., 1945, Report prepared for U.S. Army Ordinance Department under Contract W-670-ORD-4926.

[56] E. A. Vittoz, M. G. R. Degrauwe, and S. Bitz, "High-performance crystal oscillator circuits: Theory and application," *IEEE J. Solid-State Circuits*, vol. 23, no. 3, pp. 774–783, Jun. 1988.

[57] C. T.-C. Nguyen and R. T. Howe, "An integrated CMOS micromechanical resonator high-Q oscillator," *IEEE J. Solid-State Circuits*, vol. 34, no. 4, pp. 440–455, Apr. 1999.

[58] B. Razavi, "A 300-GHz fundamental oscillator in 65-nm CMOS technology," *IEEE J. Solid-State Circuits*, vol. 46, no. 4, pp. 894–903, Apr. 2011.

[59] Y. Wan, X. Lai, and J. Roychowdhury, "Understanding injection locking in negative-resistance LC oscillators intuitively using nonlinear feedback analysis," in *Proc. IEEE Custom Integr. Circuits Conf.*, Sep. 18–21, 2005, pp. 729–732.

[60] T. Wang, K. Aadithya, B. Wu, J. Yao, and J. Roychowdhury, "MAPP: The Berkeley model and algorithm prototyping platform," in *Proc. IEEE Custom Integr. Circuits Conf.*, Sep. 28–30, 2015.

[61] MAPP: The Berkeley Model and Algorithm Prototyping Platform. [Online]. Available: http://mapp.eecs.berkeley.eduhttp://mapp.eecs.berkeley.edu

[62] A. Ushida and L. O. Chua, "Frequency-domain analysis of nonlinear circuits driven by multi-tone signals," *IEEE Trans. Circuits Syst.*, vol. CAS-31, no. 9, pp. 766–778, Sep. 1984.

[63] K. S. Kundert, J. K. White, and A. Sangiovanni-Vincentelli, *Steady-State Methods for Simulating Analog and Microwave Circuits*. Norwell, MA, USA: Kluwer, 1990.

[64] A. Agarwal and J. Lang, *Foundations of Analog and Digital Electronic Circuits*. Amsterdam, The Netherlands: Elsevier Science, 2005.

[65] D. Middleton, *An Introduction to Statistical Communication Theory*. New York, NY, USA: Wiley/IEEE Press, 1996.

[66] A. Slavin, "Microwave sources: Spin-torque oscillators get in phase," *Nature Nanotechnol.*, vol. 4, no. 8, pp. 479–480, Aug. 2009.

[67] Z. Zeng *et al.*, "Ultralow-current-density and bias-field-free spin-transfer nano-oscillator," *Sci. Rep.*, vol. 3, 2013, DOI:10.1038/srep01426.

[68] T. Wang and J. Roychowdhury, "Design tools for oscillator-based computing systems," in *Proc. IEEE Design Autom. Conf.*, 2015, pp. 188:1–188:6.

[69] A. H. Taub, Ed., John von Neumann: *Collected Works. Volume V: Design of Computers, Theory of Automata and Numerical Analysis*. New York, NY, USA: Pergamon Press, 1963.

[70] R. Adler, "A study of locking phenomena in oscillators," *Proc. IEEE*, vol. 61, no. 10, pp. 1380–1385, Oct. 1973.

[71] N. B. Stern, *From ENIAC to UNIVAC: An Appraisal of the Eckert-Mauchly Computers*. Bedford, MA, USA: Digital Press, 1981.

[72] B. Randell, Ed., *The Origins of Digital Computers: Selected Papers*, 3rd ed. Berlin, Germany: Springer-Verlag, 1982.

[73] M. D. Godfrey, Ed., "J. von Neumann: First draft of a report on the EDVAC," *IEEE Ann. History Comput.*, vol. 15, no. 4, pp. 28–75, Oct.–Dec. 1993.

[74] P. Laplante, Ed., *Great Papers in Computer Science*. Silver Spring, MD, USA: IEEE Computer Society Press, 1996.

[75] A. H. Taub, Ed., John von Neumann: *Collected Works: Volume I: Logic, Theory of Sets and Quantum Mechanics*. New York, NY, USA: Pergamon Press, 1961.

[76] A. H. Taub, Ed., John von Neumann: *Collected Works. Volume II: Operators, Ergodic Theory and Almost Periodic Functions in a Group*. New York, NY, USA: Pergamon Press, 1961.

[77] A. H. Taub, Ed., John von Neumann: *Collected Works. Volume III: Rings of Operators*. New York, NY, USA: Pergamon Press, 1961–1963.

[78] A. H. Taub, Ed., John von Neumann: *Collected Works. Volume IV: Continuous Geometry and Other Topics*. New York, NY, USA: Pergamon Press, 1962.

[79] A. H. Taub, Ed., John von Neumann: *Collected Works. Volume VI: Theory of Games, Astrophysics, Hydrodynamics and Meteorology*. New York, NY, USA: Pergamon Press, 1963.

ABOUT THE AUTHOR

Jaijeet Roychowdhury (Fellow, IEEE) is a Professor in the Electrical Engineering and Computer Science Department, University of California at Berkeley, Berkeley, CA, USA. He cofounded Berkeley Design Automation (now a part of Mentor Graphics), where he currently serves as a technical advisor. His current research interests encompass novel computational architectures and paradigms, analog and mixed-signal verification, multidomain device modeling, and open-source infrastructures for reproducible research. Prior to joining academia, he was with the Research Division of Bell Laboratories, where his work on MOS homotopies was cited as an Extraordinary Achievement. Over the years, he has authored or coauthored seven best or distinguished papers in the area of analog/RF design automation.

Dr. Roychowdhury has served as an IEEE Circuits and Systems Society Distinguished Lecturer, as Program Chair of IEEE's CANDE and BMAS workshops, on the Executive Committee of ICCAD and on the Nominations and Appointments Committee of CEDA.