Boolean Computation Using Self-Sustaining Nonlinear Oscillators

Jaijeet Roychowdhury, EECS Department, University of California, Berkeley

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 traditional level-based logic.

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], 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 first review recent work that establishes that SSNOs can also serve as substrates for *general-purpose Boolean computation*. By exploiting a phenomenon known as sub-harmonic 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 CMOS ring oscillators, spin-torque nano-oscillators, synthetic biological oscillators, MEMS/NEMS-based oscillators, nanolasers and even mechanical clocks. 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 Eiichi Goto [33, 34] and John 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

---

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

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

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 re-examining 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 the paper is organized as follows. In §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 §III, energy consumption and speed in SSNO-based phase logic are explored. The superior noise immunity properties of phase encoded logic are outlined in §IV.

II. PHASE LOGIC LATCHES USING SSNOS

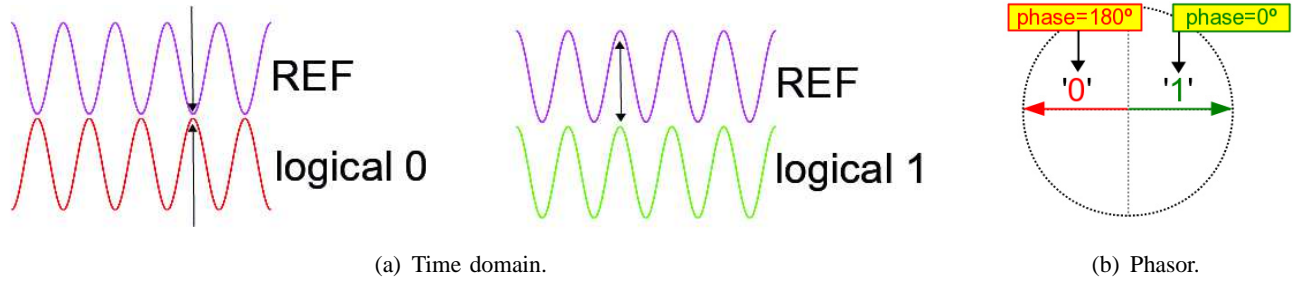
(a) Time domain.

(b) Phasor.

Fig. 1: Encoding Boolean logic using the relative phase of oscillatory signals.

Fig. 1 above 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.
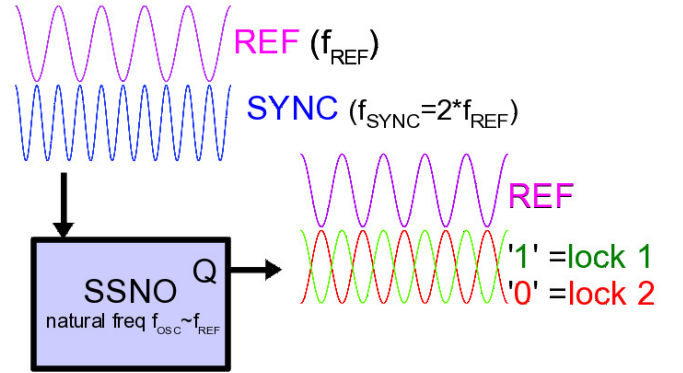

Fig. 2: SSNO serving as a bi-stable phase latch.

A. SHIL makes SSNOs phase-bistable

Fig. 2 illustrates 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

---

[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.*, can also serve as substrates for SSNO-based phase logic.

[4]Ternary and multi-state logic values can also be encoded in phase; indeed, SSNOs can serve as multi-state latches [45]. For concreteness, we focus on the binary case throughout this paper.

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

assume that a periodic REF signal with frequency f_{REF}, as shown in Fig. 1 and Fig. 2, is available. We also require another signal 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].
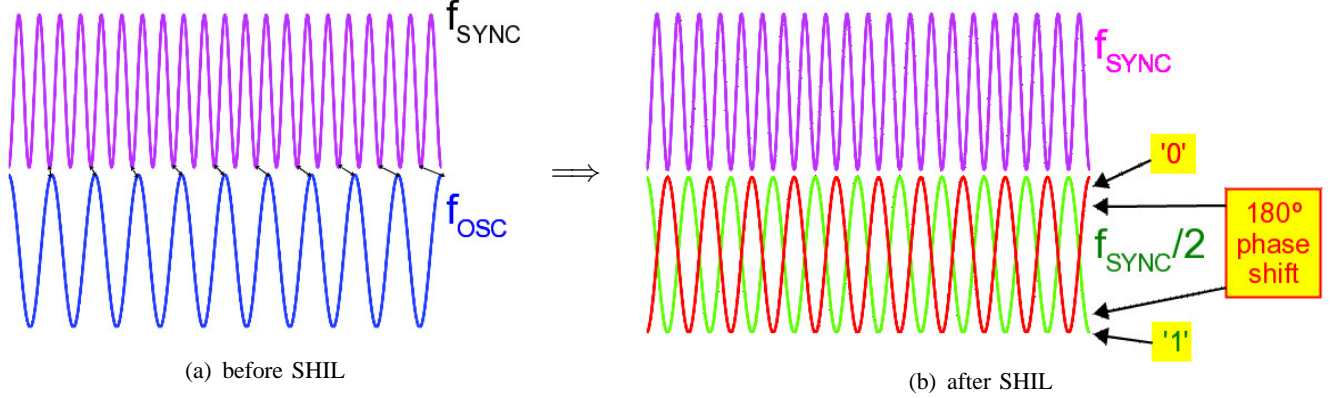

Fig. 3: Sub-harmonic injection locking in an SSNO stores phase logic states.

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 depicted 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], sub-harmonic 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 bi-stable 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 bi-stable SHIL in a CMOS ring oscillator 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. 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 bi-stable SHIL can be turned into D latches [49]. 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* [50]. For example, the Boolean function sets {AND, NOT}, {OR, NOT}, {NAND} and {NOR} are all logically complete.

---

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

[7]Which state the oscillator locks to depends on initial conditions, transients, noise, *etc.*., during circuit startup. §II-C below describes circuits and techniques for setting and manipulating the state.
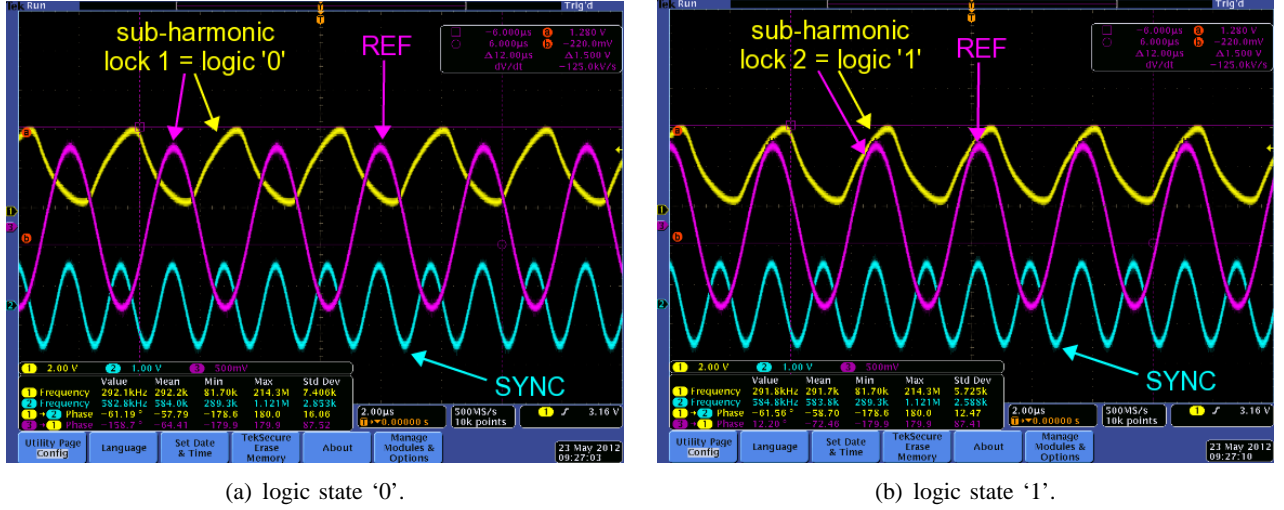
(a) logic state '0'. (b) logic state '1'.

Fig. 4: Oscilloscope traces showing bi-stable SHIL in a CMOS ring-oscillator with SYNC injection.

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 3-input majority operation, returning whichever Boolean value occurs more than once amongst its three inputs.[8] For example, MAJ(0, 0, 1) returns 0; MAJ(1, 0, 1) returns 1.
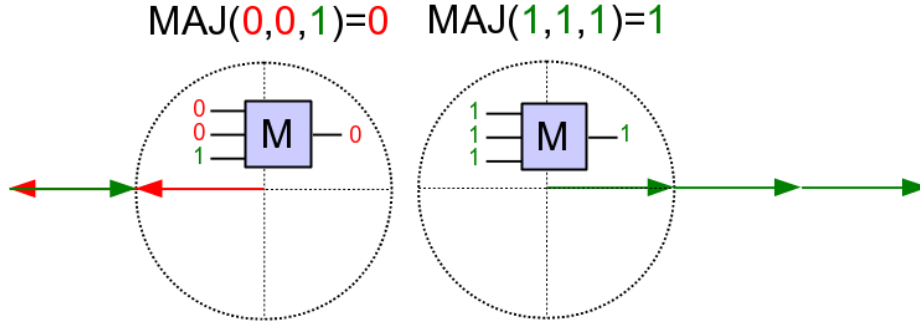

Fig. 5: Examples illustrating MAJ(A, B, C) in phase logic.

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 (*e.g.*, a standard CMOS inverter topology serves for use with CMOS ring SSNO phase latches; see §II-C below). 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.

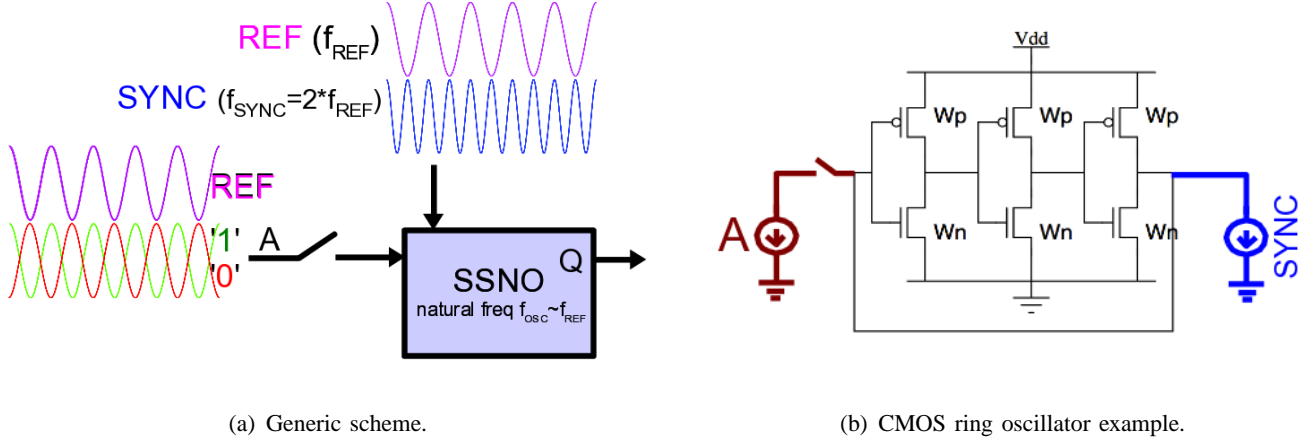(a) Generic scheme.

(b) CMOS ring oscillator example.

Fig. 6: Controlling the lock state of a SSNO under SHIL.

C. Setting and resetting SSNO SHIL logic states; phase based D-latches

To exploit SSNO bi-stability under SHIL (§II-A) for general purpose computation, it is necessary to control the SSNO's SHIL state. The basic mechanism by which this 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 [51] 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 bi-stability under SYNC injection and sets the oscillator's phase close to that of A [51, Figure 4];[9] when A is removed, bi-stability 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 labelled SYNC). The oscillator responds within about 2 cycles by changing its frequency to $f_{\text{SYNC}}/2$ 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 1 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 bi-stable 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 [51] into a gated D latch (D latch with Enable) [49] with the help of the 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 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

---

[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).

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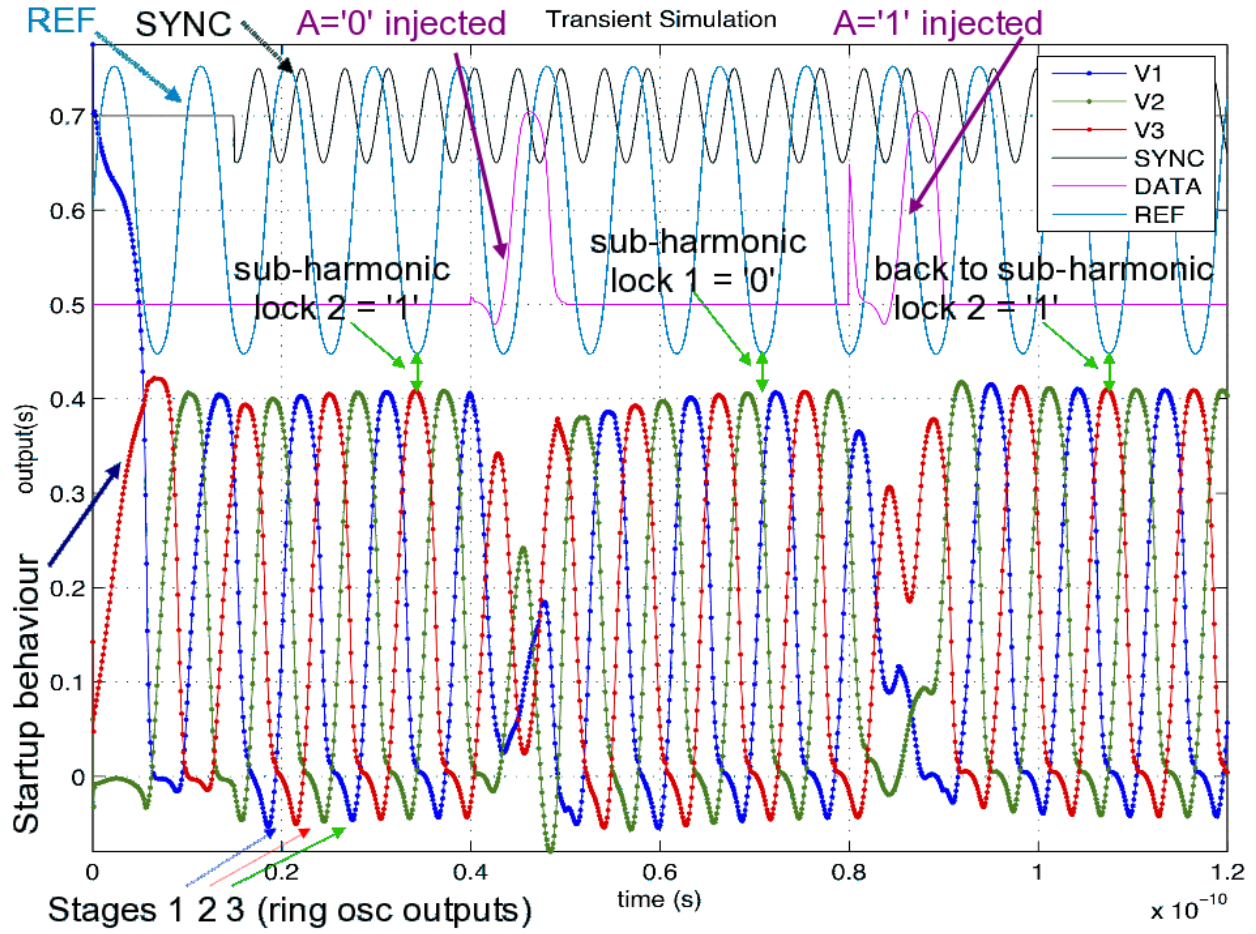
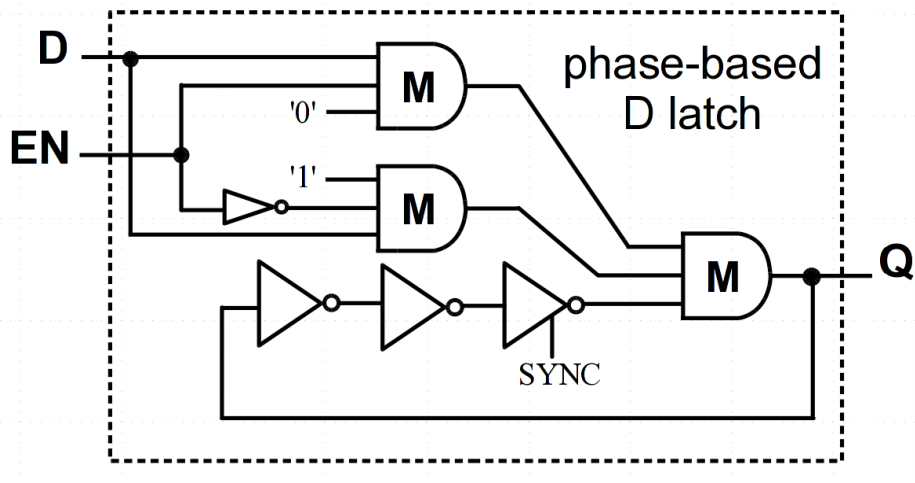**Fig. 7:** Transient simulation of the circuit in Fig. 6(b).

Fig. 8: Phase based D latch with enable using a CMOS ring oscillator [51]. The gates marked M are 3-input majority gates.

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 bi-stability and retaining the previously set state.

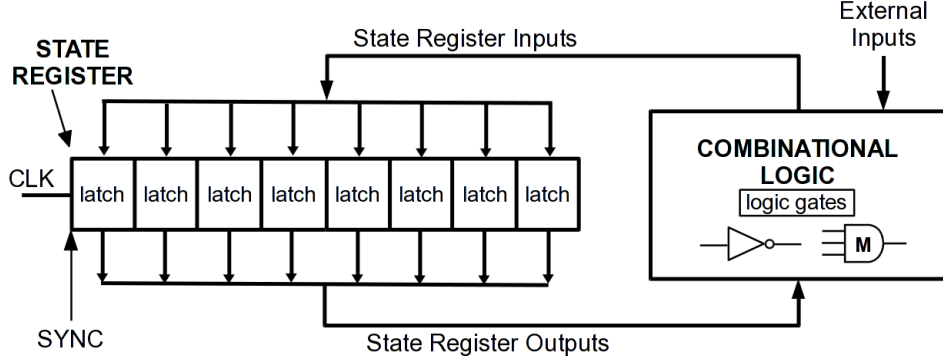### D. State machines using SSNO-based phase logic

Fig. 9: Structure of a state machine using SSNO-based phase latches and {NOT, MAJ} based combinational 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 logic. One of the most important units of a computer is the finite state machine (FSM), used for, *e.g.*, the control unit [49, 55] 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; however, a level-based clock signal will not benefit from the increased noise immunity of phase-based encoding (see §IV, below).

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 [51]. The latch is constructed using two of the D latches shown in Fig. 8, arranged in a master-slave [49] 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'→'1' state transition can only occur if **a**=**b**='1', and the '1'→'0' transition if **a**=**b**='0'. The complete state transition diagram of the FSM is shown in Fig. 10(a).
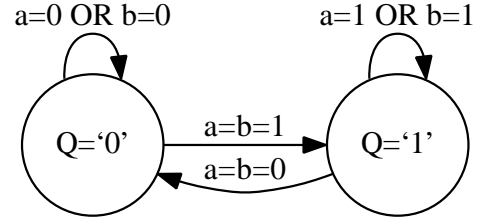
(a) State Transition Graph.
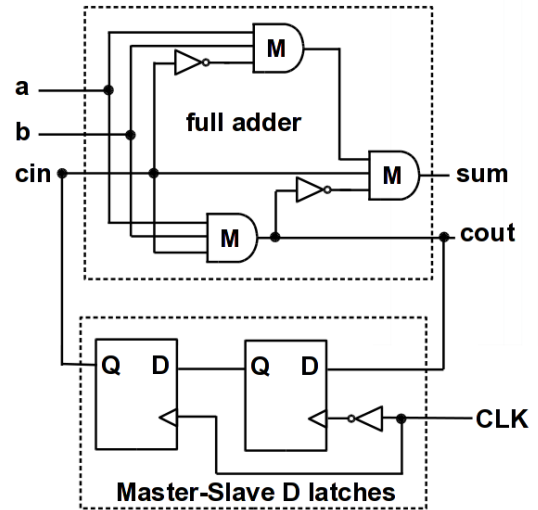
(b) Implementation with phase D latches and {NOT, MAJ} gates.

Fig. 10: SSNO-based 1-bit state machine example [51].

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 $\frac{1}{2}CV_{\mathrm{DD}}^2$ per bit flip. Just to maintain oscillation, a minimum energy of $3CV_{\mathrm{DD}}^2$ is dissipated per cycle by the 3-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 (*e.g.*, [56, 57]) are inherently energy efficient, dissipating only about $\frac{1}{Q}$ of the energy stored in the LC tank[12] 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.[13] With Q factors of 10^2-10^6 readily achievable, 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 counter-intuitive, since amplitude changes in high-Q oscillators are very slow on account of their necessarily involving 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 slow amplitude transients with time constants of the order of Q cycles of oscillation.

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

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.[14] 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
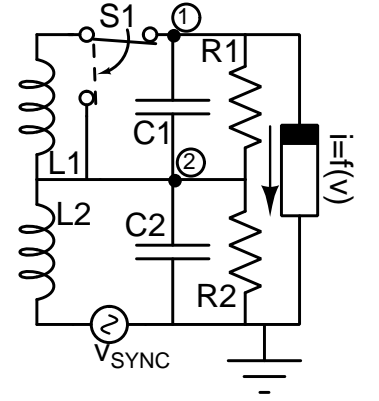
---

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

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

[12]$\frac{1}{2}CV_{\mathrm{osc}}^2$, where V_{osc} is the peak amplitude of oscillation.

[13]indeed, possibly at far higher speeds than level based logic is currently capable of, depending on the oscillator's frequency.

[14]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.

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 $f_{\text{REF}} = f_{\text{SYNC}}/2$. The single-pole double-throw (SPDT) switch S1, normally kept closed in the position shown, is used for phase logic bit flipping, as described below.

To facilitate sub-harmonic 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 $\frac{R_2}{2\pi f_{\text{SYNC}} L_2}$, 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. It has the current-voltage characteristic

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

where

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

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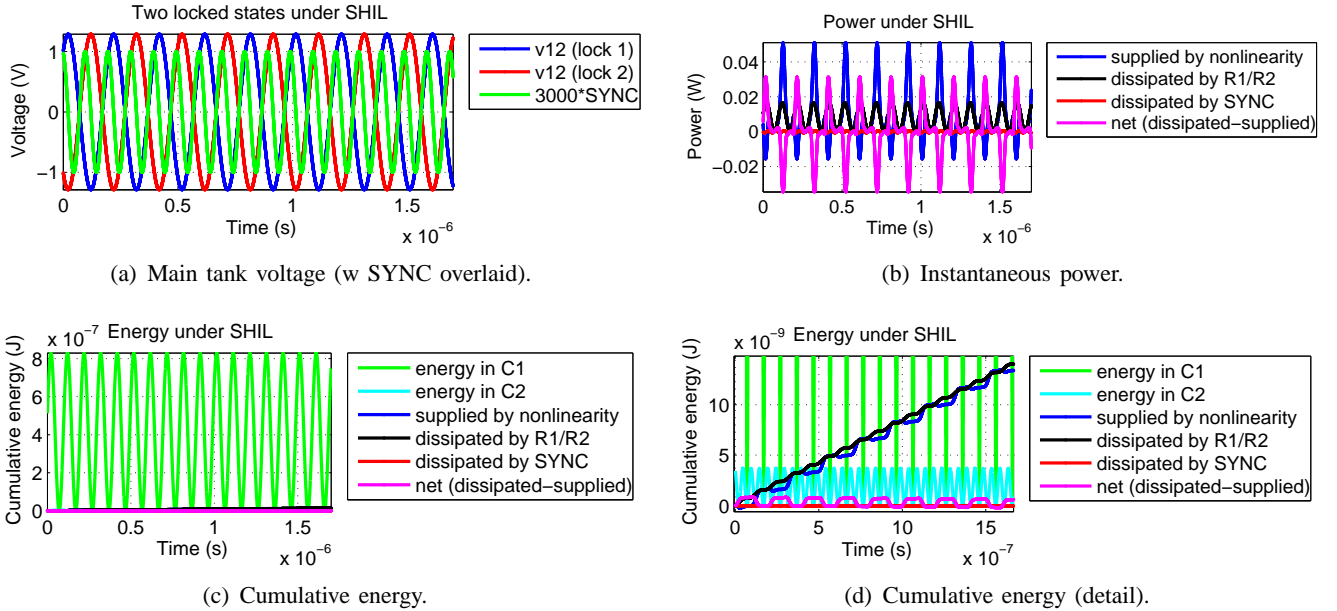

(a) Main tank voltage (w SYNC overlaid).

(b) Instantaneous power.

(c) Cumulative energy.

(d) Cumulative energy (detail).

Fig. 12: Voltages, power and energy consumption of LC oscillator under SHIL.

For natural oscillation to occur (in the absence of any injection at v_{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_{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:

$$L_1 = 1\text{nH}, C_1 = 1\mu\text{F}, R_1 = 100\Omega, L_2 = \frac{L_1}{2}, C_2 = \frac{C_2}{2},$$

$$R_2 = 90\Omega, k_1 = \frac{1}{30}, k_2 = \frac{0.0102}{k_1}, k_3 = 40k_1k_2, A = 0.9. \tag{4}$$

The switch S_1 was modelled with on resistance 0Ω and off resistance $10\text{k}\Omega$. With these parameters, $f_{\text{OSC}} \simeq \frac{1}{2\pi\sqrt{L_1C_1}} \sim 5.03292$ MHz. f_{REF} was taken to be 5.0328 MHz, with $f_{\text{SYNC}} = 2f_{\text{REF}}$. The SYNC injection was

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

Fig. 12(a) shows the voltage of the main tank of the oscillator under SHIL.[15] The two locks, representing logic levels '0' and '1', can be seen to be exactly $180°$ out of phase, as predicted by theory [45, 51].

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 resistances of the switch S_1 dissipate power, but these amounts are negligible. Fig. 12(c) and Fig. 12(d) depict 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 $0.829\mu\text{J}$.

(a) Main tank voltage overlaid on '0'/'1' waveforms.

(b) Instantaneous power.

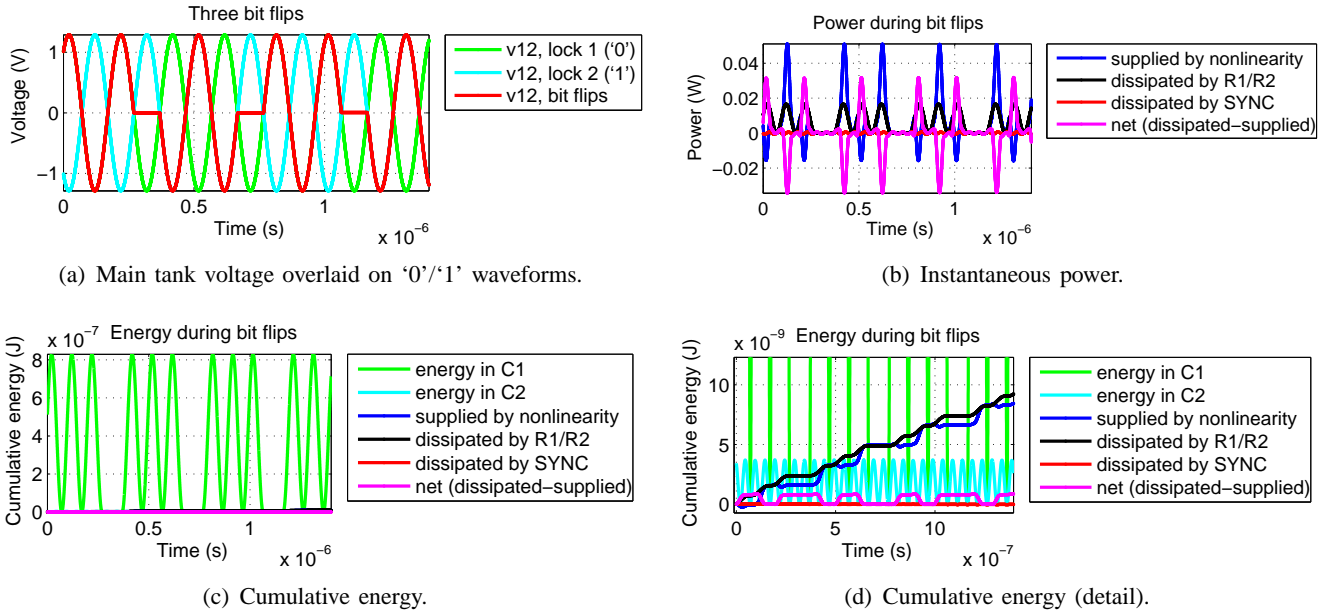(c) Cumulative energy.

(d) Cumulative energy (detail).

Fig. 13: Voltages, power and energy consumption of LC oscillator undergoing bit flips.

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 1.685nJ, implying an effective Q factor[16] of about 492 in sub-harmonically injection locked operation.

---

[15]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.

[16]Because of the nonlinear resistor, the Q of a self-sustaining oscillator is typically lower – by about $6\times$ in this case – than the ideal Q factor of the linear tank alone [64].

C. Speed and energy during bit flips

The SPDT switch S_1 in Fig. 11 can be used to transition the oscillator 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.[17] S_1 is flipped for half a cycle three times (starting around 0.27μs, 0.67μs, and 1.06μs), 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 Fig. 13(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

(a) Small noise amplitude.
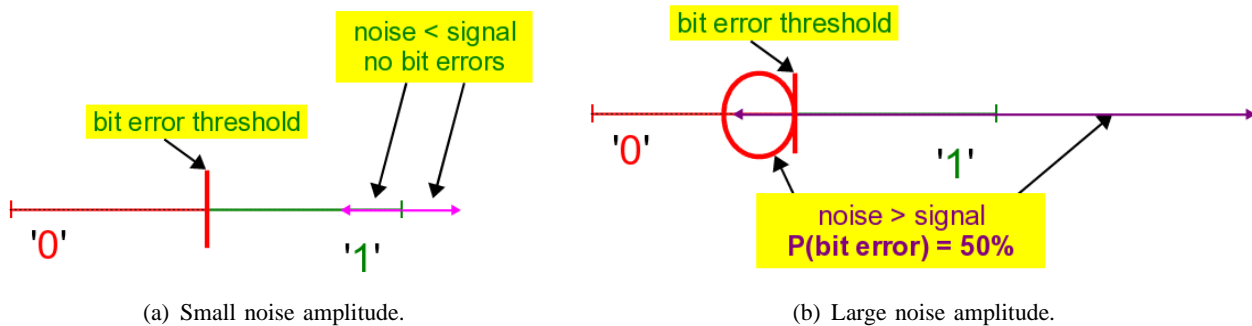
(b) Large noise amplitude.

Fig. 14: Level-based logic encoding: bit error rates for small and large noise amplitudes.

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

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. The bit error threshold in the presence of noise is zero. In Fig. 14(a), the impact of adding fixed-amplitude "small" noise (*i.e.*, the noise is less than 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 amplitude, 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 phasors, at the *same frequency*. In this case, the bit error thresholds are the vertical phasors at $\pm 90°$, *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 $[0°, 360°]$, 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.

---

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

(a) Small noise amplitude.
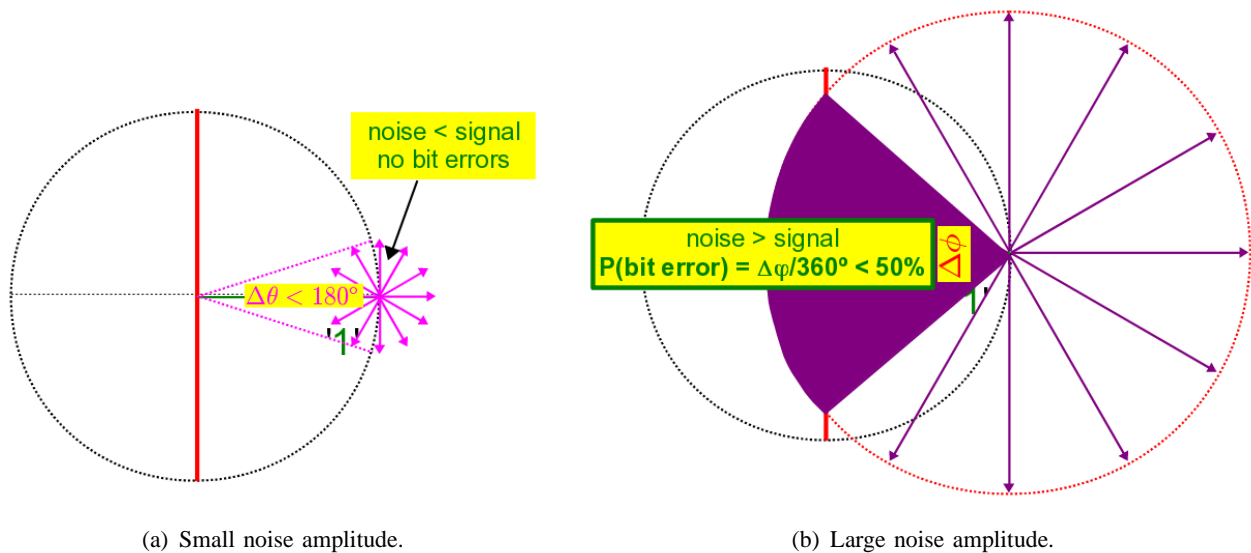
(b) Large noise amplitude.

Fig. 15: Phase-based logic encoding: bit error probabilities for small and large noise amplitudes.

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]; in particular, the above reasoning is essentially identical to that establishing the superior noise performance of BPSK (binary phase shift keying) over BASK (binary amplitude shift keying). 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 relevant as an alternative computational scheme for today's nanoscale integration era. The fact that almost any SSNO can serve as a phase logic latch implies that many new substrates for phase-based logic (such as spin-transfer nano-oscillators (STNOs) [66]) can potentially be exploited. 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, provide incentives for exploring its use.

REFERENCES

[1] Christian Huygens. *Horologium Oscillatorium*. Apud F. Muget, Paris, France, 1672. (Observations of injection locking between grandfather clocks).

[2] M. D. Feuer, R. H. Hendel, R. A. Kiehl, J. C. M Hwang, V. G. Keramidas, C. L. Allyn, and R. Dingle. High-speed low-voltage ring oscillators based on selectively doped heterojunction transistors. *IEEE Electron Device Letters*, EDL-4(9):306–307, September 1983.

[3] M. Jamal Deen, Mehdi H. Kazemeini, and Sasan Naseh. Performance characteristics of an ultra-low power VCO. In *Proc. IEEE ISCAS*, May 2003.

[4] S. Farzeen, Guoyan Ren, and Chunhong Chen. An ultra-low power ring oscillator for passive UHF RFID transponders. In *Proc. IEEE MWSCAS*, pages 558–561, Aug 2010.

[5] Paul Horowitz and Winfield Hill. *The Art of Electronics*. Cambridge University Press, Cambridge, UK, 1995.

[6] S. Kaka, M. R. Pufall, W. H. Rippard, T. J. Silva, S. E. Russek, and J. A. Katine. Mutual phase-locking of microwave spin torque nano-oscillators. *Nature*, 437:389–392, September 2005.

[7] Q. Mistral, J. V. Kim, T. Devolder, P. Crozat, C. Chappert, J. A. Katine, M. J. Carey, and K. Ito. Current-driven microwave oscillations in current perpendicular-to-plane spin-valve nanopillars. *Applied Physics Letters*, 88(19), 2006.

[8] D. Houssameddine, Ebels, B. U Delae, and et. al. Spin-torque oscillator using a perpendicular polarizer and a planar free layer. *Nature Materials*, pages 447–463, 2007.

[9] T. Devolder, A. Meftah, K. Ito, and et. al. Spin transfer oscillators emitting microwave in zero applied magnetic field. *Journal of Applied Physics*, page 063916, 2007.

[10] D. Houssameddine, S. H. Florez, J. A. Katine, and et. al. Spin transfer induced coherent microwave emission with large power from nanoscale MgO tunnel junctions. *Applied Physics Letters*, page 022505, 2008.

[11] Anthony E. Siegman. *Lasers*. University Science Books, 1986.

[12] S. Kobayashi and T. Kimura. Injection locking characteristics of an AlGaAs semiconductor laser. *IEEE J. Quantum Electron.*, 16:915–917, September 1980.

[13] R.G. Hunsperger. *Integrated Optics: Theory and Technology*. Advanced texts in physics. Springer, 2009.

[14] Clark T-C. Nguyen. Vibrating RF MEMS for Next Generation Wireless Applications. In *Proc. IEEE CICC*, May 2004.

[15] XL Feng, CJ White, A Hajimiri, and Michael L Roukes. A self-sustaining ultrahigh-frequency nanoelectromechanical oscillator. *Nature nanotechnology*, 3(6):342–346, 2008.

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

[17] M. B. Elowitz and S. Leibler. A synthetic oscillatory network of transcriptional regulators. *Nature*, 403(6767):335–338, Jan 2000.

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

[19] Steven Strogatz. *Sync: The Emerging Science of Spontaneous Order*. Theia, March 2003.

[20] Steven H. Strogatz. From Kuromoto to Crawford: exploring the onset of synchronization in populations of coupled oscillators. *Physica D*, 2000.

[21] S. H. Strogatz and I. Stewart. Coupled oscillators and biological synchronization. *Scientific American*, 269(6):102–109, Dec 1993.

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

[23] A. N. Zaikin and A. M. Zhabotinsky. Concentration wave propagation in two-dimensional liquid-phase self-oscillating system. *Nature*, 225:535–537, 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 ASP-DAC*, pages 273–278, January 2006.

[25] P. Bhansali, S. Srivastava, X. Lai and J. Roychowdhury. Comprehensive Procedure for Fast and Accurate Coupled Oscillator Network Simulation. In *Proc. ICCAD*, pages 815–820, November 2008.

[26] J. Buck and E. Buck. Synchronous fireflies. *Scientific American*, 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.*, 3(4), April 2007.

[28] J.C. Leloup and A. Goldbeter. Toward a detailed computational model for the mammalian circadian clock. *PNAS*, pages 7051–7056, June 2003.

[29] M. A. Kramer, H. E. Kirsch and A. J. Szeri. Pathological pattern formation and cortical propagation

of epileptic seizures. *Journal of the Royal Society of London, Interface 2*, 2:113–127, 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*, pages 577–591, 2002.

[31] T. Nishikawa, Y-C. Lai, and F.C. Hoppensteadt. Capacity of oscillatory associative-memory networks with error-free retrieval. *Physical Review Letters*, 92(10):273–279, March 2004.

[32] T. Roska, A. Horvath, A. Stubendek, F. Corinto, G. Csaba, W. Porod, T. Shibata, and G. Bourianoff. An associative memory with oscillatory cnn arrays using spin torque oscillator cells and spin-wave interactions architecture and end-to-end simulator. In *Cellular Nanoscale Networks and Their Applications (CNNA), 2012 13th International Workshop on*, pages 1–3, Aug 2012.

[33] Eiichi Goto. New Parametron circuit element using nonlinear reactance. *KDD Kenyku Shiryo*, October 1954.

[34] Eiichi Goto. On the application of parametrically excited nonlinear resonators. *Denki Tsushin Gakkai-shi*, October 1955.

[35] John von Neumann. Non-linear capacitance or inductance switching, amplifying and memory devices. Basic paper for Patent 2,815,488, filed April 28, granted December 3, 1957, and assigned to IBM. Reprinted in [68, Paper 11, pp. 379–419], 1954.

[36] R. L. Wigington. A New Concept in Computing. *Proceedings of the Institute of Radio Engineers*, 47:516–523, April 1959.

[37] S. Oshima. Introduction to Parametron. *Denshi Kogyo*, 4(11):4, December 1955.

[38] S. Muroga. Elementary principle of Parametron and its application to digital computers. *Datamation*, 4(5):31–34, September/October 1958.

[39] Parametron (web page). http://www.thocp.net/hardware/parametron.htm.

[40] Parametron (Wikipedia web page). http://en.wikipedia.org/wiki/Parametron.

[41] Eiichi Goto. *Dc Flux Parametron: A New Approach to Josephson Junction Logic*, volume 6 of *World Scientific Series in Computer Science*. World Scientific, 1986.

[42] Willy Hoe and Eiichi Goto. *Quantum Flux Parametron: A Single Quantum Flux Superconducting Logic Device*, volume 2 of *Studies in Josephson Supercomputers*. 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. *Proceedings of the IEEE*, 98(12):2005–2014, Dec 2010.

[44] N. Shanbhag, S. Mitra, G. de Veciana, M. Orshansky, R. Marculescu, J. Roychowdhury, D. Jones and J. Rabaey. The Search for Alternative Computational Paradigms. *IEEE Design and Test of Computers*, 25(4):334–343, July-Aug 2008.

[45] A. Neogy and J. Roychowdhury. Analysis and Design of Sub-harmonically Injection Locked Oscillators. In *Proc. IEEE DATE*, Mar 2012.

[46] Raymond A. DeCarlo and Pen-Min Lin. *Linear Circuit Analysis: Time Domain, Phasor, and Laplace Transform Approaches*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 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 *Radio Frequency Integrated Circuits (RFIC) Symposium, 2006 IEEE*, pages 4 pp.–, June 2006.

[48] B. Razavi, K.F. Lee and Ran-Hong Yan. A 13.4-GHz CMOS frequency divider. In *Solid-State Circuits Conference, 1994. Digest of Technical Papers. 41st ISSCC., 1994 IEEE International*, pages 176–177, Feb 1994.

[49] Albert Paul Malvino. *Digital Computer Electronics*. Tata McGraw-Hill, 1995.

[50] William Wernick. Complete sets of logical functions. *Transactions of the American Mathematical Society*, 51(1):pp. 117–132, 1942.

[51] T. Wang and J. Roychowdhury. PHLOGON: PHase-based LOGic using Oscillatory Nanosystems. In *Proc. Unconventional Computation and Natural Computation: 13th International Conference, UCNC 2014, London, ON, Canada, July 14-18, 2014*, LNCS sublibrary: Theoretical computer science and general issues. Springer, 2014.

[52] R. Adler. A study of locking phenomena in oscillators. *Proceedings of the I.R.E. and Waves and Electrons*, 34:351–357, June 1946. Reprinted as [69].

[53] X. Lai and J. Roychowdhury. Capturing injection locking via nonlinear phase domain macromodels. *IEEE Trans. Microwave Theory Tech.*, 52(9):2251–2261, September 2004.

[54] P. Bhansali and J. Roychowdhury. Gen-Adler: The generalized Adler's equation for injection locking analysis in oscillators. In *Proc. IEEE ASP-DAC*, pages 522–227, January 2009.

[55] John von Neumann. First draft of a report on the EDVAC. Technical report, University of Pennsylvania, 1945. Report prepared for U.S. Army Ordinance Department under Contract W-670-ORD-4926. Reprinted in [70, pp. 177–246], [71, pp. 399–413], [72], and [73].

[56] E.A. Vittoz, M.G.R. Degrauwe, and S. Bitz. High-performance crystal oscillator circuits: theory and application. *IEEE J. Solid-State Ckts.*, 23(3):774–783, June 1988.

[57] Clark T-C Nguyen and Roger T Howe. An integrated CMOS micromechanical resonator high-Q oscillator. *IEEE J. Solid-State Ckts.*, 34(4):440–455, 1999.

[58] Behzad Razavi. A 300-GHz fundamental oscillator in 65-nm CMOS technology. *IEEE J. Solid-State Ckts.*, 46(4):894–903, 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 CICC*, pages 729–732, 18-21 Sept. 2005.

[60] T. Wang, K. Aadithya, B. Wu, J. Yao, and J. Roychowdhury. MAPP: The Berkeley Model and Algorithm Prototyping Platform. In *Proc. IEEE CICC*, 28-30 Sept. 2015.

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

[62] A. Ushida and L.O. Chua. Frequency-domain analysis of nonlinear circuits driven by multi-tone signals. *IEEE Transactions on Circuits and Systems*, CAS-31(9):766–778, Sep 1984.

[63] K.S. Kundert, J.K. White, and A. Sangiovanni-Vincentelli. *Steady-state methods for simulating analog and microwave circuits*. Kluwer Academic Publishers, 1990.

[64] A. Agarwal and J. Lang. *Foundations of Analog and Digital Electronic Circuits*. The Morgan Kaufmann Series in Computer Architecture and Design. Elsevier Science, 2005.

[65] David Middleton. *An Introduction to Statistical Communication Theory*. Wiley-IEEE, New York, 1996.

[66] Zhongming Zeng, Giovanni Finocchio, Baoshun Zhang, Pedram Khalili Amiri, Jordan A Katine, Ilya N Krivorotov, Yiming Huai, Juergen Langer, Bruno Azzerboni, Kang L Wang, et al. Ultralow-current-density and bias-field-free spin-transfer nano-oscillator. *Scientific reports*, 3, 2013.

[67] T. Wang and J. Roychowdhury. Design Tools for Oscillator-based Computing Systems. In *Proc. IEEE DAC*, pages 188:1–188:6, 2015.

[68] A. H. Taub, editor. *John von Neumann: Collected Works. Volume V: Design of Computers, Theory of Automata and Numerical Analysis*. Pergamon Press, New York, NY, USA, 1963. See also volumes I–IV, VI [74–78].

[69] R. Adler. A study of locking phenomena in oscillators. *Proc. IEEE*, 61:1380–1385, 1973. Reprinted from [52].

[70] Nancy B. Stern. *From ENIAC to UNIVAC: an appraisal of the Eckert–Mauchly computers*. Digital Press history of computing series. Digital Press, 12 Crosby Drive, Bedford, MA 01730, USA, 1981.

[71] Brian Randell, editor. *The Origins of Digital Computers: Selected Papers*. Texts and monographs in computer science. Springer Verlag, Berlin, Germany / Heidelberg, Germany / London, UK / etc., third edition, 1982.

[72] John von Neumann. First draft of a report on the EDVAC. *IEEE Annals of the History of Computing*, 15(4):28–75, October/December 1993. Edited and corrected by Michael D. Godfrey.

[73] Phillip Laplante, editor. *Great papers in computer science*. IEEE Computer Society Press, 1109 Spring Street, Suite 300, Silver Spring, MD 20910, USA, 1996.

[74] A. H. Taub, editor. *John von Neumann: Collected Works: Volume I: Logic, Theory of Sets and Quantum Mechanics*. Pergamon Press, New York, NY, USA, 1961. See also volumes II–VI [68, 75–78].

[75] A. H. Taub, editor. *John von Neumann: Collected Works. Volume II: Operators, Ergodic Theory and Almost Periodic Functions in a Group*. Pergamon Press, New York, NY, USA, 1961. See also volumes I, III–VI [68, 74, 76–78].

[76] A. H. Taub, editor. *John von Neumann: Collected Works. Volume III: Rings of Operators*. Pergamon Press, New York, NY, USA, 1961–1963. See also volumes I–II, IV–VI [68, 74, 75, 77, 78].

[77] A. H. Taub, editor. *John von Neumann: Collected Works. Volume IV: Continuous Geometry and Other Topics*. Pergamon Press, New York, NY, USA, 1962. See also volumes I–III, V–VI [68, 74–76, 78].

[78] A. H. Taub, editor. *John von Neumann: Collected Works. Volume VI: Theory of Games, Astrophysics, Hydrodynamics and Meteorology*. Pergamon Press, New York, NY, USA, 1963. See also volumes I–V [68, 74–77].