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 possible. Such phase-encoded logic 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 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 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. We also outline how phase encoding has 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 an oscillatory signal (relative to another oscillatory signal, the reference) was used to encode Boolean logic states, combinational operations could 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. 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 computers that were the mainstay of 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 such as inductors 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 barriers today, and with noise and interference from densely packed circuitry an important concern, there is an ongoing search for alternative computational paradigms \cite{43, 44}. In this context, the facts that phase-encoded logic better resists noise than level-encoded logic and that any SSNO can be used as a phase-based latch\cite{3} provide considerable motivation for re-examining phase-based computing as a candidate technology for the post-CMOS era.

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. The superior noise immunity properties of phase encoded logic are outlined in §III.

II. PHASE LOGIC LATCHES USING SSNO

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\footnote{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 \cite{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 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\footnote{We 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 \cite{47}, or REF from SYNC by frequency division \cite{48}.}

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

\footnote{\cite{48}} hereby removing an important limitation that prior phase-based logic schemes have faced; the many types of nanoscale SSNOs that are currently available become potential candidates for phase latches. In this paper, we use CMOS ring 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.

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

\footnote{A mathematical proof of this fact for a generic SSNO is available in \cite{45}.}
The SSNO being used as a phase logic latch needs to have a natural frequency close to that of REF, i.e., $f_{OSC} \approx f_{REF}$, or $f_{OSC} \approx f_{SYNC}/2$. 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 $180^\circ$ 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. Note that 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). Note also that the frequency of the oscillator’s output 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.

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

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

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

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.

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.\(^7\) For example, MAJ(0, 0, 1) returns 0; MAJ(1, 0, 1) returns 1.

\[
\text{MAJ}(0,0,1) = 0 \\
\text{MAJ}(1,1,1) = 1
\]

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.

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 the oscillator locks to depends randomly on transients during circuit startup. However, this can be controlled (see II-C below).

\(^7\)Which state the oscillator locks to depends randomly on transients during circuit startup. However, this can be controlled (see II-C below).

\(^8\)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).
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 \[51, \text{Figure 4}\] 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 easily be 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 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.

\[9\] 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.
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 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 §III 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).

III. NOISE IMMUNITY OF PHASE-ENCODED LOGIC

Phase-encoded logic is an intriguing alternative for computation in today’s nanoscale integration era because encoding in phase offers intrinsic noise immunity advantages over level-based logic. The underlying reason for this noise immunity is easy to appreciate graphically.
Fig. 11: Level-based logic encoding: bit error rates for small and large noise amplitudes.

Fig. 12: Phase-based logic encoding: bit error probabilities for small and large noise amplitudes.

The situation when logic is encoded in phase is depicted in Fig. 12. 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 ±90°, i.e., the phase halfway between the ‘0’ and ‘1’ states. Fig. 12(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 Δθ, 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.

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. 12(b). The shaded region depicts the range of noise phases (Δφ) that lead to a bit error. Importantly, Δφ 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 [56]; 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.

IV. CONCLUSION

Two 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) [57], though even standard CMOS easily serves) can potentially be exploited. The realization that phase-encoded logic brings noise immunity benefits provides an incentive for exploring its use, especially in noisy nanoscale or radiation-prone environments.

Acknowledgments

We gratefully acknowledge support from the United States National Science Foundation (Grant CCF-1111733).

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. Herzl 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 Kenkyu 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 [58, 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] Eiichi Goto. *Dc Flux Parametron: A New Approach to Josephson Junction Logic*, volume 6 of *World Scientific Series in Computer Science*. World Scientific, 1986.

[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](http://www.thocp.net/hardware/parametron.htm).

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

[41] Eiichi Goto. *Quantum Flux Parametron: A Single Quantum Flux Superconducting Logic Device*, volume 2 of *Studies in Josephson Supercomputers*. World Scientific, 1991.

[42] T.N. Theis and P.M. Solomon. In Quest of the Next Switch: Prospects for Greatly Reduced
[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 [59].

[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 [60, pp. 177–246], [61, pp. 399–413], [62], and [63].

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

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

[58] 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 [64–68].

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

[60] 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.

[61] 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.

[62] 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.

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

[64] 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 [58, 65–68].

[65] 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 [58, 64–66–68].
[66] 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 [58, 64, 65, 67, 68].

[67] 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 [58, 64, 66, 68].

[68] 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 [58, 64, 67].