

Delay-Insensitive Ternary Logic

Ravi Sankar Parameswaran Nair and Scott C. Smith

Asynchronous Digital Design Laboratory
Department of Electrical Engineering
University of Arkansas
Fayetteville, AR 72701, USA

rparames@uark.edu and smithsco@uark.edu

Jia Di

Trustable Logic Circuit Design Laboratory
Department of Computer Science & Computer Engineering
University of Arkansas
Fayetteville, AR 72701, USA

jdi@uark.edu

Abstract—This paper develops a delay-insensitive (DI) digital design paradigm that utilizes ternary logic as an alternative to dual-rail logic for encoding the DATA and NULL states. This new Delay-Insensitive Ternary Logic (DITL) paradigm is compared with other DI paradigms, such as Pre-Charge Half-Buffers (PCHB) and NULL Convention Logic (NCL), showing that DITL significantly outperforms PCHB and NCL in terms of energy consumption, and is also more area efficient than NCL. Utilizing the DITL paradigm for designing secure hardware applications is then discussed.

Keywords—digital logic; asynchronous; delay-insensitive; clockless; ternary; PCHB; NCL; DITL; secure hardware

I. INTRODUCTION

For the last three decades, the focus of digital design has been primarily on synchronous, clocked architectures. However, as clock rates have significantly increased while feature size has decreased, clock skew has become a major problem. High performance chips must dedicate increasingly larger portions of their area for clock drivers to achieve acceptable skew, causing these chips to dissipate increasingly higher power, especially at the clock edge, when switching is most prevalent. As these trends continue, the clock is becoming more and more difficult to manage, while clocked circuits' inherent power inefficiencies are emerging as the dominant factor hindering increased performance. These issues have caused renewed interest in asynchronous digital design.

Asynchronous, clockless circuits require less power, generate less noise, and produce less electro-magnetic interference (EMI), compared to their synchronous counterparts, without degrading performance. Furthermore, delay-insensitive (DI) asynchronous paradigms have a number of additional advantages, especially when designing complex circuits, like Systems-on-Chip (SoCs), including substantially reduced crosstalk between analog and digital circuits, ease of integrating multi-rate circuits, and facilitation of component reuse.

As demand increases for designs with higher performance, greater complexity, and decreased feature size, asynchronous paradigms will become more prevalent in the multi-billion dollar semiconductor industry, as predicted by the International Technology Roadmap for Semiconductors (ITRS), which envisions a likely shift from synchronous to asynchronous design styles in order to increase circuit robustness, decrease

power, and alleviate many clock-related issues [1, 2]. ITRS shows that asynchronous circuits accounted for 11% of chip area in 2008, compared to 7% in 2007, and estimates they will account for 23% of chip area by 2014, and 35% of chip area by 2019 [3].

Section II provides an overview of the previous work in asynchronous logic and ternary logic. Section III develops the DITL paradigm. Section IV compares DITL to other DI paradigms; and Section V draws conclusions and presents areas for future work.

II. PREVIOUS WORK

Asynchronous circuits can be grouped into two main categories: bounded-delay and delay-insensitive models. Bounded-delay models, such as micropipelines [4], assume that delays in both gates and wires are bounded. Delays are added based on worst-case scenarios to avoid hazard conditions. This leads to extensive timing analysis of worst-case behavior to ensure correct circuit operation. On the other hand, delay-insensitive circuits, like NULL Convention Logic (NCL) [5] and Pre-Charge Half-Buffers (PCHB) [6], assume delays in both logic elements and interconnects to be unbounded, although they assume that wire forks within basic components, such as a full adder, are isochronic [7], meaning that the wire delays within a component are much less than the logic element delays within the component, which is a valid assumption even in future nanometer technologies. Wires connecting components do not have to adhere to the isochronic fork assumption. This implies the ability to operate in the presence of indefinite arrival times for the reception of inputs. Completion detection of the output signals allows for handshaking to control input wavefronts. Delay-insensitive design styles therefore require very little, if any, timing analysis to ensure correct operation (i.e., they are correct by construction), and also yield average-case performance rather than the worst-case performance of bounded-delay and traditional synchronous paradigms.

A. NULL Convention Logic (NCL)

NCL uses dual-rail signals to achieve delay-insensitive behavior. A dual-rail signal, D , consists of two wires, D^0 and D^1 , which may assume any value from the set {DATA0, DATA1, NULL}. The DATA0 state ($D^0 = 1, D^1 = 0$) corresponds to a Boolean logic0, the DATA1 state ($D^0 = 0,$

$D^1 = 1$) corresponds to a Boolean logic1, and the NULL state ($D^0 = 0, D^1 = 0$) corresponds to the empty set meaning that the value of D is not yet available. The two rails are mutually exclusive, so that both rails can never be asserted simultaneously; this state is an illegal state.

NCL differs from other gate-level DI paradigms [8-12] in that these other paradigms only utilize one type of state-holding gate, the C-element [13]. A C-element behaves as follows: when all inputs assume the same value then the output assumes this value, otherwise the output does not change. On the other hand, all NCL gates are state-holding. Thus, NCL circuits have a greater potential for optimization than other gate-level DI paradigms [14].

NCL uses threshold gates for its basic logic elements [15]. The primary type of threshold gate is the TH m n gate, where $1 \leq m \leq n$, as depicted in Fig. 1. TH m n gates have n inputs. At least m of the n inputs must be asserted before the output will become asserted. Because NCL threshold gates are designed with hysteresis, all asserted inputs must be de-asserted before the output will be de-asserted. This ensures a complete transition of inputs back to NULL before asserting the output associated with the next wavefront of input DATA. Therefore, a TH n n gate is equivalent to an n -input C-element and a TH1 n gate is equivalent to an n -input OR gate. In the representation of a TH m n gate, each of the n inputs is connected to the rounded portion of the gate; the output emanates from the pointed end of the gate; and the gate's threshold value, m , is written inside of the gate.

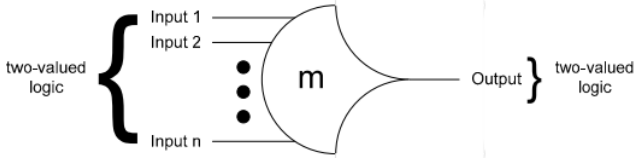


Figure 1. TH m n gate.

By employing threshold gates for each logic rail, NCL is able to determine the output status without referencing time. DI circuits communicate using request and acknowledge signals, K_i and K_o , respectively, as shown in Fig. 2, to prevent the current DATA wavefront from overwriting the previous DATA wavefront, by ensuring that the two DATA wavefronts are always separated by a NULL wavefront [5]. The acknowledge signal from the receiving circuit is the request signal to the sending circuit. When the receiver circuit latches the input DATA, the corresponding K_o signal will be logic0, indicating a *request-for-NULL* (*rfn*); and when it latches the input NULL, the corresponding K_o signal will be logic1, indicating a *request-for-DATA* (*rfd*). When the sending circuit receives a *rfd/rfn* on its K_i input, it will allow a DATA/NULL wavefront to be output, respectively. This delay-insensitive handshaking protocol coordinates DI circuit behavior, analogous to coordination of synchronous circuits by a clock signal. Additionally, delay-insensitivity requires a circuit to be *input-complete*, which means that all outputs may not

transition from NULL to DATA until all inputs have transitioned from NULL to DATA, and that all outputs may not transition from DATA to NULL until all inputs have transitioned from DATA to NULL [14]. In circuits with multiple outputs, it is acceptable according to Seitz's "weak conditions" of delay-insensitive signaling [9], for some of the outputs to transition without having a complete input set present, as long as all outputs cannot transition before all inputs arrive.

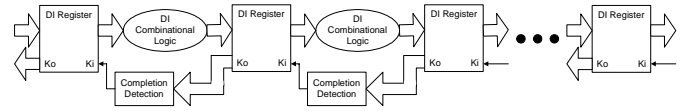


Figure 2. NCL system framework: input wavefronts are controlled by local handshaking signals and Completion Detection instead of a global clock.

B. Pre-Charge Half-Buffer (PCHB)

PCHB circuits [6] are designed at the transistor level, utilizing dynamic CMOS logic, instead of targeting a predefined set of gates like the previously mentioned DI paradigms [5, 8-12]. PCHB circuits have dual-rail data inputs and outputs, and combine combinational logic and registration together into a single block, as shown in Fig. 3, yielding a very fine-grain pipelined architecture. The dual-rail output is initially pre-charged to NULL. When request (R_{ack}) and acknowledgement (L_{ack}) are both *rfd*, the specific function will evaluate when the inputs, X and/or Y , become DATA, causing the output, F , to become DATA. L_{ack} will then transition to *rfn* only after all inputs and the output are DATA. When R_{ack} is *rfn* and L_{ack} is *rfd*, or vice versa, the output will be floating, so weak inverters must be used to hold the current output value. After both R_{ack} and L_{ack} are *rfn*, the output will be pre-charged back to NULL. After all inputs become NULL and the output changes to NULL, L_{ack} will change back to *rfd*, and the next DATA wavefront can evaluate after R_{ack} becomes *rfd*.

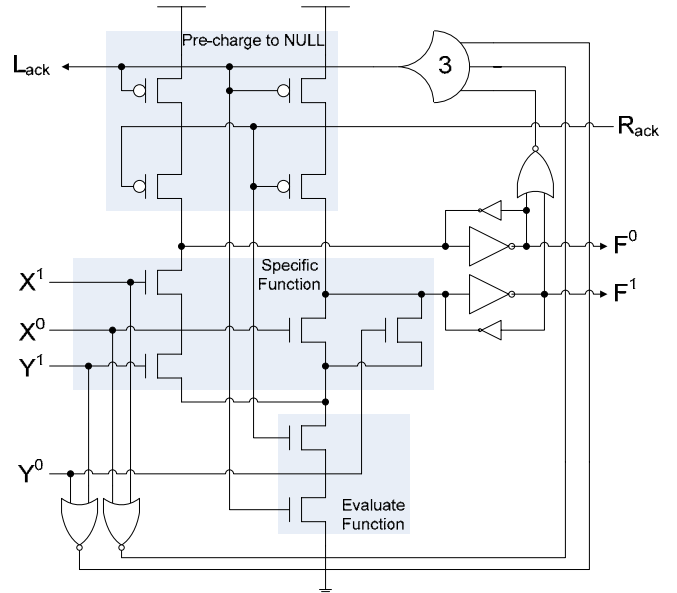


Figure 3. PCHB NAND2 circuit.

C. Ternary Logic

Ternary logic utilizes three distinct voltage values per wire, $0V$, $\frac{1}{2} V_{dd}$, and V_{dd} , whereas binary logic utilizes two distinct voltage values, $0V$ and V_{dd} . Hence, ternary logic can be used as an alternative to dual-rail logic to represent the three logic states (i.e., DATA0, DATA1, and NULL), requiring only one wire per bit. V_{dd} is used to represent DATA1, $0V$ to represent DATA0, and $\frac{1}{2} V_{dd}$ to represent NULL, which yields maximum noise margin with minimum switching power dissipation, since each wire always switches to NULL between every two DATA states, such that the voltage swing is always $\frac{1}{2} V_{dd}$.

[16, 17] develop a ternary logic completion detection circuit for use with a bounded-delay self-timed paradigm; and [18, 19] develops a ternary bounded-delay self-timed paradigm, which is similar to micropipelines [4]. However, as mentioned at the beginning of Section II, delay-insensitive paradigms have many more advantages compared to their bounded-delay counterparts. [20] develops a delay-insensitive ternary logic transmission system, called Asynchronous Ternary Logic Signaling (ATLS), which converts dual-rail signals into ternary logic for transmission over a bus, in order to decrease transmission area and power. However, all of the logic processing is still done using dual-rail logic. [21, 22] develop a circuit called a Watchful as part of their proposed delay-insensitive ternary logic paradigm. However, as shown in the timing diagram in Fig. 4, their approach is not delay-insensitive because it assumes that the input will transition to NULL before *clear* is asserted, causing *full* to be deasserted. In order to be delay-insensitive, *full* must not be deasserted until both *clear* is asserted and *in* transitions to NULL. Otherwise, if *in* remained at one DATA value (e.g., if no additional DATA needed to be processed at this time), this DATA value would continue to be utilized in subsequent operations instead of causing the system to become idle.

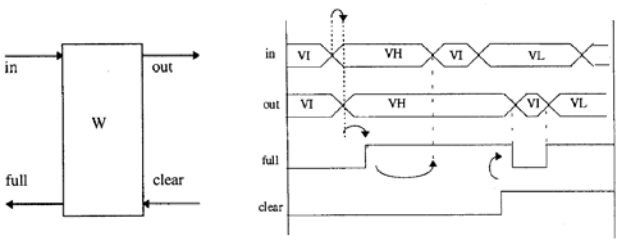


Figure 4. Watchful timing diagram [19].

[23] utilizes shifted-threshold transistors in special inverters to detect logic0 and logic1 for a ternary logic input, as shown in Fig. 5. For Detect0, *in* must be lower than $-2 \times V_{tP}$ for the PMOS transistors to turn on and pull *out* to V_{dd} . Similarly, for Detect1, *in* must be higher than $2 \times V_{tN}$ for *out* to be pulled down to $0V$. The truth table for Detect0 and Detect1 is provided in Table I.

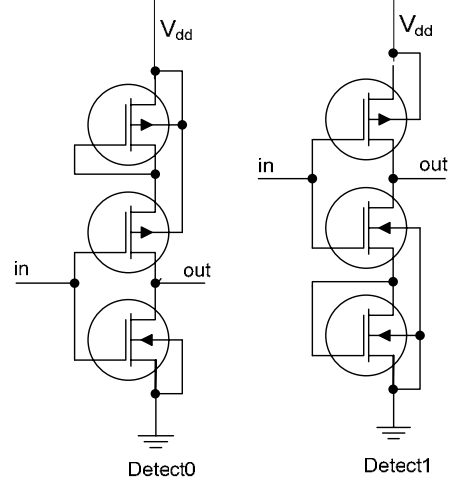


Figure 5. Original ternary logic detect circuits [23].

TABLE I. TRUTH TABLE FOR DETECT CIRCUITS

Input	Detect0 output	Detect1 Output
Gnd or DATA0	1	1
$\frac{1}{2} V_{dd}$ or NULL	0	1
V_{dd} or DATA1	0	0

III. DELAY INSENSITIVE TERNARY LOGIC (DITL)

The Delay-Insensitive Ternary Logic (DITL) paradigm developed in this paper utilizes three distinct voltage levels, $0V$, $\frac{1}{2} V_{dd}$, and V_{dd} , to encode the three DI logic states, DATA0, NULL, and DATA1, respectively, on a single wire, similar to other asynchronous ternary logic paradigms described in Section II.C. The motivations for utilizing ternary logic for delay-insensitive circuit design include reducing area, since only half the number of wires are required for each bit compared to dual-rail logic, and reducing power/energy, since each transition (i.e., NULL to DATA or vice-versa) only requires a $\frac{1}{2} V_{dd}$ swing compared to a full V_{dd} swing for dual-rail logic.

The DITL paradigm is based on the PCHB paradigm [6], shown in Fig. 3, where each component is designed at the transistor level, and consists of dual-rail data inputs and outputs, with registration included in every combinational logic component. Like PCHB, DITL circuits are designed at the transistor level, but consist of ternary data inputs and outputs and binary handshaking signals. As shown in Fig. 6, when R_{ack} and L_{ack} are both *rfd* and the inputs, X and Y , are both DATA, the specific function will evaluate, causing the output, F , to become DATA, which will then transition L_{ack} to *rfn*. When L_{ack} is *rfn* and R_{ack} is still *rfd*, the specific function is floating, so the output needs to be held at its proper DATA value, either DATA0 or DATA1, which is done by the Hold 0 and Hold 1 circuitry, respectively. After R_{ack} changes to *rfn*, the output will be pre-charged to NULL (i.e., $\frac{1}{2} V_{dd}$), through N-fets for increased speed. After all inputs become NULL and the output

changes to NULL, L_{ack} will change back to rfd , and the next DATA wavefront can evaluate after R_{ack} becomes rfd and the inputs change to DATA. If R_{ack} changes to rfd before the inputs become NULL, if the inputs become NULL before R_{ack} changes to rfd , or if both R_{ack} and L_{ack} are rfd but the inputs are still NULL, the pre-charge to NULL logic will no longer be conducting, so the NULL output must be maintained through the Hold NULL circuitry. Note that the Is DATA component used in the DITL architecture, shown in Fig. 8, has a D output that is logic1 when the input is either DATA0 or DATA1, and is logic0 when the input is NULL; $Is0$ is logic1 when the input is DATA0 and logic0 when the input is either NULL or DATA1; and $Is1$ is logic1 when the input is DATA1 and logic0 when the input is either NULL or DATA0, as summarized in Table II. Fig. 7 shows the Cadence simulation of the DITL NAND function, using the 1.2V, 0.13 μ m IBM 8RF-DM process.

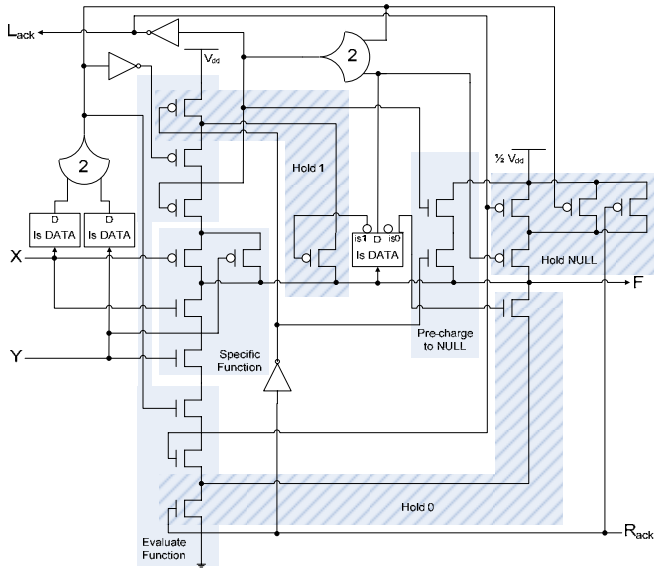


Figure 6. Version I of DITL NAND2 circuit.

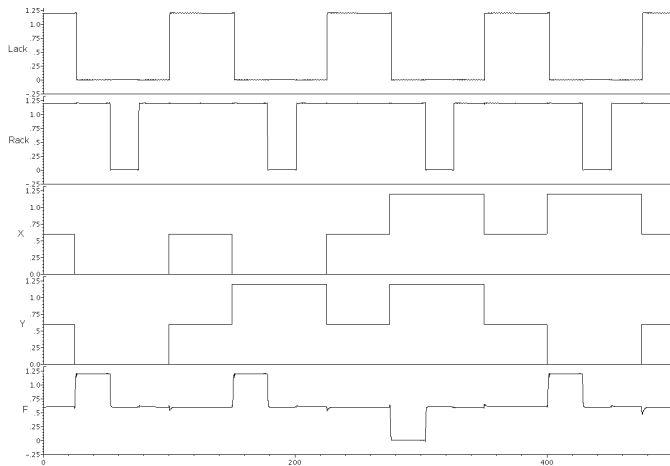


Figure 7. Cadence simulation of DITL NAND2 circuit.

TABLE II. TRUTH TABLE FOR IS DATA COMPONENT

Input	D	Is1	Is0
DATA0 (0V)	1	0	1
NULL ($\frac{1}{2} V_{dd}$)	0	0	0
DATA1 (V_{dd})	1	1	0

The DITL Is DATA component utilizes Detect0 and Detect1 circuits, as discussed in Section II.C; however, in lieu of a 3rd transistor to effectively increase the transistor threshold voltage, reverse body bias (RBB) [24–26] is used, as shown in Fig. 8, in order to reduce static power consumption. Specifically, all transistors in the previous detect circuits [23] are partially on for a NULL ($\frac{1}{2} V_{dd}$) input, which consumes significant static power: 31.8 nW for Detect0 and 5.5 nW for Detect1. Using the following body biases: $VBp0 = +4V$; $VBn0 = 0V$; $VBp1 = +1.5V$; and $VBn1 = -2.4V$, significantly reduces the 2-transistor static power to 1.13 nW for Detect0 and 0.98 nW for Detect1. Additionally, the 3-transistor detect circuits require output inverters to properly shape the outputs; otherwise the output is only 1.07V instead of 1.2V for Detect0 with an input of 0V, and 0.17V instead of 0V for Detect1 with an input of 1.2V. The 2-transistor detect circuits are also faster than their 3-transistor counterparts (i.e., average propagation delay of 0.37 ns vs. 0.45 ns for Detect0 and 0.33 ns vs. 0.65 ns for Detect1). The above analysis for the detect circuits was performed using the 1.2V, 0.13 μ m IBM 8RF-DM process.

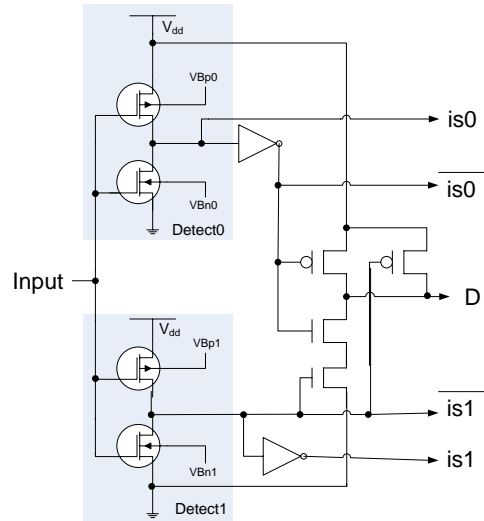


Figure 8. Is Data component using reverse body bias detect circuits.

As an alternative to Version I of the DITL circuit architecture, Version II is shown in Fig. 9, where the *Specific Function* inputs come from the input Is DATA components instead of the external inputs, X and Y . Version II requires one additional inverter for each data input (in the Is DATA component for the $is1$ output), but the advantage is that each data input drives exactly one Is DATA component for each DITL circuit to which it is an input, such that the capacitance driven by a particular signal only depends on the number of circuits to which the signal is an input, and not on the type of

circuits it drives (e.g., if signal *A* is an input to an XOR2 and NOR3 circuit and signal *B* is an input to a NAND4 and OR2 circuit, both drive the same amount of capacitance because they both drive two *Is DATA* components).

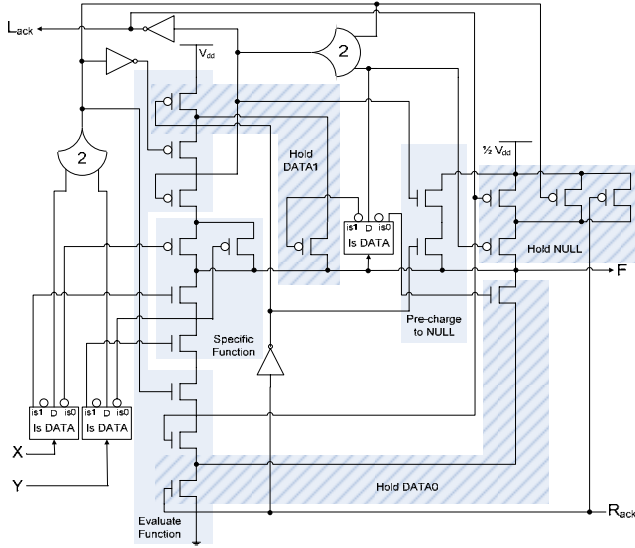


Figure 9. Version II of DITL NAND2 circuit.

IV. COMPARISON RESULTS

Cadence simulations of NAND2 circuits for both versions of DITL as well as PCHB and NCL were performed, and the results listed in Table III. Note that the NCL NAND2 circuit also includes input and output registers to make it comparable with DITL and PCHB, which both include registration within each combinational circuit. DITL Version I is slightly slower, but requires slightly less area and energy compared to Version II. Compared to PCHB, DITL is 21% slower, 74% larger, but requires 68% less energy. Compared to NCL, DITL is 50% slower, but requires 38% less energy and is 89% smaller. Therefore, DITL has a significant energy advantage compared to PCHB and NCL, and is also more area efficient than NCL. Additionally, as circuit size increases, DITL and PCHB circuits increase at a much smaller rate than NCL circuits (e.g., for a NAND2 vs. a NAND4 circuit, the area increase is 42% for DITL, 70% for PCHB, and 94% for NCL).

TABLE III. NAND2 COMPARISON

	Avg. DATA-NULL Cycle (ns)	Avg. Energy per Operation (fJ)	Area (# transistors)
DITL V1	5.43	50.3	78
DITL V2	5.40	52.3	82
PCHB	4.49	86.3	46
NCL	3.61	70.8	151

One potential application for DITL Version II is secure hardware, where the objective is to balance power, timing, and

electromagnetic (EM) emissions to prevent side-channel attacks [27-29]. Dual-rail asynchronous circuits have been shown to possess significant advantages for secure hardware compared to their synchronous counterparts, because the circuit switches from NULL (N) to DATA (D) (either DATA0, D0, or DATA1, D1) and back to NULL for each operation, regardless of the current or previous data pattern [30]. Since DITL only has one output wire, compared to two output wires for dual-rail logic, timing, power, and emissions can be more easily balanced because each signal will only drive a single capacitance, and a gate's output will always make a $\frac{1}{2} V_{dd}$ transition every DATA and NULL cycle, regardless of the DATA value (i.e., $\frac{1}{2} V_{dd} \rightarrow V_{dd} \rightarrow \frac{1}{2} V_{dd}$ for a N→D1→N transition and $\frac{1}{2} V_{dd} \rightarrow 0 \rightarrow \frac{1}{2} V_{dd}$ for a N→D0→N transition).

In general, for secure hardware applications, a cell library consisting of various-input gates, balanced for timing and power based on the number and type of gates being driven, would need to be developed. Since each DITL gate input always drives exactly one *Is DATA* component, as shown in Fig. 9, the type of gate being driven will not affect the load capacitance, such that the selection of the properly balanced DITL gate only depends on the number of gates it drives, which substantially reduces the number of balanced gates needed for a balanced gate library. To balance timing and power, transistors are sized to yield similar output rise and fall times, propagation delays, peak current spike during transitions, and energy, for all possible transitions.

As proof of concept, a series of full adders (FAs) have been designed in Boolean, NCL, and DITL, using the 1.2V, 0.13 μ m IBM 8RF-DM process. The Boolean FA is a standard gate-level design consisting of five logic gates, as shown in Fig. 10. The DITL FA also consists of five gates, including three different types balanced for timing/power through proper transistor sizing: an XOR2 that drives 2 gates, a NAND2 that drives 1 gate, and a NAND2 that drives 2 gates. For the NCL FA, two versions have been designed: one is a 10-threshold-gate design that utilizes complete logic functions to directly implement Fig. 10, denoted as NCL-10G; the other is an optimized 4-threshold-gate design [14], denoted as NCL-4G. As summarized in Table IV, these four FAs, simulated in Cadence Spectre, are compared in five categories: “*Sum/C_{out}* transition slope” is the combined rise/fall time during each transition for *Sum* and *C_{out}* outputs, respectively; “delay” is the total time for a N→D→N cycle; “peak current spike” is the magnitude of the supply voltage current spike during each transition; and “energy” is the total energy consumed during each transition. Table IV shows the maximum variance percentage of each parameter among all possible input combinations.

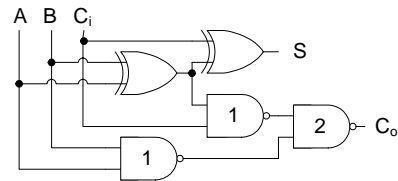


Figure 10. Full adder circuit.

TABLE IV. FULL ADDER COMPARISON

Full Adder	Maximum Variance Percentage				
	Sum Transition Slope	Cout Transition Slope	Delay	Peak Current Spike	Energy
Boolean	27.8%	11.4%	93.6%	221.4%	313.4%
NCL 4-G	21.0%	13%	105.3%	51%	32.0%
NCL 10-G	12.9%	58.4%	19.0%	47.2%	10.4%
DITL	8.5%	5.6%	13.8%	18.1%	7.4%

Although NCL as a dual-rail asynchronous logic is well-known to be more side-channel attack resistant compared to Boolean logic, the DITL design exhibits the least variations in all parameters, as shown in Table IV. Since power (energy and current spike) and timing (slope and delay) are significantly more balanced for DITL, differential power attacks and timing attacks will be much more difficult to succeed.

As for EM attack resistance analysis, the EM data is usually generated from fabricated chip testing and is very difficult to accurately simulate. However, some preliminary analysis can be done through general Maxwell equations for electric and magnetic fields [31]. Since the attacker's antenna is in a fixed position during each attack period, its distance and angles to the target chip can be viewed as constants. Therefore, the equations can be simplified as shown in (1), where E and H are the electric and magnetic fields, respectively, i is the current magnitude, f is the current frequency, and all other parameters are constants.

$$E = \sqrt{(A \cdot i)^2 + \left(\frac{3}{r}\right)^2 - (C \cdot i)^2 + \left[D \cdot i \cdot \left(\frac{E}{f} - H \cdot f\right)\right]^2}, \quad H = \sqrt{(E \cdot i)^2 + (L \cdot i \cdot f)^2} \quad (1)$$

From (1), both electric and magnetic fields are functions of two variables: how much the current changes (i), and how fast this change occurs (f). Similar to the power/timing fluctuations among processing different data, i and f in an unprotected circuit are also strongly correlated to the data, which leaks information to EM attackers. Such correlation can be clearly seen in Table IV for Boolean and NCL circuits, where the Peak Current Spike shows how much the current changes, and the Transition Slope and Delay show how fast the current change occurs. Note that NCL as dual-rail asynchronous logic is known to be resistant to EM attacks [32]. However, Table IV shows that these parameters are much more balanced in the DITL circuit. Therefore, it can be expected that DITL circuits will render EM attacks much less effective.

V. CONCLUSIONS AND FUTURE WORK

This paper developed the delay-insensitive DITL paradigm, which utilizes ternary logic instead of dual-rail logic to encode the DATA0, DATA1, and NULL states. DITL was then compared to two popular dual-rail delay-insensitive paradigms, PCHB and NCL, showing that DITL has significant advantages compared to NCL for area, and both NCL and PCHB for energy. DITL was then compared to Boolean and NCL for a secure hardware application, showing that DITL is expected to be much less susceptible to power, timing, and EM-based

attacks. Future work in this area will include developing and fabricating a large DITL circuit, such as a microprocessor, such that the physical chip can be tested for resistance to power, timing, and EM-based attacks.

REFERENCES

- [1] <http://www.itrs.net/Links/2003ITRS/Design2003.pdf> (available April 2009).
- [2] http://www.itrs.net/Links/2007ITRS/2007_Chapters/2007_Design.pdf (available April 2009).
- [3] <http://www.itrs.net/Links/2008ITRS/Home2008.htm> (available April 2009)
- [4] Ivan E. Sutherland, "Micropipelines," *Communications of the ACM*, Vol. 32/6, pp. 720-738, 1989.
- [5] K. M. Fant and S. A. Brandt, "NULL Convention Logic: A Complete and Consistent Logic for Asynchronous Digital Circuit Synthesis," *International Conference on Application Specific Systems, Architectures, and Processors*, pp. 261-273, 1996.
- [6] A. J. Martin and M. Nystrom, "Asynchronous techniques for system-on-chip design," *Proceedings of the IEEE*, pp. 1089 – 1120, Vol. 94, No. 6, June 2006.
- [7] K. Van Berkel, "Beware the Isochronic Fork," *Integration, the VLSI Journal*, Vol. 13/2, pp. 103-128, 1992.
- [8] I. David, R. Ginosar, and M. Yoeli, "An Efficient Implementation of Boolean Functions as Self-Timed Circuits," *IEEE Transactions on Computers*, Vol. 41/1, pp. 2-10, 1992.
- [9] C. L. Seitz, "System Timing," in *Introduction to VLSI Systems*, Addison-Wesley, pp. 218-262, 1980.
- [10] J. Sparsø, J. Staunstrup, M. Dantzer-Sorensen, "Design of Delay Insensitive Circuits using Multi-Ring Structures," *Proceedings of the European Design Automation Conference*, pp. 15-20, 1992.
- [11] T. S. Anantharaman, "A Delay Insensitive Regular Expression Recognizer," *IEEE VLSI Technical Bulletin*, Sept. 1986.
- [12] N. P. Singh, *A Design Methodology for Self-Timed Systems*, Master's Thesis, MIT/LCS/TR-258, Laboratory for Computer Science, MIT, 1981.
- [13] D. E. Muller, "Asynchronous Logics and Application to Information Processing," in *Switching Theory in Space Technology*, Stanford University Press, pp. 289- 297, 1963.
- [14] S. C. Smith, R. F. DeMara, J. S. Yuan, D. Ferguson, and D. Lamb, "Optimization of NULL Convention Self-Timed Circuits," *Integration, the VLSI Journal*, Vol. 37/3, pp. 135-165, August 2004.
- [15] G. E. Sobelman and K. M. Fant, "CMOS Circuit Design of Threshold Gates with Hysteresis," *IEEE International Symposium on Circuits and Systems (II)*, pp. 61- 65, 1998.
- [16] C. L. Connell and P.T. Balsara, "A new ternary MVL based completion detection method for the design of self-timed circuits using dynamic CMOS logic," *Proceedings of the 45th Midwest Symposium on Circuits and Systems MWSCAS-2002*, Vol. 1, pp. 503-506, Aug. 2002.
- [17] C. L. Connell and P.T. Balsara, "A novel single-rail variable encoded completion detection scheme for self-timed circuit design using ternary multiple valued logic," *Proceedings of the IEEE 2nd Dallas CAS Workshop on Low Power/Low Voltage Mixed-Signal Circuits and Systems*, pp. 7 – 10, March 2001.
- [18] Y. Nagata and M. Mukaidono, "Design of an asynchronous digital system with Bternary logic," *Proceedings of the 27th International Symposium on Multiple-Valued Logic*, pp. 265 – 271, May 1997.
- [19] Y. Nagata, D.M. Miller and M. Mukaidono, "B-ternary logic based asynchronous micropipeline," *Proceedings of the 29th IEEE International Symposium on Multiple-Valued Logic*, pp. 214 – 219, May 1999.
- [20] T. Felicijan and S.B Furber, "An Asynchronous Ternary Logic Signaling system," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 11, Issue 6, pp. 1114 – 1119, Dec. 2003.
- [21] R. Mariani, R. Roncella, R. Saletti and P. Terreni, "On the Realisation of Delay-Insensitive Asynchronous Circuits with CMOS Ternary logic,"

Third International Symposium on Advanced Research in Asynchronous Circuits and Systems (ASYNC '97), 1997.

- [22] R. Mariani, R. Roncella, R. Saletti and P. Terreni, "A useful application of CMOS ternary logic to the realisation of asynchronous circuits," *Proceedings of the 27th International Symposium on Multiple-Valued Logic*, pp. 203 – 208, May 1997.
- [23] J. L. Huertas and J. M. Carmona, "Low-power Ternary CMOS Circuits," *IEEE Proceedings of ISMVL*, pp. 170-174, 1979.
- [24] A. Keshavarzi, S. Ma, S. Narendra, B. Bloechel, K. Mistry, T. Ghani, S. Borkar, V. De, "Effectiveness of reverse body bias for leakage control in scaled dual Vt CMOS ICs," *Proceedings of the 2001 international symposium on Low power electronics and design*, pp. 207-212, August 2001.
- [25] K. Nose, M. Hirabayashi, H. Kawaguchi, S. Lee, and T. Sakurai, "VTH-Hopping Scheme to Reduce Subthreshold Leakage for Low-Power Processors," *IEEE Journal of Solid-State Circuits*, Vol. 37, No. 3, March 2002.
- [26] J. Tschanz, J. Kao, S. Narendra, R. Nair, D. Antoniadis, A. Chandrakasan, and V. De, "Adaptive Body Bias for Reducing Impacts of Die-to-Die and Within-Die Parameter Variation on Microprocessor Frequency and Leakage," *ISSCC Digest of Technical Papers*, pp. 412--413, Feb. 2002.
- [27] P. Kocher, J. Jaffe, and B. Jun "Differential Power Analysis," *Springer-Verlag, LNCS 1666, Crypto '99*, pp. 388-397, 1999.
- [28] P. Kocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems," *16th Annual International Cryptology Conference on Advances in Cryptology*, pp. 104-113, 1996.
- [29] E. Mulder, S. Ors, B. Preneel, and I. Verbauwhede, "Differential Electromagnetic Attack on an FPGA Implementation of Elliptic Curve Cryptosystems," *WAC*, pp. 1-6, 2006.
- [30] Z. Yu, S. Furber, L. Plana, "An Investigation into the Security of Self-timed Circuits," *9th International Symposium on Asynchronous Circuits and Systems*, pp. 206-215, 2003.
- [31] T. S. Rappaport, *Wireless Communications: Principles and practice*, 2nd Ed., Prentice Hall, 2002.
- [32] C. Hanken, J. Le, T. S. Fiez, and K. Mayaram. "Simulation and modeling of substrate noise generation from synchronous and asynchronous digital logic circuits," *CICC*, pp. 845-848, Sept. 2007.