

# Multi-Threshold Asynchronous Circuit Design for Ultra-Low Power

Andrew Bailey<sup>1</sup>, Ahmad Al Zahrani<sup>1</sup>, Guoyuan Fu<sup>2</sup>, Jia Di<sup>1,\*</sup>, and Scott Smith<sup>2</sup>

<sup>1</sup>*Department of Computer Science and Computer Engineering, University of Arkansas, Fayetteville, AR 72701, USA*

<sup>2</sup>*Department of Electrical Engineering, University of Arkansas, Fayetteville, AR 72701, USA*

(Received: 14 June 2008; Accepted: 9 October 2008)

This paper presents an ultra-low power circuit design methodology which combines the Multi-Threshold CMOS (MTCMOS) technique with quasi delay-insensitive (QDI) asynchronous logic, in order to solve the three major problems of synchronous MTCMOS circuits: (1) Sleep signal generation, (2) storage element data loss during sleep mode, and (3) sleep transistor sizing. In contrast to most power reduction methods that result in area overhead, the QDI asynchronous MTCMOS circuits are usually smaller than their original versions. Moreover, QDI circuits utilize handshaking protocols instead of clocks for circuit control, resulting in flexible timing requirements, which yields increased circuit robustness and allows for extreme supply voltage scaling to subthreshold region for further power reduction, without requiring any circuit modifications. This QDI asynchronous MTCMOS methodology is used to design a 4-stage pipelined 8-bit  $\times$  8-bit unsigned multiplier, which is then compared against the original QDI design (i.e., without incorporating MTCMOS) and its synchronous version. All designs use the IBM 8RF-DM 0.13  $\mu\text{m}$  process. Results show 150 $\times$  and 1.8 $\times$  leakage power and active energy reductions on average in the QDI asynchronous MTCMOS design compared to the original QDI version, respectively.

**Keywords:** Ultra-Low Power, Quasi Delay-Insensitive Asynchronous Logic, MTCMOS, NULL Convention Logic.

## 1. INTRODUCTION

With the current trend of semiconductor devices scaling into deep submicron region, design challenges that were previously minor issues now become increasingly important. Where in the past dynamic power has been the major factor in CMOS digital circuit power consumption, recently with the dramatic decrease of supply and threshold voltages, a significant growth in leakage power demands new design methodologies for digital integrated circuits (ICs) to meet the new power constraints. As one of the major components of leakage power, subthreshold leakage is caused by the current flowing through a transistor even though it is supposedly turned off. The shrinking in transistor feature size exponentially increases the impact of subthreshold leakage.

Many techniques have been proposed to control or minimize leakage power in deep submicron technology.

In Ref. [1] several leakage minimization techniques are discussed and compared. Super Cutoff CMOS (SCCMOS)<sup>2</sup> under-drives (or over-drives) the sleep transistors, which gate the power to the circuit, to reduce leakage. Forced Transistor Stacking<sup>3</sup> takes advantage of stack effect, i.e., leakage current decreases due to two or more series transistors that are turned off, by replacing a single transistor by two transistors with half width. Sleepy Stack<sup>4</sup> combines forced stacking and sleep transistors to reduced delay. Input Vector Control<sup>5</sup> analyzes the input vector dependence to shut down the circuit in standby mode. Optimal supply and threshold voltage scaling to achieve minimum energy operation is discussed in Ref. [6]. Variable Threshold voltage subthreshold CMOS (VT-sub-CMOS) and subthreshold Dynamic Threshold voltage MOS (sub-DTMOS) are introduced in Ref. [7] to extend VTCMOS and DTMOS concepts to subthreshold region. Adaptive Body Bias (ABB) is discussed in Ref. [8] to provide different biasing voltage to the bulk nodes of the transistors to change the threshold voltages ( $V_t$ ). Pseudo-NMOS logic for subthreshold leakage reduction is discussed in Ref. [9].

\*Author to whom correspondence should be addressed.  
 Email: jdi@uark.edu

In addition to the leakage reduction methods above, Multi-Threshold CMOS,<sup>10</sup> which reduces leakage power by disconnecting the power supply from the circuit during the standby (or sleep) mode while maintaining high performance in the active mode, has been widely adopted in industry. MTCMOS incorporates transistors with two or more different threshold voltages in a circuit. Low threshold transistors offer fast speed but cost high leakage. In contrast, high threshold transistors suffer from reduced speed, but leak less current when turned off. MTCMOS combines these two types of transistors by utilizing low threshold voltage transistors for circuit switching to preserve performance and high threshold voltage transistors to gate the supply power in order to suppress the subthreshold leakage. Section 2 describes MTCMOS implementation in synchronous circuits in more detail.

Another area of interest in low power research is asynchronous logic, which uses handshaking protocols instead of clocks to control circuit behavior. As clock rates have significantly increased while feature size has decreased, the clock has become a major problem of synchronous circuits. Hence, the 2005 edition of the International Technology Roadmap for Semiconductors (ITRS) report predicts that asynchronous (clockless) paradigms will become more widely used in the industry to increase circuit robustness, decrease power, and alleviate many clock related issues; and the 2007 edition shows that asynchronous circuits account for 11% of chip area in 2008, compared to 7% in 2007, and estimates that asynchronous circuits will account for 22% of chip area within the next 5 years, and 30% of chip area within the next 10 years. The advantages of asynchronous circuits include no clock tree, high power efficiency, flexible timing requirement, robust circuit operation, and low noise/emission. Asynchronous circuits, especially quasi delay-insensitive asynchronous circuits, allow the power supply to be scaled to extremely low voltages, in some cases way below the threshold voltages of the transistors, while maintaining the correct circuit operation, and therefore have the potential to achieve ultra-low power consumption. As shown in this paper, by incorporating the MTCMOS technique into quasi delay-insensitive asynchronous logic, the three primary drawbacks of synchronous MTCMOS circuits, namely, *Sleep* signal generation, storage element data loss during sleep mode, and sleep transistor sizing, can be eliminated. By implementing a performance enhancing technique named Early Completion, a low active/leakage power, zero or even negative area overhead (i.e., the MTCMOS circuit is even smaller than its original version), and robust digital circuit architecture can be achieved.

This paper is organized as follows: Section 2 describes the synchronous MTCMOS architecture and its drawbacks; Section 3 introduces NULL Convention Logic (NCL), the quasi delay-insensitive asynchronous paradigm

used in the designs; Section 4 discusses the implementation of MTCMOS in NCL circuits; Section 5 presents the designed circuits for comparison, the simulation results, and the result analysis; and Section 6 draws conclusions and describes related future research areas.

## 2. MULTI-THRESHOLD CMOS FOR SYNCHRONOUS CIRCUITS

There are multiple ways to implement MTCMOS in synchronous circuits. One method is to use low threshold (low- $V_t$ ) transistors to build the circuit units on critical paths, while those on non-critical paths use high threshold (high- $V_t$ ) transistors. This allows the critical paths of a circuit to retain high speed, but use less leaky transistors in portions of the circuit with lower speed requirements. In addition to this path replacement methodology, there are two other architectures of implementing MTCMOS. A large-scale technique investigated in Ref. [11] is to use low threshold logic for all circuit functions and to gate the logic with high threshold sleep transistors between the logic and the power source, as shown in Figure 1. The sleep transistors are controlled by the *Sleep* signal. During the active mode, the *Sleep* signal is deasserted, causing both high- $V_t$  transistors to turn on and provide a virtual power and ground to the low- $V_t$  logic. When the circuit is inactive, *Sleep* signal is asserted, forcing both high- $V_t$  transistors

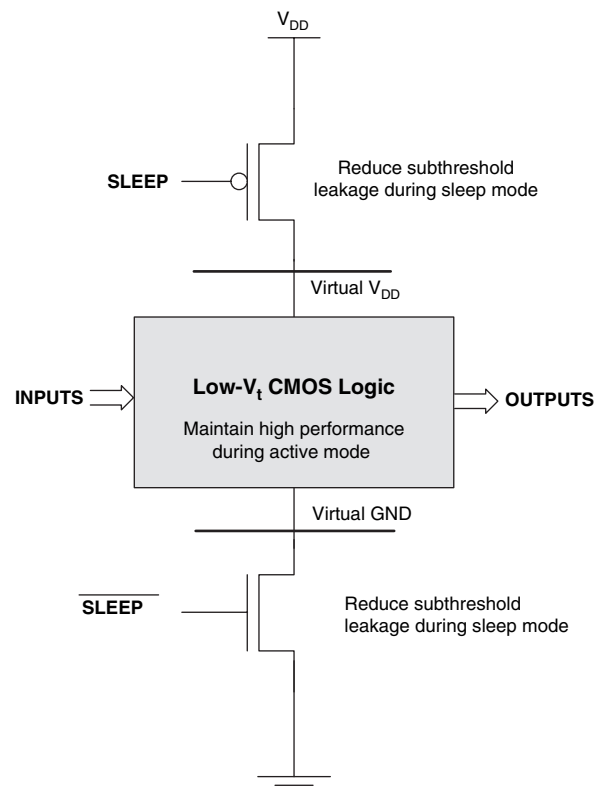


Fig. 1. General MTCMOS circuit architecture.

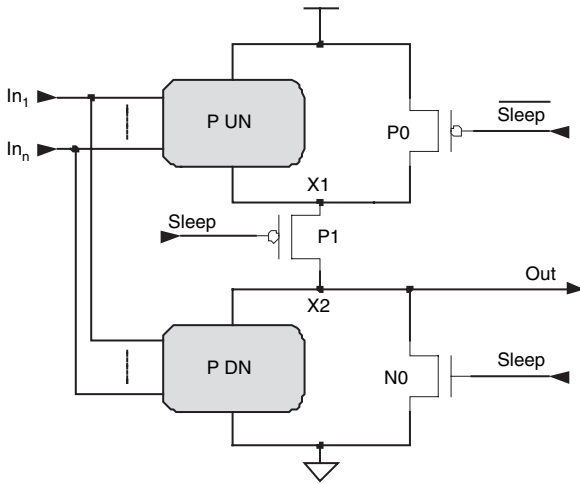


Fig. 2. Implementing MTCMOS in synchronous cells.

to cut-off and disconnect the power lines from the low- $V_t$  logic; this results in a very low sub-threshold leakage current from power to ground when the circuit is in standby mode. One drawback of this method is that partitioning and sizing of the sleep transistors is difficult for large circuits. The other architecture option, shown in Figure 2, is more fine-grained, and modifies each individual cell in the library,<sup>12</sup> using low threshold transistors for both pull-up and pull-down networks in each cell and high threshold transistors to gate the leakage power between the networks. Two extra low threshold transistors are included in parallel with the pull-up and pull-down networks to maintain equivalent voltage potential. Implementing MTCMOS in each cell eases the sleep transistor sizing; however, it also causes large area overhead.

In general, three serious drawbacks hinder the widespread use of MTCMOS in synchronous circuits:<sup>11</sup> (1) the generation of *Sleep* signals is timing critical, often requiring complex logic circuits; (2) synchronous circuits lose data when the power transistors are turned off; and (3) proper sizing of the sleep transistors is a very difficult task, which is critical for correct circuit operation. However, all three of these drawbacks can be eliminated by utilizing quasi delay-insensitive asynchronous logic in conjunction with the MTCMOS technique, as shown in this paper.

### 3. NULL CONVENTION LOGIC (NCL)

#### 3.1. NCL System Architecture and Dual-Rail Encoding

Quasi delay-insensitive (QDI) design styles, like NULL Convention Logic (NCL), require very little, if any, timing analysis to ensure correct operation (i.e., they are correct by construction). NCL circuits utilize multi-rail signals, such as dual-rail logic, to achieve delay-insensitivity. A dual-rail signal,  $D$ , consists of two wires,  $D^0$  and  $D^1$ . The DATA0 state ( $D^0 = 1, D^1 = 0$ ) corresponds to a Boolean logic 0; the DATA1 state ( $D^0 = 0, D^1 = 1$ ) corresponds to a Boolean logic 1; 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.<sup>13</sup> The two rails are mutually exclusive, such that both rails can never be asserted simultaneously; this state is defined as an illegal state.

The NCL system architecture consists of QDI combinational logic sandwiched between QDI registers, as shown in Figure 3, which is very similar to synchronous systems, such that the automated design of NCL circuits can follow the same fundamental steps as synchronous circuit design automation. The Completion Detection block of each stage ensures that all outputs of the corresponding QDI registers become DATA (or NULL) before it sends the handshaking signal to the previous stage allowing for the next NULL (or DATA) state, as described in Section 3.3.

#### 3.2. Threshold Gates

NCL circuits are comprised of 27 fundamental gates, called threshold gates,<sup>14</sup> which comprise the set of all functions of four or fewer variables. The primary type of threshold gates, shown in Figure 4, is the TH $_{mn}$  gate, where  $1 \leq m \leq n$ . TH $_{mn}$  gates have  $n$  inputs; at least  $m$  of the  $n$  inputs must be asserted before the output will become asserted; and NCL threshold gates are designed with *hysteresis* state-holding capability, such that after the output is asserted, all inputs must be deasserted before the output will be deasserted. Hysteresis ensures a complete transition of inputs back to NULL before asserting the output associated with the next wavefront of input data.

The CMOS implementation of each threshold gate consists of four major blocks and an output inverter with

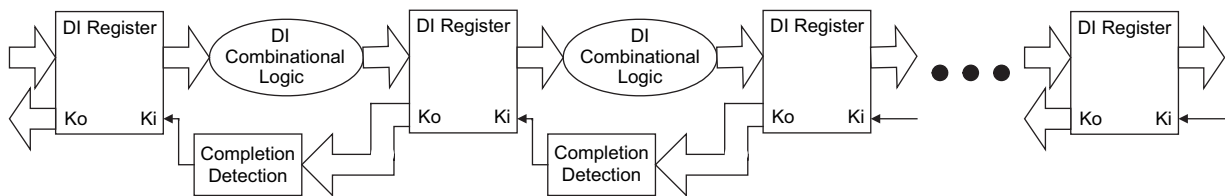


Fig. 3. NCL system architecture: Input wavefronts are controlled by local handshaking signals and completion detection instead of by a global clock signal.

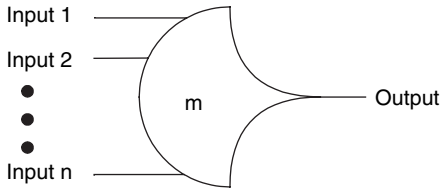


Fig. 4. THmn threshold gate.

feedback, as shown in Figure 5(a). The “Go-to-NULL” block forces the output to ‘0’ when activated; the “Hold-NULL” block makes sure the output keeps ‘0’ when the number of logic high inputs does not meet or exceed the gate’s threshold; the “Go-to-DATA” block forces the output to ‘1’ when the gate’s threshold is met; and the “Hold-DATA” block makes sure the output keeps ‘1’ until all inputs return to ‘0’. As an example, Figure 5(b) shows a TH23 gate and the corresponding four functional blocks. If all inputs are ‘0’, the three PMOS transistors in the “Go-to-NULL” block will be turned on and the output will become ‘0’; if two or more inputs are ‘1’, at least one of the three paths formed by the five NMOS transistors in the “Go-to-DATA” block will be turned on and the output will become ‘1’; otherwise, depending on the previous output, one of the six paths, including three paths formed by the seven PMOS transistors in the “Hold-NULL” block and the other three formed by the five NMOS transistors in the “Hold-DATA” block, will be turned on and the output will be held at either ‘0’ or ‘1’.

NCL threshold gate variations include *resetting* THnn and *inverting* TH1n gates. Circuit diagrams designate resettable gates by either a *d* or an *n* appearing inside the gate, along with the gate’s threshold. *d* denotes the gate as being reset to logic 1; *n*, to logic 0. Both resettable and inverting gates are used in the design of NCL registers.<sup>15</sup>

### 3.3. Regular and Early Completion Schemes

Two adjacent register stages interact through their request and acknowledge signals,  $K_i$  and  $K_o$ , respectively, 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. The acknowledge signals are combined in the Completion Detection circuitry, as shown in Figure 3, to produce the request signal to the previous register stage. When all current register outputs are DATA, the corresponding completion detection signal will be logic 0, indicating a “request-for-NULL”; and when all current register outputs are NULL, the corresponding completion detection signal will be logic 1, indicating a “request-for-DATA.” After receiving the request signal, the previous register will allow the corresponding NULL/DATA wavefront to pass to the combinational logic block between the two registers. This handshaking protocol coordinates NCL circuit behavior, analogous to coordination of synchronous circuits by a clock signal.

For the MTCMOS NCL version, to be introduced in Section 4, the standard NCL pipeline architecture, shown in Figure 3, needs to be slightly modified by utilizing the Early Completion technique,<sup>16</sup> shown in Figure 6, in order to maintain quasi delay-insensitivity. Otherwise, if the output of stage<sub>*i*</sub>’s original NCL completion component was used to sleep stage<sub>*i*</sub>’s combinational logic, all circuitry would become logic 0 whenever the completion component requested NULL, without first waiting for the circuit inputs to become NULL, as required for DI signaling. Early Completion utilizes the inputs of register<sub>*i-1*</sub> along with the  $K_i$  request to register<sub>*i-1*</sub> to generate the request signal to register<sub>*i-2*</sub>. Now this request signal to register<sub>*i-2*</sub> can be used to sleep the combinational circuitry in stage<sub>*i*</sub> without

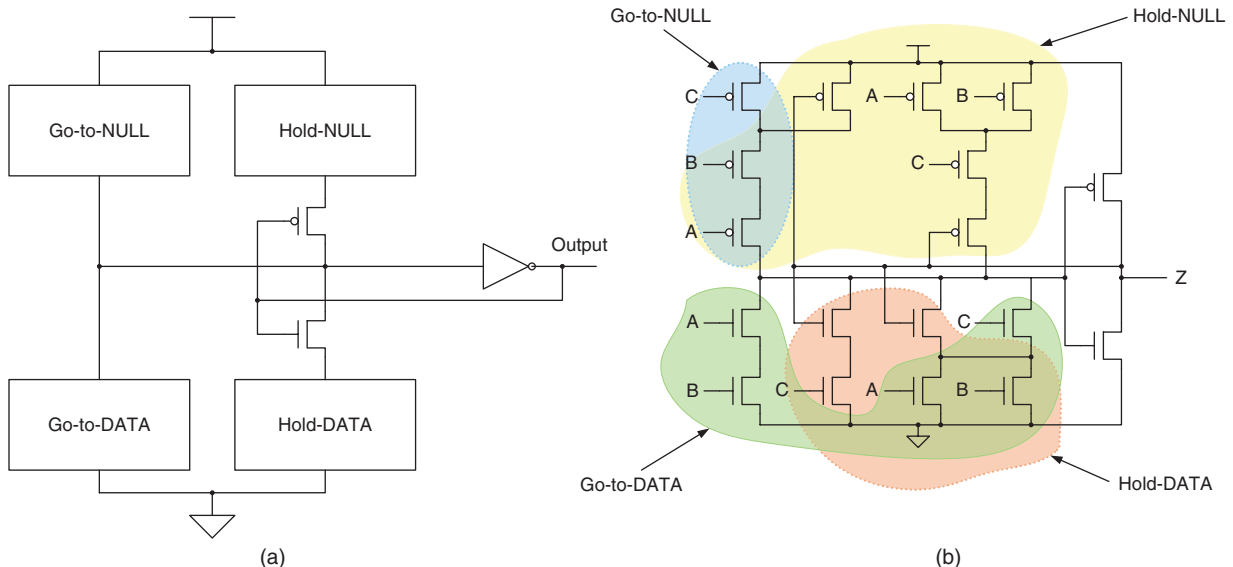


Fig. 5. (a) General circuit structure of NCL threshold gates (b) TH23.

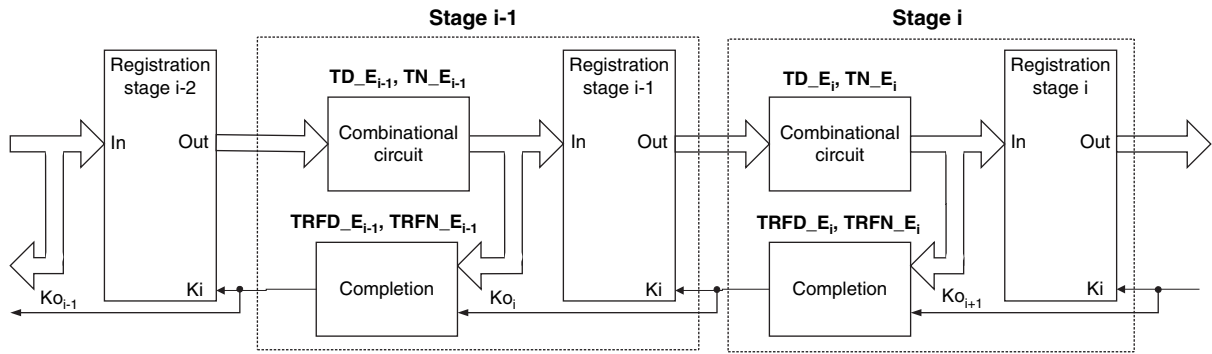


Fig. 6. NCL pipeline utilizing early completion.<sup>16</sup>

compromising quasi delay-insensitivity, since stage<sub>*i*</sub> will only be put to sleep when both its inputs are NULL and it is requesting NULL.

## 4. IMPLEMENTING MTCMOS IN NCL CIRCUITS

### 4.1. MTCMOS Threshold Gates

As stated before, incorporating MTCMOS architecture into each gate eases the sizing of sleep transistors. In contrast to synchronous MTCMOS gates, which incorporate large area overhead, for most NCL threshold gates, their MTCMOS

versions are actually smaller than the original designs. In fact, among the 27 threshold gates in the NCL logic family, only three gates have increased transistor count after modification from their original version to their MTCMOS version; all other 24 gates have reduced transistor counts except for one where the transistor count of the two versions is the same. There are two reasons for this area reduction. (1) NCL threshold gates are usually much larger and more powerful than Boolean gates. Table I presents the complete list of NCL threshold gates including the Boolean function for each threshold gate. Note that due to the hysteresis feature of NCL, the Boolean function only determines when the output of the gate will become logic 1. The

Table I. Original and MTCMOS threshold gates size comparison.

Threshold gate	Boolean function	Transistor count (original version)	Transistor count (MTCMOS version)	Area overhead
TH12	$A + B$	6	11	+5
TH22	$AB$	12	11	-1
TH13	$A + B + C$	8	13	+5
TH23	$AB + AC + BC$	18	17	-1
TH33	$ABC$	16	13	-3
TH23w2	$A + BC$	14	13	-1
TH33w2	$AB + AC$	14	13	-1
TH14	$A + B + C + D$	10	15	+5
TH24	$AB + AC + AD + BC + BD + CD$	26	24	-2
TH34	$ABC + ABD + ACD + BCD$	24	23	-1
TH44	$ABCD$	20	15	-5
TH24w2	$A + BC + BD + CD$	20	20	0
TH34w2	$AB + AC + AD + BCD$	22	21	-1
TH44w2	$ABC + ABD + ACD$	23	19	-4
TH34w3	$A + BCD$	18	15	-3
TH44w3	$AB + AC + AD$	16	15	-1
TH24w22	$A + B + CD$	16	15	-1
TH34w22	$AB + AC + AD + BC + BD$	22	19	-3
TH44w22	$AB + ACD + BCD$	22	20	-2
TH54w22	$ABC + ABD$	18	15	-3
TH34w32	$A + BC + BD$	17	15	-2
TH54w32	$AB + ACD$	20	15	-5
TH44w322	$AB + AC + AD + BC$	20	19	-1
TH54w322	$AB + AC + BCD$	21	19	-2
THxor0	$AB + CD$	20	15	-5
THand0	$AB + BC + AD$	19	17	-2
TH24comp	$AC + BC + AD + BD$	18	15	-3
Average transistor count per gate		18	16	-2

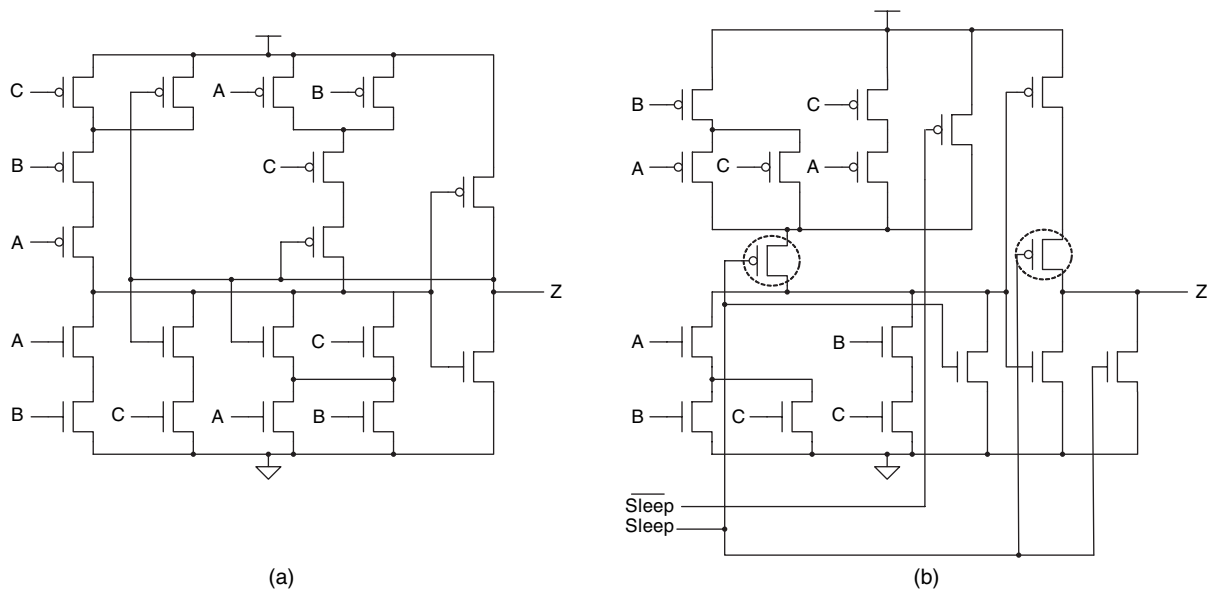


Fig. 7. Static TH23 gate (a) original version, (b) MTCMOS version (circled transistors are high- $V_t$ ).

output only switches to logic 0 when all inputs are logic 0. It is clear from Table I that NCL threshold gates are capable of implementing much more complex logic functions compared to Boolean gates. The average transistor count of the original threshold gates is 18. Compared to the total number of transistors in a threshold gate, the added MTCMOS transistors only occupy a small portion. (2) Among the four functional blocks of each threshold gate shown in Figure 5(a) (i.e., Go-to-NULL, Hold-NULL, Go-to-DATA, and Hold-DATA), the Go-to-NULL block can be omitted because all threshold gates will be forced to enter sleep mode after each DATA state, which has the same effect as undergoing a NULL state, i.e., the output becomes logic zero; the Hold-DATA block can also be omitted because all threshold gates in a stage will become logic 0 when put to sleep; hence, hysteresis is no longer needed.

Similar to the method in Ref. [12], a single high threshold PMOS transistor is inserted between the pull-up and pull-down networks, which are composed of low threshold transistors. Extra NMOS and PMOS transistors are added in parallel with the pull-up and pull-down networks to keep the voltage potential between the source node of the high threshold transistor and power supply ( $V_{DD}$ ), and the drain node of the high threshold transistor and ground, at equivalent levels to minimize leakage in sleep mode. In addition, most threshold gates contain a logic portion and an output/feedback inverter, which needs to incorporate the same MTCMOS circuit structure. The only difference is that the PMOS transistor connected in parallel with the pull-up network can be eliminated for the inverter since it provides very minimal leakage reduction in comparison to its added power consumption overhead. As an example, Figure 7 shows the original and MTCMOS versions of a TH23 gate. Table I shows the transistor count comparison of all threshold gates between their original and MTCMOS versions.

The fewer number of transistors in each threshold gate and the robust timing requirement of NCL largely ease the sleep transistor sizing. Tradeoffs exist between higher power for upsizing and longer delay for downsizing. A series of simulations has been performed to determine the optimal size of high threshold transistors. The technology used is the IBM 8RF-DM 0.13  $\mu\text{m}$  CMOS process, which has a designated  $V_{DD}$  of 1.2 V, and offers both high (0.47 V  $V_{sat}$  for NMOS and  $-0.465$  V  $V_{sat}$  for PMOS) and low threshold (0.155 V  $V_{sat}$  for NMOS and  $-0.275$  V  $V_{sat}$  for PMOS) transistors. Since NCL circuit behavior is not affected by different gate delays, all transistors are minimum-sized, i.e., 160 nm channel width, except the sleep transistors. Through thorough simulation, it was determined that each high threshold PMOS transistor should have a gate width of 220 nm. An example is shown in Figure 8.

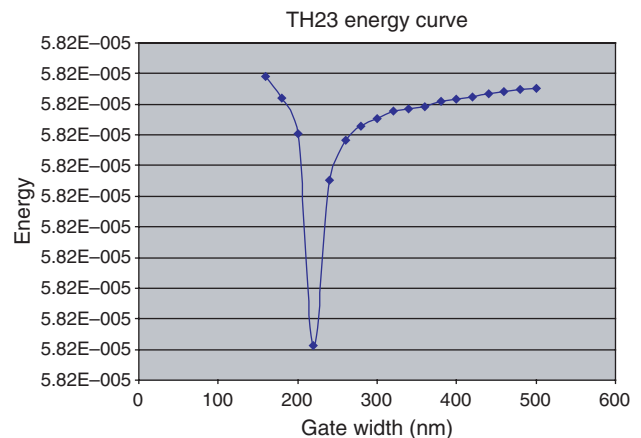
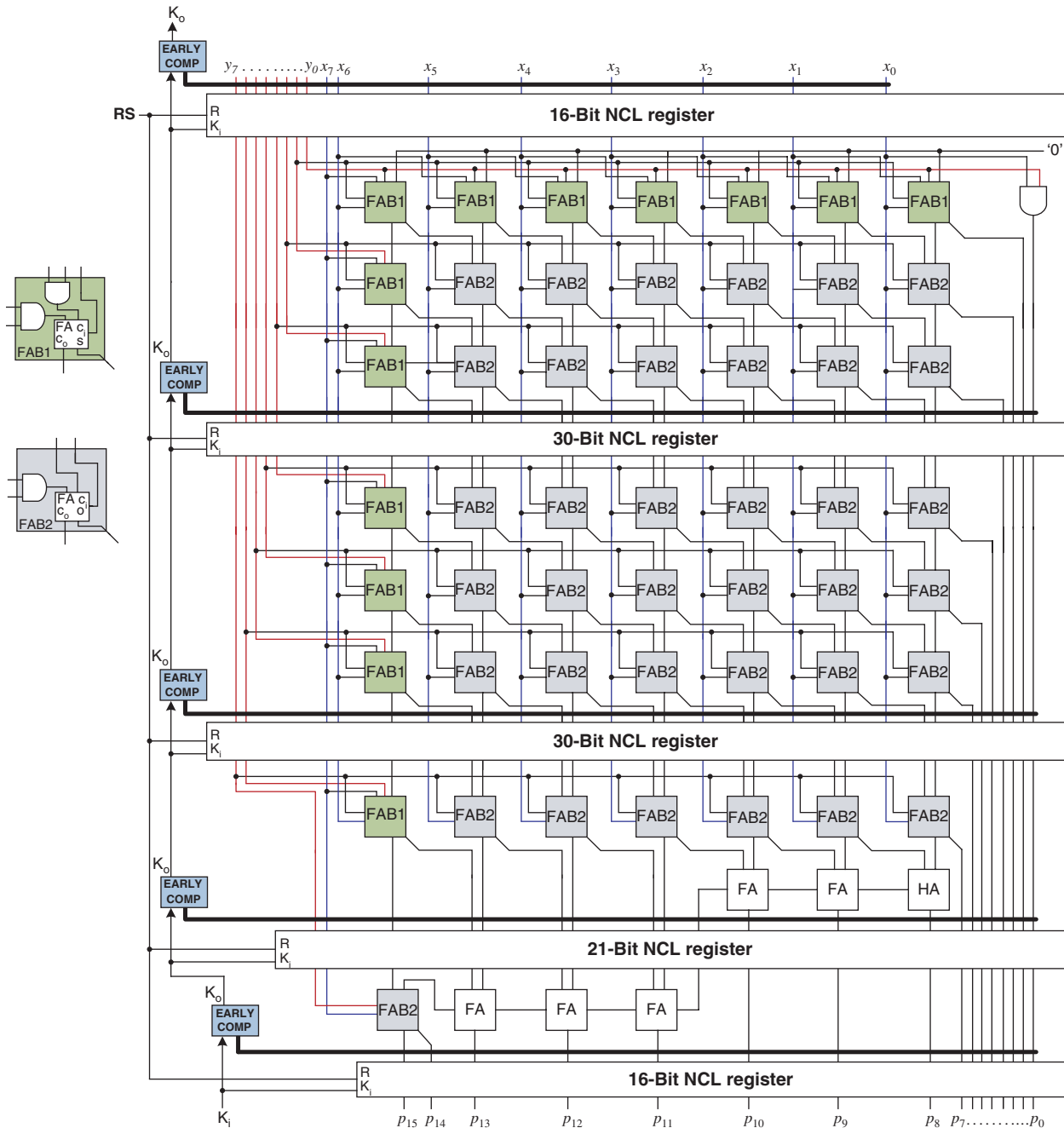


Fig. 8. Sleep transistor sizing results of TH23 gate.

### 4.2. Sleep Control

To build a MTCMOS NCL circuit, it is very important to consider how to generate the *Sleep* signal for each gate in order for the circuit to retain quasi delay-insensitivity. *Sleep* signal generation is a major concern for MTCMOS synchronous circuits, requiring additional logic with carefully analyzed timing relation to avoid glitches and potential circuit malfunction. However, for MTCMOS NCL circuits, it is very simple, or natural, to generate all *Sleep* signals. As shown in Figures 3 and 6, NCL pipeline stages

use handshaking signals to communicate with each other. These handshaking signals (i.e., completion detection or  $K_i$  and  $K_o$  signals) control the previous register of each pipeline stage to be able to pass either DATA or NULL. During a NULL cycle, the outputs of all threshold gates in the combinational logic block within a pipeline stage become logic 0; therefore, it is appropriate to force these gates to output logic 0 using the sleep mechanism, in order to reduce leakage power and improve performance. The corresponding completion detection signal naturally serves as the inverted *Sleep* signal to the NCL gates, without



requiring any additional sleep generation hardware, except for extra *Sleep* signal buffers, which are also required in synchronous MTCMOS circuits. Moreover, due to the delay-insensitive nature of NCL circuits, there is no timing requirement as to what order the completion detection signal arrives at the previous register versus the combinational logic block. However, isochronic forks<sup>17, 18</sup> must be assumed in the *Sleep* signal buffer distribution network in order for the QDI property to remain valid. Therefore, hardware overhead as well as timing analysis requirements are greatly reduced when applying MTCMOS to NCL versus synchronous circuits. Additionally, NCL circuits do not lose data during sleep mode, since sleep mode is applied in lieu of the NULL cycle, which has the same effect, causing all gates to return to zero. Note that the data has already been latched by the subsequent register prior to the NULL cycle/sleep mode. Thus, the two remaining drawbacks of synchronous MTCMOS circuits, i.e., *Sleep* signal generation and data loss, are eliminated.

## 5. DESIGN EXAMPLES, RESULTS, AND ANALYSIS

To evaluate the effectiveness of the MTCMOS NCL methodology in terms of leakage power reduction, three  $8 \times 8$  pipelined unsigned array multipliers were designed using the IBM 8RF-DM 0.13  $\mu\text{m}$  CMOS process: synchronous low- $V_t$  design, original NCL low- $V_t$  design, and MTCMOS NCL design. Buffers for the *Sleep* signals

have been included in each design. All transistor-level circuit schematics were designed in Cadence Virtuoso. Simulations were performed using the Spectre simulator for accurate timing and power analysis. Measurements include data for supply voltage scaling, leakage and active power/energy, and delay.

### 5.1. Synchronous Array Multiplier

The designed synchronous array multiplier is an  $8 \times 8$  array multiplier pipelined into 4 stages. The array consists of standard static CMOS full adders and AND gates. Global clock and reset signals are provided to each register block. The circuit structure is the same as the classic array multiplier.<sup>19</sup>

### 5.2. Original NCL Array Multiplier

Given that NCL utilizes dual-rail encoding for each signal, the NCL array multiplier is designed in such a way that fewer and shorter wires between the logic blocks are needed. As shown in Figure 9, all NCL AND gates that generate  $x_i y_i$  are embedded inside their corresponding full adder logic blocks, FBA1 and FBA2. The design is pipelined into four stages by inserting NCL registers after every three FBA rows, except for the last two rows which are arranged differently in order to balance the latency between stages. The circuit implements the Early Completion scheme introduced in Section 3.3 in every stage to achieve better performance. Figure 10 shows an

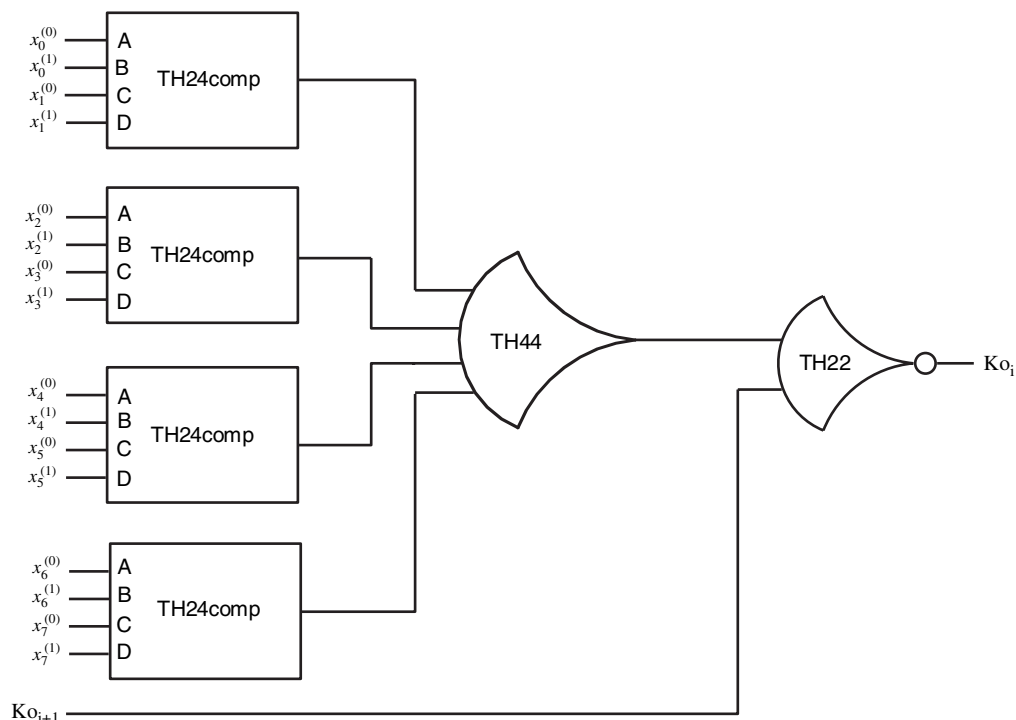


Fig. 10. Early completion block example.

example of early completion blocks for an 8-bit dual-rail datapath. Every TH24comp gate has the Boolean function  $(A + B) \cdot (C + D)$  to detect whether both inputs are DATA or NULL. When all circuit inputs are NULL and the

subsequent stage early completion block is also requesting NULL, the circuit requests DATA; and when all inputs are DATA and the subsequent stage early completion block is also requesting DATA, the circuit requests NULL.



Fig. 11. 8 × 8 MTCMOS NCL array multiplier.

### 5.3. MTCMOS NCL Array Multiplier

The circuit structure of the MTCMOS NCL array multiplier is shown in Figure 11. The circuit is similar to the original NCL version except that all combinational logic blocks between the NCL registers are built using MTCMOS threshold gates. For each pipeline stage, when all register inputs are NULL and the subsequent stage is requesting NULL, all threshold gates of the combinational logic circuit in the subsequent stage are put to sleep and become NULL, until all register inputs become DATA and the subsequent stage requests DATA. Early Completion blocks are used to assert the *Sleep* signals for the MTCMOS logic elements. As stated before, this *Sleep* signal generation mechanism in MTCMOS NCL circuits does not require any complex timing analysis or additional hardware units as in the case of MTCMOS synchronous designs. Moreover, since the logic elements in each stage are forced to sleep only when they are going to undergo a NULL cycle, no data is lost during sleep mode.

## 5.4. Results and Analysis

### 5.4.1. Supply Voltage Scalability Comparison

Supply voltage scalability of each circuit is evaluated by decreasing  $V_{DD}$  while increasing data/clock pulse width, until the outputs of the circuit become erroneous. Exhaustive simulations demonstrate that the synchronous low- $V_t$  multiplier is able to function correctly at 0.15 V  $V_{DD}$  at the lowest, while the original NCL low- $V_t$  design is able to operate at 0.11 V  $V_{DD}$ . This is because of the flexible timing requirement of quasi delay-insensitive asynchronous circuits, which makes them more robust to transistor speed variations due to different  $V_{DD}$ . The MTCMOS NCL circuit functions at 0.25 V  $V_{DD}$ . This is due to the fact that the MTCMOS NCL threshold gates contain multiple high- $V_t$  transistors. However, if the sleep transistor size is enlarged, the MTCMOS NCL circuit is able to operate correctly at lower supply voltages.

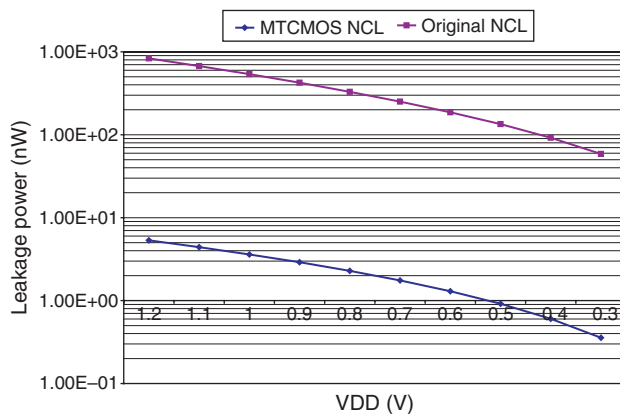


Fig. 12. Leakage power comparison between MTCMOS and original NCL circuits.

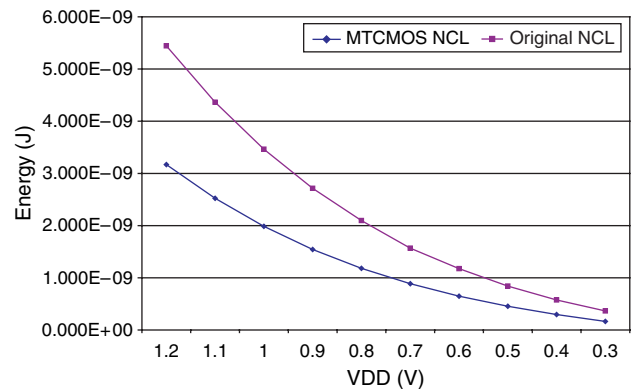


Fig. 13. Active energy comparison between MTCMOS and original NCL circuits.

### 5.4.2. Leakage, Active, and Total Power/Energy Comparison

The two NCL multipliers are simulated to evaluate the effectiveness of the MTCMOS NCL methodology in reducing power/energy consumption. Four parameters, namely, leakage power, active energy, total average energy, and delay, are simulated, recorded, and compared.

Figure 12 shows the leakage power comparison. The leakage power is calculated while providing NULL to all circuit primary inputs for both designs, thus activating the *Sleep* signals of the MTCMOS NCL circuit. From Figure 12, it is clear that the MTCMOS NCL circuit has significant leakage reduction compared to its original counterpart at all supply voltage testing points. The differences are from 143 $\times$  to 164 $\times$ , with an average of 150 $\times$ . This is due to the MTCMOS circuit structure in each threshold gate, and the reduced total transistor count. Note that the Y-axis of Figure 12 is logarithm-scale.

Figure 13 shows the comparison of active energy, which is measured by integrating the supply current over the time period during which each circuit undergoes a full DATA cycle, and multiplying by the corresponding  $V_{DD}$ . As shown in Figure 13, the MTCMOS NCL design still outperforms

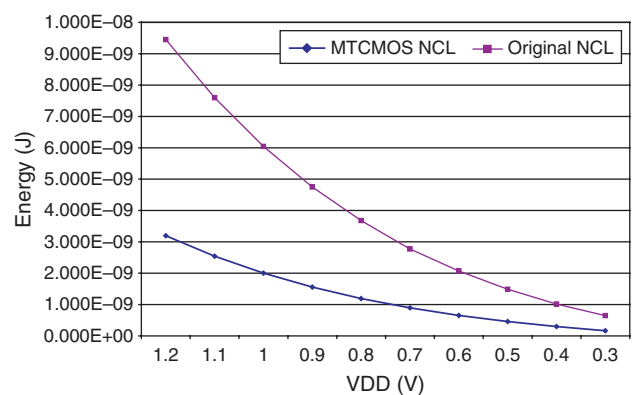
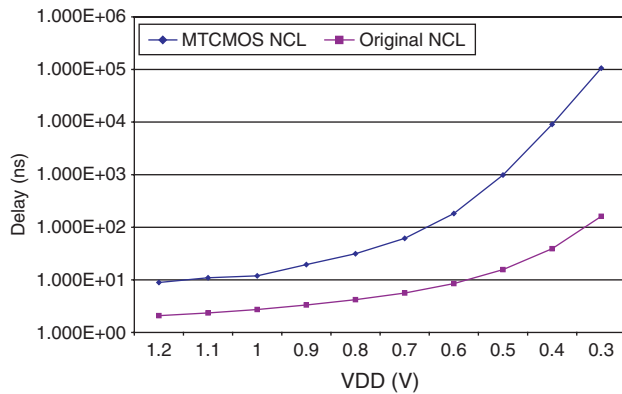


Fig. 14. Total energy comparison between MTCMOS and original NCL circuits.



**Fig. 15.** Delay comparison between MTCMOS and original NCL circuits.

its original NCL counterpart at each testing point, with differences between  $1.3\times$  to  $2.2\times$ . The average difference is  $1.8\times$ . This is due to the reduced transistor count in each threshold gate.

As a pseudo-comprehensive comparison, Figure 14 shows the data of total energy consumption, which is calculated by integrating the supply current over the same time period during which each circuit undergoes a full DATA-NULL cycle, and multiplying by the corresponding  $V_{DD}$ . The length of the NULL cycle (the sleep time period of the MTCMOS NCL design) is the same as the DATA cycle. As can be seen in Figure 14, since both leakage power and active energy of the MTCMOS NCL design are smaller than those of the original NCL design, total energy reduction between  $3.0\times$  to  $3.9\times$  with an average of  $3.2\times$  are achieved.

Figure 15 shows the delay comparison. As expected, the MTCMOS NCL circuit is slower than the original NCL version at all testing points. This is due to the use of high- $V_t$  transistors in the MTCMOS NCL circuit, which are slower than the low- $V_t$  ones in the original NCL circuit.

## 6. CONCLUSION

This paper introduces an ultra-low power circuit design methodology combining the MTCMOS technique with quasi delay-insensitive asynchronous logic, in order to solve the three major problems of synchronous MTCMOS circuits. In the resulting MTCMOS NCL circuits: (1) the completion detection handshaking signals are directly used as the *Sleep* signals, such that no additional hardware is required; (2) circuit stages are put to sleep in lieu of the NULL cycle, both of which cause all gates in the stage to become logic 0; hence, operation remains the same and no data is lost; (3) the NCL gate library is modified by adding a *Sleep* signal to each gate, such that transistor sizing is done for each NCL gate, and is no longer circuit dependent. In addition, QDI asynchronous MTCMOS circuits are usually smaller than their original versions, causing zero or even negative area overhead. Moreover, the

quasi delay-insensitive nature of NCL circuits results in flexible timing requirements, which yields increased circuit robustness and allows for extreme supply voltage scaling to subthreshold operation for further power reduction, without requiring any circuit modifications. The proposed MTCMOS NCL methodology was compared to the original NCL methodologies showing  $150\times$  and  $1.8\times$  leakage power and active energy reduction compared to the original NCL version, respectively. Future work includes optimizing threshold gate designs to further reduce leakage current, incorporating transistors with three or more threshold voltages to minimize energy consumption, and analyzing the possible use of ternary logic to decrease area.

## References and Notes

1. B. Deepaksubramanian and A. Nunez, Analysis of subthreshold leakage reduction in CMOS digital circuits. *50th Midwest Symposium on Circuits and Systems*, August (2007), pp. 1400–1404.
2. H. Kawaguchi, K. Nose, and T. Sakurai, A super cut-off CMOS (SCCMOS) scheme for 0.5-V supply voltage with picoampere stand-by current. *IEEE Journal of Solid-State Circuits* 35 (2000).
3. A. Chandrakasan, W. J. Bowhill, and F. Fox, Design of High-Performance Microprocessor Circuits, Wiley-IEEE Press (2000).
4. J. C. Park and V. J. Mooney III, Sleepy stack leakage reduction. *IEEE Transactions on VLSI* 14, 1250 (2006).
5. A. Abdollahi, F. Fallah, and M. Pedram, Leakage current reduction in CMOS VLSI circuits by input vector control. *IEEE Transactions on VLSI* 12, 140 (2004).
6. A. Wang and A. Chandrakasan, Optimal supply and threshold scaling for subthreshold CMOS circuits. *IEEE Computer Society Annual Symposium on VLSI* (2002).
7. H. Soeleman, K. Roy, and B. Paul, Robust subthreshold logic for ultra-low power operation. *IEEE Transactions on VLSI* 9, 90 (2001).
8. 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 variations on microprocessor frequency and leakage. *IEEE Journal of Solid-State Circuits* 37, 1396 (2002).
9. H. Kim and K. Roy, Ultra-low power DLMS adaptive filter for hearing aid applications. *International Symposium on Low Power Electronics and Design* (2001).
10. S. Mutoh, T. Douseki, Y. Matsuya, T. Aoki, S. Shigematsu, and J. Yamada, 1-V power supply high-speed digital circuit technology with multithreshold-voltage CMOS. *IEEE Journal of Solid-State Circuits* 30, 847 (1995).
11. VSDCAD J. T. Kao, and A. P. Chandrakasan, Dual-threshold voltage techniques for low-power digital circuits. *IEEE Journal of Solid-State Circuits* 35, 1009 (2000).
12. P. Lakshmikanthan, K. Sahni, and A. Nunez, Design of ultra-low power combinational standard library cells using a novel leakage reduction methodology. *IEEE International SoC Conference* (2006).
13. S. C. Smith, R. F. DeMara, J. S. Yuan, D. Ferguson, and D. Lamb, Optimization of NULL convention self-timed circuits. *Elsevier's Integration, The VLSI Journal* 37/3, 135 (2004).
14. G. E. Sobelman and K. M. Fant, CMOS circuit design of threshold gates with hysteresis. *IEEE International Symposium on Circuits and Systems (II)* (1998), pp. 61–65.
15. 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* (1996), pp. 261–263.

16. S. C. Smith, Speedup of self-timed digital systems using early completion. *The IEEE Computer Society Annual Symposium on VLSI*, April (2002), pp. 107–113.
17. A. J. Martin, Programming in VLSI: From communicating processes to delay-insensitive circuits. *Developments in Concurrency and Communication, UT Year of Programming* Institute on Concurrent Programming, Addison-Wesley (1990), pp. 1–64.
18. K. Van Berkel, Beware the Isochronic Fork. *Integration, the VLSI Journal* 13/2, 103 (1992).
19. B. Parhami, *Computer Arithmetic: Algorithms and Hardware Designs*, Oxford University Press (1999).

### **Andrew Bailey**

Andrew Bailey received B.S. degree in Computer Engineering from the University of Arkansas in 2007. He is now working towards his M.S. degree in Computer Engineering in the Computer Science and Computer Engineering department of the University of Arkansas. His research interests include ultra-low power digital design, multi-threshold digital circuits, and asynchronous digital systems.

### **Ahmad Al Zahrani**

Ahmad Al Zahrani received his B.S. degree from the Electrical and Computer Engineering department of Umm Al-Qura University, Saudi Arabia, in 2007. He is now working towards his M.S. degree in Computer Engineering in the Computer Science and Computer Engineering department of the University of Arkansas. His research interests include digital design, computer architecture, parallel computation, and hardware encryption.

### **Guoyuan Fu**

Guoyuan Fu received his B.S. in Physics in July 1990 and a M.S. in Optoelectronic Engineering in June 1999 from Chongqing University, P. R. China. He received a M.S. in Electrical Engineering in August 2007 and a Ph.D. in Microelectronics-Photonics in December 2007 from University of Arkansas. His research interests include photorefractive effects in semiconductor materials, space electronics, ultra-low power circuit designs, and NCL circuit designs.

### **Jia Di**

Jia Di received his B.S. and M.S. degrees from Tsinghua University, Beijing, P. R. China, in 1997 and 2000, respectively, and his Ph.D. in Electrical and Computer Engineering from the University of Central Florida in the year of 2004. He is currently an Assistant Professor in the Computer Science and Computer Engineering department of the University of Arkansas. His research interests include asynchronous logic, low power digital circuits and systems design, power estimation, FPGA synthesis, system security, and ASIC.

### **Scott Smith**

Scott Smith received his B.S. degrees in Electrical Engineering and Computer Engineering, and M.S. degree in Electrical Engineering, from University of Missouri-Columbia in 1996 and 1998, respectively, and his Ph.D. in Computer Engineering from the University of Central Florida in 2001. He is currently an Associate Professor in the Electrical Engineering department of the University of Arkansas. His research interests include NULL Convention Logic, asynchronous logic, embedded systems, and FPGAs.