

# Reconfigurable magnetoelectronic circuits for threshold logic

Nicholas P. Carter<sup>\*,†</sup> and Steve P. Ferrera

*Center for Reliable and High-Performance Computing, Department of Electrical and Computer Engineering,  
University of Illinois at Urbana-Champaign, 1308 W. Main St. Urbana, IL 61801, U.S.A.*

## SUMMARY

Magnetoelectronic devices, which combine ferromagnetic materials with conventional silicon structures, offer the potential to add non-volatile storage to electronic systems, eliminating their vulnerability to data loss due to power supply interruptions. We present a set of circuits, based on the Hybrid Hall Effect device, that combine logic with non-volatile storage. These circuits can be configured on a cycle-by-cycle basis to compute different functions of their inputs, store their outputs indefinitely even in the absence of system power, and can be easily integrated into CMOS systems to provide non-volatile operation without requiring additional supply voltages or other global signals. They exploit the properties of the Hybrid Hall Effect device to efficiently implement threshold logic functions and thereby reduce the number of gates required to implement most Boolean expressions.

In this paper, we describe our reconfigurable magnetoelectronic circuits and the interfaces that make them compatible with CMOS systems. The results presented are based on data from fabricated experimental devices, and we discuss how they can be expected to improve as devices scale to nanometer dimensions. Finally, we consider how magnetoelectronic circuits might be integrated into system designs to deliver high performance while tolerating power supply interruptions. Copyright © 2004 John Wiley & Sons, Ltd.

KEY WORDS: magnetoelectronics; reconfigurable logic; threshold logic; non-volatile circuits

## 1. INTRODUCTION

*Magnetoelectronic* devices, which combine ferromagnetic materials with semiconductor structures, have a significant advantage over more-conventional electronic devices: they can retain state without power. When exposed to a sufficiently-large magnetic field, the atoms in a ferromagnetic material align themselves with the field, causing the material to develop a magnetic field of its own. Once magnetized in a given direction, the material will retain its magnetization state indefinitely unless a differently-oriented magnetic field of sufficient strength is applied to the device. This magnetization process can be repeated many times without

---

\*Correspondence to: N. P. Carter, Center for Reliable and High-Performance Computing, Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, 1308 W. Main St. Urbana, IL 61801, U.S.A.

†E-mail: npcarter@crhc.uiuc.edu

Contract/grant sponsor: Office of Naval Research; contract/grant number: N00014-02-1-1038

degrading the material, making magnetoelectronic devices very attractive for applications in which state changes frequently.

A number of non-volatile magnetoelectronic memory devices have been studied in recent years [1, 2], all of which use the magnetization state of a ferromagnetic element to store data. To read the state of their ferromagnetic elements, many of these devices rely on the giant magnetoresistance (GMR) effect, in which the resistance through a structure containing multiple magnetic materials changes depending on whether the magnetic materials are magnetized in the same or opposite directions, while others make use of the Hall effect, in which a current passing through a magnetic field generates a voltage perpendicular to both the current and the field. Magnetic memories (MRAMs) based on these devices have been used in the design of systems that must operate in high-radiation environments, and MRAMs with capacities and power consumption comparable to commercial DRAMs are expected to become widely available in the near future.

In this paper, we describe a set of reconfigurable logic circuits based on a novel magnetoelectronic circuit element: the hybrid Hall effect (HHE) device [3, 4]. Unlike previous magnetoelectronic systems, HHE-based circuits combine logic and storage, computing a configurable function of their inputs and retaining their output values indefinitely even in the absence of external power. These circuits compute *threshold* logic functions of their inputs, a superset of AND/OR logic, and are CMOS-compatible in both their interfaces and their fabrication, allowing them to be added to conventional semiconductor systems without redesigning the entire system.

The body of this paper begins with a description of the HHE device and a discussion of its expected scaling with improvements in fabrication technology. We then present our reconfigurable circuits and discuss how they might be used in system-level designs. Finally, we describe related and future work, and conclude.

## 2. THE HYBRID HALL EFFECT DEVICE

Figure 1 shows an atomic force micrograph of a hybrid Hall effect device that was fabricated at the Naval Research Laboratory [3, 4]. The HHE device consists of a  $2.2\mu \times 0.5\mu$  ferromagnetic bar that is fabricated on top of a high-mobility InAs/GaAsSb mesa. Four insulating legs in the mesa, created by ion damage, create a Hall cross structure that can be thought of as two intersecting wires: a bias line and an output line. A write wire (not shown in the figure) runs over the ferromagnetic bar, perpendicular to the orientation of the bar, so that current flowing through the write wire induces a magnetic field parallel to the length of the bar.

Figure 2 illustrates the operation of an HHE device. Current  $I_{in}$  flowing through the write wire at the top of the figure induces a magnetic field  $M$  in the ferromagnetic bar at the middle of the figure. If the magnitude of the induced magnetic field exceeds a threshold value, it magnetizes the ferromagnetic bar in the direction of the field. Since the ferromagnetic bar will retain its magnetization state when the induced magnetic field is removed, it can be used to encode a bit of data, with magnetization in one direction representing a logical '0' and magnetization in the other direction representing a logical '1.' For simplicity, the figure shows only one input wire, although multiple-wire designs are possible. Our circuit designs assume two input wires per gate, one of which flows current in the direction that changes the output from '0' to '1,' while the other flows current in the opposite direction.

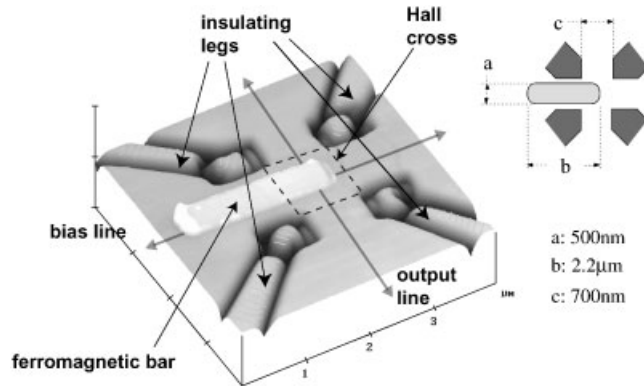


Figure 1. Atomic force micrograph of an HHE device (figure courtesy Mark Johnson, Naval Research Lab).

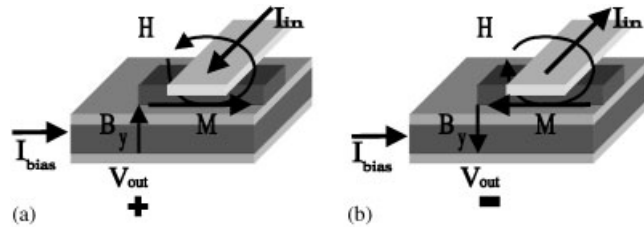


Figure 2. HHE device physical structure showing the two possible magnetization states  $M$ : (a) right magnetization state; and (b) left magnetization state.

Once the ferromagnetic bar has been magnetized, it generates a magnetic field of its own. At the left edge of the bar, this field is mostly vertical, with the direction of the field (up or down) being determined by the magnetization state of the bar. To read the state of the HHE device, a bias current  $I_{\text{bias}}$  is passed through the high-mobility mesa, parallel or anti-parallel to the magnetization direction of the bar. When this current intersects the vertical magnetic field at the edge of the ferromagnetic bar, the Hall effect [5] induces a voltage  $V_{\text{out}}$  perpendicular to the direction of current flow. The sign of this voltage is determined by the direction of the magnetic field, and the magnitude of the voltage is linearly related to the magnitude of  $I_{\text{bias}}$  by the *Hall resistance*, which is a function of the device structure and the materials used.

### 2.1. Device characteristics and scaling

The parameters of an HHE device that are of greatest interest to circuit designers are the device's physical size, the input current required to change the magnetization state of the ferromagnetic element, the device's Hall resistance, and the time required to perform an operation. The speed of HHE devices is limited by the time required to change the magnetization state of their ferromagnetic elements. Prototype devices have operated correctly with cycle times of 2 ns, and future devices are expected to operate at Gigahertz rates.

To date, the size of HHE devices has been limited by the quality of the available fabrication equipment, not the inherent physics of the device. For example, the results we present are based on the measured characteristics of the device shown in Figure 1, which was approximately four microns on a side and incorporated a  $2.2\mu \times 0.5\mu$  ferromagnetic element. Given access to state-of-the-art and future fabrication processes, HHE devices a few nanometers on a side should be feasible, subject to the limitations that the ferromagnetic bar must be large enough to display the proper hysteresis behavior, and the write wire must be large enough to carry the required input currents without exceeding the electromigration limits of the fabrication process.

Input current is a significant issue in most magnetoelectronic devices, and HHE gates are no exception. The devices used in this study require input currents of approximately 10 mA to change the magnetization state of their ferromagnetic elements, which is similar to the current levels required by other ferromagnetic devices [2]. Fortunately, the input current required by an HHE device is strongly affected by the size of the ferromagnetic element, and improves as the device shrinks. For example, an earlier HHE device, whose ferromagnetic bar measured  $1.5\mu \times 7.5\mu$ , required 100 mA of input current to change its state [4], a factor of 10 greater than the current required by the devices used in this paper. This input current scaling is expected to continue as devices shrink, further reducing power consumption.

An HHE device's Hall resistance is also an extremely important parameter, as it determines the amount of bias current required to produce a given output voltage swing. Prototype hybrid Hall effect devices have had Hall resistances of approximately 10 ohms, requiring significant bias currents to generate CMOS-compatible output voltages. For this reason, our designs use an SRAM cell as an output amplifier, reducing the amount of current required and the duration over which this current must flow. Researchers [6] are also experimenting with HHE devices that incorporate a metal gate between the ferromagnetic bar and the conductor that carries the bias current. If the proper voltage is applied to this gate, the Hall resistance of the device is expected to increase to 10–100 K $\Omega$ , leading to proportional decreases in bias currents. These same researchers are fabricating HHE devices on CMOS substrates, making them more practical for integrated circuit applications.

Overall, the characteristics of HHE devices are expected to improve significantly with advances in fabrication process and device design. In particular, reductions in input current levels and increased Hall resistance should greatly reduce power consumption, making integration of HHE gates into large systems more practical.

## 2.2. HHE circuit designs

The initial difficulty that arises in the design of HHE-based circuits is input–output compatibility. As described above, an HHE device takes currents as inputs, but generates an output voltage, requiring some amount of support circuitry between each pair of HHE devices to convert between these two types of signals. In our previous work [7, 8], we examined a number of designs for this interface before developing the structure shown in Figure 3.

Rather than translating directly between the output voltage of the HHE gate and the required input currents, this design takes CMOS-compatible input voltages and generates CMOS output levels to support integration of HHE-based circuits into CMOS designs. In the example input circuitry shown in the left half of the figure, the three pull-down transistors are sized so that two or more of them must be turned on for enough current to flow to magnetize the

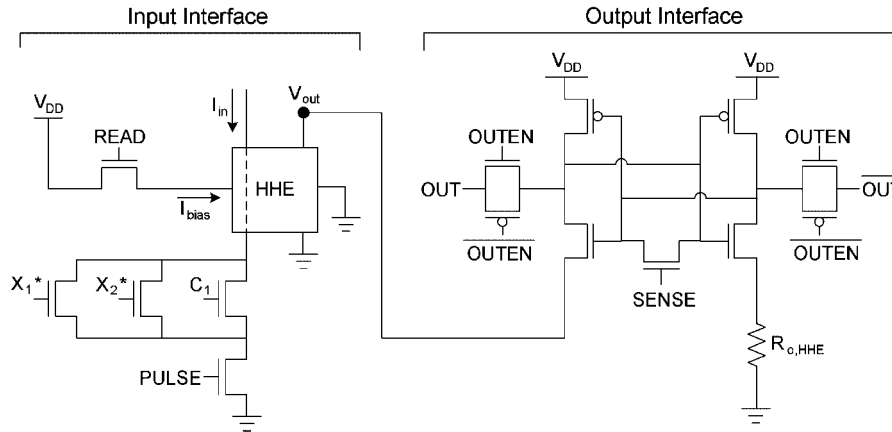


Figure 3. CMOS-compatible input and output interfaces.

ferromagnetic element and set the state of the device to '1'. This allows the  $C_1$  input to control whether the gate computes the AND or the OR of the  $X_1$  and  $X_2$  inputs. A complimentary pull-up chain (not shown in the figure) allows enough current to flow in the opposite direction through another input wire to set the state of the device to '0' when appropriate. To reduce power consumption, the PULSE input limits current flow to a short period after the start of each clock cycle. In many of our designs, such as the one shown in Figure 5, the output of the gate or its complement are ANDed with the PULSE signal so that current only flows through the input wire that could change the state of the device, further reducing power consumption.

The output interface shown in the right half of the figure both latches and amplifies the output of the HHE gate to reduce the duration of the bias current and its amplitude. The interface is a cross-coupled inverter pair, with the HHE device connected between the NFET of one inverter and ground, while the other inverter's NFET connects to ground through a resistor sized to match the output resistance of the HHE gate. To latch the output of the gate, the SENSE signal is asserted to bring the cross-coupled inverter pair into its metastable state, and the READ signal is asserted to start a bias current flowing through the HHE device. Once the bias current begins flowing, the output voltage of the HHE device becomes positive or negative, depending on the state of the ferromagnetic element. SENSE is then de-asserted, and the cross-coupled inverter pair assumes one of its two stable states, depending on the output of the HHE device, at which point READ is de-asserted to turn off the bias current. The output enable (OUTEN) input serves to isolate the cross-coupled inverter pair from the remainder of the circuit while it is in its metastable state, preventing other devices from affecting the output of the gate.

These input and output interfaces make it possible to connect HHE-based gates with each other and with other CMOS circuits. The control inputs to the interface logic (READ, PULSE, SENSE, and OUTEN) can be generated off of a standard two-phase clock using a small number of inverters as delay buffers, and can be shared across multiple HHE gates to reduce their area impact. Detailed designs for the logic that generates these control signals are presented in Reference [8].

## 3. RECONFIGURABLE THRESHOLD LOGIC

HHE-based reconfigurable gates were first proposed in Reference [7], where we presented a set of reconfigurable AND/OR/NAND/NOR gates that could be configured on a cycle-by-cycle basis to compute different functions of their inputs. However, the HHE device can be used to implement a much wider range of logic functions, in particular *threshold* logic functions. Threshold logic [9] is a generalization of AND/OR logic in which at least  $T$  inputs of an  $N$ -input gate must be logic 1 for the gate's output to be logic 1. Threshold logic is a strict superset of AND/OR logic, as an  $N$ -input AND gate can be implemented as an  $N$ -input threshold gate where  $T = N$ , and an  $N$ -input OR gate can be implemented using an  $N$ -input threshold gate where  $T = 1$ . Because HHE gates inherently change state when the magnitude of their input current(s) exceeds a threshold value, HHE-based threshold logic circuits are quite simple, requiring only  $2 * \text{Log}_2(N)$  more transistors to implement an  $N$ -input threshold logic gate than are required to implement an  $N$ -input HHE AND/OR gate.

Figure 4 shows a block diagram of a four-input reconfigurable threshold gate. In this diagram, inputs  $C_2$  and  $C_1$  are interpreted as a two-bit configuration value that sets the threshold of the logic gate, whose output is '1' if the number of inputs  $X_{1-4}$  that are asserted plus the value of  $C_{2,1}$  is four or more. The capabilities of this gate can be further extended by adding circuitry that can be configured to selectively invert, ground, or tie to  $V_{DD}$  each of the inputs, allowing the gate to compute inverted threshold functions and/or act as a gate with fewer than four inputs.

An HHE-based implementation of this gate is shown in Figure 5. This design assumes an HHE device with two input wires, as described earlier. An output interface circuit like the one shown in Figure 3 is also assumed, but has been left out of the figure to highlight the details of the threshold logic. In this figure, current flows through the left-hand input wire only in the direction that will set the output of the gate to '1,' and the right-hand input wire flows current in the opposite direction. As described earlier, the output of the gate and the PULSE signal are used to ensure that current only flows through the input wire that could change the gate's output and to limit the duration of current flow through that wire.

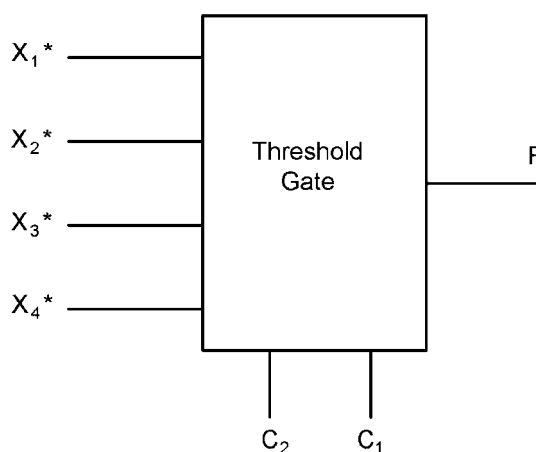


Figure 4. Threshold gate block diagram.

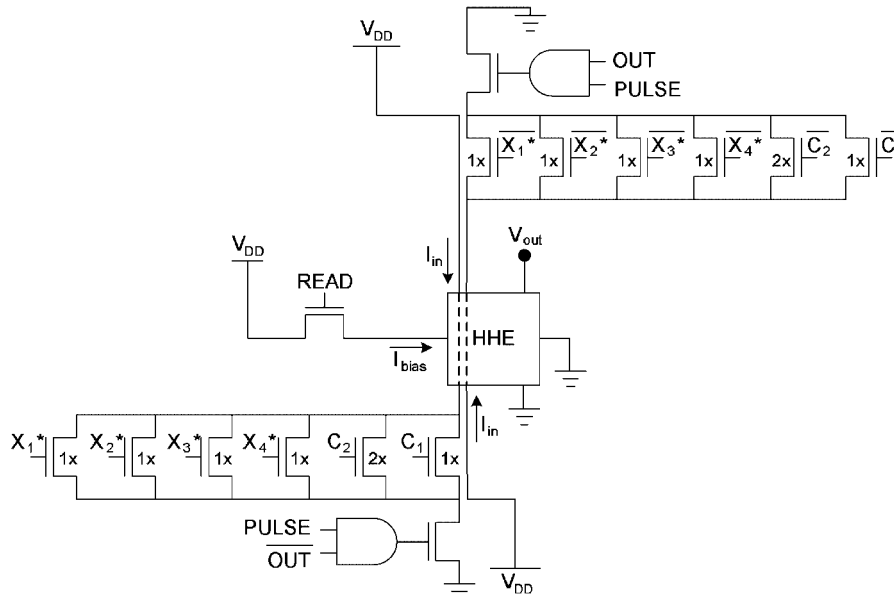


Figure 5. Reconfigurable threshold gate.

To implement threshold logic, the transistors connected to the  $X_{1-4}$  inputs and their complements are sized to flow slightly more than 25% of the current required to set the magnetization state of the HHE device. The transistors connected to the  $C_1$  input and its complement are equally sized, while the transistors connected to the  $C_2$  input are twice as large. Thus, the left-hand pull-down chain conducts enough current to set the output of the gate to logic 1 if  $X_1 + X_2 + X_3 + X_4 + 2 * C_2 + C_1 \geq 4$ , and the right-hand chain conducts enough current to set the output to logic 0 if the inverse is true.

This circuit can be generalized to larger numbers of inputs by adding additional input transistors and appropriately-sized configuration transistors in parallel with the ones in the existing pull-down chains, although the maximum number of inputs that a gate can support will be limited by fabrication variations. As the number of inputs increases, the margin between the amount of current that flows through an input wire when  $T - 1$  inputs are asserted and the amount that flows when  $T$  inputs are asserted (for a gate with threshold  $T$ ) shrinks, eventually becoming less than the expected variation in current between different transistors of the same drawn size on different chips. While the exact number of inputs that can be supported will be a function of the fabrication process, we estimate that gates with 15 or more inputs are likely to be feasible, because the variation in drive current between different transistors of the size required by these designs is typically a relatively small fraction of the expected current. In Reference [8], we present extensions to our designs that help to reduce the impact of fabrication variations, allowing the implementation of gates with large numbers of inputs.

### 3.1. A reconfigurable threshold logic macrocell

In AND/OR logic, complex Boolean expressions are commonly expressed in sum-of-products (SOP) form, which allows any Boolean function to be implemented in two levels of logic

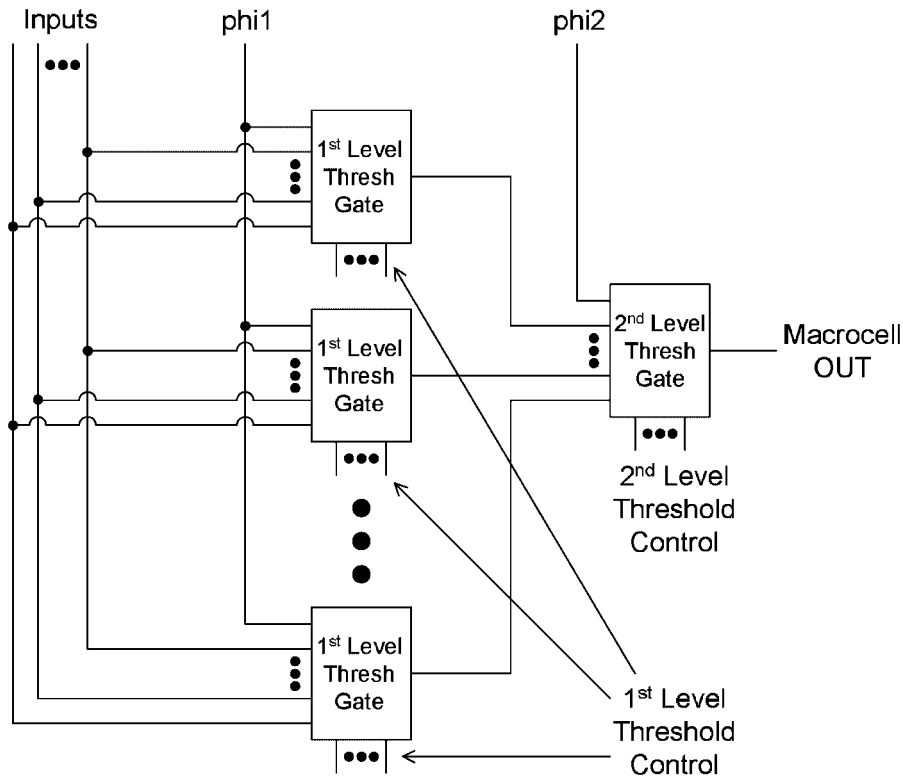


Figure 6. Two-level threshold macrocell.

if the gates at each level support enough inputs. Threshold logic gates can be connected in a similar two-level structure, as shown in Figure 6, although the increased flexibility of threshold logic gates often allows functions to be implemented in fewer gates than is possible in AND/OR logic. This figure illustrates the design of a two-level threshold logic macrocell, in which each gate can be configured to implement a different threshold logic function of its (potentially inverted) inputs. Each gate in the macrocell generates its control inputs off of the rising and falling edges of a clock input, and the clock inputs to each level of the macrocell are offset sufficiently to allow data to propagate through the first level of the macrocell before the inputs to the second level are inverted.

### 3.2. Comparison with AND/OR logic

To illustrate the benefits of threshold logic, we used the LSAT [10] and espresso [11] synthesis tools to generate two-level implementations of all possible functions of four input variables in both threshold logic and AND/OR logic. Figure 7 shows the results. On average, approximately one fewer gate is required to implement each function in threshold logic than in AND/OR logic, with XOR functions showing the largest reductions in gate count.

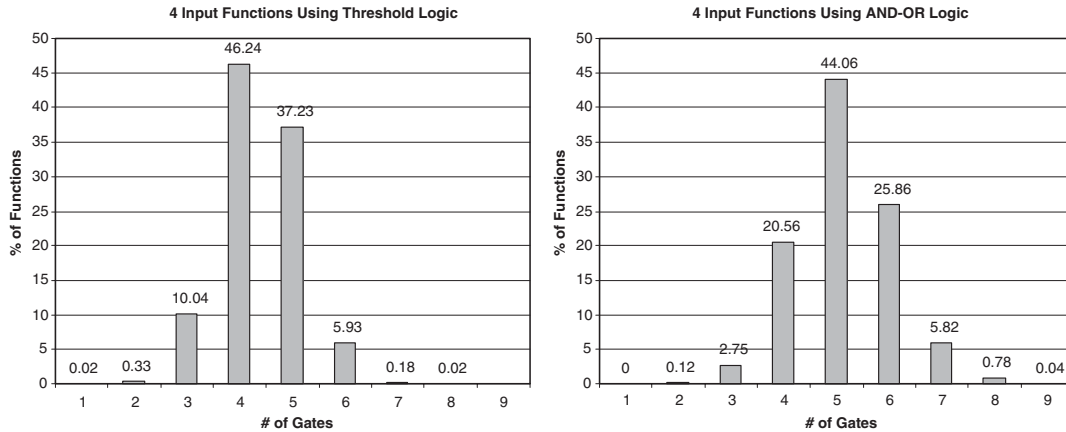


Figure 7. Number of gates required to implement four-input functions in threshold and AND/OR logic.

### 3.3. Simulation results

Using the HSPICE circuit simulator, we have developed a circuit model of the HHE device using the techniques presented in Reference [12] and have used this model to simulate our designs. We model the I–V characteristics and the magnetization state of the HHE device based on measurements of the device shown in Figure 1, although our model can be configured with different parameters for the switching threshold, input resistance, and output resistance to model different HHE devices.

Figure 8 shows the simulated output of a four-input two-level HHE threshold logic macrocell of the type shown in Figure 6. Using a 20 ns clock period, the macrocell is configured on a cycle-by-cycle basis to implement four different functions: AND, XOR, symmetry (determines whether the input vector is a palindrome), comparison (true if  $X_{1,2} \geq X_{3,4}$ ), and majority (true if  $X_1 + X_2 + X_3 + X_4 \geq 3$ ). Note that the output of the macrocell is delayed by one clock cycle from the inputs and configuration, so the input changes at  $T = 30$  ns cause the output to change at  $T = 50$  ns.

Figure 9 shows how our designs respond to power supply interruptions. At the start of simulation, the state of the HHE gate has been set to ‘1,’ and the SENSE input to the output interface is asserted, driving the output metastable. When SENSE is de-asserted, the output of the HHE gate converges to logic 1, showing that the output interface correctly latches the output of the HHE device. At time  $T = 60$  ns, power is removed from the device ( $V_{DD} = 0$ ), and the output voltage quickly decays to 0 V. At  $T = 140$  ns, power is restored and the SENSE input asserted, returning the output to its metastable state. 10 ns later, SENSE is de-asserted and the output converges to logic 1 again. This shows that HHE-based gates can be used to build systems that recover near-instantly from power supply interruptions, retaining their state when power is removed and quickly returning to normal operation when it returns.

### 3.4. Logic synthesis to minimize power

Power consumption is a major issue in HHE-based designs because of their high input and bias currents. In general, minimizing the number of gates required to implement a circuit

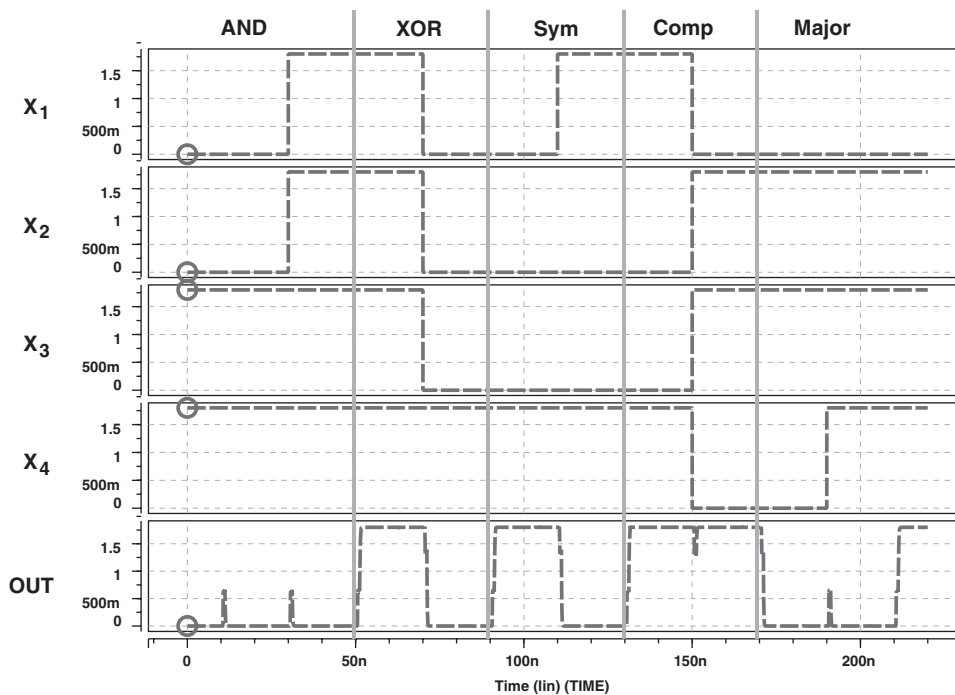


Figure 8. Simulation results showing cycle-by-cycle reconfiguration of the threshold macrocell. Outputs are one clock period (20 ns) delayed from input changes.

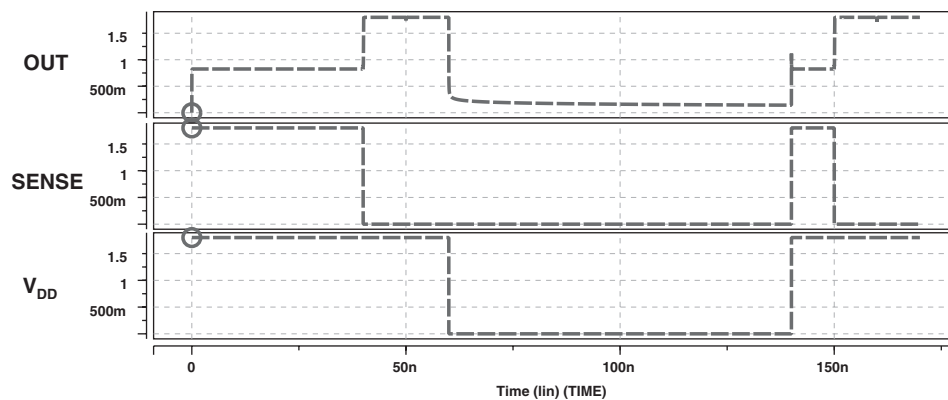


Figure 9. Recovery from power failure.

will reduce power consumption, but there is often more than one implementation of a circuit that requires the minimum number of gates. We have developed algorithms that estimate the power consumption of different HHE-based designs, allowing us to choose an implementation that minimizes power as well as gate count.

To estimate the input power consumption of an HHE-based macrocell, we start with the assumption that the inputs to the circuit are random, and compute the probability that each threshold gate's output will be '0' or '1' on any cycle. With this information, we can compute the average number of input transistors in each threshold gate that will be conducting current on a given cycle, taking into account that we only allow current flow through the input wire that could change a gate's output, and scaling our estimates based on the size of each transistor in the gate. Summing these average-on counts for each of the gates in the macrocell gives an estimate of the amount of input power that the macrocell will consume.

We have applied this technique to multiple implementations of a four-input XOR function on our macrocell, and found an implementation whose predicted number of 'on' input transistors was 86% as large as that of the implementation originally generated by LSAT. Simulating these two designs, we find that the new implementation consumes 89% of the power consumed by the LSAT implementation, showing that this on-transistor estimate is a good predictor of total power consumption. The simulated results do not exactly match the predicted results because the predicted results do not account for power consumed in the output stage of the device, which will be relatively constant from design to design. Given the high power consumption of HHE gates, we believe that power-reducing synthesis techniques such as these will be critical to the success of our designs.

#### 4. COMPARISON WITH OTHER TECHNOLOGIES

In order to fully evaluate circuits based on hybrid Hall effect devices, it is important to understand how they compare to both other non-volatile storage devices and other implementations of reconfigurable logic. A number of non-volatile memory technologies are either available today or expected to enter the market in the next few years, including magnetoelectronic RAMs, FLASH memories, FeRAMs, and ovonic memories. One key advantage of the HHE device over all of these alternatives is its ability to combine logic with non-volatile storage, rather than requiring separate circuitry to compute a logic function and store the result.

##### 4.1. Alternate non-volatile technologies

FLASH/EEPROM memories [13, 14] provide non-volatile data storage by tunneling charge onto and off of a floating gate structure. While high-capacity FLASH memory chips are available today, FLASH memories have disadvantages that make them an unsuitable technology for non-volatile logic. First, they are very slow compared to HHE devices, requiring microseconds or even milliseconds to change state. Second, charging/discharging the floating-gate capacitor in a FLASH memory cell damages the gate oxide. This limits the lifetime of FLASH memories to approximately  $10^6$  read/write cycles, potentially causing them to wear out in only a few seconds when written at their maximum rate.

As might be expected from the fact that they rely on similar physical phenomena, the performance of HHE devices is very similar to that of other magnetoelectronic memory devices, such as those based on giant magnetoresistance [1, 2]. Both HHE and GMR devices require relatively high input currents, retain data indefinitely, and can be read and written an arbitrary number of times without device degradation. Because they only require one layer of ferromagnetic material, magnetoelectronic devices based on the Hall effect may be

somewhat easier/cheaper to fabricate than those based on giant magnetoresistance, as these devices require multiple layers of ferromagnetic material. GMR-based devices have been used to implement look-up tables that retain their state without power and non-volatile logic gates [12], but these designs require multiple magnetoelectronic devices for each input bit. In addition, the current pulses required to program these circuits are more complex than the ones used in HHE-based designs, increasing the complexity of the support circuitry required around the gate.

Ferroelectric memories (FeRAMs) [15], which store data by altering the capacitance of a ferroelectric material, have the potential to be a serious competitor to magnetoelectronic devices because they operate at similar speeds and require much less current to change their state. However, current FeRAM structures suffer from either limited data retention time or limited durability. FeRAM devices have been reported that can handle up to  $10^{12}$  read/write cycles with long data retention times, which is more than sufficient for many applications. However, using such a structure in a system that operated at Gigahertz speeds would lead to device failure in approximately 100 s, which is unacceptable.

Ovonic memories [16] rely on thermal, rather than electrical, processes to store information. An ovonic memory cell consists of a layer of chalcogenide material and a small electrical heater. Flowing current through the heater melts the chalcogenide. When the heater is turned off, the chalcogenide solidifies in either a crystalline or an amorphous state depending on whether the heater is turned off gradually or abruptly. The resistance of the chalcogenide varies significantly depending on whether it is crystalline or amorphous, making it possible to detect the state of the bit by measuring current flow through the material. Unfortunately, the need to cool the chalcogenide relatively slowly limits the speed of ovonic devices to 10s of nanoseconds per write cycle. This makes them attractive competitors to DRAM memories, which have similar access times, but makes them too slow for logic applications.

Of the currently-available or near-market non-volatile memory technologies, magnetoelectronic devices are the only ones that display both the performance and the durability required for logic applications. While other magnetoelectronic structures have been used in reconfigurable logic, HHE devices require fewer fabrication steps than GMR-based devices, reducing manufacturing complexity. In addition, the current pulses required by HHE-based devices are less complex than the ones used in GMR devices, making them easier to integrate into designs.

#### *4.2. Other implementations of reconfigurable logic*

SRAM-based look-up tables and EEPROM wired-and logic represent two extremes of the technologies currently used to implement reconfigurable logic. Table I compares these technologies in terms of volatility, write time and wear, power, functional coverage, and fan-in.

In a design based on SRAM LUTs, each logic block contains a  $2^n$ -bit memory that can be programmed to compute any one-bit function of  $n$  inputs, treating the inputs as the address lines of the LUT. SRAM LUTs give high performance, can be reprogrammed an arbitrary number of times, and consume little power. However, they are extremely volatile, losing their configuration and their outputs without power. Non-volatile LUT designs that retain their configuration without power do exist, such as those based on antifuse technologies, but these systems are generally 'write-once' devices, meaning that they cannot be reconfigured to implement a different circuit once they have been configured.

Table I. Properties of three different reconfigurable programming technologies.

Programming technology	Non-volatile config.	Non-volatile outputs	Cell write time	Write wear	Power per cell	Functional coverage	Typical fan-in
SRAM LUT	No	No	1–10 ns	No	Low	100%	4–6 inputs
EEPROM Wired-AND	Yes	No	10–100 $\mu$ s	Yes	Medium	Depends on No. of product term gates $M$	30 inputs
HHE Threshold	Yes	Yes	2 ns	No	Very High	For same $M$ , better than EEPROM	8–16 inputs

EEPROM wired-AND logic is implemented using a set of EEPROM transistors connected in parallel, with a ratioed pull-up resistor. By varying the amount of charge stored on their floating gates, the EEPROM transistors can be configured to be ‘on,’ meaning that a high voltage at their gate input causes current to flow between the source and drain, or ‘off,’ in which case no current flows through the device, regardless of the voltage on the gate. This allows each wired-AND gate to compute the NOR of a subset of its inputs. Two levels of wired-AND logic can be used to implement sum-of-products (SOP) or product-of-sums (POS) functions, similar to the two-level threshold macrocell presented earlier.

EEPROM-based systems retain their configurations without power, making them useful in control state machines. However, their long programming times (microseconds or higher) and the fact that they wear out after a limited number of write cycles makes it impractical to store computation state in EEPROM-based systems, leaving them vulnerable to loss of data on power failures. Wired-AND circuits also have the disadvantage that they consume static power whenever any of their inputs are asserted. Dynamic precharge/evaluate techniques have been proposed to address this issue [17], and we assume their use in our simulations.

HHE-based threshold gates fall between LUTs and wired-AND gates in terms of flexibility. An  $N$ -input threshold gate can implement a superset of the functions realizable by an  $N$ -input wired-AND gate, but fewer functions than a LUT with the same number of inputs. Like EEPROM circuits, HHE gates can retain their configuration state without power (assuming that their configuration bits are stored on HHE devices), but they also provide non-volatile storage of their outputs.

Both HHE and wired-AND gates can support large numbers of inputs effectively. Adding inputs to a wired-AND gate requires only one additional EEPROM transistor per input, while an  $N$ -input HHE gate needs  $2N$  input transistors and  $2 \log_2(N)$  configuration transistors. In contrast, LUT-based gates require exponentially more bits of SRAM as their number of inputs increases, making direct implementation of wide-input functions unattractive. Multi-level networks of LUT-based gates can implement wide-input functions reasonably efficiently, however.

As a qualitative comparison between these three technologies, Figure 10 shows the power consumption of an 8-input XOR gate in each technology as a function of clock rate, averaged across all possible inputs to the gate. The EEPROM and HHE implementations use a two-level macrocell, while the LUT implementation relies on a single 8-input lookup table. SRAM LUTs are the clear winner in this comparison, because the SRAM cells used to implement

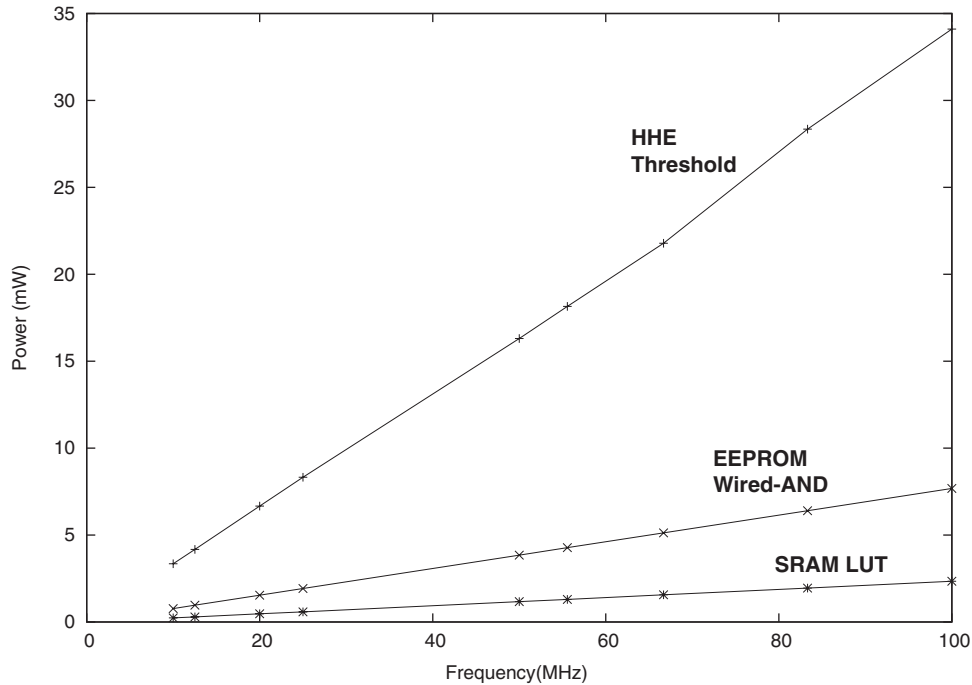


Figure 10. Power consumption comparison of different reconfigurable technologies.

them consume very little power once they have been programmed. The wired-AND design also requires less power than the HHE-based design, in spite of the fact that the HHE design requires only 9 gates to implement the XOR function while the wired-AND design requires 129, due to the magnitude of the input currents required by each HHE gate.

While HHE designs suffer in purely power-based comparisons, they do much better when both power and performance are considered. Figure 11 shows the power-delay product (PDP) of an XOR gate in each technology as a function of the number of inputs to the gate. For small numbers of inputs, the HHE design performs less well than the others, but there is a crossover point at around 10 inputs where the HHE gate becomes more efficient. This crossover occurs because the number of wired-AND gates required to implement an XOR function increases exponentially with the number of inputs, as does the number of bits of SRAM required in the LUT, while the HHE-based threshold logic only requires linearly more gates to input wider XOR functions. At some point, which may vary slightly based on the fabrication process used to implement each design, the high power/gate of the HHE design is outweighed by the reduction in the number of gates needed, making it the more efficient choice. Note that, for this experiment, we assumed a single-LUT implementation of the XOR function. More efficient multi-LUT implementations exist, but are less directly comparable to the other technologies.

In most cases, the deciding factor between SRAM LUTs, EEPROM wired-AND, and HHE gates as the implementation technology for a design will be the amount of non-volatility required by the application. If the application can tolerate loss of state on power failure and

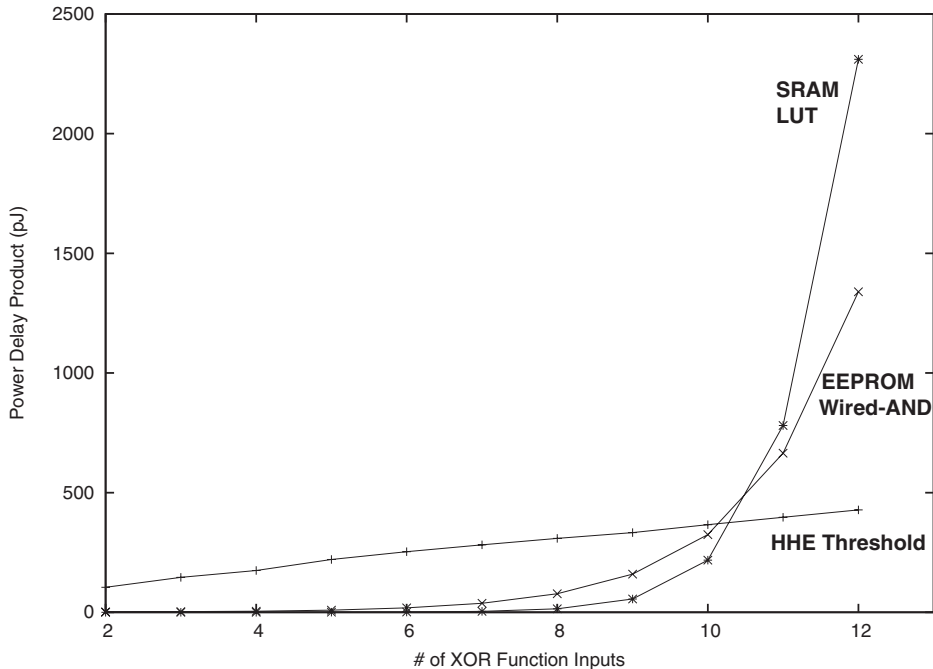


Figure 11. Power-delay product of different reconfigurable technologies.

some amount of reboot/reconfiguration time, then SRAM LUTs will almost always be the better choice because of their versatility and low power consumption. If non-volatile storage of a system's configuration is required, either HHE or wired-AND gates may be the right choice, depending on the number of inputs to the functions being computed and how often the device will be reconfigured. If it is necessary to retain both device configuration and the state of a computation across power failures, HHE gates become the only practical technology, and the question becomes how these devices should be integrated into a design to maximize performance while minimizing power consumption and recovery time from power failures. In the next section, we present some initial thoughts on system design using HHE devices.

## 5. SYSTEM-LEVEL ISSUES

As high-performance non-volatile logic gates and memories become available, designers will be able to implement a wide range of systems that recover near-instantly from power supply interruptions. These systems will have a number of advantages over current computing devices, including the ability to perform long-running calculations without risking data loss on a power failure and the ability to shut down completely to consume zero power but still power-on instantly. The key question in the design of these systems will be where non-volatile devices should be used to give the best combination of performance, power consumption, and non-volatile operation.

Unfortunately, the high power consumption of HHE gates makes it impractical to build large-scale systems out of these devices. Instead, we consider how a small number of HHE devices might be added to a conventional processor pipeline or a field-programmable gate array (FPGA) to provide non-volatile operation. In all cases, we assume that any off-chip memory is implemented using MRAMs or an equivalent non-volatile storage technology.

In our discussions, we focus on protecting the state of a computation, rather than the contents of program memory or the configuration of a device because the time required to load a program or configuration from non-volatile storage is typically very short – less than a second for current FPGAs, for example. Most of the time spent in recovering from power loss is spent restoring program state, making it the critical resource to protect. For the rare application that cannot tolerate this program/configuration loading time, it is possible to implement the program or configuration memory using HHE devices because the contents of these memories do not change during normal program execution. Instead of pulsing the input and bias currents of the HHE device every cycle, we can rely on the output interface to hold the program/configuration state except when a power failure occurs, only applying input currents when a program/configuration is first loaded onto the chip, and only applying the bias currents when recovering from power failures. This allows implementation of a non-volatile program or configuration memory whose performance and power consumption is similar to that of a conventional SRAM during normal operation, although the area per bit of the memory will be noticeably larger than SRAM due to the magnetoelectronic devices.

### 5.1. *Microprocessors*

Current microprocessors are constructed out of one or more instruction pipelines, each of which consists of a number of stages that are separated by latches. Instructions move through a pipeline at the rate of one stage per clock cycle, stalling if their input data is not yet available. Typically, the processor also contains a number of registers that provide fast access to data. From the point of view of the programmer, the state of the microprocessor consists of the values in the registers, the contents of memory, and the address of the next instruction that the program will execute. This is the state that our designs will preserve across power failures—other system state, such as the contents of branch prediction buffers, exists solely to improve performance and can be discarded on a power failure.

One approach to non-volatile processor design would be to implement each of the pipeline latches and data registers using HHE devices. In this approach, power could be removed from the processor at any time. When power was restored, the processor could resume operation immediately, with no loss of state.

While this design would give near-instant recovery from power failures, it is likely to be impractical, for several reasons. First, the latencies of HHE devices are on the order of 1–2ns, which is much longer than the 0.5–0.3ns clock cycle times of high-end microprocessors. Adding the delay of an HHE device to each pipeline stage would reduce performance by up to a factor of seven, which would be unacceptable in most cases. In addition, modern processors have tens of thousands of bits of pipeline latches and registers. Replacing all of these bits with HHE devices would unacceptably increase the power consumption of the processor.

A more attractive approach takes advantage of the structures that processors use to execute instructions out of order. Out-of-order processors allow instructions to begin execution whenever their inputs are ready, but require that they complete in the order that they appeared

in the program. This improves performance by preventing one operation, such as a memory instruction that misses in the cache, from holding up other independent operations, while ensuring that the results of a program are the same as if it were executed in order and that interrupts and exceptions behave as if instructions were executed in order.

To implement out-of-order execution, processors use structures such as reorder buffers [18] to hold instructions that have gone through the pipeline but that are not allowed to complete because instructions ahead of them in the program have not completed. At any point in time, there is an instruction in the program that is the most-recently completed instruction. Any instructions before that instruction are guaranteed to have completed, and no instructions after that instruction have completed. If the processor discovers that it has started to execute some instructions that it should not have, for example by starting to execute down the wrong path of a branch, it can simply discard the contents of the reorder buffer and the pipeline latches and resume execution at the instruction after the most-recently completed instruction with little loss of progress.

This concept can be extended to non-volatile processor design by implementing a second register file using HHE gates or other non-volatile storage. Since this register file would be slower than the normal register file, the results of a *block* of several instructions would be aggregated into one multi-register write operation and written into the non-volatile register file simultaneously, along with the address of the last instruction in the block. If a power failure occurs, register state and the program counter could be copied out of the non-volatile register file and execution resumed at the following instruction.

While we assume the use of non-volatile off-chip memory to retain data across power failures, almost all processors contain on-chip caches, as well. One approach to cache design for a non-volatile processor would be to require the use of write-through caches, in which all store operations write their results into the off-chip memory as well as the on-chip cache. This would guarantee that the data in the off-chip memory was always up to date, but the amount of time required to write the result of every store out to off-chip memory would reduce program performance in most cases.

An alternate approach would be to require that the results of store instructions be written to off-chip memory at the time that the block of instructions containing the stores commits its results into the non-volatile register file. This would reduce the number of accesses to off-chip memory because only the last value written to a given address in a block of instructions would have to be written out to memory. Similarly, only the last value written to a given register would be committed into the non-volatile register file.

Under this approach, on-chip caches could be made write-back, improving performance for most programs. In addition, the 'commit point,' the last instruction whose result has been written to non-volatile storage, would be the same for memory and non-memory instructions, which would be of significant benefit. We are starting to explore processor architectures that use this approach, and hope to have performance results in the near future.

## 5.2. Field-programmable gate arrays

A typical FPGA consists of an array of logic blocks that communicate with each other through a programmable network. SRAM cells in each logic block and network switch hold the configuration of the FPGA, and each logic block can typically be configured to latch its output to allow computations to be pipelined through the array. Using this structure, some or

all of the SRAM-based logic blocks can be replaced with HHE-based configurable gates to provide non-volatile storage of critical application state by designing an HHE-based gate with the same I/O interface as the original logic block.

The challenge in this design is not so much in the hardware architecture, but in determining which pieces of an application's state need to be kept in non-volatile storage and communicating this to the synthesis tools so that they map these pieces onto HHE gates. Little to no work has been done in this area, and one of the goals of our future research will be to develop these techniques.

Overall, we believe that HHE-based non-volatile logic gates can be of significant benefit to system designers if used carefully. Designs that incorporate these devices will have to balance the amount of non-volatile state they incorporate against power consumption issues and performance needs, but will be able to handle power supply interruptions much more gracefully than current systems.

## 6. RELATED WORK

A number of research efforts have explored threshold logic circuits, most recently based on modified CMOS transistor designs that rely on capacitive sum coupling between several inputs and transistor gates. Proposed devices include  $\mu$ MOS transistors [19] and capacitive-threshold (CTL) gates [20]. These devices can be fabricated in standard CMOS processing technology, making them easy to integrate into semiconductor systems, and have been used to implement reconfigurable threshold logic gates [21].

These designs have several disadvantages, however. First,  $\mu$ MOS and CTL gates require periodic resets to prevent capacitive discharge effects from causing their threshold values to vary over time. Second, they require an analog reference voltage to determine their threshold values, which can be difficult to generate and distribute, particularly when process variations are taken into account. Finally, these devices are volatile, losing their state when power is removed.

HHE-based threshold gates have none of these disadvantages. Because their threshold values are determined digitally, they do not require the generation of analog signals, and their threshold values will remain stable across effectively infinite numbers of evaluation cycles. They provide non-volatile storage of their output values, and are likely to display substantial resistance to soft errors caused by particle strikes.

## 7. FUTURE WORK

The SPICE models used in our circuit designs are highly empirical, based on measurements of a small number of fabricated HHE devices. While these models have been very useful, one of our immediate goals is to add accurate models of magnetic phenomena to our SPICE models, in order to better predict the delay through our circuits and allow us to estimate how performance will change with improvements in fabrication technology. These improved models will also help us study the sensitivity of our designs to fabrication variations.

Another focus of future work will be fabrication of prototype versions of our circuits to verify their functionality. The major obstacle in this effort has been locating a fabrication

facility that can build HHE devices and conventional transistors on the same wafer. Recently, we have begun discussions with a group at the University of Notre Dame that is fabricating magnetoelectronic devices on top of CMOS structures, and hope that their techniques will allow us to fabricate test circuits in the near future.

Finally, we are beginning studies of non-volatile system designs based on our circuits and other magnetoelectronic devices. Loss of data and application state on power failure is a major problem in current electronic devices, and we are developing architectures that incorporate a small number of magnetoelectronic gates to greatly reduce the amount of time required to recover from power supply interruptions. As discussed in Section 5, this will be particularly challenging for reconfigurable systems, as it will also be necessary to develop new CAD techniques that automate the design of non-volatile systems by mapping the critical portions of circuits onto magnetoelectronic devices and generating any support logic required to recover from power failures.

## 8. CONCLUSION

This paper has outlined a set of reconfigurable magnetoelectronic circuits for threshold logic based on the hybrid Hall effect device. These circuits can be configured on a cycle-by-cycle basis to compute different functions of their inputs, and will retain their output values indefinitely, even in the absence of an external power supply. To facilitate integration with conventional semiconductor designs, our circuits are CMOS-compatible, using input and output interfaces that limit the duration of current flow to reduce power consumption.

Our work is based on the results of micron-scale device fabrication experiments at the Naval Research Laboratory. HHE devices are expected to scale to dimensions of a few nanometers on a side as fabrication technology improves, with significant reductions in input current levels as devices shrink. In addition, new device structures are being explored that greatly increase the Hall resistance of HHE devices, reducing the amount of bias current required to sense the state of each device.

Because of their high performance compared to other non-volatile devices, stability over large numbers of read-write cycles, and easy integration with CMOS designs and fabrication, HHE-based circuits are an attractive building block for non-volatile electronic systems. We are currently exploring system architectures based on our circuits that recover near-instantly from power supply interruptions, both for embedded applications where instant-on application and zero standby power are essential and for computing applications where loss of state due to power failures is a significant issue. Because magnetoelectronic devices are generally significantly less sensitive to single-event upsets caused by high-energy particles than CMOS circuits, HHE-based systems may also find applications in systems that must operate in high-radiation environments. These systems could handle radiation-induced soft errors in the same way they handle power supply failures, restoring themselves to a magnetoelectronically-stored state when an error is detected by parity logic.

## ACKNOWLEDGEMENTS

The authors would like to thank Mark Johnson of the Naval Research Lab for his help in this work, including the data used in our device models.

This material is based upon work supported by the Office of Naval Research under Award No. N00014-02-1-1038. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the ONR.

## REFERENCES

1. Johnson M. Magneto-electronic memories last and last. *IEEE Spectrum*, 2000; 33–40.
2. Zhu J-G, Zhang Y, Prinz GA. Ultrahigh density vertical magnetoresistive random access memory. *Journal of Applied Physics* 2000; **87**(9):6668–6673.
3. Johnson M, Bennett BR, Yang MJ, Miller MJ, Shannabrook BV. Hybrid hall effect device. *Applied Physics Letters* 1997; **71**:974–976.
4. Johnson M, Bennett BR, Hammar PR, Miller MM. Magneto-electronic latching boolean gate. *Solid-State Electronics* 2000; **44**(6):1099–1104.
5. Streetman BG, Banerjee S. *Solid State Electronic Devices*. Prentice-Hall: Upper Saddle River, NJ, 2000.
6. Alexi Orlov. Personal communication, 2004.
7. Steve P. Ferrera, Nicholas P. Carter, Reconfigurable circuits using hybrid hall effect devices. In *Proceedings of the 13th International Conference on Field Programmable Logic*, 2003; 1–10.
8. Ferrera SP. Reconfigurable circuit designs and applications for magneto-electronic logic. *Master's Thesis*, University of Illinois at Urbana-Champaign, 2004.
9. Muroga S. *Threshold Logic and its Applications*. Wiley-Interscience, New York, 1971.
10. Oliveira AL, Sangiovanni-Vincentelli A. LSAT—an algorithm for the synthesis of two level threshold gate networks. In *International Conference on Computer-Aided Design* 1991; 130–133.
11. Brayton RK, Hachtel G, McMullen C, Sangiovanni-Vincentelli A. *Logic Minimization Algorithms for VLSI Synthesis*. Kluwer Academic Publishers: Boston, 1984.
12. Black Jr WC, Das B. Programmable logic using giant-magnetoresistance and spin-dependent tunneling devices. *Journal of Applied Physics* 2000; **87**(9):6674–6679.
13. Sasago Y, Kurata H, Arigane T, Oguta K, Kobayashi T, Ikeda Y, Fukumura T, Narumi S, Sato A, Terauchi T, Shimizu M, Noda S, Kozaki K, Tsuchiya O, Furusawa K. 90-nm-node multi-level AG-AND type flash memory with cell size of true  $2f^2$ /bit and programming throughput of 10 MB/s. In *IEEE International Electron Devices Meeting*, 2003; 34.2.1–34.2.4.
14. Yong-Sik Yim, Kwang-Shik Shin, Sung-Hoi Hur, Jae-Duk Lee, Ihn-Gee Baik, Hong-Soo Kim, Soo-Jin Chai, Eun-Young Choi, Min-Cheol Park, Dong-Seok Ekn, Sung-Bok Lee, Hye-Jin Lim, Sun-Pil Youn, Sung-Hun Lee, Tae-Jung Kim, Han-Soo Kim, Kyu Charn Park, Ni-Nam Kim. 70nm NAND flash technology with  $0.025\mu\text{m}^2$  cell size for 4Gb flash memory. In *IEEE International Electron Devices Meeting*, 2003; 34.1.1–34.1.4.
15. Oh S-H, Hong S-K, Noh K-H, Kweon S-Y, Kim N-K, Yang Y-H, Kim J-G, Seong J-Y, Jang I-W, Park S-H, Bang K-H, Lee K-N, Jeong H-J, Son J-H, Lee S-S, Choi E-S, Sun H-J, Yeom S-J, Ban K-D, Park J-W, Park G-D, Song S-Y, Shin J-H, Lee S-I, Park Y-J. Noble FeRAM technologies with MTP cell structure and BLT ferroelectric capacitors. In *IEEE International Electron Devices Meeting*, 2003; 34.5.1–34.5.4.
16. Manzur Gill, Tyler Lowrey, John Park. Ovonic unified memory—a high-performance nonvolatile memory technology for stand-alone memory and embedded applications. In *International Solid-State Circuit Conference*, 2002; 1:202–203.
17. Wang C-C, Wu C-F, Hwang R-T, Kao C-H. A low-power and high-speed dynamic PLA circuit configuration for single-clock cmos. *IEEE Transactions on Circuits and Systems I* 1999; **46**(7):857–861.
18. James E. Smith, Andrew R. Plezkun. Implementation of precise interrupts in pipelined processors. In *International Symposium on Computer Architecture*, 1985; 36–44.
19. Shibata T, Ohmi T. A functional MOS transistor featuring gate-level weighted sum and threshold operations. *IEEE Transactions on Electron Devices* 1992; **39**(6):1444–1455.
20. Özdemir H, Kepkep A, Pamir B, Leblebici Y, Çilingiroğlu U. A capacitive threshold-logic gate. *IEEE Journal of Solid-State Circuits* 1996; **31**(8):1141–1150.
21. Aoyama K, Sawada H, Nagoya A, Nakajima K. A threshold logic-based reconfigurable logic element with a new programming technology. In *International Conference on Field Programmable Logic and Applications* 2000; 665–674.