A Magnetoelectronic Macrocell Employing Reconfigurable Threshold Logic

Steve P. Ferrera
University of Illinois
Urbana-Champaign, IL
ferrera@crhc.uiuc.edu

Nicholas P. Carter
University of Illinois
Urbana-Champaign, IL
npcarter@crhc.uiuc.edu

ABSTRACT

In this paper, we introduce a reconfigurable fabric based around a new class of circuit element: the hybrid Hall effect (HHE) magnetoelectronic device. Because they incorporate a ferromagnetic element, HHE devices are inherently non-volatile, retaining their state without a power supply. In addition, HHE devices are extremely well-suited to implementing threshold logic circuits, which allows many complex logic functions to be implemented in fewer gates than are required in systems based on AND-OR logic. We present the design of an HHE-based reconfigurable macrocell based on two-level threshold logic that can be configured on a cycle-by-cycle basis while internally storing non-volatile configuration data and computation state. The performance of this macrocell is characterized, and compared to that of competing technologies, showing that it has a significantly better power-delay product when implementing complex functions of many inputs.

Categories and Subject Descriptors

B.6.1 [**Logic Design**]: Design Styles – *combinational logic*; B.7.1 [**Integrated Circuits**]: Types and Design Styles – *advanced technologies*

General Terms

Performance, Design, Theory

Keywords

Magnetoelectronic circuits, lookup table, non-volatility, PLA/CPLD, threshold logic, wired-AND logic.

1. INTRODUCTION

Since the discovery of the giant magnetoresistance (GMR) effect, a number of magnetoelectronic devices have been developed that take advantage of the properties of ferromagnetic materials to provide non-volatile data storage [1,2]. Recently, researchers at the Naval Research Lab have developed a new class of magnetoelectronic device, the hybrid Hall effect (HHE) device, which can be reprogrammed from cycle to cycle to implement a variety of logic functions with non-volatile storage of the result.

In [3], we presented a set of circuit designs for reconfigurable logic gates based on HHE devices, including interface logic that allows these circuits to be integrated into CMOS systems. These circuits, which are based on a single HHE device and a small number of CMOS transistors, can be reconfigured to implement AND, OR, NAND, and NOR gates with multiple inputs. In addition, we presented circuits that use HHE devices to provide non-volatile storage for conventional SRAM cells.

However, HHE devices are more versatile than these circuits would indicate. State switching in an HHE device is dependent on whether or not the magnitude of the input current to the device is large enough to generate a magnetic field capable of changing the magnetization state of the device's ferromagnetic element. This permits HHE-based circuits to efficiently implement reconfigurable threshold logic, which often allows the implementation of complex logic functions in fewer gates than AND-OR designs.

In this paper, we present a set of designs for reconfigurable threshold logic based on HHE devices. These circuit designs occupy an intermediate point between current SRAM-based reconfigurable logic and EEPROM wired-AND systems. At one extreme, designs based on SRAM lookup tables (LUTs) provide extremely high functional coverage, being able to implement any Boolean function of their inputs. However, LUTs suffer from geometric increases in area as their number of inputs increases, and do not provide any non-volatile storage. At the opposite end of the spectrum, EEPROM based programmable logic arrays

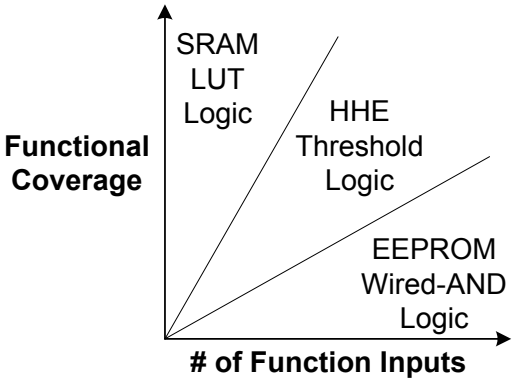


Figure 1. Functional coverage vs # of function inputs.

FPGA'04, February 22–24, 2004, Monterey, California, USA

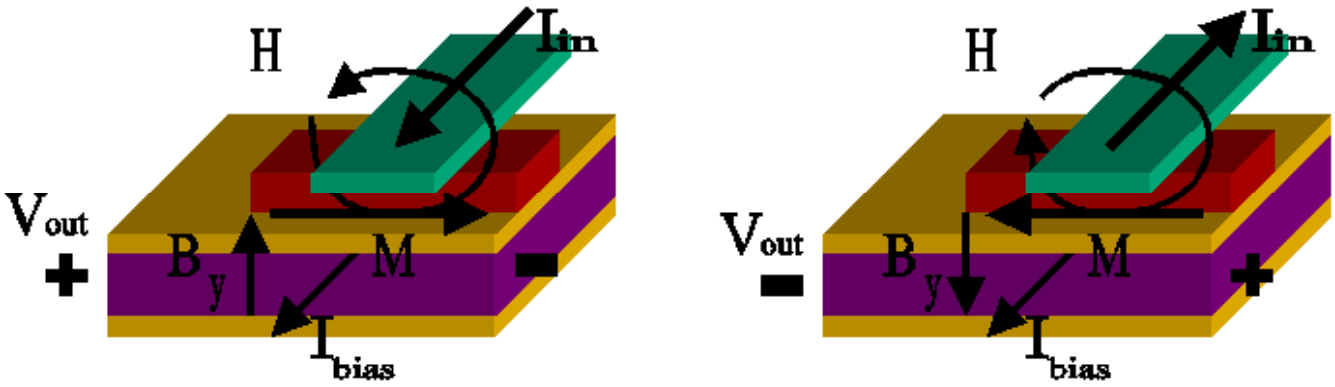


Figure 2. HHE device physical structure showing two possible magnetization states M.

(PLAs) retain their configuration without power, and efficiently implement functions with many inputs, but are limited in the set of functions they can implement. As diagrammed in Figure 1, HHE-based threshold logic is capable of efficiently implementing functions with more inputs than LUT-based designs can efficiently support, but can implement a wider range of functions than PLA-based systems.

In the next section, we describe the HHE device and its operation. In section 3, we introduce the concepts behind threshold logic. Section 4 discusses the implementation behind an HHE-based reconfigurable macrocell. Section 5 presents SPICE simulation results for our designs, including comparisons with other technologies. In sections 6 and 7, we discuss related work and propose future work respectively. Lastly, we conclude in section 8. Although we mainly focus on implementations employing HHE devices, the concepts presented in this paper can be transferred to designs utilizing other magnetoelectronic devices.

2. HHE DEVICE

2.1 Device Description and Operation

Figure 2 shows an HHE device [4,5] consisting of one or more input wires (the top bar in the figure) that pass over a region of ferromagnetic material (the middle bar in the diagram). If the magnitude of the current along the input wire is sufficiently large, the magnetic field it generates will magnetize the ferromagnetic element in either the left or right direction, depending on the direction of current flow. Figure 2 shows both possible current directions and their associated magnetization states.

Since ferromagnetic materials retain their magnetization state in the absence of an external magnetic field, the magnetization state of the ferromagnetic element can be used to store a binary value, interpreting one direction of magnetization as a logic 1 and the other as a logic 0. To observe the magnetization state of the ferromagnetic element, a bias current I_{bias} is passed through the conductor at the base of the device in a direction perpendicular to the magnetic field. As specified by the Hall Effect [6], the interaction of this current and the magnetic field generated by the ferromagnetic element produces a voltage perpendicular to the direction of the bias current. The magnitude of this voltage is determined by the Hall resistance of the device and the magnitude of the bias current, allowing trade-offs between the sensitivity required in the circuitry that reads the output voltage and the amount of bias current, and therefore bias power, required.

One issue in the design of HHE-based systems is their need for interface circuitry to translate the inputs and outputs of the device to and from CMOS logic levels. To reduce the amount of bias current required to generate CMOS output levels, an amplifier can be added to the output of the HHE device, and the "zero" value of the output voltage can be adjusted during fabrication so that the output either oscillates around 0 volts or ranges between 0 volts and some maximum. This allows the use of a wide range of output amplifiers. In this paper, we assume the use of an SRAM cell as an output buffer. The output voltage of an HHE device is sufficient to determine which output state an SRAM cell will converge to if forced into its metastable state, making it an effective output buffer for our designs [3]. A more thorough study of the area/power tradeoffs of different output circuitry is intended for future work.

A key difference between HHE devices and CMOS transistors is that the inputs to HHE devices are currents, not voltages. Rather than having a critical input voltage at which the device turns on or off, HHE devices change their state if the magnitude of the input current exceeds a specific value. This makes it easy to implement threshold logic with HHE devices by constructing circuits in which some fraction of the inputs need to be high for the input current to exceed the switching threshold. As we will show in later sections, these threshold logic circuits are extremely well-suited to reconfigurable implementations.

Similar to CMOS technology, HHE devices experience noise. In CMOS, bit flips and parasitic capacitive coupling between lines can unintentionally change the voltage on a node. In HHE technology, magnetic fields due to flowing currents and the magnetization state of neighboring HHE devices may affect the state of any given HHE device. However, since HHE devices require locally strong magnetic fields in order to change state and since there is a $1/r^2$ decrease in magnetic field strength with distance, the parasitic interactions between neighboring HHE devices are insignificant in actuality. Also, noise tolerant techniques for layout can be achieved in which undesired wires are never placed vertically above ferromagnetic elements. As long as the total noise seen by an HHE device is below its switching threshold, noise is rejected, and the HHE device will regenerate to its original state.

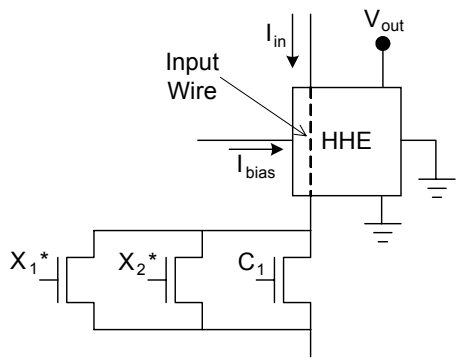
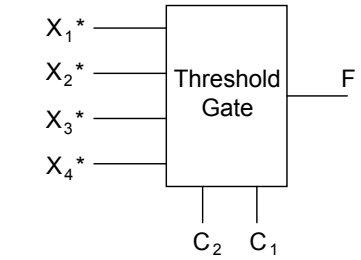


Figure 3. Input interface circuitry for HHE device.



F=1 if X_1*+X_2*+X_3*+X_4* >= 4-(2C_2+C_1)
F=0 if X_1*+X_2*+X_3*+X_4* < 4-(2C_2+C_1)

Figure 4. Reconfigurable threshold gate.

2.2 HHE Circuit Designs

An HHE-based reconfigurable gate was first proposed in [5]. Circuit designs integrating HHE devices with conventional CMOS switching transistors have been designed and simulated in [3]. The basic HHE structure for a reconfigurable gate relies on the sum of input currents surpassing the switching threshold T of the device. Figure 3 depicts the basic concepts behind the interface circuitry required for converting between CMOS voltage levels and the input currents required by the HHE device. If the 3 parallel transistors are sized such that at least 2 of them must be conductive to produce an input current larger than the switching threshold, then the control signal C_1 determines whether the structure computes the AND or OR of its input signals. Detailed circuit designs for HHE reconfigurable gates are depicted in the next section.

3. THRESHOLD LOGIC

3.1 Basics

Threshold logic [7] is a generalization of conventional AND-OR logic. A threshold gate is one in which at least T inputs must be logic 1 in order to produce a logic 1 at the gate output. T is an integer value representing the threshold of the gate. The following function describes the operation of a threshold function, where X_1, X_2, …, X_n are inputs, F is the gate output, and T is the threshold.

$$\mathbf{F = 1 \ \ if \ \ X_1+X_2+\ldots+X_n \geq T}$$

$$\mathbf{F = 0 \ \ if \ \ X_1+X_2+\ldots+X_n < T}$$

A reconfigurable threshold gate may incorporate several control signals in order to affect the threshold value seen by the gate inputs. Figure 4 depicts a reconfigurable threshold gate with a

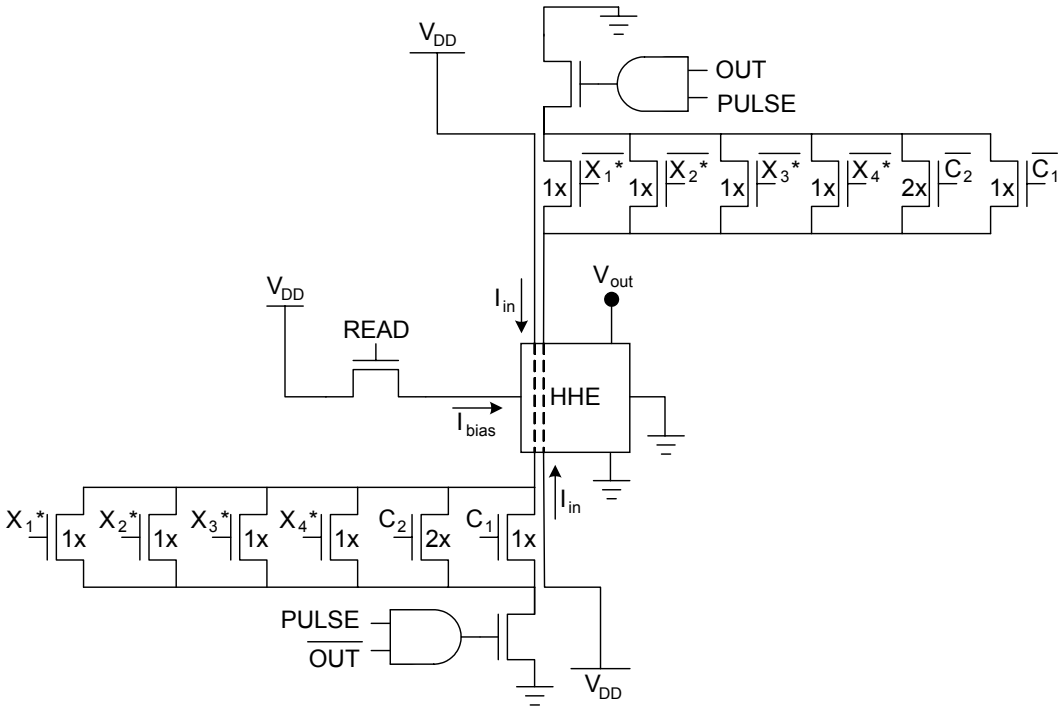


Figure 5. Input interface circuitry for reconfigurable threshold gate.

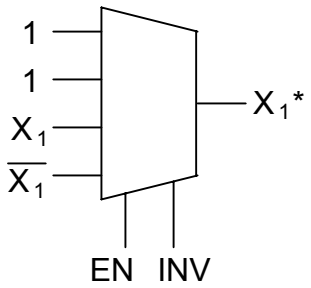


Figure 6. Input Multiplexor.

total of six inputs. Four of the inputs (X_1, X_2, X_3, and X_4) are the input signals, while C_2 and C_1 are configuration signals. In this diagram, C_2 has a weight of two, contributing twice as much as any of the other inputs to the threshold equation. This reduces the number of configuration bits required.

If the threshold value T is set to the number of gate input signals (in this case 4), the static assignment of control signals C_2 and C_1 determines the function of the gate. For example, if C_2C_1=11, then the gate functions as an OR gate. If C_2C_1=00, then the gate functions as an AND gate. Other threshold functions exist between these two scenarios. In general, the effective threshold seen by the gate input signals is $4-(2C_2+C_1)$.

One of the advantages of threshold logic is that Boolean functions can potentially be implemented with fewer threshold gates than with conventional AND-OR gates. This is especially true for complex functions. For example, using only 2 levels of logic, a 4-input XOR function can be implemented with a total of 5 threshold gates versus the 9 that are required for AND-OR logic (8 minterms plus an OR gate). The benefits of threshold gate implementations become even greater as the number of function inputs increases. In a subsequent section, reconfigurable threshold logic will be compared against reconfigurable wired-AND logic, which is commonly used in PLA/CPLD systems.

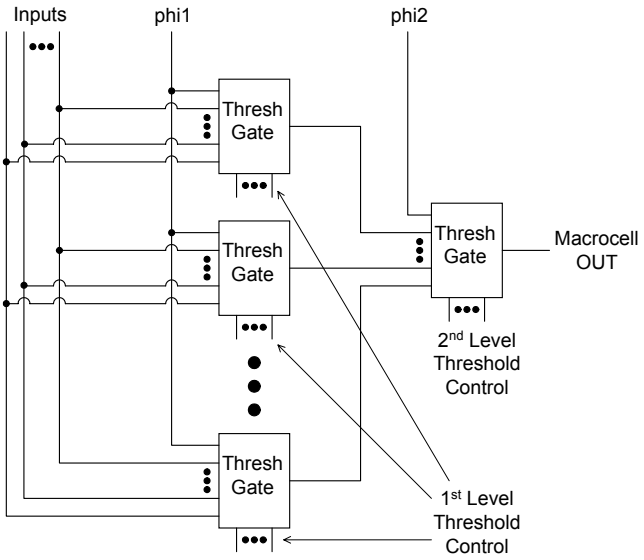


Figure 7. Macrocell design based on two-level threshold gate logic.

3.2 Circuit Implementation

We introduce circuit designs for reconfigurable threshold gates based on HHE devices. Figure 5 shows a circuit design for a four-input reconfigurable gate equivalent to the diagram in Figure 4. This design can be generalized to threshold gates with larger or smaller numbers of inputs.

In this diagram, two input wires exist across the HHE device. These input wires are vertically stacked in different metal layers. To reduce the complexity of the input circuitry, current is only allowed to flow in one direction on each wire. Effectively, one input wire controls whether the output of the gate will be set to logic 1 in a given cycle, and the other controls whether the output will be set to logic 0. A set of CMOS transistors controls whether the current along each wire is greater or less than the threshold value of the HHE device. In the figure, a sizing of "1x" represents a transistor that will conduct 25% of the threshold current of the HHE device, so four of the transistors of size "1x" must be on to switch the magnetization state of the device. The transistor connected to configuration input C_2 is made twice as large to give it twice the weight in the threshold computation.

To reduce power consumption, current is only allowed to flow through one of the input wires at any time, that being the one that could change the state of the HHE device away from its current state. This is accomplished by connecting the true and inverted values of the gate's outputs to the gating transistor for each wire. This particular circuit design is analogous to static CMOS design and has been shown to reduce the power consumption during gate evaluation when compared to circuit designs based on a reset-evaluate methodology [3].

To further reduce power consumption, a pulse signal is used to limit static input currents to a small fraction of the clock period. The lower bound of these pulse widths is constrained to the switching time of the device, which is approximately 2ns for current HHE device technology. Once the magnetization state of the device has been set, the bias current is applied by asserting the read signal, and the device's state is latched into the SRAM output buffer (not shown). If power is removed from the device, the HHE device will retain its state, which can be latched into its output buffer by again applying bias current through it.

A multiplexor, shown in Figure 6, is added before each input transistor in order to provide a greater range of reconfigurability. Configuration signals INV and EN determine whether the input should be inverted (input inversion) and whether the input should be enabled (input enabling) and can be used as selection signals into each multiplexor. The configuration bits for input inversion and input enabling may be stored in a non-volatile fashion within separate HHE devices, as may the configuration bits for the threshold gate. HHE threshold gates with this added reconfigurable flexibility are used in building a fully reprogrammable HHE-based macrocell.

4. HHE-BASED MACROCELL

4.1 Two-Level Threshold Gate Macrocell Design

In this section, we describe a two-level threshold logic implementation that has a similar structure to two-level AND-OR logic. In reconfigurable systems such as PLAs and CPLDs, the

inputs to each first level AND gate and connections to the second level OR gate may be configured. In the case of threshold logic, the same reconfiguration techniques apply, except the threshold T of each gate may also be reconfigured. This provides greater flexibility in reconfiguration and allows complex functions with many product terms to be implemented with a small number of threshold gates. Figure 7 shows a block diagram of a reconfigurable two-level threshold logic macrocell implemented using HHE devices.

Within the macrocell, each threshold gate takes a clock input to control the timing of its input enable and bias current pulses. Logic within the gate generates these pulses on the rising and falling edges of the clock input, separating them by enough time to ensure that the output of the HHE device has stabilized before the bias current is asserted. A four-phase clocking scheme is used for the macrocell, which has two non-overlapping clock inputs φ_1 and φ_2.

4.2 Two-Level Threshold Logic Example

In this section, we provide an example of how a 3-input XOR function is implemented using 2 levels of threshold gate logic. We use the LSAT synthesis tool [8] which is specifically designed for minimizing two-level threshold logic networks. The minimization algorithm is an expansion of the well-known boolean minimizer, espresso [9].

Using LSAT, we derive a threshold logic implementation of a 3-input XOR function, which is shown in Figure 8. This implementation requires a total of 4 threshold gates. For each 1st level threshold gate of Figure 8a, a Karnaugh map is shown representing the threshold equation that it implements. For example, gate F of Figure 8a has a threshold value of 2 and has an output of 1 if the gate inputs *abc* are either 000, 100, 010, or 001. The minterms a'b'c', ab'c', a'bc', or a'b'c satisfy gate F's threshold equation; hence, the Karnaugh map cells corresponding to these values are labeled with a logic 1 value.

Using LSAT, the 2nd level threshold gate was determined to have a threshold of 2. As a result, a minterm must appear in at least two of the 1st level Karnaugh maps in order be a minterm for the overall function. These minterms are circled in each 1st level Karnaugh map, and these circled minterms determine the overall

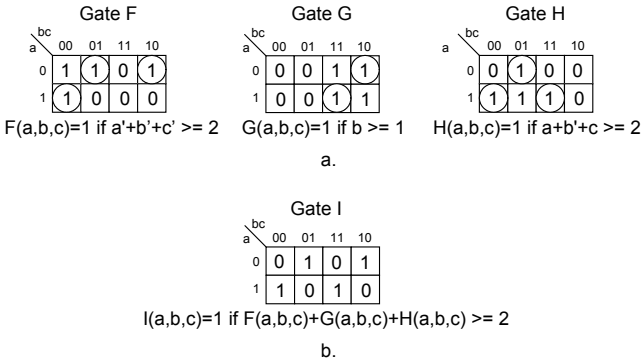


Figure 8. Example of a XOR function implemented with two-level threshold logic.

XOR function that is depicted in Figure 8b. A total of 4 threshold gates are required for the example of Figure 8, which is less than an AND-OR implementation of 5 gates.

4.3 Advantages Over Conventional AND-OR Logic

Figure 9 illustrates the density advantages of threshold-based logic over AND-OR logic by plotting the number of gates required to implement all of the possible functions of four inputs in each methodology. In the figure, the height of each bar represents the fraction of 4-input functions that can be implemented with that many gates. We use LSAT and espresso to find efficient implementations using threshold logic and AND-OR logic respectively. One should note that neither LSAT nor espresso guarantee a logic implementation with a minimum number of gates. Instead, they use a number of heuristics to arrive at a near-optimum solution. As shown in the figure, threshold logic on average requires fewer gates than AND-OR logic to implement Boolean functions. In fact, the maximum number of gates required to realize any function of *n* inputs using threshold logic is less than or equal to that required by AND-OR logic since AND-OR logic is a subset of threshold logic. Also, if we restrict the number of gates that may be used, the functional coverage of threshold logic is much better than AND-OR logic. Although not experimentally verified due to the exponential increase in the

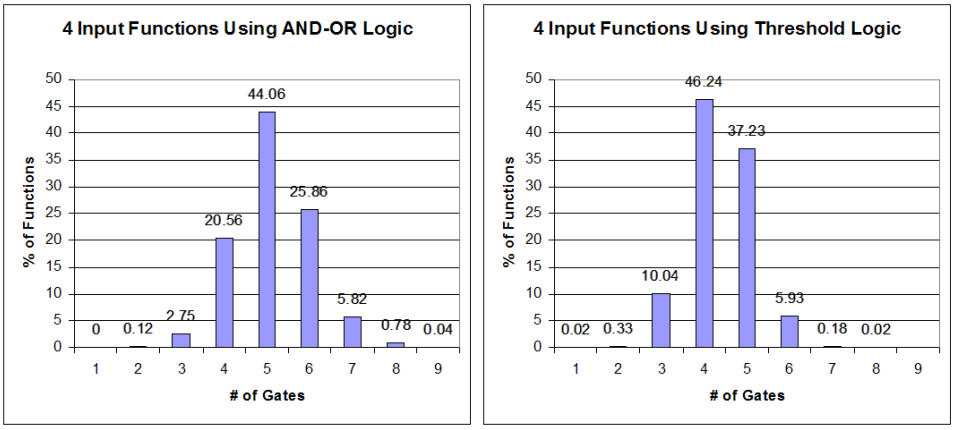


Figure 9. Distribution of the number of gates required for implementing a 4-input function using AND-OR and threshold logic.

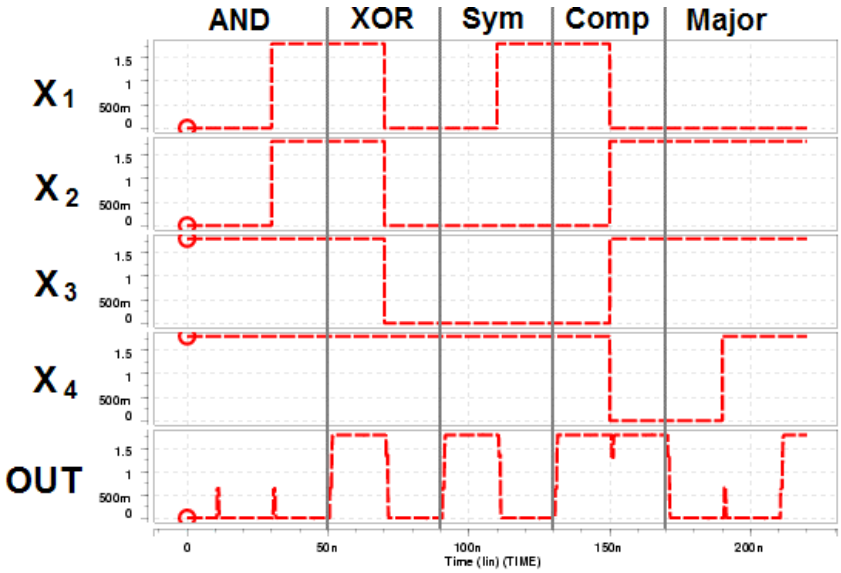


Figure 10. HHE threshold gate reconfigured on a cycle-by-cycle basis. Outputs are one clock period (20ns) delayed from inputs.

number of possible functions, it can be extrapolated from these examples that threshold logic would still require fewer gates and would have better functional coverage for functions with 5 or more inputs.

5. SIMULATION RESULTS

Using the HSPICE™ circuit simulator, we developed a circuit model of the HHE device based on the techniques presented in [10]. We model the IV characteristics and the magnetization state of the HHE device based on experimental data. The model can be configured with different parameters such as switching threshold, input resistance, and output resistance. Each of the designs in the following subsections incorporates .18µ CMOS transistors in a 1.8V technology.

5.1 Reconfiguration of the HHE Macrocell

In Figure 10, we simulate the operation of a 4-input HHE-based reconfigurable macrocell. The macrocell is configured as several different functions during the course of the simulation. With a cycle period of 20ns, this simulation shows that the HHE macrocell can be configured on a cycle-by-cycle basis. In the simulation, the outputs occur one clock period after the inputs become available. A description of each configured function follows.

AND: implements an AND function

XOR: implements an XOR function

Sym: determines whether the input vector is a palindrome, $X_1X_2=X_4X_3$

Comp: determines whether $X_1X_2 \geq X_3X_4$

Major: implements a majority gate, $X_1+X_2+X_3+X_4 \geq 3$

5.2 Programming Technology Comparisons

In this subsection, we evaluate the tradeoffs between SRAM LUT, EEPROM wired-AND, and HHE threshold logic. The basic properties of each programming technology are listed in Table 1.

For a LUT, the basic programming element is an SRAM cell. A LUT may be considered as a single-bit access memory in which the inputs to the LUT are the address lines. A tree decoder or pass gate multiplexer is used to select a specific SRAM cell from the LUT to drive the output. For an *n*-input LUT, there are a total of 2^n SRAM cells to accommodate the output of every possible input combination.

For wired-AND reconfigurable logic, a common programming element is an EEPROM floating-gate transistor. This device may be programmed "off" such that the transistor is non-conducting

Table 1. Properties of three different reconfigurable programming technologies.

Programming Technology	Non-Volatile	Cell Write Time	Write Wear	Power per Cell	Functional Coverage	Typical Fanin
SRAM LUT	No	1-10ns	no	Low	100%	4-6 inputs
EEPROM Wired-AND	Yes	10-100us	yes	Medium	Depends on # of product term gates M	~30 inputs
HHE Threshold	Yes	2ns	No	Very Large	For same M, better than EEPROM	8-16 inputs

under the presence of any input. Several programmed and unprogrammed EEPROM devices may be connected together in parallel with a ratioed pull-up resistor. This provides a NOR gate structure. Two-level NOR-NOR logic can hence be implemented to realize a POS (or SOP) form of a function. Although ratioed logic can be used to implement wired-AND logic, it consumes considerable power due to static currents. Hence, dynamic techniques involving precharge and evaluation phases have been proposed [11], and these techniques are the basis of our wired-AND simulations.

SRAM LUT technology and EEPROM wired-AND technology can be considered to be at two opposite ends of the programming technology spectrum. SRAM LUTs can implement any function of n inputs. However, the area and power penalty become exponentially large as the number of inputs increase. EEPROM wired-AND logic can accommodate wide fan-in functions. However, since wired-AND logic realizes a SOP form of a function, it is unable to efficiently realize complex functions that have a significant number of product terms. FPGAs based on LUT technology are very useful at realizing random logic such as datapaths. PLA/CPLDs based on wired-AND logic are more suited for wide input functions seen in control logic and state machines, especially when non-volatility is required.

The introduction of HHE threshold logic represents a middle point in the programming technology spectrum. An HHE macrocell based on threshold logic is capable of providing some of the advantages of both SRAM LUT and EEPROM wired-AND technology. Hence, datapath logic and control logic may be efficiently implemented within an HHE-based reconfigurable system. Also, since HHE devices are inherently non-volatile, configuration data as well as state information are retained whenever power is removed.

Using SPICE, we compare each of these programming technologies in terms of power and clock frequency (delay). Figure 11 shows the relative power consumption of each technology as a function of clock frequency, while Figure 12 shows the power-delay product for each technology as a function of the number of inputs. To generate the graph in Figure 11, we simulated an 8-input XOR function implemented in each programming technology to determine the technology's efficiency in implementing complex functions. A complete input vector set (2^8 vectors) was used. This graph shows that SRAM LUT technology remains the best implementation in terms of power. Although wired-AND logic is inefficient at implementing complex functions since it requires 128 8-input AND gates, it still results in less power consumption than HHE threshold logic. Although HHE threshold logic realizes an n-input XOR function with n+1 gates [7], it is unable to compete with the other programming technologies for an 8-input XOR function. The static power consumption of each individual gate outbalances the advantage of a reduced gate count implementation. Hence, the main drawback of an HHE macrocell is power consumption since static currents are required.

As we increase the number of XOR function inputs from 8 up to 12, there is a change among the 3 programming technologies in terms of efficiently implementing complex functions. Figure 12 illustrates the power-delay product (PDP), or energy per operation, of each programming technology as the number of function inputs increase. PDP is a well-known metric in determining the advantages of different circuit technologies. We compare the relative PDP performance of each programming technology in implementing an n-input XOR function. Although a 20ns clock was employed for these simulations, the PDP value is generally independent of clock period and even gate delay as can be seen by the linear nature of the graphs in Figure 11.

Due to the exponential increase in the number of SRAM bit cells, LUT technology consumes more power than the other programming technologies when implementing wide fan-in functions. Also, since EEPROM wired-AND gates cannot efficiently implement complex functions, they also show a significant increase in power consumption. The only technology that is not severely affected is HHE threshold logic. As the number of function inputs increase, the benefit of a reduced gate count overcomes the inherent static power consumption of the HHE macrocell. Our simulations show that beyond 10 inputs, a HHE-based macrocell can efficiently implement complex functions in terms of power. A similar observation may be made regarding the area advantages of the HHE-based macrocell, except that the graphs in Figure 12 may have relative vertical shifts depending on the unit area of each programming technology.

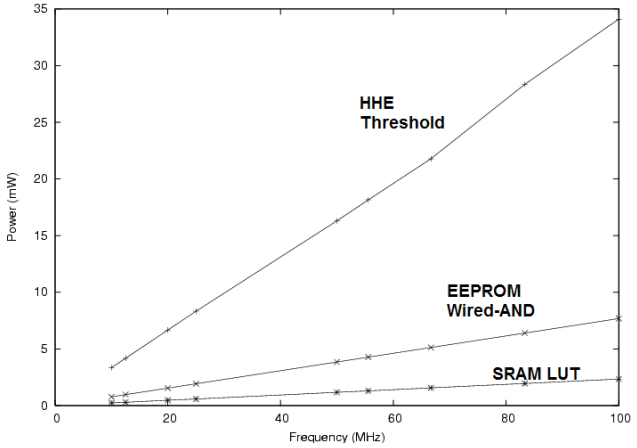


Figure 11. Power vs clock frequency for each programming technology

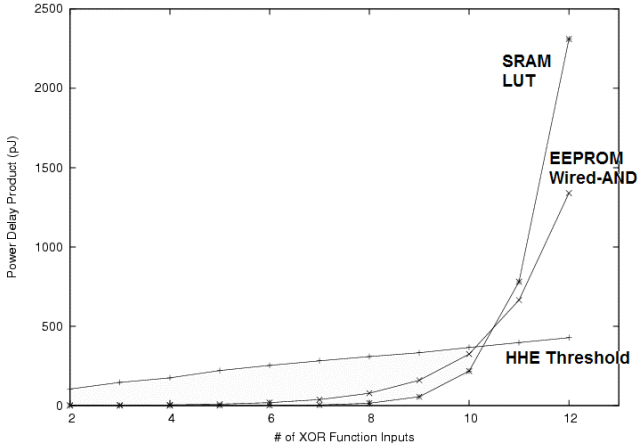


Figure 12. Power-delay product vs # of XOR function inputs for each programming technology.

The comparative simulations presented here are initial studies that show the basic properties of mapping a particular logic function onto a single programming block. If multiple programming blocks are employed, a reduction in implementation cost can be achieved. For example, a 12-input XOR function can be realized with four small 4-input LUTs rather than a very large 12-input LUT. By consuming multiple programming blocks, a reduction in area and power is possible. Similar observations are made if multiple wired-AND blocks or multiple threshold macrocells are used. Comparative simulations for entire systems incorporating multiple programming blocks are intended for future work to determine how area, power, and functional coverage are affected.

6. RELATED WORK

In the past, there have been several investigations of threshold logic. The more recent developments come from modified CMOS transistor designs that rely on capacitive sum coupling between several inputs and the transistor gate. A few examples of these threshold devices are μMOS transistors [12] and capacitive-threshold logic (CTL) gates [13]. These devices have been previously proposed as the basis for reconfigurable logic elements employing threshold logic [14]. Since these devices are capable of varying the threshold seen by the gate inputs, they are reconfigurable threshold gates. Also, these implementations are easy to integrate with conventional digital CMOS designs since the process technology is the same. However, several disadvantages exist for these devices.

First, due to capacitive discharging effects, a μMOS/CTL gate requires a periodic reset to alleviate the effects of a roaming threshold value T over time. Also, an analog voltage V_{ref} is necessary to implement the threshold value T, which requires a more complex design style beyond digital logic.

A magnetoelectronic threshold gate avoids all of the aforementioned requirements. It is also potentially more reliable than other technologies since magnetoelectronic devices are less susceptible to soft bit errors caused by alpha particles. Lastly, a reconfigurable threshold gate based on magnetoelectronic devices provides inherent non-volatile configuration and state. Hence, it is well suited for the applications that are handled by current PLA/CPLD systems.

7. FUTURE WORK

In the future, we intend to perform a more detailed analysis of the tradeoffs among power, functional coverage, and area for each of the reconfigurable technologies (LUT, wired-AND, threshold logic). This will involve mapping a broad range of real circuits such as counters and simple ALUs onto each technology. Also, CAD layout studies will be performed to determine the area of the primitive cells as well as the interconnect area of each programming technology.

We also will tackle the issue of magnetoelectronic reliability. Since input current margins decrease as the number of gate inputs increases, we need to account for intra-die process variation on the saturation currents I_{DSAT} of large CMOS transistors. We must determine how these variations consequently affect our designs. Also, we intend to develop more reliable circuit designs for magnetoelectronic devices. At the logic level, we will also develop automatic test pattern generation algorithms (ATPG) for reconfigurable threshold gates.

Our eventual goals are to develop a complete PLA/CPLD architecture composed of reconfigurable threshold gate logic based on magnetoelectronic devices. We plan to run architectural studies to determine system parameters such as the number of first-level threshold gates per macrocell required to guarantee a certain level of functional coverage. We also will explore heterogeneous reconfigurable architectures. Since a magnetoelectronic macrocell consumes significant power, it would be advantageous to include EEPROM wired-AND logic in such an architecture to handle simple functions. Complex functions with wide fan-in can then be handled by the magnetoelectronic macrocells. Due to its fast programming time, a magnetoelectronic macrocell also has the potential for dynamic reconfiguration, which has been altogether impossible in PLA/CPLD systems based entirely on EEPROM technology. Further investigations in heterogeneous architectures composed of LUTs, wired-AND, and threshold logic will be performed to see if such a system can benefit from the advantages of all three programming technologies.

8. CONCLUSION

In this paper, we have presented the first known work that proposes using magnetoelectronic devices as the underlying technology for reconfigurable threshold gates. Since power consumption is the major drawback of this technology, we introduced novel circuit techniques that reduce the static power consumption inherent in magnetoelectronic devices. We proposed a new reconfigurable macrocell that implements 2-level threshold logic, and we simulated this design in HSPICE™.

Comparisons among three programming technologies (LUT, wired-AND, and threshold logic) were made to determine the relative strengths and weaknesses of each. Although HHE threshold logic shows very promising results, there are significant barriers to its success, such as power consumption. However, as HHE device properties improve due to advances in process fabrication, an HHE-based macrocell may become a reality as a more favorable option to EEPROM or SRAM-based programming technologies.

9. ACKNOWLEDGEMENTS

The authors would like to thank Mark Johnson of the Naval Research Lab for information regarding the properties of HHE devices and for the experimental data used in our modeling. This material is based on 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 authors and do not necessarily reflect the views of the Office of Naval Research.

10. REFERENCES

[1] G.A. Prinz, "Magnetoelectronics," *Science*, vol. 282, pp. 1660-1663, Nov. 1998.

[2] J. De Boeck, "Technology and Materials Issues in Semiconductor-Based Magnetoelectronics," Semiconductor Science and Technology 17 (2002), 342-354

[3] S. Ferrera, N. Carter, "Reconfigurable Circuits Using Hybrid Hall Effect Devices," Field Programmable Logic (2003) 3-12

[4] M. Johnson, B. Bennett, M.J. Yang, M.M. Miller, and B.V. Shanabrook, "Hybrid Hall Effect Device," Applied Physics Letters 71 (1997) 974-976

[5] M. Johnson, "Magnetoelectronic Latching Boolean Gate," Solid-State Electronics 44 (2000) 1099-1104

[6] J. Rabey, *Digital Integrated Circuits*. Upper Saddle River, NJ: Prentice Hall, 1996, pp. 190-202.

[7] S. Muroga, *Threshold Logic and Its Applications*. New York: Wiley-Interscience, 1971, pp. 3-17.

[8] A.L. Oliveira and A. Sangiovanni-Vincentelli, "LSAT – An Algorithm for the Synthesis of Two Level Threshold Gate Networks," ICCAD (1991) pp. 130-133

[9] R.K. Brayton, Logic *Minimization Algorithms for VLSI Synthesis*. Boston: Kluwer Academic Publishers, 1984.

[10] B. Das and W.C. Black Jr., "A Generalized HSPICE™ Macro-Model for Pinned Spin-Dependent-Tunneling Devices," IEEE Transactions on Magnetics (1999), pp. 2889-2891.

[11] C. Wang, C. Wu, R. Hwang, and C. Kao, "A Low-Power and High-Speed Dynamic PLA Circuit Configuration for Single-Clock CMOS," IEEE Transactions on Circuits and Systems, Vol. 46, No. 7, (1999), pp. 857-861.

[12] T. Shibata, T. Ohmi, "A Functional MOS Transistor Featuring Gate-Level Weighted Sum and Threshold Operations," IEEE Trans. on Electron Devices (1992), pp. 1444-1455, June 1992

[13] H. Özdemir, A. Kepkep, B. Pamir, Y. Leblebici, U. Cilingiroğlu, "A Capacitive Threshold-Logic Gate," IEEE Journal of Solid-State Circuits (1996), pp. 1141-1150

[14] K. Aoyama, H. Sawada, A. Nagoya, K. Nakajima, "A Threshold Logic-Based Reconfigurable Logic Element with a New Programming Technology," Field Programmable Logic (2000) pp. 665-674