# *A Polymorphic Circuit Interoperability Framework*

Timothy Dunlap and Gang Qu

*Department of Electrical and Computer Engineering and Institute for Systems Research*
*University of Maryland*
College Park, USA
{tdunlap, gangqu}@umd.edu

Jinmei Lai

State Key Library of ASIC and System
Fudan University
Shanghai, China

jmlai@fudan.edu.cn

***Abstract*—The Polymorphic Circuit Interoperability Framework is presented in this paper. This framework separates the polymorphic component (called the polymorphic element) from the functional gates (called the switchable gate). The requirement of the framework is that the polymorphic element outputs a non-empty set of signals that change based on the polymorphic effect desired. In this paper, single output polymorphic elements based on voltage and clock speed are shown, and a polymorphic element based on temperature is theoretically adapted from existing literature [5]. A switchable gate that implements NAND/NOR functionality is shown and used with these polymorphic elements to test the framework for polymorphic functionality. The results are presented and polymorphic functionality is successfully demonstrated.**

***Keywords—polymorphic gate; circuit design; multifunctional circuits***

## I. INTRODUCTION

Polymorphic gates are logic gates that can change their functions based on some external factors. Since the 1970s, polymorphic circuits have been theorized, created, and their effects studied. Some possible polymorphic gates are gates that change their function based on voltage (for example, a gate that is a NAND gate when VDD is 1V and becomes a NOR gate at 3V), temperature, or an external signal [3][4][5]. Historically, these gates have been difficult to design with one common design methodology known as evolutionary algorithms [1] [2]. These algorithms attempt to discover full polymorphic gates (such as the NAND/NOR gate mentioned above) by finding the topology and the values of various parameters for a given number of transistors.

One of the very first such gates was the AND/OR gate reported by Stoica et al in 2001, where both SPICE simulation and experiments by putting an FPTA (field programmable transistor array) chip in a temperature chamber indicated that the gate behave like an AND gate at one temperature, but changes to an OR gate at a different temperature. This interesting feature, namely one logic gate can implement multiple functions, is made possible because polymorphic gates have unconventional structure at the transistor level. They can be implemented with FPTA and CMOS, emerging devices such as silicon nanowire (SiNW), and ambipolar devices.

When Stoic et al. reported the first polymorphic circuit, their goal was to reduce the hardware cost (or number of transistors) to implement multiple functions. Recently, polymorphic circuits find many applications where a few predefined functions have to be implemented and a global control signal is used to select the function, such as multifunctional adaptive systems, multifunctional image filters, finite impulse response filter, and self-checking circuits. With these emerging applications, synthesis and testing techniques for polymorphic circuits have also been proposed.

In [8], the authors implemented a 48-bit chip ID on the reconfigurable polymorphic REPOMO32 chip. Their basic idea is to use the different switching time of the polymorphic gate caused by fabrication variation. In [9], the authors demonstrate how the emerging nano structures can be used as hardware security primitives where they discussed five circuit structures including polymorphic gates based on silicon nanowire (SiNW) and how to use them for circuit obfuscation. Replacing selective standard logic cells by polymorphic gates can be used for watermarking [5] and fingerprinting [6].

In this paper, we propose a Polymorphic Circuit Interoperability Framework for the design and construction of polymorphic gates. This framework separates the creation of the polymorphic functionality (such as switching due to the supplied voltage) from the gate functionality (such as NAND and NOR). This paper also presents a method for reconfiguring currently existing polymorphic gates into this framework.

The rest of the paper is organized as follows. In Section II, we propose the polymorphic interoperability framework with a detailed explanation on its two main segments: the polymorphic element and the switchable gate. We demonstrate how to apply this framework to a known polymorphic gate and report the simulation results in Section III before conclude in Section IV.

## II. POLYMORPHIC INTEROPERABILITY FRAMEWORK

The proposed Polymorphic Interoperability Framework contains two main segments: the polymorphic element and the switchable gate. These two segments are combined together with an interface consisting of a series of signals (at least one signal is required). In this section, we elaborate these two segments.

### *A. Polymorphic Element*

The polymorphic element exists as the portion of the system that is acted on by the environment to change state. This change is output to the switchable gate through the interface. A polymorphic element that switches state based on the voltage of the system is shown in Figure 1, where this element uses a single

output through the interface. However, it is possible to use multiple outputs through the interface, utilizing either multiple levels of the same polymorphic element, or combining multiple polymorphic elements into a single group.

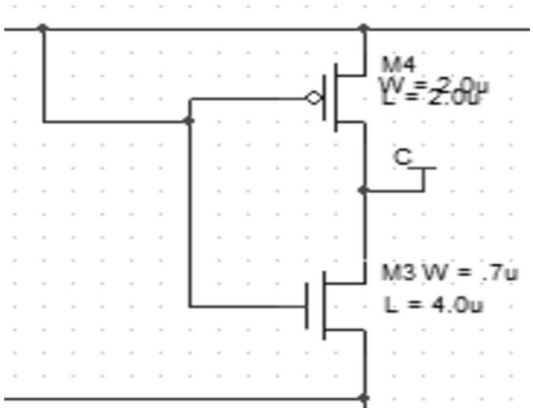



FIGURE 1. EXAMPLE SHOWING A POLYMORPHIC ELEMENT WHERE 'C' DEPENDS ONLY ON THE VOLTAGE DIFFERENCE BETWEEN THE TOP AND BOTTOM RAILS.

The polymorphic element is the crux of the framework and makes the gate polymorphic. Currently existing methods can be employed to create this polymorphic element can be found in [1] which discusses the synthesis techniques of polymorphic circuits using NAND/NOR gates. Of these methods, we will elaborate two: an evolutionary algorithm and a skilled designer.

One of the most common methods of polymorphic circuit design is to use evolutionary algorithms [1]. These algorithms start by randomly combining transistors, testing them, and combining different partial solutions to eventually create a circuit that completes the needed functions [5]. In order to use an evolutionary algorithm with this framework, the target design must be set to interact with the interface (which is commonly just a single signal or a small number of signals, depending on the type of switchable gate). This requirement is notably simpler to design when compared to a standard polymorphic gate, since only a single output condition must be created instead of every output of a gate's truth table. This also highlights one of the major benefits of this framework: existing circuit reusability.

Much work has been done towards developing different polymorphic circuits [1] [2] [3]. To leverage as much previous work as possible, it is important for this framework to be able to make use of currently existing polymorphic circuits. In order to convert a currently existing polymorphic gate to the framework, an input combination that expresses a different output must be forced. For example, in [5] a NAND/NOR temperature polymorphic is displayed. If input A is forced high and B is forced low the output switches exactly with the polymorphic state. This concept can be extrapolated to a generic gate because by definition a polymorphic gate must be a combination of two distinct gates, and by definition of distinct gates there must exist some input combination that causes the output to differ. Using this principle, polymorphic elements can be crafted from existing gates described in the literature.

Another option for creation of these polymorphic elements is for a skilled designer to invent them [1]. Designing the polymorphic element is an easier task when compared to designing the entire gate since the design must only conform to the interface, or a very sparse set of signals, and not handle the inputs that a gate would. Two of the polymorphic elements described and simulated in the section below have been designed by hand.

### *B. Switchable Gates*

Switchable gates are gates whose function depends on the input from external signals. The external signal, in this case, comes from the polymorphic element through the interface. Examples of switchable gates can be seen in any previous work that depends on an external signal, and these circuits can, in general, be used interchangeably with the switchable gates presented here.

In this paper, one switchable NAND/NOR gate will be discussed. This circuit was created with standard CMOS creation methods based on the truth table, with a control signal from the polymorphic element. When the control signal is 0, the circuit works as a NAND gate and otherwise it works as a NOR gate. The resulting circuit is shown in Figure 2. A particular benefit of this framework is that the switchable gates are able to be minimized easily using standard circuit minimization techniques, such as using a Karnaugh map, given the small size of the switchable gates.

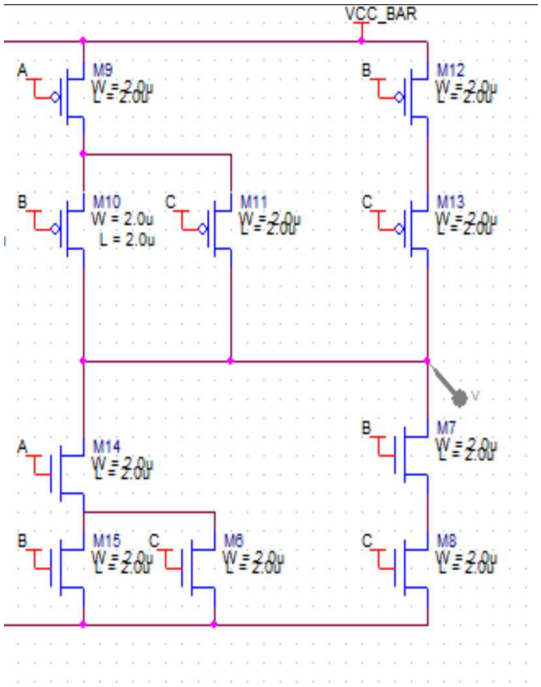



FIGURE 2. SWITCHABLE GATE TO IMPLEMENT NAND/NOR GATES. SWITCHES FROM NAND TO NOR BASED ON 'C' GOING FROM LOW TO HIGH.

## III. CIRCUITS AND SIMULATION

The polymorphic element designed to change result based on clock speed is shown in Figure 3. The gate on M3 is constantly held at 0.7V, which allows relatively small amount of current to be pulled to ground from the clock signal. This forces the value of the clock to remain close to zero for high clock speeds because it cannot charge the capacitor in the time that the clock

signal is high. This forces the output signal (C) to be high, which goes to the interface and the switchable gate.

The polymorphic element created for voltage-level based switching is shown in Figure 4. This circuit has varied W/L values for the M3 transistor, which causes the output to switch around 1.2V, as shown in Figure 5. This switching voltage becomes the threshold for the output changing from 1 to 0 and is the signal that is given to the interface.

The switchable gate designed for testing these polymorphic elements is a NAND/NOR gate for the control signal being 0/1, respectively. This circuit becomes a majority wins circuit where if two or more low signals are detected, the circuit output is high. The circuit used is shown in Figure 2. When 'C' is tied to the interface, the circuit behaves as expected, as shown in Figure 6.

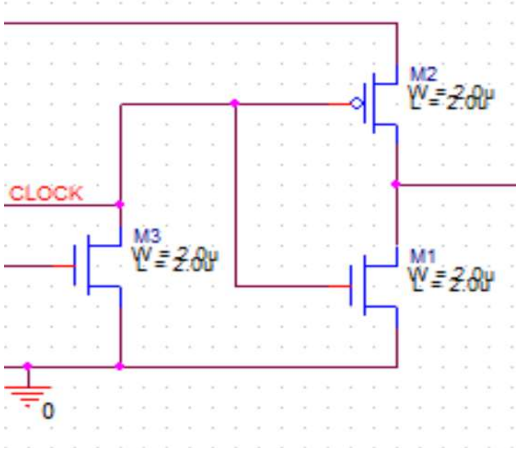



FIGURE 3. POLYMORPHIC ELEMENT TO SWITCH BASED ON THE CLOCK SPEED RUNNING ON THE 'CLOCK' SIGNAL.

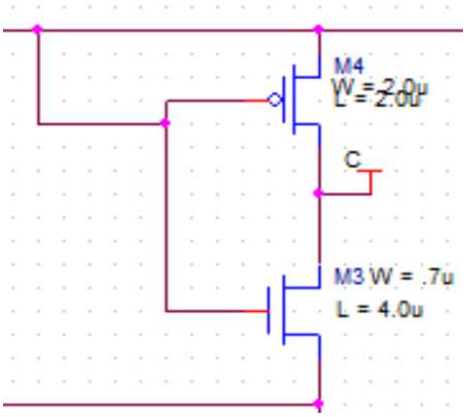



FIGURE 4. POLYMORPHIC ELEMENT TO SWITCH BASED ON THE VOLTAGE APPLIED ACROSS THE TOP AND BOTTOM RAILS. OUTPUT 'C' SWITCHES AROUND 1.2 VOLT.

The polymorphic elements are connected to the switchable gate via the interface (signal 'C') and simulated across all A/B input possibilities and the polymorphic parameter. The results were as expected and documented in Table 1 as well as shown in Figure 7. In Figure 7, A and B show the result of the voltage polymorphic element while C and D show the result of the clock speed polymorphic element. The voltage simulation was run at 5 volts and 1 volt for A and B respectively. The clock speed was run at 50 MHz and 50Hz, respectively. The signal should be sampled on the rising edge of the clock.

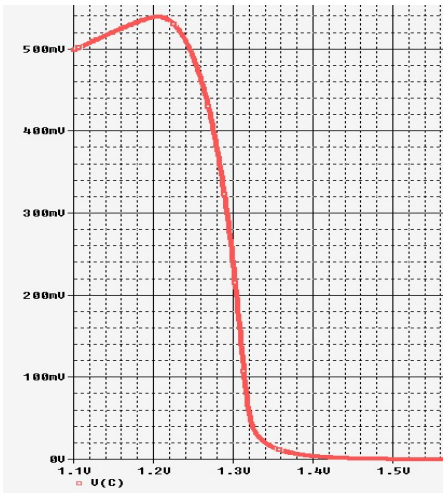
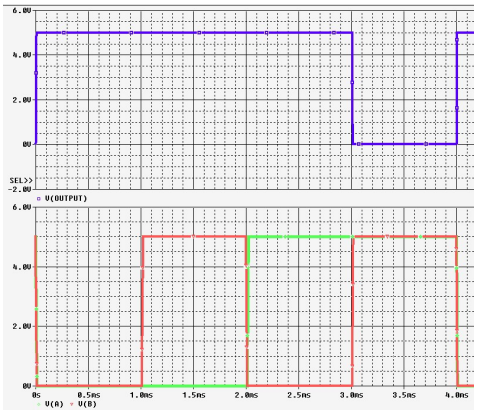
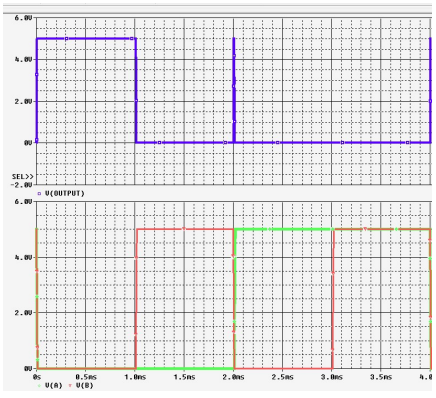



FIGURE 5. GRAPH OF VOLTAGE DEPENDENT POLYMORPHIC ELEMENT SIMULATION.

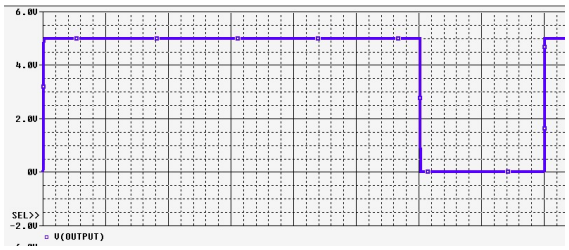



(a)

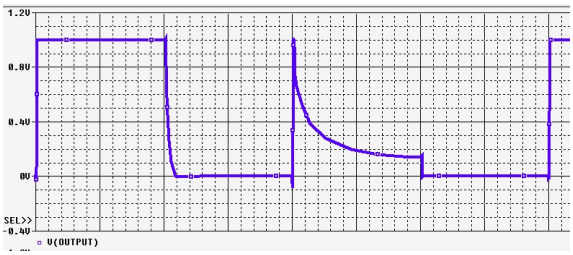



(B)

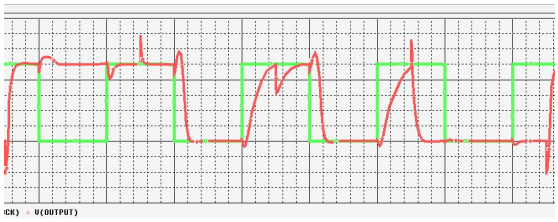
FIGURE 6. SHOWS THE OUTPUT OF THE SWITCHABLE NAND/NOR GATE. A IS WITH 'C' LOW AND B IS WITH 'C' HIGH



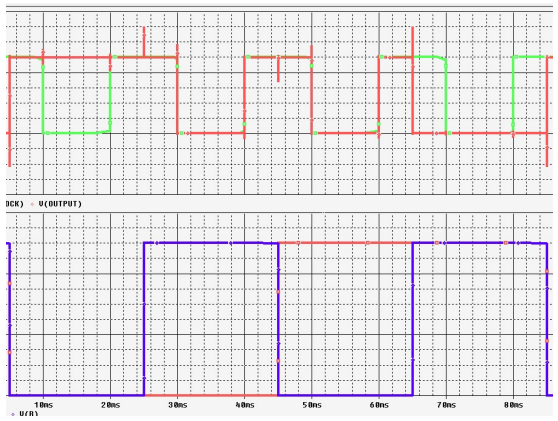

(A)




(B)




(C)




(D)

FIGURE 7. A AND B SHOW THE RESULT OF THE VOLTAGE DEPENDANT POLYMORPHIC ELEMENT TIED TO THE SWITCHABLE GATE. C AND D SHOW THE RESULT OF THE CLOCK SPEED DEPENDENT POLYMORPHIC ELEMENT.

## IV. CONCLUSION

A framework for the creation and modification of polymorphic circuits is introduced in this paper. The framework is built around the idea of separating the functional gates from their polymorphic components. Once these elements are separated, they can be interchanged at will to create new polymorphic gates. This allows optimizations created for one gate to be applied across numerous different polymorphic gates. Examples given are based around a NAND/NOR switchable gate and multiple polymorphic elements: voltage switching, clock speed switching, and temperature. Simulation results of this process show expected results that confirm the polymorphic behavior of this framework. The next steps in developing this framework include creating and simulating multiple signal interfaces as well as the corresponding polymorphic elements and switchable gates. Further work also will include applications of this framework to include circuit obfuscation/encryption and circuit fingerprinting/watermarking.


## Acknowledgments

Timothy Dunlap and Gang Qu were supported in part by the National Science Foundation under grant CNS1745466 and by a research agreement between the University of Maryland and the Laboratory for Physical Sciences.


## REFERENCES


[1] A. Crha, R. Ruzicka and V. Simek, "Synthesis Methodology of Polymorphic Circuits Using Polymorphic NAND/NOR Gates," *2015 17th UKSim-AMSS International Conference on Modelling and Simulation (UKSim)*, Cambridge, 2015, pp. 612-617.

[2] A. Stoica, R. Zebulum, D. Keymeulen, J. Lohn, "On polymorphic circuits and their design using evolutionary algorithms," Proc. of Lasted International Conference on Applied Informatics (AI2002), 2002.

[3] A. Stoica, R. S. Zebulum, X. Guo, D. Keymeulen, M. I. Ferguson and V. Duong, "Taking evolutionary circuit design from experimentation to implementation: some useful techniques and a silicon demonstration," in *IEE Proceedings - Computers and Digital Techniques*, vol. 151, no. 4, pp. 295-300, 18 July 2004.

[4] A. Suarez, H. Oro, L. Peñaredonda, R. Anacan and M. N. Pangilinan, "Design of a New External Signal Controlled Polymorphic Gates," *2016 7th International Conference on Intelligent Systems, Modelling and Simulation (ISMS)*, Bangkok, 2016, pp. 413-418.

[5] T. Wang, X. Cui, D. Yu, O. Aramoon, G. Qu, and X. Cui, "Polymorphic gate based IC watermarking techniques," *2018 23rd Asia and South Pacific Design Automation Conference (ASP-DAC)*, Jeju, 2018, pp. 90-96.

[6] T. Wang, X. Cui, D. Yu, O. Aramoon, T. Dunlap, G. Qu and X. Cui. "A novel polymorphic gate based circuit fingerprinting technique", *28th IEEE/ACM Great Lakes Symposium on VLSI (GLSVLSI'18)*, pp. 141-146, May 2018.

[7] A. Stoica, R. Zebulum, D. Keymeulen, "Polymorphic electronics", Proc. of Evolvable Systems: From Biology to Hardware Conference, LNCS 2210, Springer 2001, pp. 291–302.

[8] Sekanina, Lukas, et al. "Implementing a unique chip ID on a reconfigurable polymorphic circuit." Information Technology And Control, 42.1 (2013): 7-14.

[9] Bi Y, Shamsi K, Yuan J S, et al, "Emerging technology-based design of primitives for hardware security," ACM Journal on Emerging Technologies in Computing Systems (JETC), 2016, 13(1): 3.