Present and the Future of Chaos Computing

Behnam Kia, Vivek Kohar, William Ditto

Department of Physics, North Carolina State University, Raleigh, North Carolina 27695-8202 USA

Abstract We study chaos computing as a new approach for reconfigurable computing and present some of our latest results and discuss what this new direction to computing means and implies. We discuss the advantages and challenges that come with this new paradigm of computing and envision its future.

1 Introduction

Nonlinear dynamics can be considered as a rich library of different behaviors. More specifically, by changing a bifurcation parameter of a nonlinear system, the dynamics of the system qualitatively changes. For example, one bifurcation parameter value puts the nonlinear system in a periodic regime, whereas another parameter value can move the system to a chaotic regime [1]. Also, when a nonlinear system is in a chaotic regime, its chaotic attractor is composed of infinite number of different unstable periodic orbits, where each can be dynamically selected and stabilized [1]. This rich dynamics of a nonlinear system, combined with our ability to control and program it, enables us to introduce fascinating applications for it that otherwise are not possible through conventional engineering or by linear systems.

Chaos computing is a term that refers to the methods and approaches to utilize nonlinear dynamics and chaos to perform computing [2,3]. The main advantage of chaos computing is that the same circuit can be programmed to implement different types of computation [4,5]. In this approach, since the nonlinear dynamics, which contains many different behaviors, is performing the computation, different types of computation coexist within the system and we can dynamically select and pick up different types of computation from the system [4,5].

This capability of a circuit to implement different types of functions is of crucial importance in today's troubled semiconductor industry, where we cannot shrink the size of the transistors anymore to integrate more transistors into a mi-

croprocessor [6]. In this paper, we discuss this problem and argue how our reconfigurable computing can address this issue.

In section 2, we will discuss the Moore's law, and describe the current challenges that the semiconductor industry is facing, and what is available beyond the Moore's law. In section 3 we review chaos computing, and in section 4 we present some of our circuit designs. In section 5, we discuss challenges facing chaos computing, the compatibility of chaos computing with conventional technology process, and the future of chaos computing. The paper is concluded in section 6.

2 The Moore's Law and Beyond

Historically, the number of transistors on microprocessors has been doubling every two years. This rule of thumb, which is called the Moore's law, has been used as a roadmap for semiconductor industry, and this exponentially increasing number of transistors on microprocessors has resulted in greater performance in each generation of processors compared to the previous generation [7]. The main method to integrate more and more transistors on a chip was to shrink the size of the transistors. But the problem is that the size of the transistors is getting so small that such devices are no longer governed by the rules of classical mechanics, instead, they enter the realm of the quantum mechanics [8]. For example, when the size of channel, the distance between source and drain in a Metal Oxide Semiconductor Field Effect Transistor (MOSFET) shrinks below 10nm, there will be only a few silicon atoms remaining between the source and the drain. At such small feature sizes and with so few atoms, it is the quantum mechanics that governs such devices. As a result, there will be uncertainty in the operation of the device and it will operate in a way that is beyond the classical operation of a MOSFET device. For example, when the device is off, the classical mechanics tells us that there should be no current between the terminals of the device, however, due to quantum tunneling there would be carriers tunneling through the channel and it will create a leakage current that is beyond the scope and explanation of classical mechanics. Long story short, a MOSFET device with a very small feature size will not operate as a perfect switch and this is a challenge that the semiconductor industry is facing. Furthermore, even if the semiconductor industry can come up with a way to control the leakage current, still the shrinking of the transistors may not be scalable any more because we are left with very few silicon atoms. With further shrinking of the transistors, assuming that leakage current can be controlled and managed, we will end up with a channel that is only one atom thick.

There are ongoing research approaches to address the aforementioned challenges of the semiconductor industry. For example, if after further shrinking of transistors it is the quantum mechanics that is governing these devices, why not use the rules and laws of quantum mechanics to design new type of devices that operate based on tunneling effect. Such devices that are supposed to operate based

on quantum tunneling are called Tunneling Field Effect Transistors, or in short TFET [9]. Or if the silicon atoms in the channel are numbered and we might reach to a single atom device, why not use alternative materials that have smaller atom size to give us more number of atoms to work with? Or what about sub-atomic devices that use electrons or protons as their main building blocks? These are all ongoing research fields, but despite the tremendous amount of resources allocated to these fields, it is unknown when the resulting device would be operational and ready for commercialization.

There is another approach to enhance the performance of microprocessors that focuses more on the better utilization of the transistors instead of increasing their number on the chip. In this approach, the focus is on increasing the average amount of computation that is performed by each transistor on the processor. For example, a common idea is that rather than doubling the number of transistors on a processor what if we increase the performance of the current transistors so that they operate as if there are twice of them on the chip. Such approaches are commonly called design based equivalent scaling, alluding to the fact that the result of such design based enhancement is equivalent to scaling the transistors and integrating more of them into the microprocessor [8]. Our dynamics based approach is a type of design based equivalent scaling, in the sense that it utilizes the complex dynamics of fewer transistors to perform multiple different functions that otherwise in conventional technology had to be fabricated individually and separately with more number of transistors. We study this approach in this paper.

3 How Chaos Computing Performs Reconfigurable Computing

A function is a machine that maps its inputs to outputs. A dynamical system maps its initial state to future states. Therefore, a dynamical system is an embodiment of a function. The real world signals and systems are mostly analog, meaning that dynamical variables are continuous in value. In this paper we focus on implementing digital functions, as a result, we have to bridge the digital and analog signals together. Figure 1 shows a dynamical system, how it maps its initial state to future states, and how we translate digital inputs to an analog initial condition of the dynamical function, and how we decode a digital output from the final analog state of the dynamical system. Here we use a very simple Digital to Analog Convertor, DAC, to convert the digital inputs to an analog initial condition, and a simple threshold mechanism to convert the analog output to a digital output. Note that in this specific block diagram we have assumed that the dynamical system is 1-D, therefore it has just one single state variable. If we use a dynamical system of higher dimension, e.g. m, then up to m different digital data inputs can be directly used to initialize an m-dimensional state with no need for a DAC to convert m binary inputs to a single analog value.

A simple threshold mechanism produces the output O from the final state x_t based on a threshold value τ:

$$\mathbf{O} = \begin{cases} 0 & x_t < \tau \\ 1 & \tau \leq x_t \end{cases} \quad (1)$$

Nonlinear dynamics contains many different dynamical behaviors, where each of them is dynamically selectable. As a result, by selecting a different behavior from the rich library of a nonlinear dynamical system, one can implement different mappings between inputs and outputs, and as a result different functions. This is the main idea behind chaos computing. One circuit can implement different functions as different functions coexist within the dynamics of the same circuit. Furthermore, programming the circuit to implement different functions can be extremely quick, and indeed in our latest fabrications, the chaos computing circuit was able to implement a different function at each clock cycle with no need for a halt in-between for reprograming.

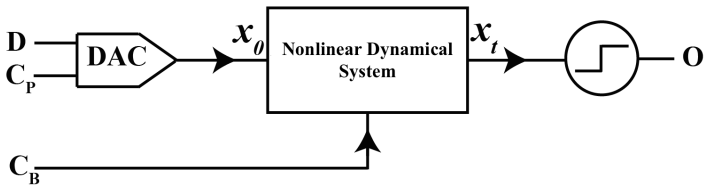

Fig. 1: Block diagram of a 1-*D* chaos computing system

In Fig. 1. Block diagram, binary data inputs, denoted by **D**, are encoded as an initial condition of the nonlinear system, x_0. And an output O is decoded from the final state of the nonlinear system, x_t. Two control inputs C_B and C_P reconfigure and program the nonlinear dynamical system to implement different functions. C_B sets the bifurcation parameter of the nonlinear dynamical system and changes the type of dynamics of the system and as a result, changes the function the nonlinear dynamical system implements.

Chaos control has taught us how to control and tame chaos and how to stabilize different behaviors within the chaotic attractor. However, a majority of these methods are feedback control methods to detect and stabilize the unstable periodic orbits. These chaos control methods can be slow in the sense that they might need some time until they can stabilize the desired orbit, and in addition, they can be complex to implement. Each chaos computing gate will need an instance of these control modules, and it needs to be very quick, otherwise the processing can be slow. As a result, we cannot afford to have a large and slow control circuit for every single chaos gate, especially by considering that we might need to have millions of such chaos gates on the chip. Here we use a much simpler feed-forward mechanism to pick up different behaviors from the chaotic system. This mechanism is a simple bias voltage addition, C_P, which shifts the produced initial condition to dif-

ferent parts of the attractor and therefore changes the behavior of the orbits and the type of the functions that the circuit implements.

4 Circuit Design

A question at this point could be how to get nonlinearity and chaos at the circuit level? The truth is that nonlinearity is already there; all physical systems by nature are nonlinear, it is actually linearity that is hard to find and create. The intrinsic nonlinearity of transistors is enough to create the nonlinear dynamics at the circuit level, and that is how we have implemented chaos computing.

We have designed different versions of chaos computing with different nonlinear map circuits. For example, in [4] we used a simple three-transistor nonlinear map. And in [5] we designed and fabricated a slightly larger, but comparatively robust nonlinear map. This nonlinear map circuit is shown in Fig. 2, and Fig. 3 shows the circuit design for a chaos computing logic block. This circuit is capable of implementing different two-input one-output combinational functions. By changing the bifurcation inputs C_3 and C_4 and perturbation inputs C_1 and C_2 one can change the type of digital function that the Fig. 3 can implement [5]. For example, when the control bits C_1 C_2 C_3 C_4=0011, the O_1 output of Fig. 3 circuit gives AND of inputs I_1 and I_2 when the iterated map iterates just once, or when the control bits C_1 C_2 C_3 C_4=0110, the O_1 output of Fig. 3 circuit gives NOR of inputs I_1 and I_2 when the iterated map iterates once. For more details please refer to [5].

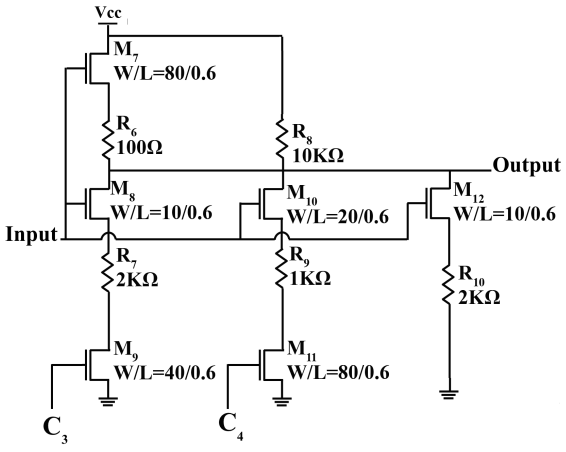

Fig. 2: Nonlinear Map Circuit, NMC. Two bifurcation control bits C_3 and C_4 shape the nonlinearity of the map, and how Input is mapped to the Output.

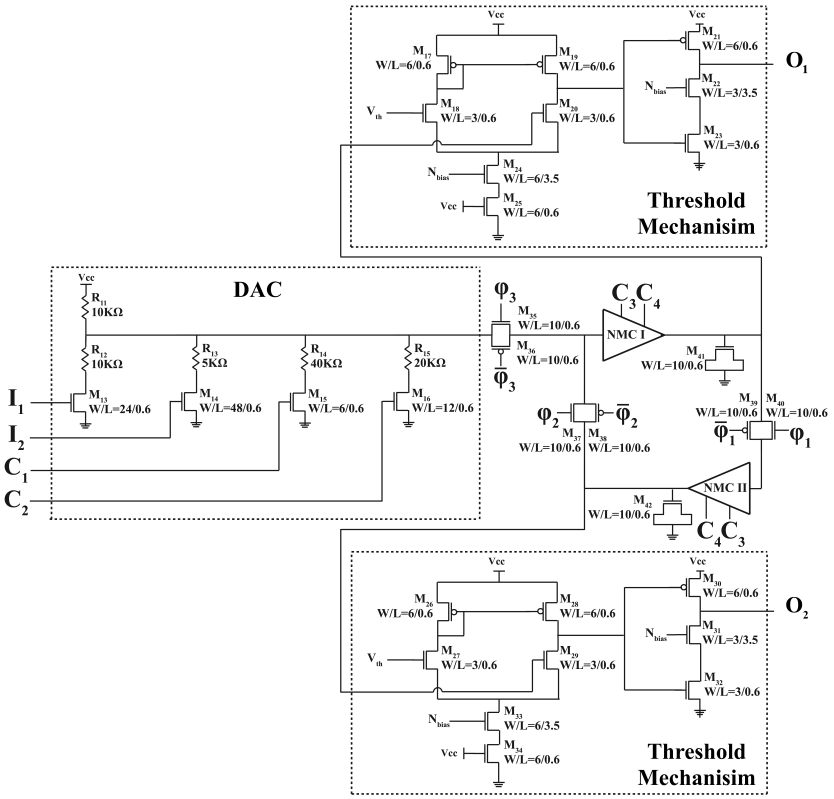

Fig. 3: Two NMCs are chained together to create an iterated map. DAC maps inputs I_1 and I_2, biased with control inputs C_1 and C_2, to an initial condition for the iterated maps. Two threshold circuits produce binary outputs, O_1 and O_2, from the final state of each NMC.

5 Challenges, Compatibility, and the Future

5.1 Challenges

Robustness of nonlinear circuits discussed in section 2 against variabilities such as temperature variations, fabrication variations, and source voltage variations are of great importance. If there is a variation in a fabrication parameter or the operation condition of the chip, there is a possibility that it can change the dy-

namics of the circuit, and because the computation is dynamics based in these systems, the type of computation that these circuits can implement might change as well. As a result, we have to make sure that the dynamics of the circuit does not change by such variations, or at least if the dynamics changes, the type of computation that the dynamics implements does not change.

We used CAD tools to simulate many of published chaotic circuits to investigate their robustness. It turned out that the input-output characteristics of these circuits vary with change in temperature. Chaos computation requires that the characteristics of the nonlinear circuit remain constant with temperature fluctuations, otherwise, the implemented function will depend on the temperature of the chip as well. This is a fairly new specification for nonlinear, chaotic circuit design, and many published circuits are not intended for this requirement. In previous papers, the chaotic circuits were intended for random number generation, and they were robust enough to insure that the circuits remained in the chaotic regime despite the temperature variations. But for our application, the circuit characteristics had to remain fixed – to the extent that the type of functions does not change due to temperature variations. The specific circuits that we have designed and fabricated in this cycle are robust, and we expect further research can extend the level of robustness of these circuits.

5.2 Compatibility

One of the major challenges in adoption of a new technology is its compatibility with existing infrastructure, fabrication facilities and process, CAD tools, and the background and skill set of the personnel in the industry. A new technology that can contribute a slight improvement in performance over the existing systems, but requires a completely different fabrication process, needs an absolutely different CAD tools and software to operate, and the entire personnel and workforce has to be retrained to use the new technology, is not a technology that can be easily adopted by the industry. The revenue or extra value that comes out of a new technology has to justify the amount of investment for new fabrications plants, CAD tools, and retraining the personnel. One of the major advantages of Chaos Computing is that it uses the existing technology for fabrication, and the very same CAD tools and fabrication processes that are used for conventional integrated circuit technology can be used for design and fabrication of chaos computing. Chaos computing is a logic design at the circuit level, which uses the same device technology as the conventional CMOS technology. For example, in our case, we used standard, commercial grade Cadence design suite for design and simulation of the circuits. The designed circuit was fabricated using MOSIS multi-project wafer service, and as a result, the chaos computing chip actually shared a wafer with

other conventional designs. This manifests the ultimate compatibility of this technology with conventional technology.

One of the questions about chaos computing is its future if a new generation of transistors or devices is introduced. Chaos computing is dynamics-based circuit level design method, in the sense that it is independent of a specific generation of transistors or devices. Even if a new generation of transistors replaces the existing transistor technology, we still can utilize the nonlinearity of the new devices and design and implement chaos computing based on this new generation of devices.

5.3 Future

So far our lab has successfully designed and fabricated individual chaos computing logic blocks that can be programmed perform different logic operations [5]. We are working on enhancing and improving the performance of the logic block in terms of power consumption, speed, area on the chip, and last but not the least, robustness against variabilities.

Besides optimizing an individual logic block, another important goal of our lab is to design and fabricate chaos computing systems, where rather than an individual logic block, a series of logic blocks are fabricated and work together to implement a higher level type of function, such as a 16-bit addition operation. Our vision is to transform the flexibility and reconfigurability of chaos computing from circuit level and gate level to system level, where a chaos computing system can implement a different function at each cycle. Any computer application or program is a sequential list of instructions. In our approach, this sequential instructions will be fed to the chaos computing system, one instruction at each clock cycle, and the chaos computing system at each cycle will be able to reconfigure itself to implement that exact instruction and perform that type of computation on the operands and input data of that instruction. ***I***

In conventional microprocessors, for every machine level arithmetic and logic instruction there is a separate circuit that implements that specific instruction. In our approach, there would be just one type of circuit that implements all instructions. This will cut down the number of dedicated circuits for specific functions that occupy the chip and consume power and may or may not be useful depending on the type of computation that the program or application demands. The processor will be composed of identical, but reconfigurable circuits that can be easily reconfigured to perform the required type of computation. This approach will allow us to fully utilize all the transistors on the chip, rather than the transistors that are a part of circuits that implement the desired operation at that specific time. According to Moore's law, we have been doubling the number of transistors on a microprocessor every two years, but on the other side, these many transistors on a chip produce excessive heat that we are unable to sink out of the chip. As a result, in a modern processor, we keep just a portion of the processor chip live and powered up and the rest is turned off. Chaos-based reconfigurable computing is proposing a solution to this problem. Rather than having billions of transistors on a conventional processor to perform many tasks that you may not need very frequently, implement the processor with a fraction of those transistors, and use circuit level reconfigurability to reconfigure the circuits to implement any desired task and operation. Since there will be very few transistors on the chip, heat production and sink will not be a problem anymore.

6 Conclusions

Chaos computing utilizes the rich dynamics of nonlinear systems to perform reconfigurable computation. Nonlinearity and chaos can be easily and conveniently implemented at the circuit level by utilizing the inherent, intrinsic nonlinearity of transistor circuits. The main advantage of chaos computing is that a circuit can implement many different functions. As a result, fewer transistors can implement a computing system that otherwise would have needed a much higher number of transistors if fixed, dedicated circuits were used. With this approach, we do not need to exponentially increase the number of transistors on a microprocessor over generations to provide a higher performance, rather we just need to utilize the existing transistors better by performing computation using their complex dynamics.

We also discussed that chaos computing is compatible with conventional semiconductor process and design tools, and as a result it does not require extensive change in process technology or tools. Chaos computing utilizes the conventional semiconductor technology, and it opens the doors to increase the performance of computers with no need for substantially changing the fabrication process or technology.

---

[1] Ott, Edward. Chaos in dynamical systems. Cambridge university press, 2002.

[2] T. Munakata, S. Sinha, and W. L. Ditto IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications **49**, 1629, (2002).

[3] B. Kia, J. F. Lindner, and W. L. Ditto, Frontiers in computational neuroscience **9** (2015).

[4] B. Kia, J. F. Lindner and W. L. Ditto, IEEE Transactions on Circuits and Systems II: Express Briefs, **63** (10), 944, (2016).

[5] B. Kia, K. Mobley, and W. L. Ditto. IEEE Transactions on Circuits and Systems II: Express Briefs. doi: 10.1109/TCSII.2016.2611442

[6] I. L. Markov, Nature **512**(7513), 147 (2014).

[7] C. A. Mack, IEEE Transactions on semiconductor manufacturing, **24**(2), 202 (2011).

[8] A. B. Kahng, in Proc. Design Automation Conference, (2013).

[9] B. Ganjipour, et al. ACS nano **6**(4), 3109 (2012).