Chua's hardware random number generator

Michael Battaglia and Yuan Qi Ni

17 April 2017

1 Introduction

Evolution of chaotic systems is defined by density of periodic orbits in phase space, topological transitivity, and sensitive dependence on initial conditions. The last of these conditions is equivalent to the more common statement that the measured uncertainty in initial position of a point in the system's phase space will diverge exponentially as the system evolves. The timescale after which the position of the particle becomes essentially unpredictable is the Lyapunov time $1/\lambda$, and uncertainty evolves as the Lyapunov exponent λ.

$$|\delta(t)| = e^{\lambda t}\delta(0) \tag{1.1}$$

This motivates us to use a chaotic system with a short Lyapunov time as a way to quickly draw hardware random numbers with rate proportional to the inverse of the Lyapunov time. The following document describes the design and characteristics of the Chua's Hardware Random Number Generator (HRNG). The advantage of this over thermal noise based random number generators is relative insensitivity to uncontrollable, not necessarily random, noise sources. The signal in a chaotic circuit is macroscopic and will drown this out.

2 Theory

Leon Chua (1993 [1]) describes a very simple method of observing chaos in his invention, the Chua's Circuit. It consists of only four components; the Inductor, the Resistor, the Capacitor, and the Chua's diode (Figure 1). It is described by the following system of differential equations which we easily derived using Kirchoff's loop and node analysis. Here x (voltage across C1), y (voltage across C2), and z (current through L) are observables in the circuit.

$$R \times C1\frac{dx}{dt} = \alpha(y - x - RG(x)) \tag{2.1}$$

$$R \times C2\frac{dy}{dt} = y - x + Rz \tag{2.2}$$

$$L\frac{dz}{dt} = -y \tag{2.3}$$

Leon's paper shows that one is able to observe bifurcations, period doubling, and strange attractors, all characteristics of chaotic systems. The Chua's circuit oscillator in particular is characterized by the "Double Scroll" strange attractor which can be made symmetric about the x observable. This very useful characteristic allows us to digitize a trajectory in the Chua circuit's phase space as 0 (if in the left scroll) and 1 (if in the right scroll). Sampling the trajectory at greater than lyapunov timescale allows us to essentially draw from a uniform or weighted distribution over the set 0, 1. To any precision needed, we can take corresponding number of bits in base 2 to finely partition the interval [0, 1] which allows us to draw random numbers.

Chua's Circuit (and the Van der Pol oscillator) are often taken as the canonical examples of chaos in electric circuits. MKaouar et al. (2012 [2]) and Chua el al. (1993 [1]) document in great detail how one may go about building the Chua's diode. The Chua's diode is fully characterized by its current vs voltage function G (piecewise Gb, Ga, Gb) which is exactly the G function in Chua's system of equations 2.3. Two negative impedance converters in parallel gives the nonlinear negative resistance needed. We

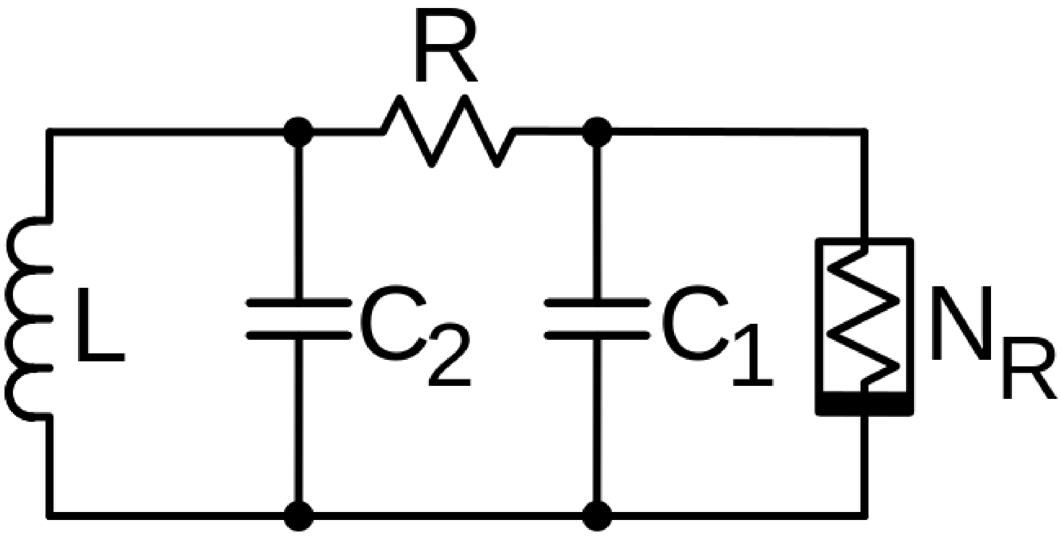

Figure 1: Chua's Circuit.

will build our Chua's diode in the same way. This function is plotted below in Figure 2. The values of Ga and Gb can be selected by the resistor values R1 to R6 as in Equation 2.6.

$$\begin{align} Ga &= -\frac{R2}{R1R3} - \frac{R5}{R4R6} \tag{2.4}\\ Gb &= -\frac{R2}{R1R3} - \frac{1}{R4} \tag{2.5}\\ & \tag{2.6} \end{align}$$

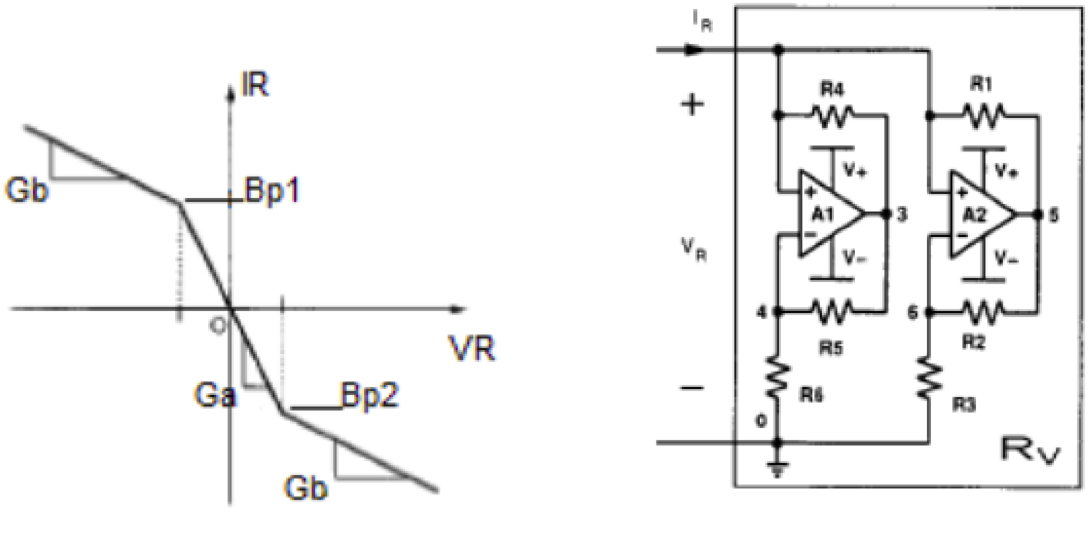

Figure 2: Chua's diode constructed using op-amps and resistors [2].

It turns out Chen et al. (2001 [3]) have already built such a device that extracts random bits from

Chua's circuit. We will take the values of capacitance, inductance, and Chua diode characteristic slopes used by Chen as typical values to be used in our Chua's circuit.

At high frequencies, inductors present significant noise. Sedra and Smith (5th ed. pg.1112 [4]) describes how to use op-amps to simulate the transfer function of an inductor (Antoniou's circuit, Figure 3).

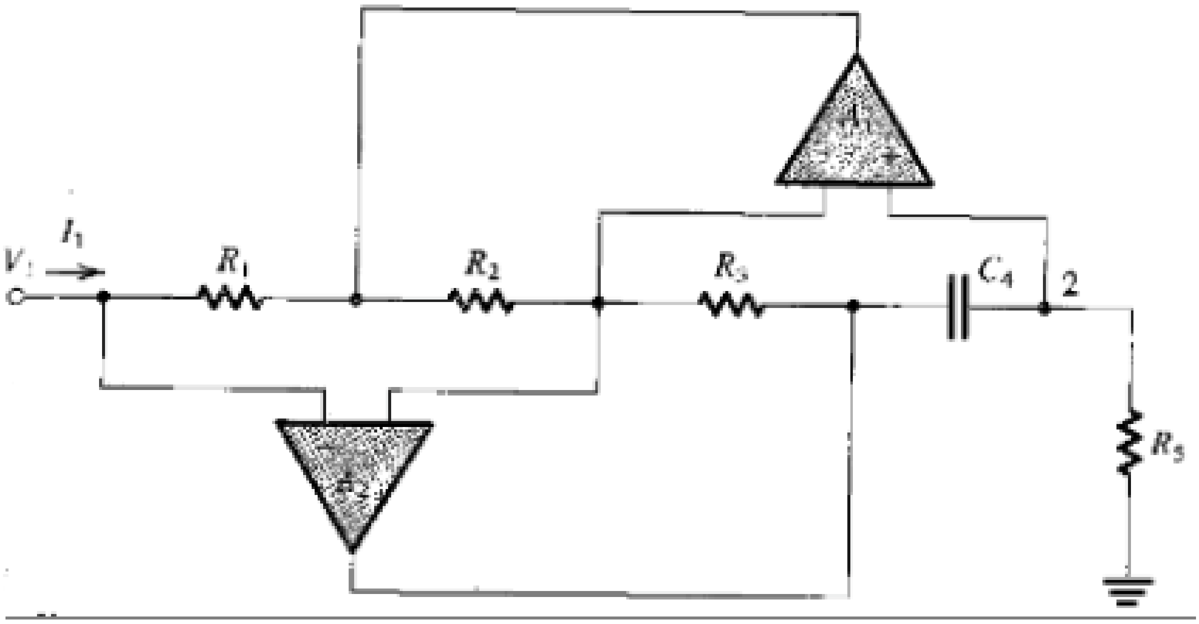

Figure 3: Antoniou's Inductor Simulator [4]

Inductance of this circuit is controlled by the various resistors and capacitors as the following.

$$L = \frac{R1R3R5C4}{R2} \tag{2.7}$$

This way we are able to control in addition to the resistance R, the inductance L which allows us to control all parameters governing the system of differential equations 2.3.

3 Prototype

KiCAD circuit schematic in Appendix C details our circuit design and components used (Figure 11). The delineated blocks separate the circuit into its constituents (Antoniou Inductor, Chua's Diode, and others). The block labelled Chua's circuit contains the chaotic oscillator. The two potentiometers allows us to search the parameter space of the system and observe bifurcations leading to chaos as we push the environment variables. The observables x, y, and z are clearly labelled in the schematic. The z observable is not quite the same as the observable z in Equation 2.3, but this new z' is it related to it by $z' = y - z \times R7$ (easy to see). The x observable is sent to the section labelled digitizer, which will convert it to a 1 or 0 based on its scroll parity. The digitizer circuit consists of an open loop op-amp which rails to positive or negative power supply voltage depending on the sign of x. A diode cuts the square wave into a positive square wave between a large positive voltage and zero. Assuming power supply voltage between 7V and 9V (typical values if 9V batteries are used), we constructed a voltage divider to output a square wave of positive swing between 3.5V and 5V. 3.0V to 5.0V are the upper and lower limits for logic high on the Arduino Uno's digital input pins, so naturally we will collect this signal using an Arduino for Monte Carlo Simulations in the next section. The circuit draws 0.657W (45mA at 9V, -28mA at -9V) at max power, so a 9V battery should last for many hours.

Below, we will characterize the various components of this circuit, starting with the Chua's diode. The values of Ga and Gb can be selected by the resistor values R1 to R6, which were chosen so that the slopes here would be close to those used by Chen et al. They agree to within about 15%.

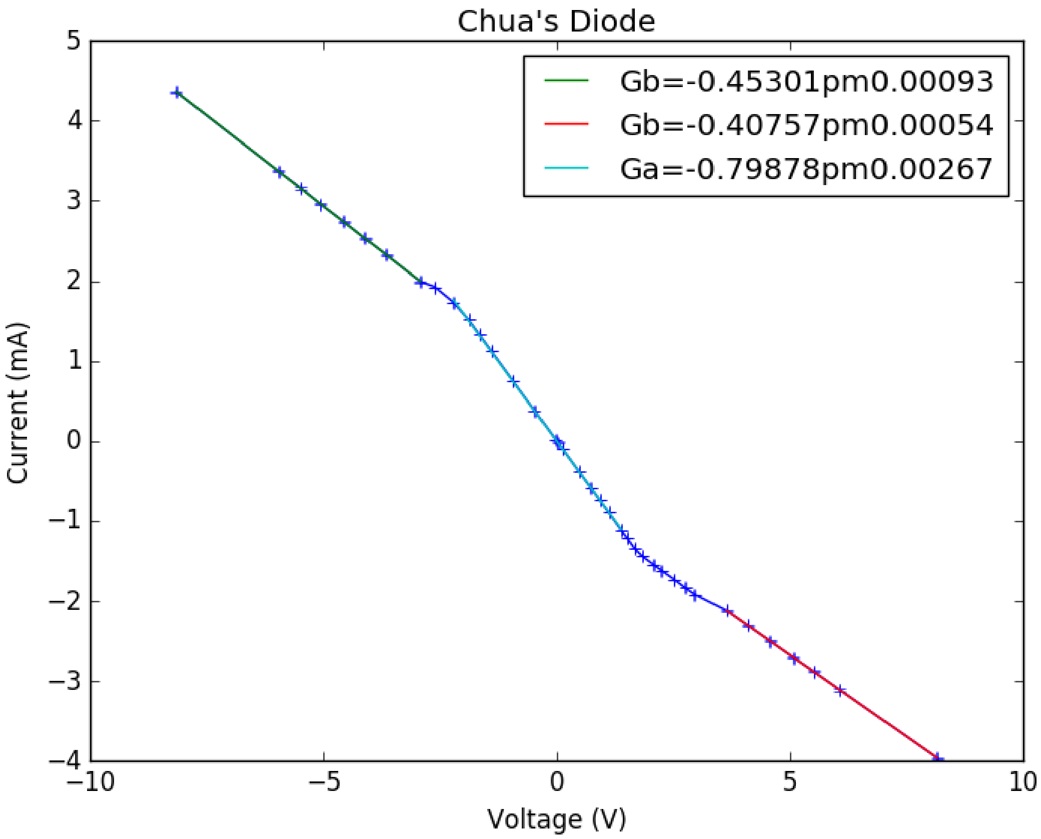

Figure 4: Current voltage function of chua diode. [3]

To test the Antoniou Inductor, we built a simple high pass filter as shown below in Figure 5 (left). The transfer function in Figure 5 (right) is consistent with an inductor of inductance L=15mH by the position of the 3dB rolloff. This is quite close to the 18mH used by Chen et al. in their HRNG experiment. For frequencies much greater than the rolloff, we observe no phase shift, while for frequencies much smaller than the rolloff, we observe phase shift of -90^o. At f3dB itself, we observe phase shift of -45^o. Magnitude and phase of the transfer function fully characterizes the behaviour of this object as an inductor whose inductance is L=15mH.

4 Analysis of Outputs

Here we will examine outputs of the prototypical Chua's HRNG. In particular, we will observe the x and y state variables, as well as the digitized output. Figure 6 shows bifurcation and period doubling as the resistance R is increased slowly from 0 up to around 1.5kR.

At R10=1.682±0.001kR (that is L=16.79mH) and R=1.574±0.001kR, we can observe the double scroll quite clearly. From this we can guess that the lyapunov time is at least around 5ms (the approximate time it takes to cross lobes). That means we should sample the digitized bit sequence at a rate slower than 200Hz.

Figure 8 below shows digitized output of x observable in aforementioned double scroll regime when running the circuit on ±7V and ±9V. It can be seen that the functional regime of the 9V battery is sufficient to power the Chua HRNG as an Arduino module.

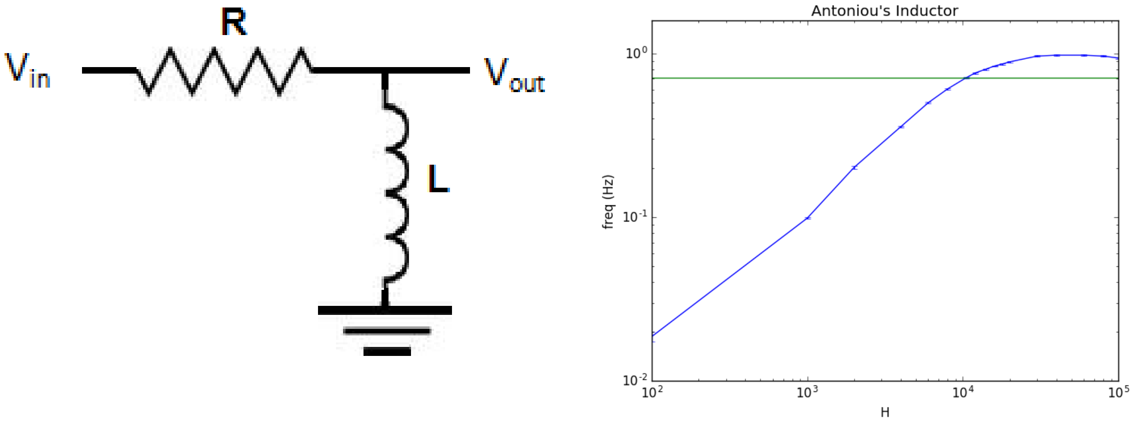

Figure 5: (left) Transfer function of Antoniou high pass filter. [3]. (right) Measured transfer function of Antoniou high pass filter, where the 3dB rolloff is 10.48±0.05kHz.

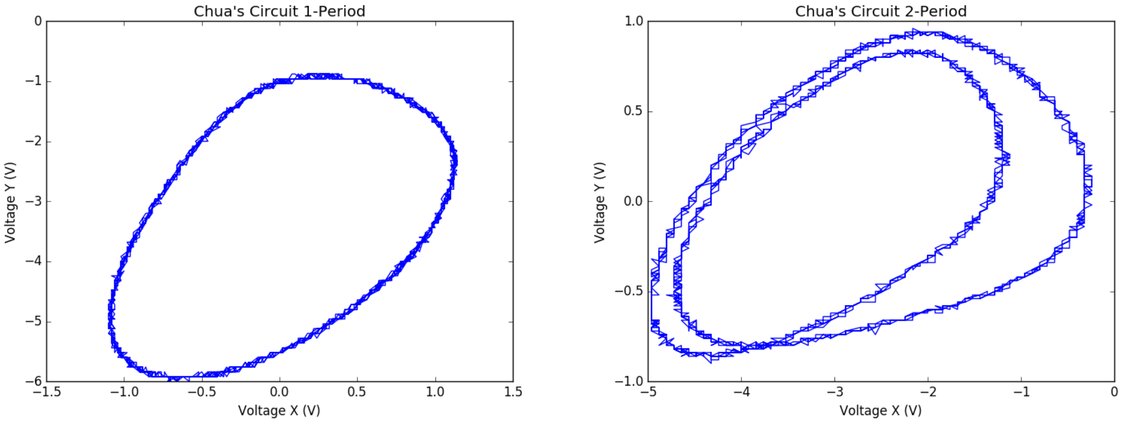

Figure 6: (left) 1-period cycle in phase space. (right) 2-period cycle in phase space.

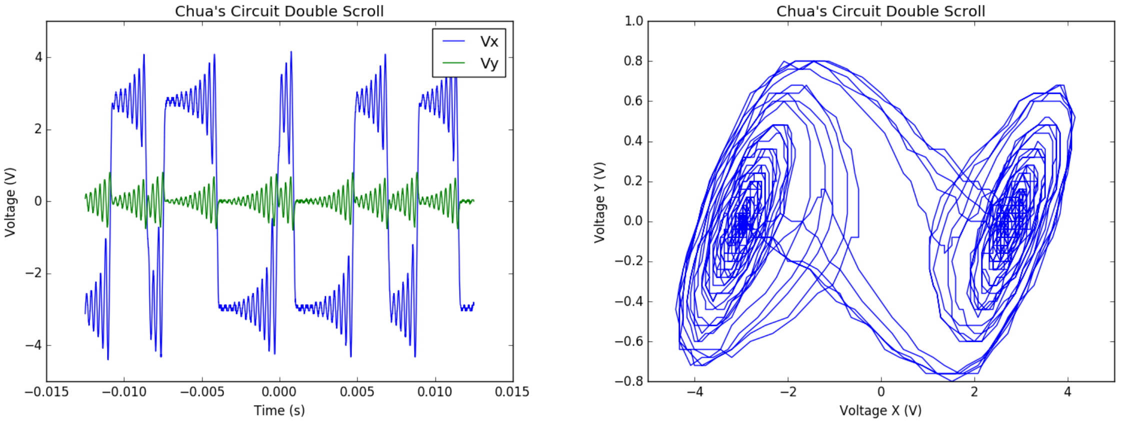

Figure 7: (left) Double scroll strange attractor. (right) graphed in time domain to illustrate Lyapunov time.

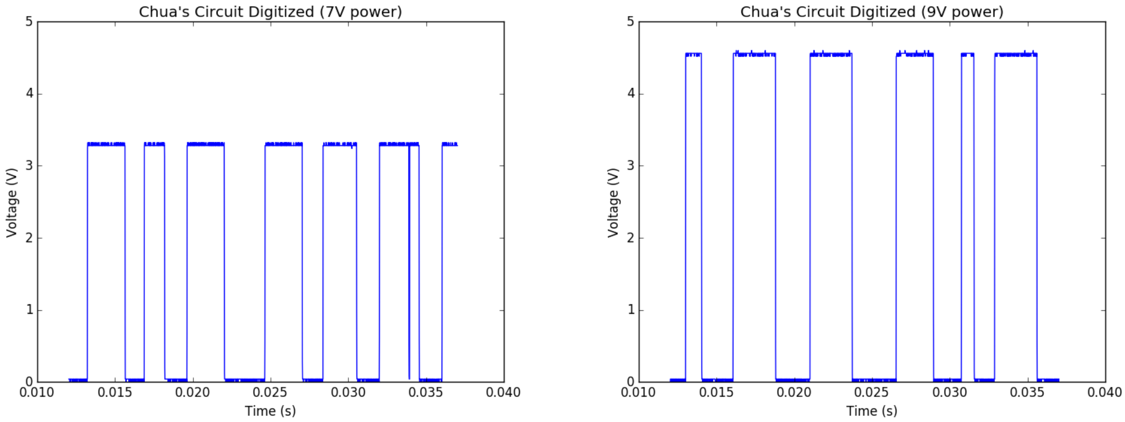

Figure 8: (left) Double Scroll digitized output at 7V power. (right) Double Scroll digitized output at 9V power.

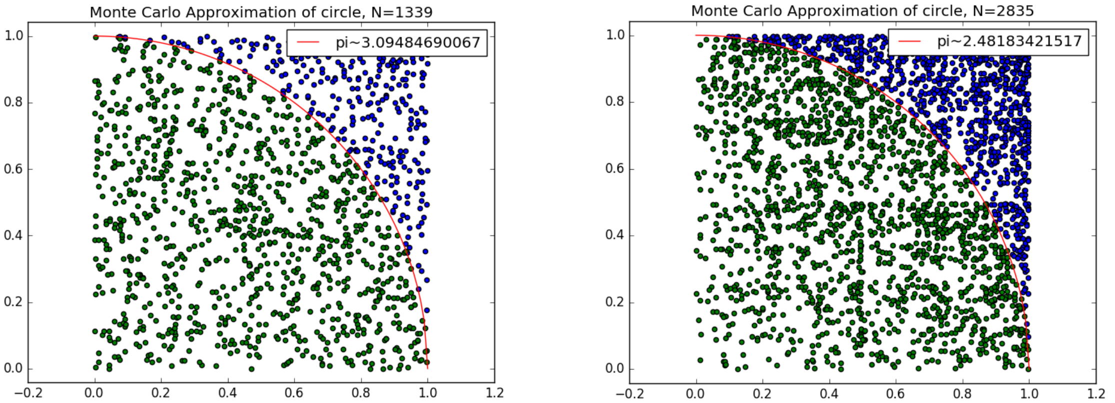

Figure 9: (left) Drawing from symmetric double scroll. (right) Drawing from asymmetric double scroll.

5 Software

The code in Appendix B takes in digitized input from the Chua's HRNG and output a bitstream which can be used to perform Monte Carlo simulations. The speed is 0.02s per bin, which is clearly faster than Lyapunov time since up to 4 scroll changes would have occurred in 20ms. Hence we can be assured that the bits will be thoroughly uncorrelated. Binning the interval [0,1] into 255 segments, we can take a length 8 binary string as representative of a decimal number in [0, 1] with resolution 0.004, and a length 16 binary would have resolution 1.5e-5. Let's use length 8 random binary numbers to populate a [0, 1]x[0, 1] square to approximate π. However, in order to get a vector in the unit square, each vector characterized by 8 bits, we would need 16 bits in total. We can expect to densely pack the square down to its resolution limit only when we have drawn $256^2/1^2 = 65536$ such vectors, and that corresponds to a total of 1 million bits, which at our rate would take 14.6 days to draw. We have tried to make the Double Scroll as symmetric as possible and drew 11119 bins (1339 vectors) and we have also performed the same experiment without symmetrizing the double scroll, drawing 45373 bins (2835 vectors). Figure 9 shows the Monte Carlo simulations.

6 Manufacturing

Our circuit is too large to fit on Phy405 standard issue breadboards, and breadboards are inelegant. We used KiCAD to generate traces for the schematic in Figure 11. The gerber file which will be used to print the PCB can be found in Figure 12 in Appendix C. Figure 10 shows 3D model of the PCB. We attempted to manufacture this using Phy405 standard issue copper boards and etchant from home

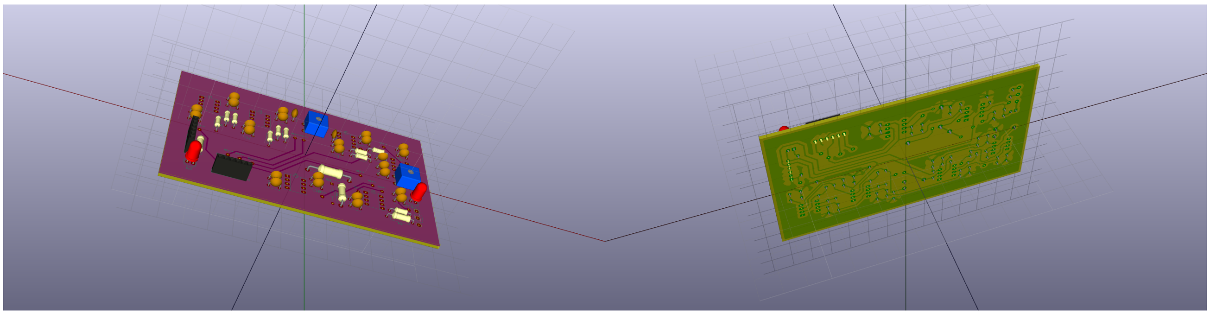

Figure 10: Chua's HRNG circuit PCB front and back.

hardware using the toner transfer method. After 6 wasted boards and hours of wasted time, we opted to order a PCB instead.

7 Conclusion

We have demonstrated proof of concept for the Chua's HRNG circuit with our prototype circuit. We have observed the characteristic bifurcations, period doubling, and double scroll strange attractor of the chaotic Chua's circuit. We have used the chaotic nature of the system to extract random bit with some weighted distribution over 0, 1 governed by the ratio of sizes of the scrolls. We have demonstrated how one might use these random bits to perform Monte Carlo simulations over real domains. We have completed the necessary preparations to manufacture the circuit, but have yet to order the PCB.

References

[1] L. Chua, C. W. Wu, and Z. G. Q., "A Universal Circuit of Generating and Studying Chaos - Part 1: Routes to Chaos," *IEEE Transactions on Circuits and Systems - 1: Fundamental Theory and Applications* **40** (Oct., 1993) .

[2] H. Mkaouar and O. Boubaker, "On electronic design of the piecewise linear characteristic of the chua's diode: Application to chaos synchronization," in *2012 16th IEEE Mediterranean Electrotechnical Conference*, pp. 197–200. March, 2012.

[3] J. Chen, L. Ran, and K. Chen, "A random sequence generator based on chaotic circuits," *Journal of Electronics (China)* **18** no. 1, (2001) 56–60. `http://dx.doi.org/10.1007/s11767-001-0008-5`.

[4] A. S. Sedra and K. C. Smith, *Microelectronic Circuits*. Oxford University Press, fifth ed., 2004.

8 Appendix A: Distriubtion of work

Chris dealt with construction of the printed board circuit, code for Arduino bit sampling, determined working sample rate, code for analysis of circuit data and Monte Carlo method. Michael dealt with the original construction of the breadboard circuit, characterization of the Chua diode current vs voltage function, characterization of Antoniou inductor transfer function, and aided in the mathematical analysis. The final writeup was a collaborative effort.

9 Appendix B

```
// ChuaHRNG.ino
// Read in a digitized double scroll observable as 1 or 0
// sampling at much slower rate than lyapunov timescale

#include <math.h>

int inPin = 2;  // Digital input pin for sampling Chua state voltage
int outPin = 13; // Digital output pin for indicator light

void setup() {
    Serial.begin(9600); // this number is the baud rate (9600 by default)
    pinMode(inPin, INPUT);
    pinMode(outPin, OUTPUT);
}

void loop() {
    int Vo;            // Integer value of voltage reading (0/1)

    Vo = digitalRead(inPin);
    if (Vo == HIGH){
      digitalWrite(outPin, HIGH);
    }
    Serial.println(Vo);
    delay(20);
    digitalWrite(outPin, LOW);
}
```

10 Appendix C

Schematics

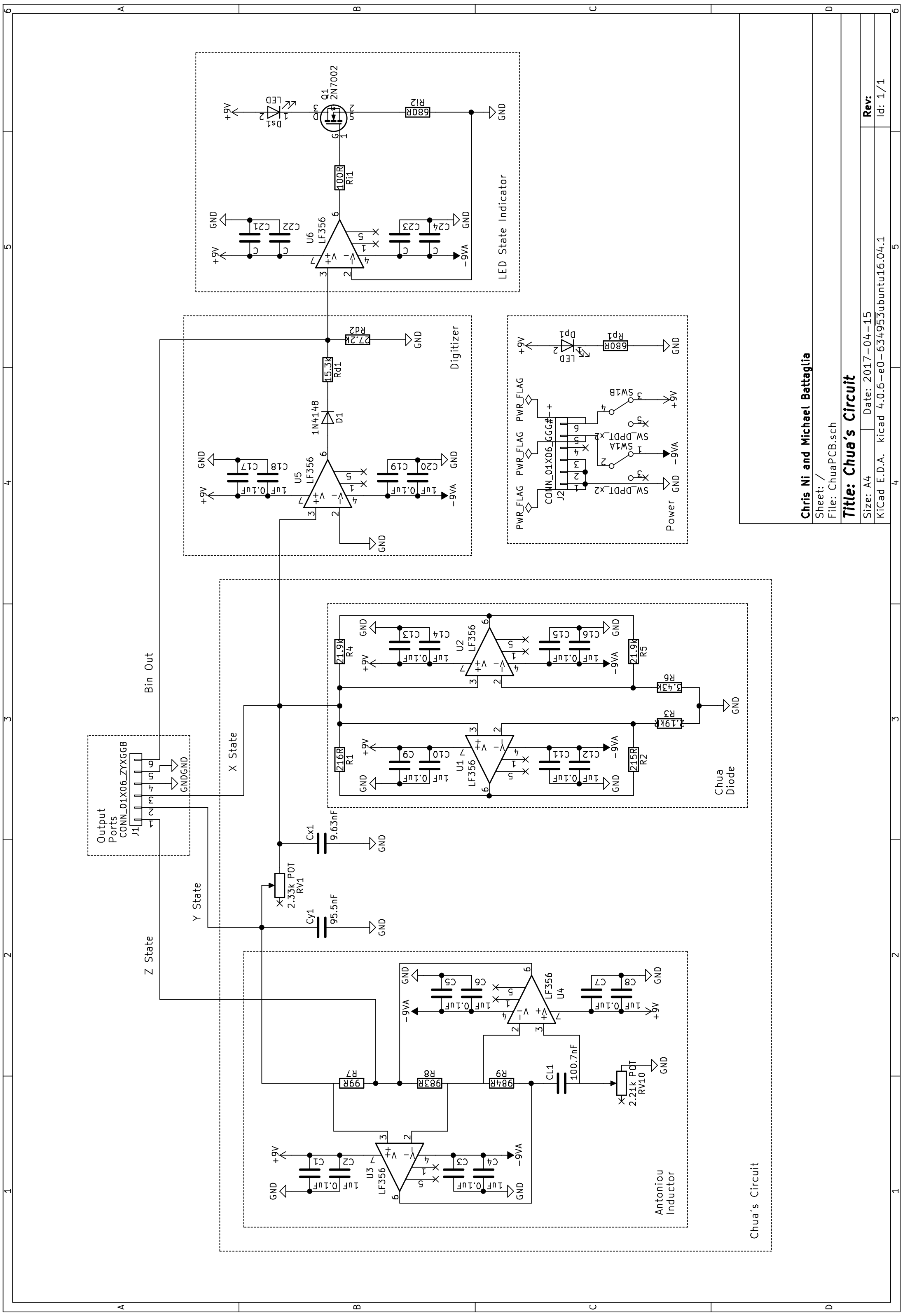

Figure 11: KiCAD schematic of Chua's circuit hardware random number generator design.

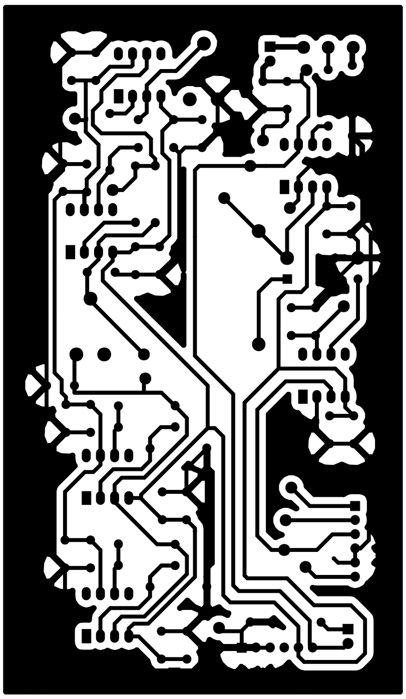

Figure 12: KiCAD gerber of Chua's circuit PCB design.