Indian Journal of Computer Science and Technology
https://www.doi.org/10.59256/indjcst.20250403041
Volume 4, Issue3 (September-December 2025), PP: 258-264.
www.indjcst.com
ISSN No: 2583-5300

An Analysis of Entropy Reduction in Digital Logic Circuits with Feedback

Sourjya Gupta[1], Sumana Sikdar[2], Anwesha Bose[3]
[1,2,3]*Department of Computer Science Engineering, Techno India University, Kolkata, West Bengal, India.*

To Cite this Article: *Sourjya Gupta[1], Sumana Sikdar[2], Anwesha Bose[3], "An Analysis of Entropy Reduction in Digital Logic Circuits with Feedback", Indian Journal of Computer Science and Technology, Volume 04, Issue 03 (September-December 2025), PP: 258-264.*

Abstract: *This paper demonstrates that a designed Boolean logic circuit with feedback connections can transform completely random binary inputs into structured output sequences exhibiting lower Shannonentropy. We establish that the circuit output follows an ergodic Markov chain, which—by the law of large numbers—converges to a non-uniform probability distribution over sequences. This convergence reduces Shannon entropy from its maximum value of 4.7004 bits (for perfectly random sequences) to lower values. We formalize the circuit architecture using combinational logic gates with controlled probabilities and demonstrate through both mathematical derivation and computational experiments that random inputs, when processed through appropriate logic arrangements, can produce increasingly deterministic patterns. This work establishes a fundamental connection between Boolean circuit topology and information-theoretic properties, suggesting that determinism can emerge from randomness through structural design.*

Key Words: *Boolean logic, Ergodicity, Law of large numbers, Markov chain, Shannon entropy.*

I. INTRODUCTION

The classic infinite monkey theorem proposes a concept where a monkey randomly typing on a typewriter. If the probability of typing any letter is equal and the monkey types for an infinite amount of time, it is almost certain to produce any given text, such as the complete works of Shakespeare. An estimate of monkey completing the Hamlet which has around 130000 letters would take 1 chance among $3.4 \times 10^{183\,946}$ possibilities, which has a very high estimated time to achieve[1].

However, there is a way to reduce the estimated time, even though there will be same number of possibilities. It will still be having random inputs, but the probabilities of the output will vary upon the previous letters. Like in Shakespeare's language he used 'thou' more often than any other author, who nearly always use 'you'. In the method given below, a logic for a typewriter can be constructed in such a way, that there will be higher probability of "h" following "t". Which would indicate that most probably in lesser number of chances the word "thou" can be achieved by the monkey.

Here the logic gates are arranged in a particular format. In the circuit, the output from one sub-circuit is depicted as a letter. This output is an input into the next sub-circuit. A letter is read if the subcircuit's output is 1, otherwise the process passes to the next sub-circuit's output, where the check is repeated.

II.METHODOLOGY

Establishing the output probabilities of Boolean logic gates.

$$P(A \otimes B = 1) = \frac{1}{2} \qquad (1)$$

$$P(A \wedge B = 1) = \frac{1}{4} \qquad (2)$$

$$P(A \vee B = 1) = \frac{3}{4} \qquad (3)$$

Combining these gates, through combinational circuit new probability of output 1 is achievable. Example of creating a probability output of 1 in Figure 1 in F is as follows.

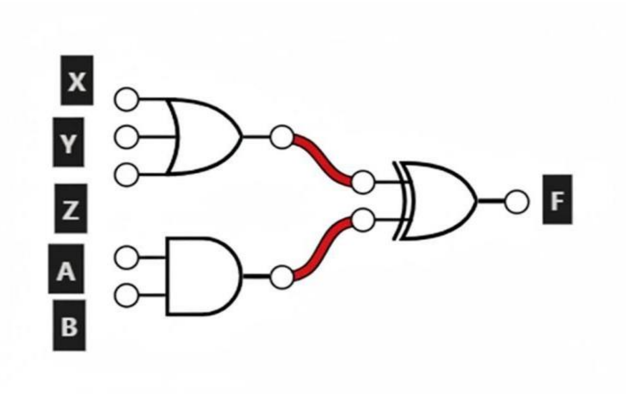

Fig 1

The probability of F = 1 depends on its inputs. We can calculate the probabilities of the intermediate signals (assuming all primary inputs X, Y, Z, A, B are independent and random):

$$P(F=1)=\tfrac{1}{2}P(X\vee Y\vee Z=1)+\tfrac{1}{2}P(A\wedge B=1) \tag{4}$$

Further we can get,

$$P(X\vee Y\vee Z=1)=\tfrac{7}{8} \tag{5}$$

And,

$$P(A\wedge B=1)=\tfrac{1}{4} \tag{6}$$

Hence,

$$P(F=1)=\tfrac{1}{2}\times\tfrac{7}{8}+\tfrac{1}{2}\times\tfrac{1}{4}=\tfrac{9}{16} \tag{7}$$

Similar to this example rational number with denominator of power 2 have probabilities of output that can be generated with a finite size combinational digital circuit. Finite size digital circuit have finite Boolean logic gates. The number of Boolean logic gates varies with the circuit diagram we make and the intended probabilistic output.

These circuits operate in parallel without any feedback or cross-connections, their outputs are statistically independent. This lack of dependency ensures that the generated sequence of letters is fundamentally random and lacks any sequential pattern or correlation. A Markov chain can be used to build a meaningful "language". Below we give a non-combinational circuit to generate a Markov chain as an example to demonstrate the "language".

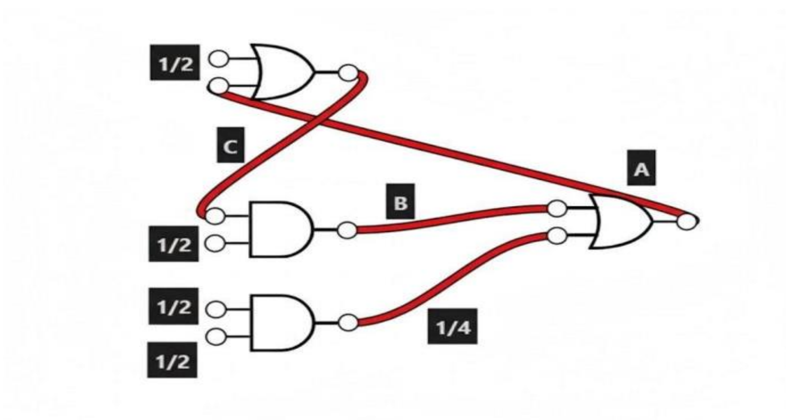

Fig 2

The circuit will contain 3 outputs which are considered in the experiment. A, B, C with each having a certain probability of output 1, and each probability influences other's probability, thus making it interdependent. With probability of A equals to 1 will determine the value of probability of B equals to 1.

To get all the probabilities a system of linear equation must be solved. $P(A=1)=a, P(B=1)=b, P(C=1)=c$

$$a = \frac{1}{4}\times\frac{3}{4} + b\times\frac{3}{4} \tag{8}$$

$$b = \frac{1}{2}\times\frac{1}{4} + c\times\frac{1}{4} \tag{9}$$

$$c = \frac{3}{4}\times\frac{1}{2} + a\times\frac{3}{4} \tag{10}$$

This can be rearranged in a matrix form, $\begin{pmatrix} 16 & -12 & 0 \\ 0 & 8 & -2 \\ -6 & 0 & 8 \end{pmatrix}\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 3 \\ 1 \\ 3 \end{pmatrix}$ (11)

On solving the matrix $a = \frac{9}{22} \approx 0.409$, $b = \frac{13}{44} \approx 0.295$, $c = \frac{15}{22} \approx 0.682$

To illustrate how this circuit's output can be viewed as a "language," we first define a simple "write rule": at each time step, if a module's (A, B, or C) output is 1, its letter is written. If the output is 0, nothing is written.

Applying our "write rule" to this sequence, we would generate the final sentence:

ABCACABCABCACABCAC

This system shows that while the long-term *average* probability of each letter is fixed (at a, b and c), the *immediate* probability of the next letter is not. Because of the circuit's feedback (e.g., the input to C depends on the output of A), the probability of the next state is dependent on the current state.

This is the formal definition of a Markov chain. The system's next state depends *only* on its present state, not on the entire history of states that preceded it. As Shannon demonstrated, a system (like our circuit) whose next state depends on the previous one state is a classic example of a first-order (or Markov) approximation of a language[2].

III.MODEL DESCRIPTION

In Figure 3, we propose a generalized model. This system consists of n number of modules, $m1,\ldots, mn$. The term mi is a probabilistic weighting parameter that acts as a scaling factor for the i-th logic module. That represents how strongly the module's inputs influence its output.

Each module mi receives an independent, random input Pi and a sequential input vi+1 from the next module (with $v1$fed back from vn, creating a ring). The output of module mi is vi..

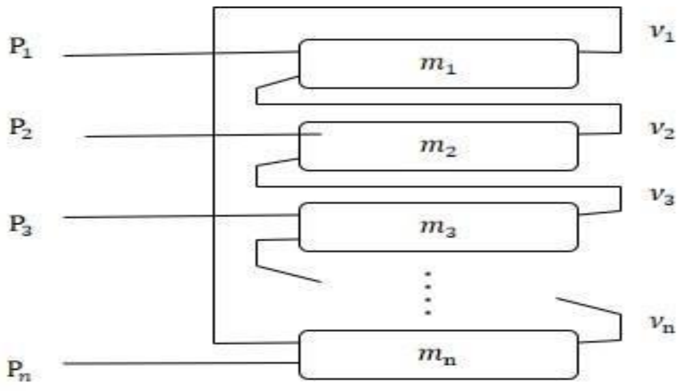

Fig 3

Let Pi be the probability of the i-th input equals 1, vi is the probability of the i-th output equals 1. Thus, the equation that relates is

$$v_i = m_i P_i + m_i v_{i+1} \tag{12}$$

$$m_i P_i = v_i - m_i v_{i+1} \tag{13}$$

This can be represented in a matrix form

$$\begin{pmatrix} 1 & -m_1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & -m_2 & 0 & \cdots & 0 \\ 0 & 0 & 1 & -m_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & 0 & \ddots & -m_{n-1} \\ -m_n & 0 & 0 & 0 & \cdots & 1 \end{pmatrix}\begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_{n-1} \\ v_n \end{pmatrix} = \begin{pmatrix} m_1P_1 \\ m_2P_2 \\ m_3P_3 \\ \vdots \\ m_{n-1}P_{n-1} \\ m_nP_n \end{pmatrix} \tag{14}$$

For the output sequence to be predictable and stable, it must follow the **Law of Large Numbers**. However, for a Markov chain it has to be ergodic to follow the law of large numbers[3]. This law states that the average result from many trials will eventually converge to the expected theoretical probability[6,8]. An ergodic chain is one that doesn't get "stuck" in a specific part of the system; it's possible to get from any state (letter) to any other state, and the system doesn't get trapped in a simple, repeating loop.

We can prove our system is ergodic using the following intuitive logic A Markov chain whose transition matrix is such that all the elements of a matrix $T0$ are strictly positive, then every power A^n (for $n > 0$) will also have all elements positive[7].

All elements of the transition matrix are strictly positive ($P_{ij} > 0$ for all i, j)

$$T_0 = \begin{pmatrix} v_{11} & \cdots & v_{1n} \\ \vdots & \ddots & \vdots \\ v_{n1} & \cdots & v_{nn} \end{pmatrix} \tag{15}$$

is a well-known condition that guarantees the chain is ergodic (specifically, it is "irreducible" and "aperiodic," which are the formal requirements for ergodicity)[4,5]

Because our circuit's Markov chain is ergodic, the Law of Large Numbers does apply. This gives us a strong mathematical guarantee that our output sequence will, over a long enough time, "almost surely converge" to the stable, expected probabilities.

According to Shannon's theory, the entropy of a system is maximized when all the outcomes are equally likely. This state of maximum entropy corresponds to the highest degree of randomness.

In Figure 3 the logic circuit, is designed to impose a non-uniform probability distribution on its outputs. We have derived an analytical model from this Boolean logic, yielding an equation that solves for each of the possible output letters. These derived probabilities, which are a direct consequence of the circuit's topology, are then applied to Shannon's entropy formula. This allows us to precisely calculate the circuit's specific information entropy, which we hypothesize will be measurably lower than the theoretical maximum entropy of a uniform system.

Shannon's entropy formula is given by $H = -\sum p_i \, log(p_i)$ (16)

As there is total 26 possibility and we are considering 1 possibility out of the 26, hence the entropy is given by

$$p_i = \frac{1}{26} \tag{17}$$

There is total 26 so summation 26 times will be

$$H = -\frac{1}{26} log\left(\frac{1}{26}\right) \times 26 \tag{18}$$

When applied to all perfectly random outputs of a letter from 26 alphabets, the entropy calculated is 4.7004 [3]

In the context of the logic circuit described, we can now precisely calculate the Shannon entropy of the output sequence.

$$v_\text{i} = m_\text{i}\text{P}_i + m_\text{i}v_{\text{i}+1} \tag{19}$$

To figure out the final entropy, we need to know the chance of each letter appearing in the final sequence. The chance of the first letter, $p1$ is its own probability, $v1$divided by the sum of all the letter probabilities

$$p_1 = \frac{v_1}{\sum v_i} \tag{20}$$

The v_i Putting the values in the equation,

$$H = -\sum p_\text{i} log(p_\text{i}) \tag{21}$$

Expanding this summation gives the full equation for the circuit's entropy, based on the contribution of each possible letter,

$$H = -[p_1 log(p_1) + p_2 \log(p_2) + p_3 \log(p_3) \ldots p_\text{n} \log(p_\text{n})] \tag{22}$$

Probability value taken for the first output,

$$v_1 = m_1\text{P}_1 + m_1 v_2 \tag{23}$$

(23)

Probability value taken for the second output,

$$v_2 = m_2\text{P}_2 + m_2 v_3 \tag{24}$$

This pattern continues around the ring until the n-th module feeds back into the first, creating a complete, interdependent system

Numerical steps to solve for the first output,

$$v_1 = m_1P_1 + m_1(m_2P_2 + m_2v_3) \tag{25}$$

$$v_1 = m_1P_1 + m_1m_2P_2 + m_1m_2v_3 \tag{26}$$

$$v_3 = m_3P_3 + m_3v_4 \tag{27}$$

$$v_1 = m_1P_1 + m_1m_2P_2 + m_1m_2(m_3P_3 + m_3v_4) \tag{28}$$

$$v_1 = m_1P_1 + m_1m_2P_2 + m_1m_2m_3P_3 + m_1m_2m_3v_4 \tag{29}$$

Repeatedly putting values,

$$v_1 = m_1P_1 + m_1m_2P_2 + m_1m_2m_{3.}\dots m_{n-1.}P_{n-1} + m_1m_2m_3\dots m_nv_1 \tag{30}$$

Solving for v_1,

$$v_1 - m_1m_2m_3\dots m_nv_1 = m_1P_1 + m_1m_2P_2 + m_1m_2m_{3.}\dots m_{n-1.}P_{n-1} \tag{31}$$

$$v_1 = \frac{m_1P_1 + m_1m_2P_2 + m_1m_2m_{3.}\dots m_{n-1.}P_{n-1}}{1 - m_1m_2m_3\dots m_n} \tag{32}$$

Or in a more compact form,

$$v_1 = \frac{\sum_{j=1}^{n}\left(P_j\left(\prod_{i=1}^{j} m_i\right)\right)}{1 - \prod_{i=1}^{n} m_i} \tag{33}$$

Thus, completing the numerical we get the formula for $v1$.
Similarly, $v2$can be calculated by cyclically shifting the indices.
To find the entropy, we do as before, taking the sum of all the possible probabilities of vi the final summation,

$$H = -\left(\frac{v_1}{\sum v_i} log\left(\frac{v_1}{\sum v_i}\right) + \frac{v_2}{\sum v_i} log\left(\frac{v_2}{\sum v_i}\right) + \frac{v_3}{\sum v_i} log\left(\frac{v_3}{\sum v_i}\right) + \cdots + \frac{v_n}{\sum v_i} log\left(\frac{v_n}{\sum v_i}\right)\right) \tag{34}$$

IV.EXPERIMENT RESULTS

Results For Experiment 1

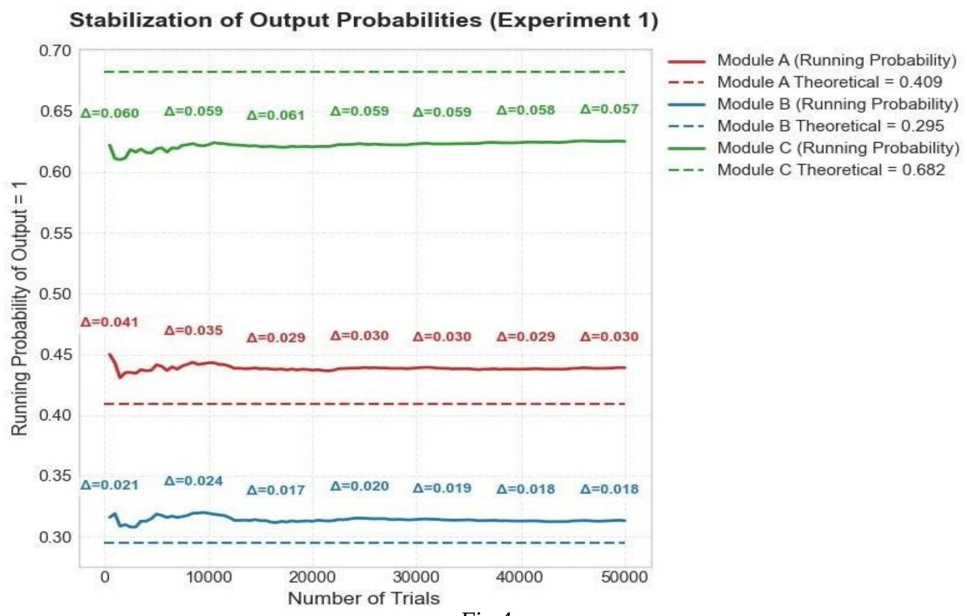

Fig 4

We conducted a computational experiment to simulate the circuit depicted in Figure 2. This experiment compares the theoretically-derived value of the probability to the calculated value and represents the diagram in the graph in Figure 4. The values are near expectation with very less deviation.

Results For Experiment 2

When we ran the simulation for an assumed circuit of the type of Figure 3, we generated a long string of letters. This string was built by checking the output of each sub-circuit (one for each letter) in sequence. A letter was only added to the string if its circuit outputted a '1'. If it outputted a '0', it was skipped, and the system moved to the next letter. We noticed that this process created 'chunks' of letters that appeared more frequently than others.

In Figure 5 we have taken the frequency of a bigram, trigram and quadrigram. To measure, how nonrandom or deterministic this was. In our model (for trigram) n = 26, there are 17,576 possible 3-letter combinations (from 'AAA' to 'ZZZ', which is $26^3 =$ 17,576). The deterministic behaviour the solid curve from the complete random behaviour shown in the horizontal dotted line, which is the distribution in a purely random case.
This exactly same pattern is found in both bigram and quadrigram.

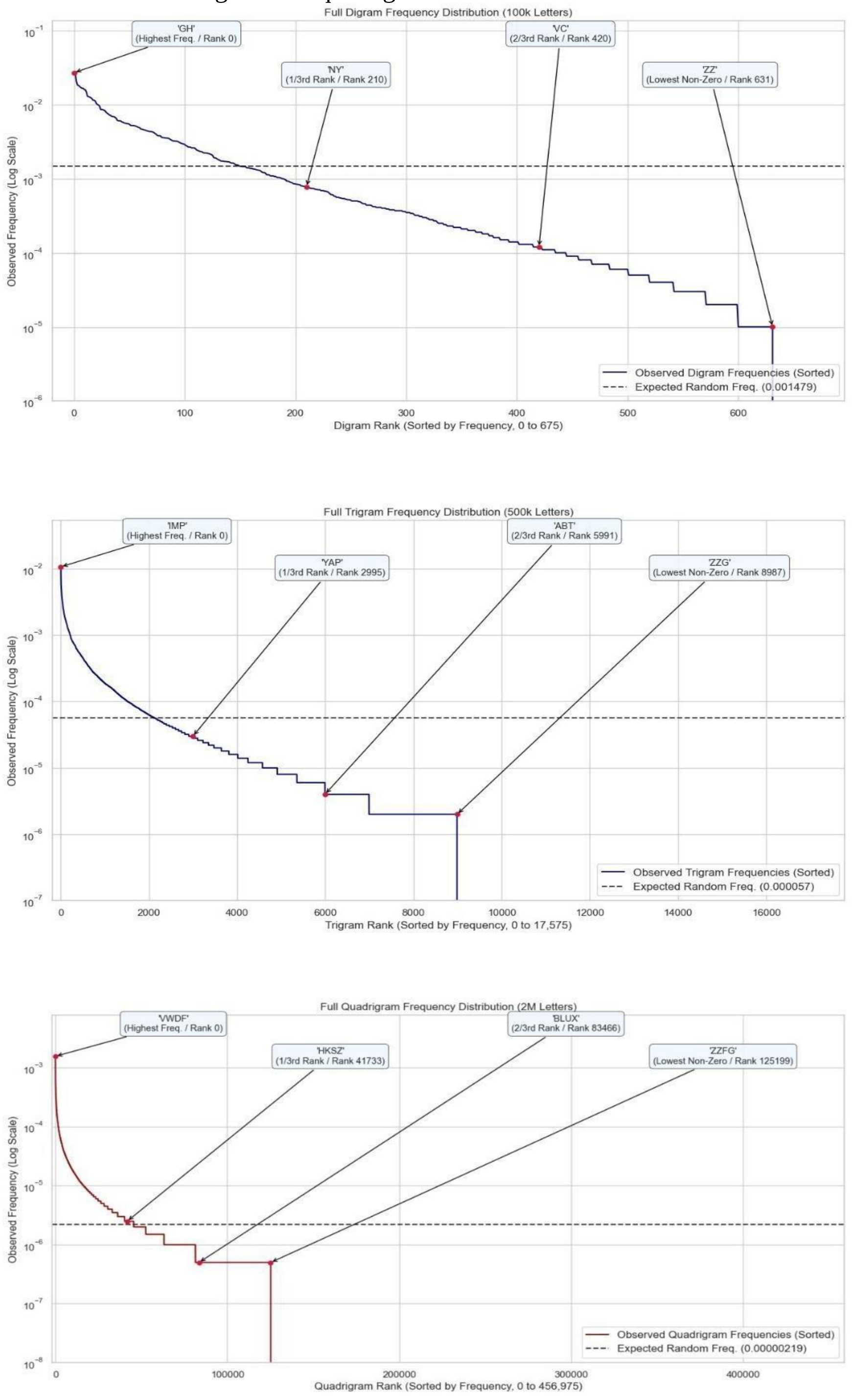

Fig 5

Results For Experiment 3

Finally, we computed the Shannon entropy (H) for the generalized model (Figure 3, n = 26) using randomly generated sets of probabilities Pi and mi. We ran 10 rounds, obtaining the following entropy values (in bits).

Entropy values (H): 4.3588, 4.3378, 4.5018, 4.2928, 4.2749, 4.3510, 4.5496, 4.4900, 4.4231, 4.4990

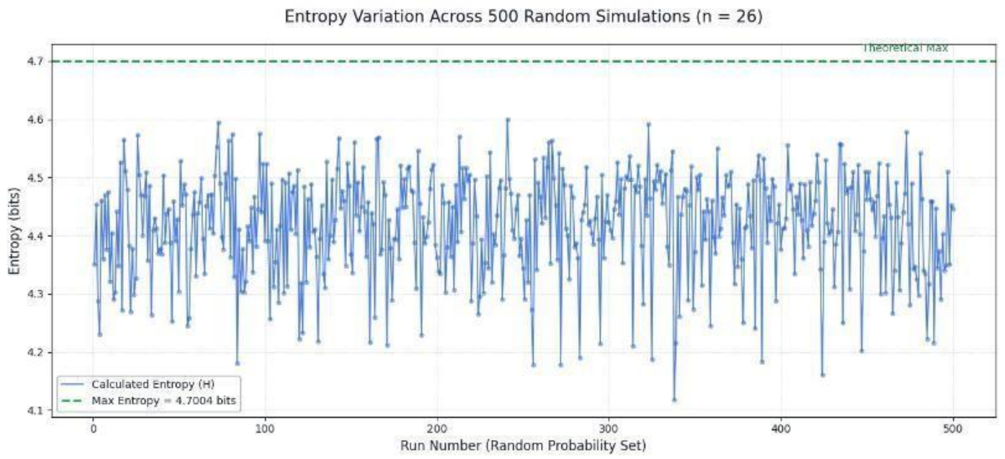

Fig 6

Figure 6 plots the calculated entropy H for 500 different random probability sets. In all 500 simulations, the resulting entropy was measurably lower than the theoretical maximum of 4.7004 bits. This strongly indicates that the circuit structure imposes a non-uniform probability distribution on the output, reducing randomness and creating more deterministic patterns.

V.CONCLUSION

By repeatedly processing random binary inputs through the described logic circuit, we generate a string of letters whose statistical properties converge, as stated by the Law of Large Numbers for ergodic Markov chains. Our experiments confirmed the theoretical derivation: the resulting sequences exhibit a measurable decrease in Shannon entropy compared to a uniformly random sequence. This implies that the topological structure of a Boolean circuit can impose a non-uniform probability distribution on its outputs, effectively filtering randomness to produce more deterministic patterns. This work establishes a clear connection between Boolean logic, circuit topology, and the information-theoretic properties of the sequences they generate.

VI.FUTURE WORKS

This work is primarily theoretical. Future research could explore practical applications of this principle, such as in designing hardware-based pseudo-random number generators with specific statistical properties or modelling emergent com plex behaviour in logical systems.

REFERENCES

1. Anderson, J. (2011), A million monkeys and Shakespeare. Significance, 8: 190-192. https://doi.org/10.1111/j.1740-9713.2011.00533.x
2. Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, *27*(3), 379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x
3. Norris, J. R. (1997). . Cambridge University Press.
4. Plemmons, R. J. (1988). Matrix Analysis (Roger A. Horn and Charles R. Johnson). *SIAM Review*, *30*(1), 153–154. https://doi.org/10.1137/1030034
5. Jolliffe, F., & Ross, S. M. (1995). Introduction to probability models. *Journal of the Royal Statistical Society Series a (Statistics in Society)*, *158*(1), 201. https://doi.org/10.2307/2983433
6. Grimmett, G., & Stirzaker, D. (2001). *Probability and random processes* (3rd ed.). Oxford University Press.
7. Pakes, A. G. (1969). Some conditions for ergodicity and recurrence of Markov chains. *Operations Research*, *17*(6), 1058-1061.
8. Breiman, L. (1960). The strong law of large numbers for a class of Markov chains. *The Annals of Mathematical Statistics*, *31*(3), 801–803. https://doi.org/10.1214/aoms/1177705810