Research Migration Project Proposal

Name of the Participant: Bandi Hema Chandu

Affiliation / Institution: Department of Electronics and Communication Engineering, Rajiv Gandhi University of Knowledge and Technologies, Nuzvid, Andhra Pradesh, India

Title of the Circuit

Design and Simulation of a Cellular Automata based Pseudo-Random Number Generator (CA-PRNG) using Rule 30

Theory / Description

A **Cellular Automata based Pseudo-Random Number Generator (CA-PRNG)** is a digital hardware system that generates a sequence of statistically random binary numbers using the mathematical principles of 1-Dimensional Cellular Automata (CA). Unlike traditional Linear Feedback Shift Registers (LFSRs), which rely purely on linear XOR operations and are vulnerable to algebraic attacks, a CA-PRNG uses **non-linear local transition rules** applied in parallel across a register of cells (D Flip-Flops).

This design implements **Wolfram's Rule 30**, where the next state of each cell `i` is determined by:

Next[i] = C[i-1] XOR (C[i] OR C[i+1])

- `C[i-1]` = State of the left neighbor cell
- `C[i]` = State of the current cell
- `C[i+1]` = State of the right neighbor cell

The system consists of a **32-bit state register** initialized with a non-zero seed (`0x00010000`). On every rising clock edge, the entire 32-bit state evolves simultaneously using the Rule 30 logic, producing a new 32-bit random output every clock cycle. **Null boundary conditions** are applied (i.e., cells at the edges treat missing neighbors as logic 0).

The design exposes two outputs:

- `prng_data_out [31:0]` — Full 32-bit parallel random output for high-bandwidth applications.
- `prng_serial_out` — Single-bit serial output from the center cell (bit 16), suitable for cryptographic stream cipher use where exposing the full state would be a security risk.

Reason to Reproduce with eSim

eSim is a free and open-source EDA tool developed by FOSSEE, IIT Bombay, which supports mixed-signal simulation using the NgVeri Verilog-to-Ngspice conversion tool. This project is ideal for eSim for the following reasons:

1. **NgVeri Integration:** The CA-PRNG Verilog module (`ca_prng.v`) can be directly converted into an Ngspice-compatible simulation block using the NgVeri tab in eSim, making it executable entirely within the FOSSEE ecosystem without any external tools.
2. **Open-Source Hardware Design:** The CA-PRNG is a purely digital, synthesizable design written in standard Verilog HDL — perfectly aligned with FOSSEE's mission of promoting open-source hardware design education.
3. **Mixed-Signal Demonstration:** Using `adc_bridge` components for clock and reset inputs and `dac_bridge` for the serial output, students can visually observe the random bitstream waveform in the Ngspice plotter, bridging the gap between digital RTL design and analog waveform analysis.
4. **Educational Value:** Reproducing this circuit in eSim provides hands-on experience in Digital IC design, PRNG theory, Cellular Automata mathematics, and FPGA-level thinking — all within a free, accessible FOSSEE tool.

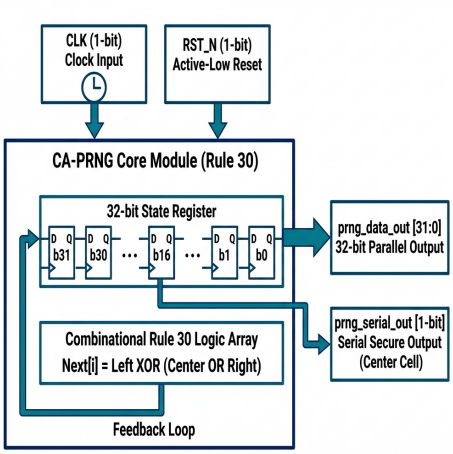
Expected Outcome / Outputs

After simulation, the following outputs are expected:

- `clk` — A clean 100 MHz square wave clock (10 ns period).
- `rst_n` — Active-low reset pulse that initializes the CA state to seed `0x00010000`.
- `prng_data_out[31:0]` — A new, unique 32-bit hexadecimal random number produced on every rising clock edge. The values will change unpredictably on every cycle (e.g., `0x00010000` → `0xAAFB4FC8` → `0x00038000`...).
- `prng_serial_out` — A single-bit signal alternating between 0 and 1 in a non-periodic, non-repeating random pattern.

Circuit Diagram

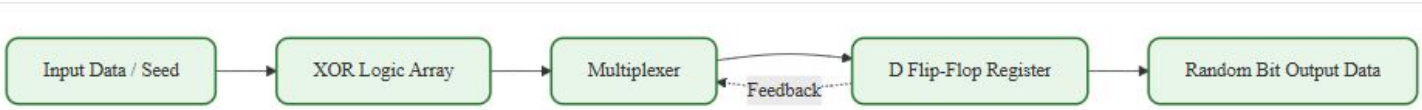
The circuit diagram below shows the CA-PRNG system architecture — the 32-bit State Register, Rule 30 Combinational Logic Array, feedback path, and dual outputs.



CA-PRNG System Circuit Diagram

Block Diagram

The top-level block diagram below shows the data flow through the CA-PRNG: clock/reset inputs → 32-bit State Register → Rule 30 Logic Array (with feedback) → 32-bit parallel output and 1-bit serial secure output.

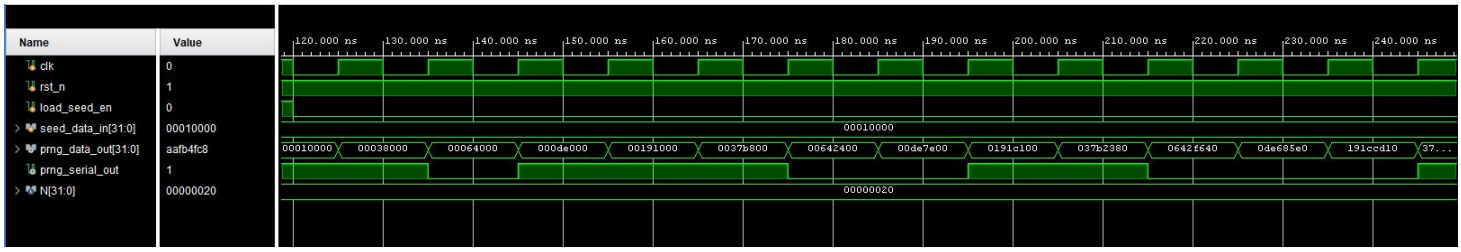


CA-PRNG Top-Level Block Diagram

Expected Results (Input/Output Waveforms)

The simulation waveform below was captured from **Xilinx Vivado Behavioral Simulator**. It demonstrates the correct operation of the CA-PRNG over multiple clock cycles and confirms:

1. The seed `0x00010000` is correctly loaded on reset assertion.
2. A unique, unpredictable 32-bit random value is produced on every rising clock edge.
3. The `prng_serial_out` signal shows a non-periodic random bit sequence — with no observable pattern.



CA-PRNG Simulation Output Waveform — Vivado Behavioral Simulator

Research Paper / Journal / Source References

1. **Title:** Random Sequence Generation by Cellular Automata.
   Author: Wolfram S
   Journal: Advances in Applied Mathematics, 7(2), 123–169, 1986.
   Link: https://doi.org/10.1016/0196-8858(86)90028-X