Polynomial Metamodel Integrated Verilog-AMS for Memristor-Based Mixed-Signal System Design

G. Zheng, S. P. Mohanty[2], E. Kougianos[3] and O. Okobiah[4]
NanoSystem Design Laboratory (NSDL, http://nsdl.cse.unt.edu)
University of North Texas, Denton, TX 76203, USA.[1,2,3,4]

Presented By
Oghenekarho Okobiah
Email:oo0032@unt.edu

Acknowledgments: This research is supported in part by NSF awards CNS-0854182 and DUE-0942629 and SRC award P10883.

Outline

- Motivation and Background
- Novel Contributions
- Proposed Memristor Model
- Case Study Design
- Metamodel Technique
- Conclusion

Background and Motivation

- Emerging integrated circuits featuring memristors
- Efficient block level representations are required for high level design space exploration

Novel Contributions

- Verilog-A model for memristor which includes threshold and non-linear dopant drift for TiO_2
- Memristor programmable Schmitt trigger oscillator design
- Polynomial metamodels for proposed oscillator
- Verilog-AMS model embedding polynomial metamodels

Memristor Model

- The Memristor Model
  - Based on coupled variable-resistor of TiO_2

$$V_M = [R_{on} x + R_{off}(1-x)] I_M, \quad (1)$$

$$\frac{dx}{dt} = \mu_v \left(\frac{R_{on}}{D^2} \right) I_M, \quad (2)$$

  - Hard switching/terminal state problem

Proposed Memristor Model

- An exponential dopant drift model expressed

$$v \approx \begin{cases} \mu_v E & \text{if } E \ll E_o, \\ \mu_v E_o e^{E/E_o} & \text{if } E \sim E_o, \end{cases} \qquad (3)$$

- Eq 2 is modified to be extended as:

$$\frac{dx}{dt} = \begin{cases} \mu_v \frac{V_p}{D^2} e^{\frac{R_{on}}{V_p} I_M} & \text{if } V_M \geq V_p, \\ \mu_v \frac{V_n}{D^2} e^{\frac{R_{on}}{V_n} I_M} & \text{if } V_M \leq V_n, \\ \mu_v \frac{R_{on}}{D^2} I_M & \text{otherwise.} \end{cases} \qquad (4)$$

- Circuit Parameters are retained

Verilog-A Memristor Model

- Spice memristor models are source type dependant
- Verilog-A is not restricted and hence more suitable.
- Memristor device model implemented in verilog-A ansd MATLAB Simulink

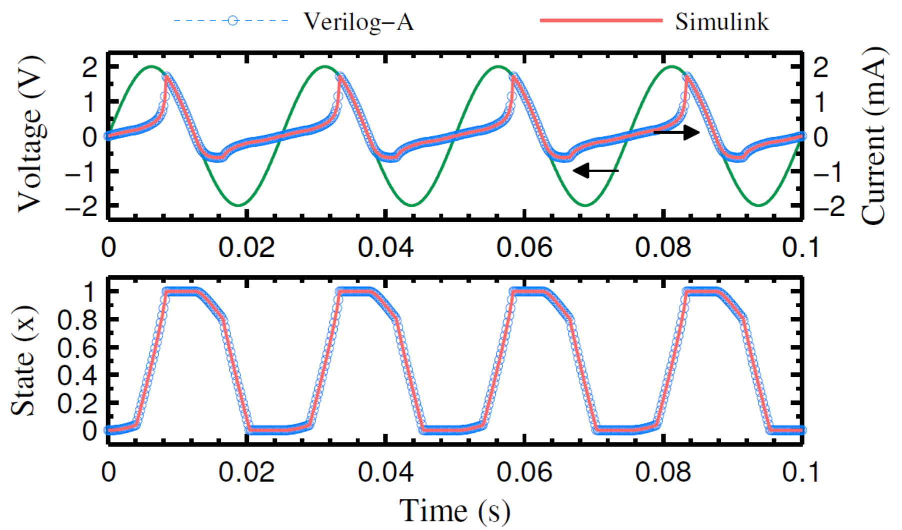


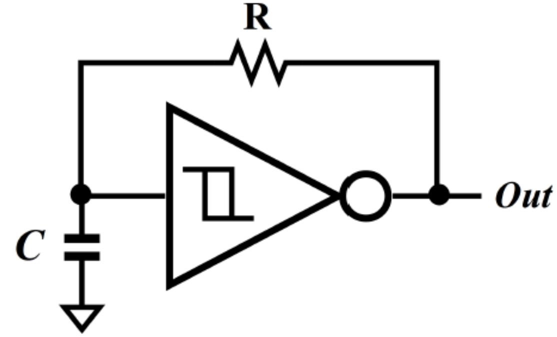
Fig. 1: Time-domain simulation of memristor. A 2-V, 40-Hz sine wave is applied to a memristor: $R_{on} = 1$ kΩ, $R_{off} = 10$ kΩ, $\mu_v = 10^{-14}$ $\mathrm{m^2/s/V}$, $V_p = 1.7$ V, $V_n = -1.7$ V.

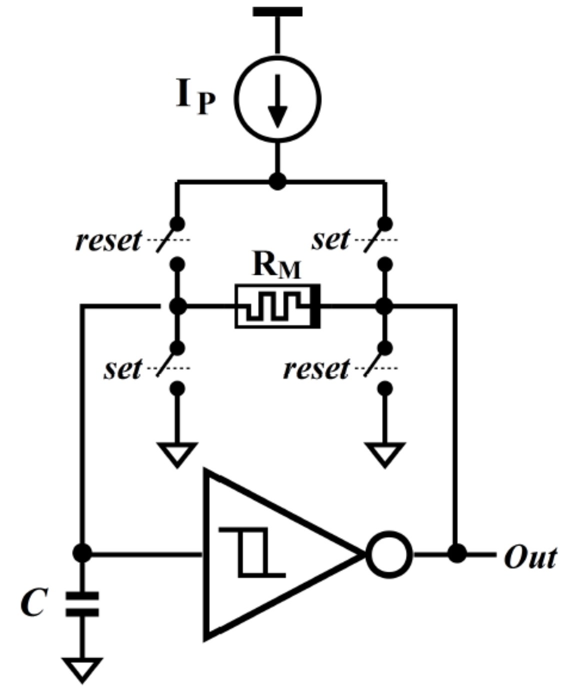
Memristor Based Programmable Oscillator

$$f_o = \frac{1}{t_1 + t_2}$$

$$t_1 = RC \cdot \ln\left|\frac{V_{SH}}{V_{SL}}\right|$$

$$t_2 = RC \cdot \ln\left|\frac{V_{DD} - V_{SL}}{V_{DD} - V_{SH}}\right|$$





(a) Schmitt trigger oscillator (b) Memristor based programmable oscillator.

Fig. 2: The conventional Schmitt trigger oscillator and the proposed one with added programmability.

Memristor Based Programmable Oscillator

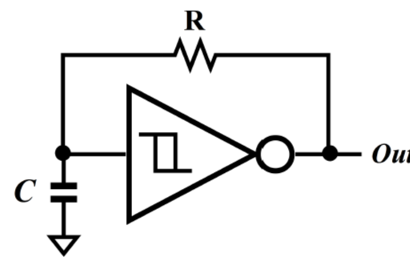
Programming Process

- **Reset:**
  - Reset closed
  - Memristor state set to 1
- **Set:**
  - Set closed
  - Set to desired state.

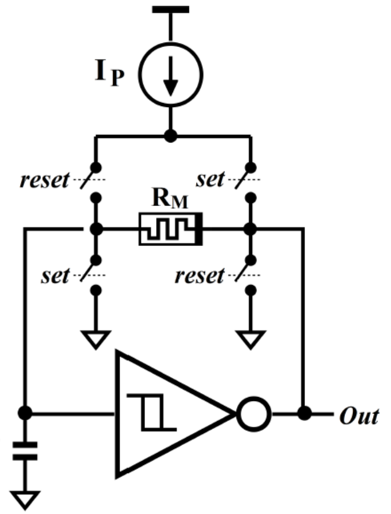
$$f_o = \frac{1}{t_1 + t_2}$$

$$t_1 = RC \cdot \ln\left|\frac{V_{SH}}{V_{SL}}\right|$$

$$t_2 = RC \cdot \ln\left|\frac{V_{DD} - V_{SL}}{V_{DD} - V_{SH}}\right|$$



(a) Schmitt trigger oscillator



(b) Memristor based programmable oscillator.

Fig. 2: The conventional Schmitt trigger oscillator and the proposed one with added programmability.

Programming Simulation

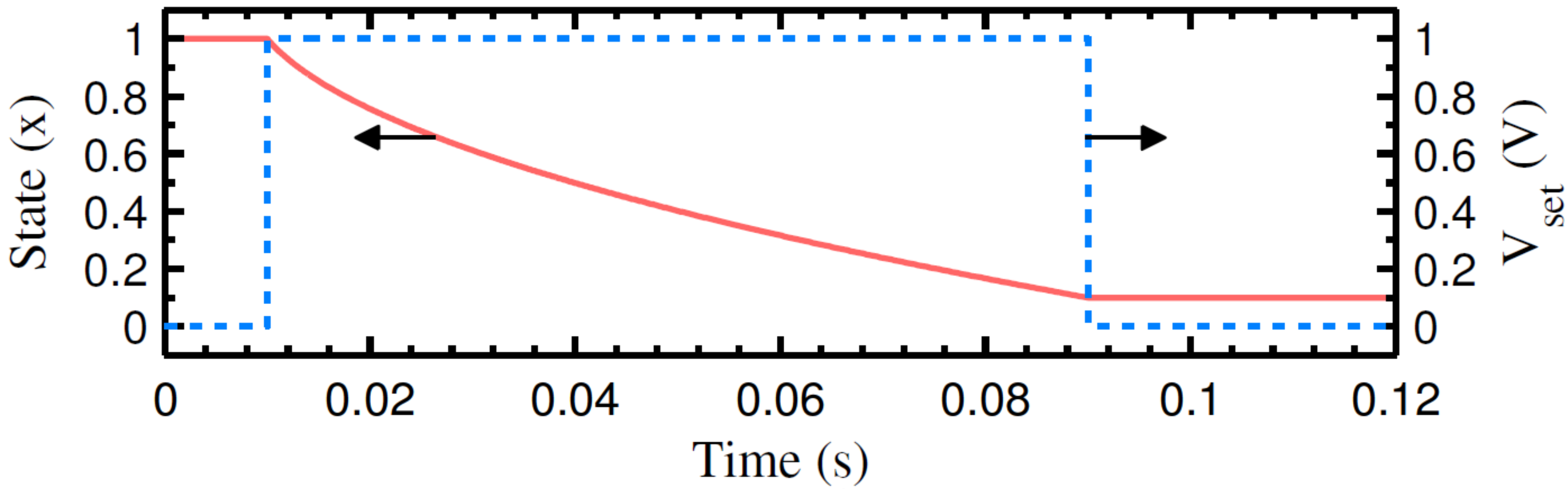


Fig. 3: Programming the memristor state.

- t_{set} = 80ms,
- x = 0.1

CMOS Implementation

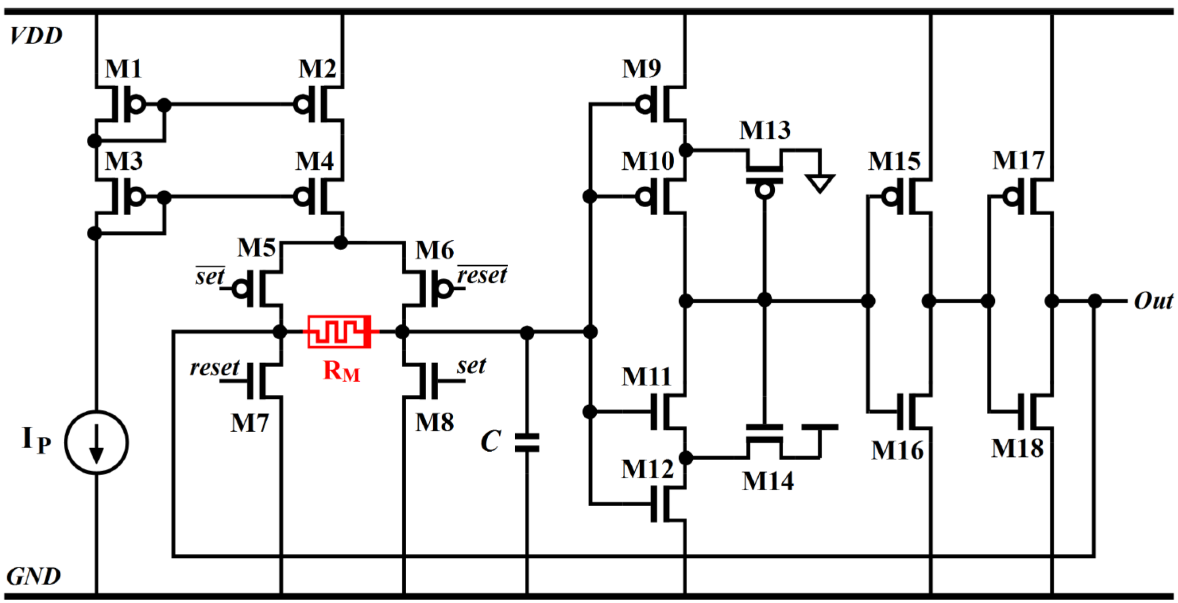


Fig. 4: Memristor based programmable oscillator design.

- 90nm technology
- 1-V Power supply
- Ln, Lp = 100nm

Oscillator Characteristics

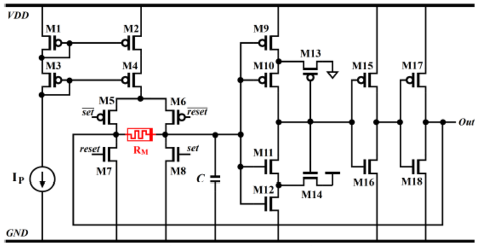
TABLE I: Oscillator Characteristics

Memristor based Programmable Oscillator			
$R_M(x)$	**Frequency**	**Jitter**	**Power**
100 kΩ (0)	34.4915 MHz	536 ps	130.4 μW
55 kΩ (0.5)	59.662 MHz	323 ps	133.9 μW
10 kΩ (1)	251.459 MHz	69 ps	161.2 μW
Schmitt Trigger Oscillator			
R	**Frequency**	**Jitter**	**Power**
100 kΩ	34.4887 MHz	615 ps	130.4 μW
55 kΩ	59.6531 MHz	257 ps	133.9 μW
10 kΩ	251.454 MHz	83 ps	161.2 μW

Power has no noticeable change

For R = 100KΩ,
Schmitt exhibits larger jitter due to domination from resistor thermal noise
For R = 50KΩ,
Resistor noise is reduced, memristor exhibits more jitter
For R = 10KΩ,
Memristor is insensitive to higher frequency, schmitt exhibits more jitter.

Metamodeling for the Memristor Based Design

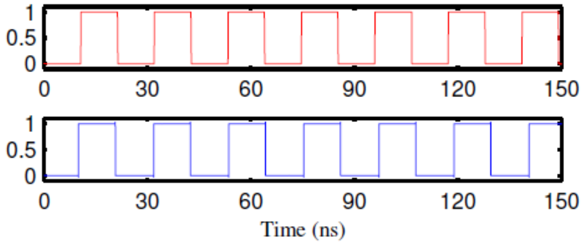


$$f_{POM}(\mathbf{v}) = \sum_{i=0}^{N_B-1} \beta_i \prod_{j=0}^{N_D-1} v_j^{p_{ij}},$$

Polynomial Metamodeling

Verilog-AMS Module

Verilog-AMS Integration



Transient Simulation

Design Parameter Ranges

TABLE II: Variables and their ranges for $x_{POM}(\mathbf{v})$.

Variable	Minimum	Maximum	Device
t_{set}	0 ms	100 ms	-
I_p	80 μA	120 μA	-
R_{on}	8 kΩ	12 kΩ	R_M
R_{off}	80 kΩ	120 kΩ	R_M
W_P	1.6 μm	2.4 μm	M9, M10
W_N	0.8 μm	1.2 μm	M11, M12
$x_{POM}(\mathbf{v}) = x_{POM}(t_{set}, I_p, R_{on}, R_{off}, W_P, W_N)$			

Memristor Surface Response

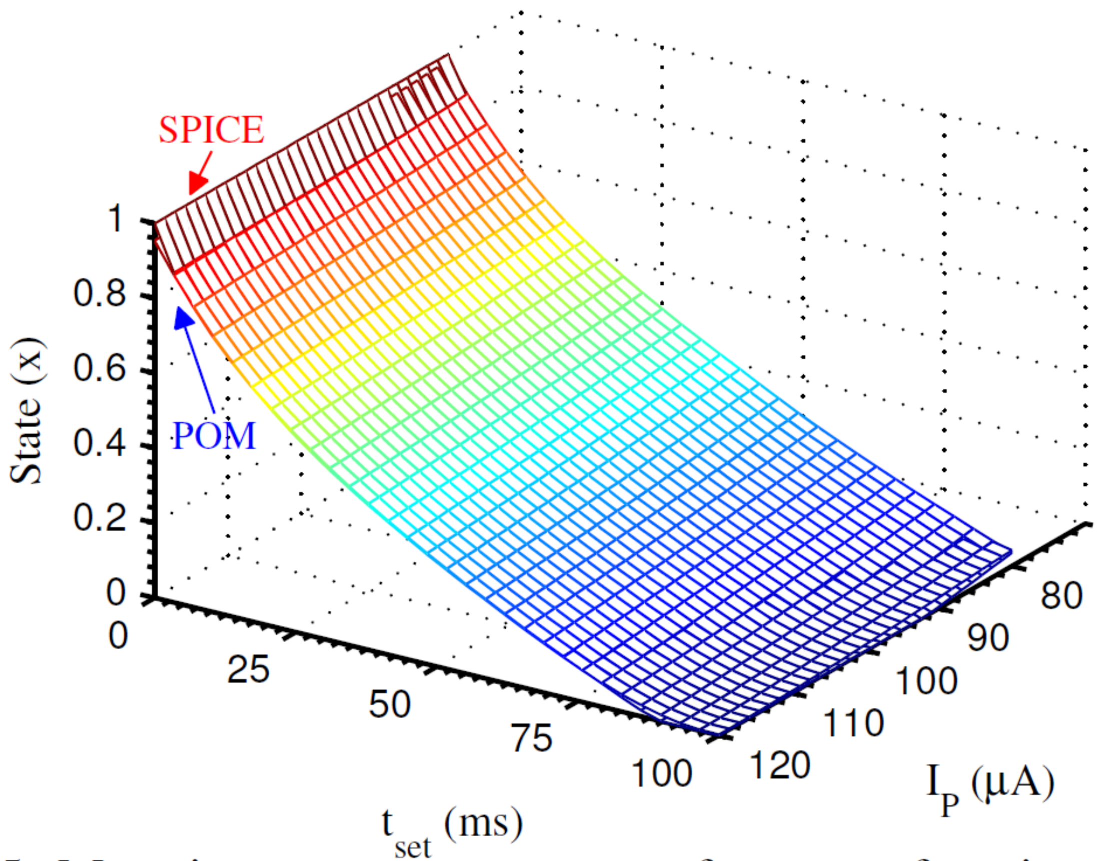


Fig. 5: Memristor state response surface as a function of the programming current and the set time.

Surface response for a subset of t_{set} and I_P

Transient Analysis of Memristor Oscillator

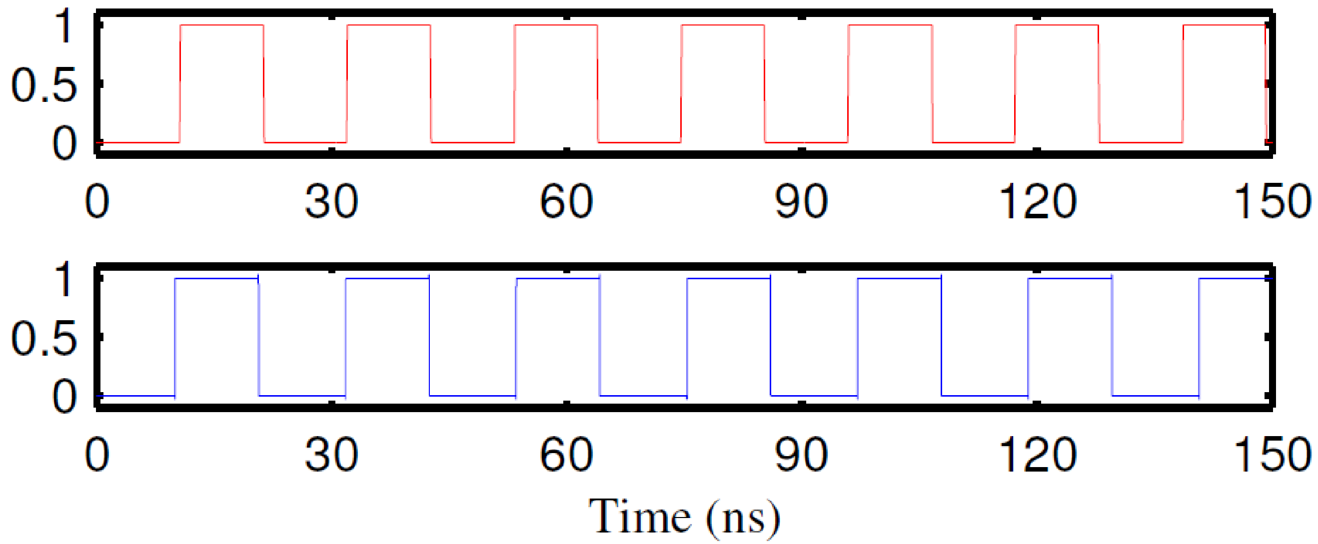


Fig. 6: Transient analyses of the memristor based oscillator using Verilog-AMS-POM (top) and SPICE netlist (bottom).

Comparison of Verilog-AMS-POM with SPICE

TABLE III: Runtime of memristor-based oscillator.

x	**Verilog-AMS-POM**	**SPICE**	**Speedup**
0.1	10.72 ms	360.31 s	33704×
0.3	11.60 ms	335.96 s	28962×
0.5	11.77 ms	349.10 s	29660×
0.7	11.73 ms	348.08 s	29674×
0.9	10.91 ms	355.97 s	32628×

Conclusion

- The proposed memristor device model is demonstrated and retains circuit parameters
- The polynomial metamodeling technique creates accurate and efficient models
- Verilog-AMS-POM improves the simulation speed significantly.

Thank you !!!