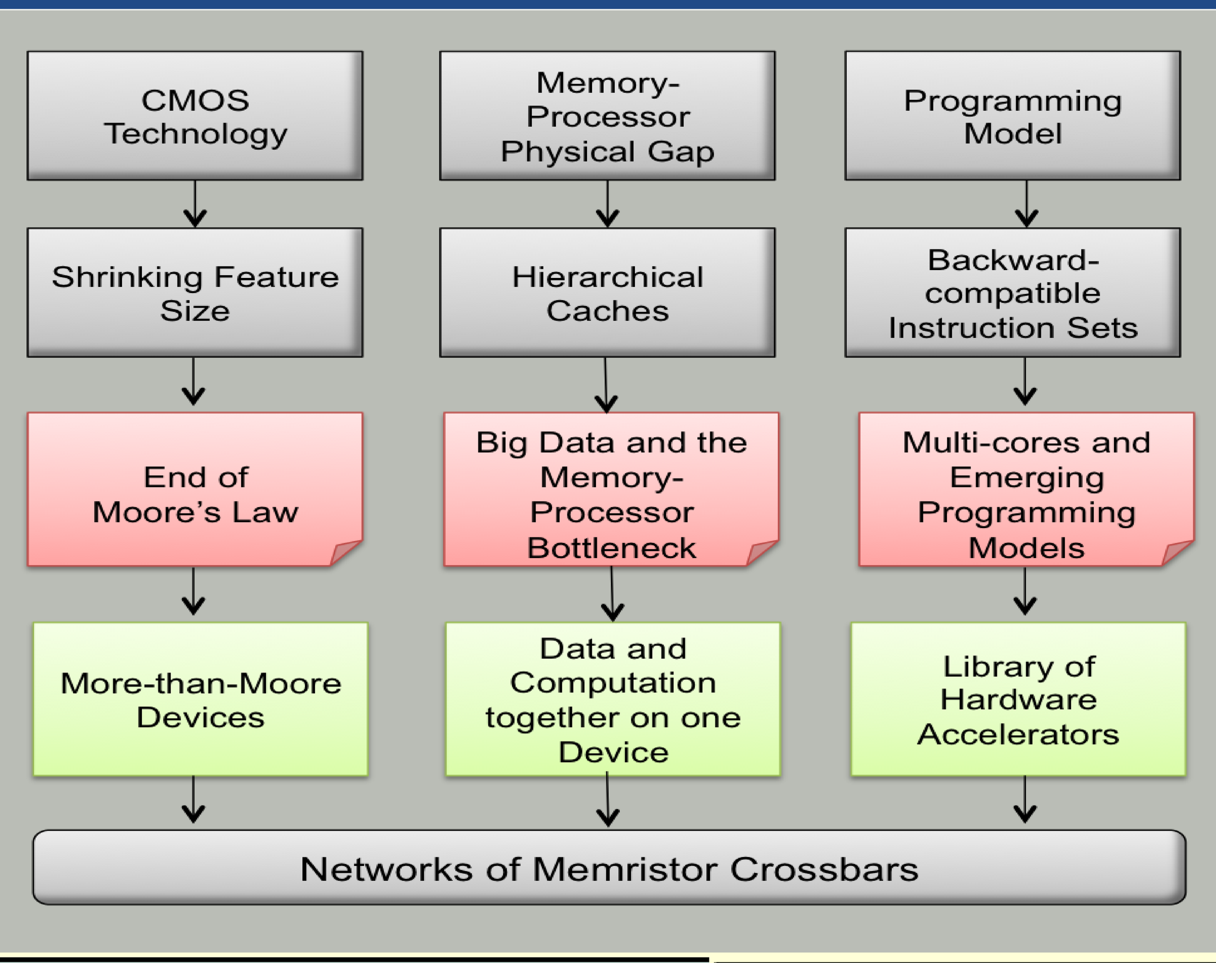
# Flow-based Digital Computing on More-than-Moore Nanoscale Crossbars: Overcoming the von Neumann Bottleneck

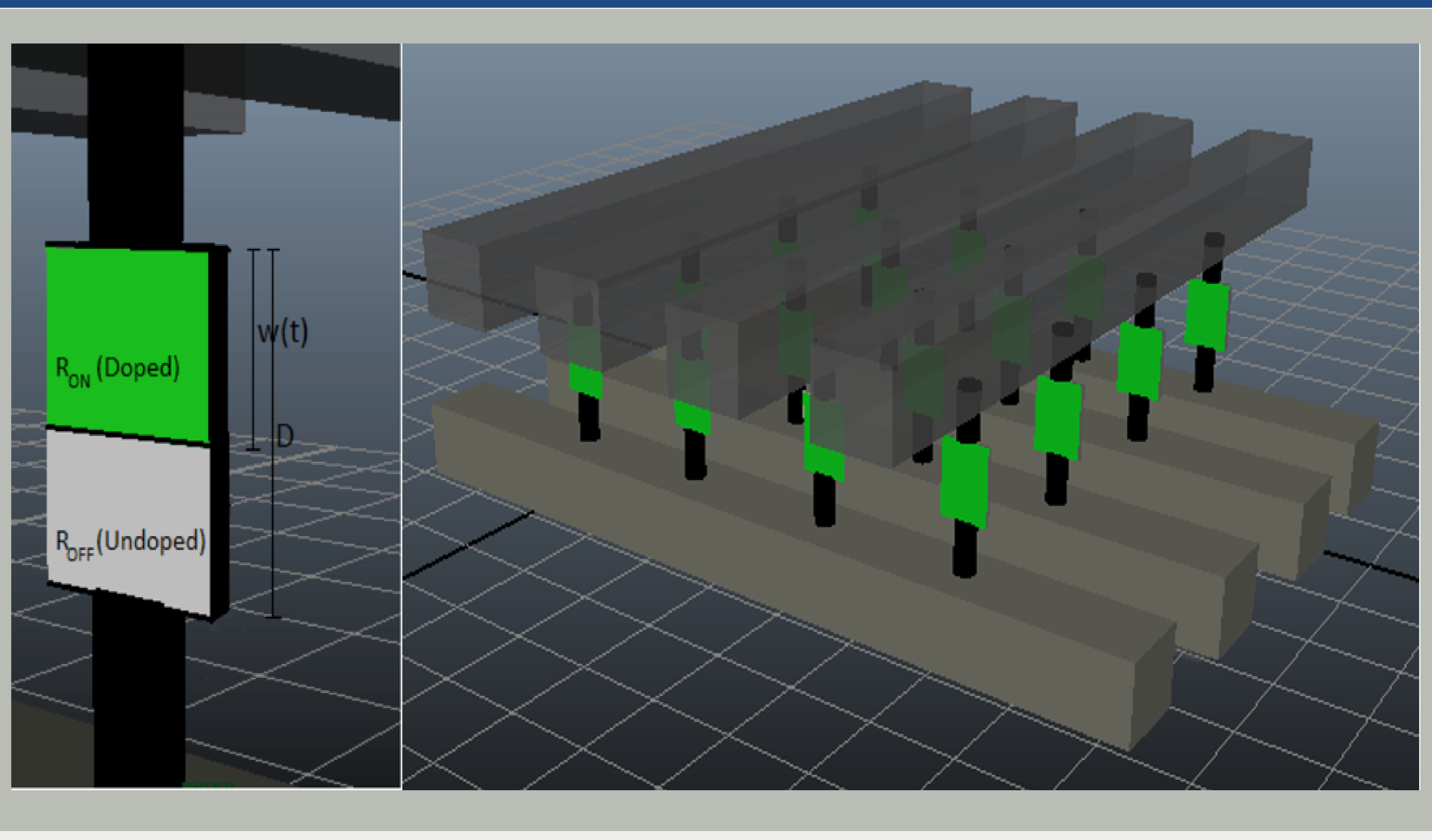
University of Central Florida — SUNY Polytechnic Institute

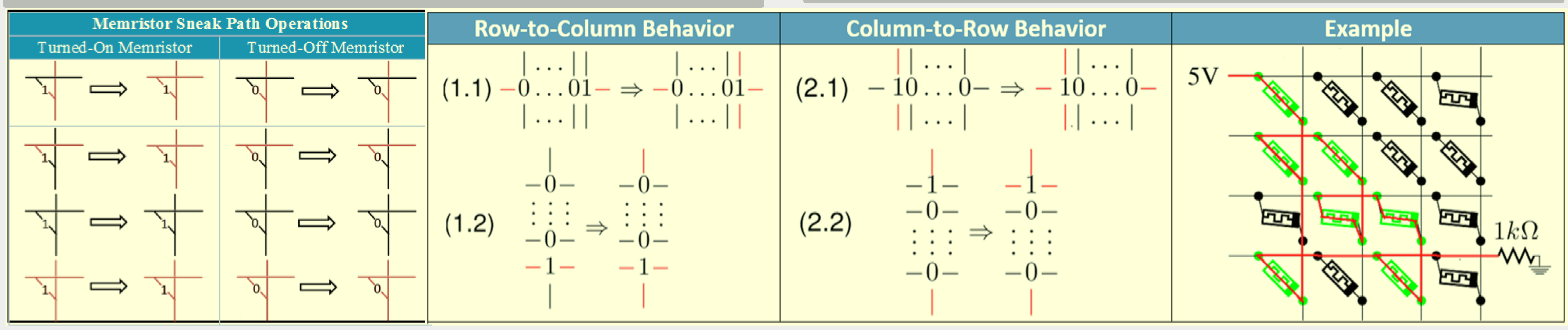
National Science Foundation Award #1438989, Principal Investigators: Sumit Kumar Jha and Nathaniel Cady

## Why compute using memristor crossbars?




## Memristor and Memristor Crossbars

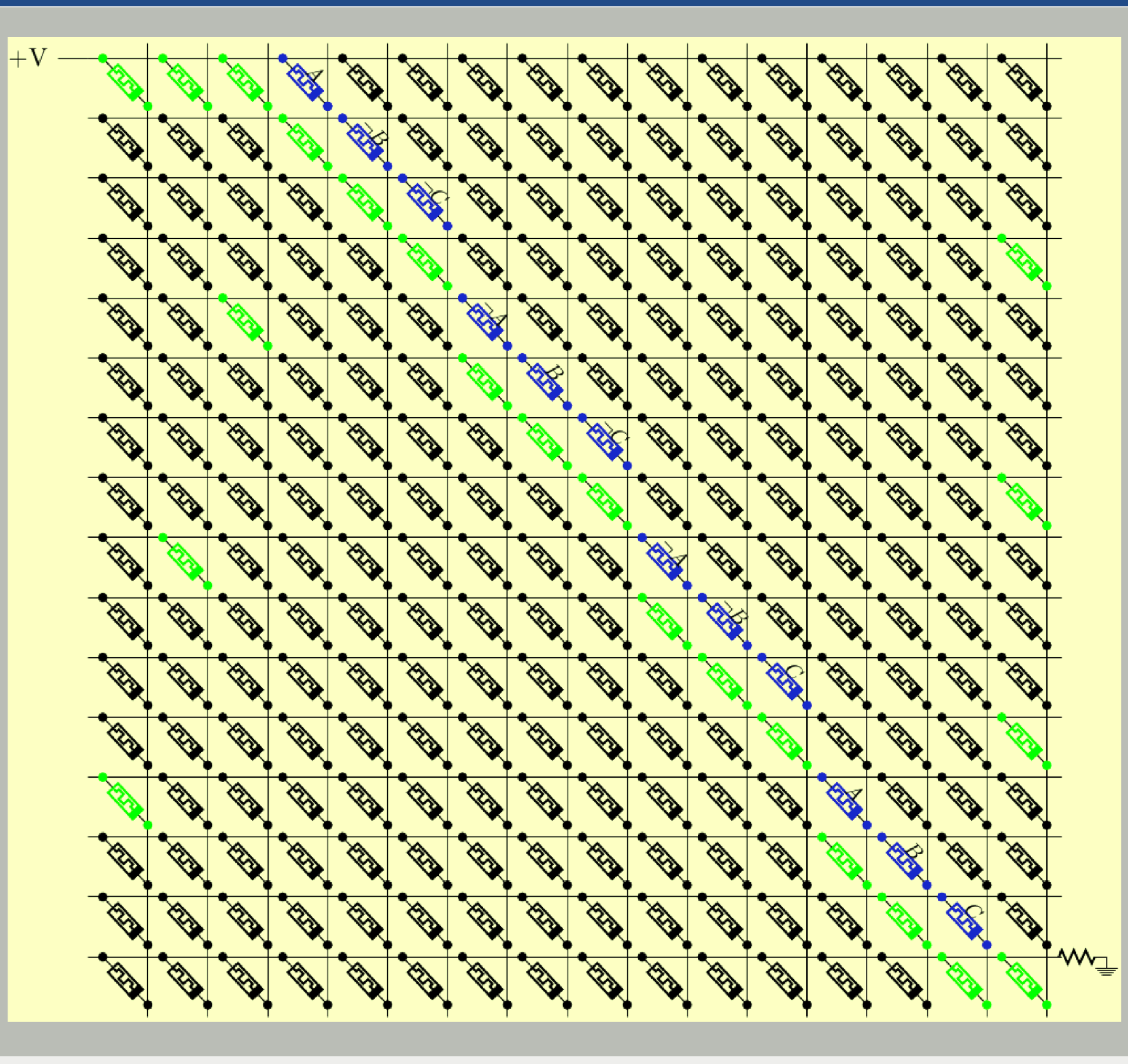



## Memristor Crossbars: Computing and Storage

- A "memory-resistor" or memristor can perform both computation and storage.
- Nanoscale memristors can be assembled into high-density 2-D arrays or crossbars.
- Flow of current through "sneak paths" in a nanoscale crossbar can be used to perform computation.
- Data can be stored on memristors - the stored data in the memristors can modulate the flow of information through "sneak-paths" in a crossbar.

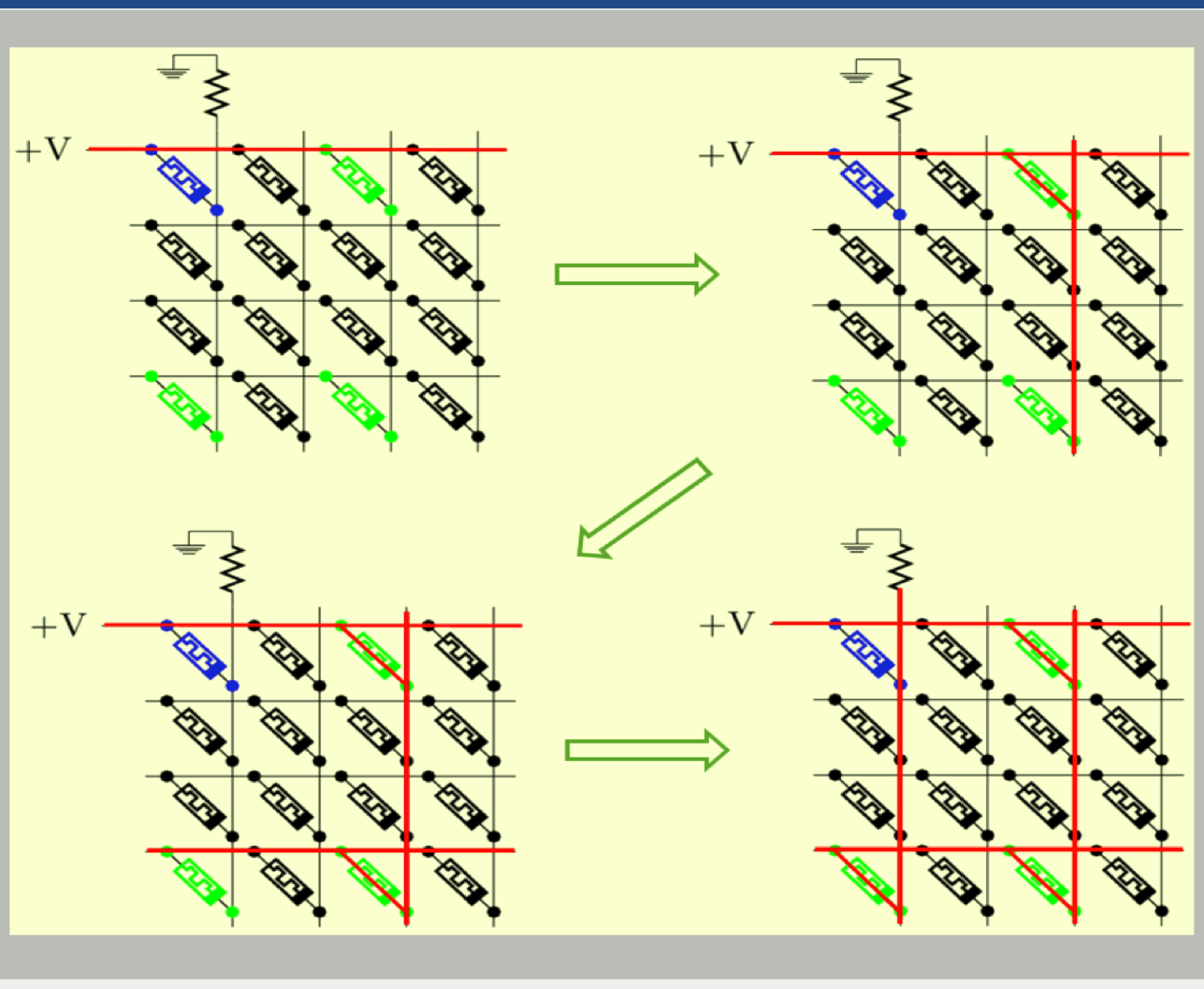
## 1-bit Addition using Sneak Paths - Rule-based Manual Design




## 1-bit Addition using Algorithmically Designed Sneak Paths

| A | B | $C_{in}$ | $S$ | $S$ (measured) | $C_{out}$ | $C_{out}$ (measured) |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0.0012 V | 0 | 0.0003 V |
| 0 | 1 | 0 | 1 | 1.4284 V | 0 | 0.0012 V |
| 1 | 0 | 0 | 1 | 1.4285 V | 0 | 0.0012 V |
| 1 | 1 | 0 | 0 | 0.0018 V | 1 | 1.6665 V |
| 0 | 0 | 1 | 1 | 1.4285 V | 0 | 0.0012 V |
| 0 | 1 | 1 | 0 | 0.0012 V | 1 | 1.5384 V |
| 1 | 0 | 1 | 0 | 0.0013 V | 1 | 1.6666 V |
| 1 | 1 | 1 | 1 | 1.3908 V | 1 | 1.7550 V |

## What is a "sneak" path?

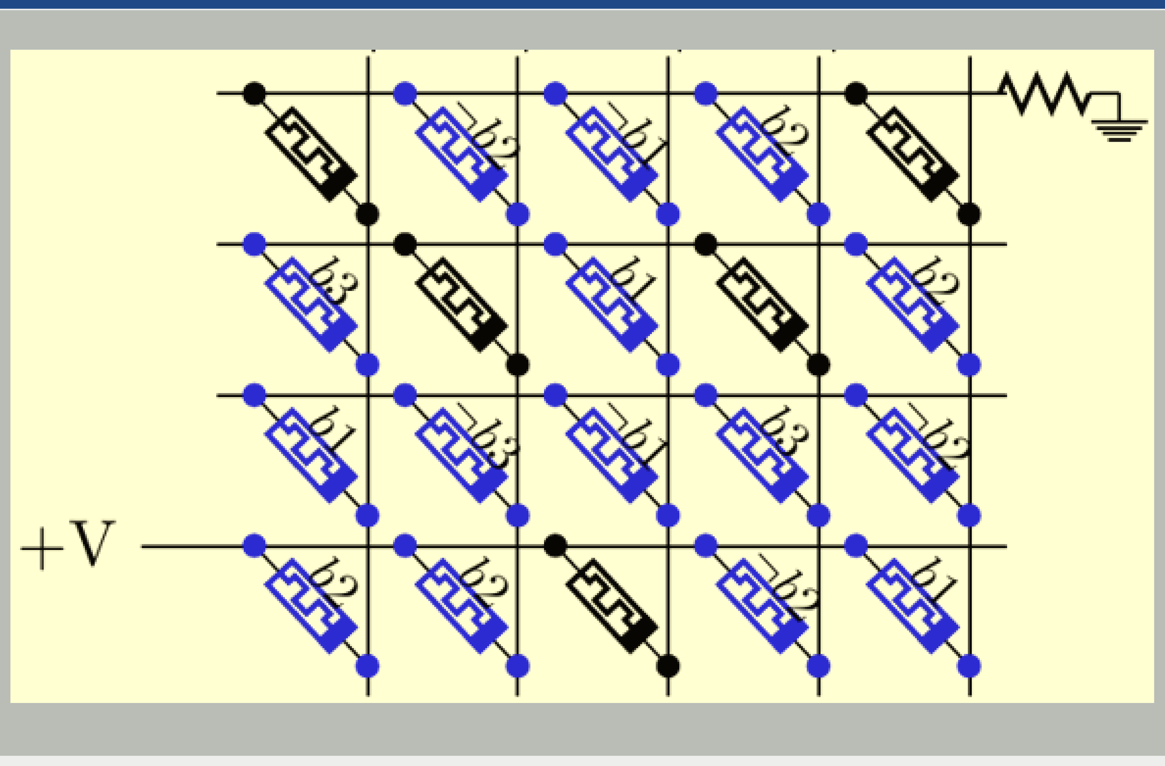



## Rules for Implementing Boolean Computations in Crossbars

$$\mathbb{M}_{\phi_1\wedge\phi_2} = \begin{pmatrix} \begin{pmatrix} & & \\ & M_{\phi_1} & \\ & & \end{pmatrix} & \begin{matrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{matrix} \\ \begin{matrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{matrix} & \begin{pmatrix} & & \\ & M_{\phi_2} & \\ & & \end{pmatrix} \end{pmatrix}$$

$$\mathbb{M}_{\phi_1\vee\phi_2} = \begin{pmatrix} \begin{matrix} 1 \\ 0 \\ \vdots \\ 0 \end{matrix} & \begin{pmatrix} & & \\ & M_{\phi_1} & \\ & & \end{pmatrix} & \begin{matrix} 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & 1 \end{matrix} \\ \begin{matrix} 1 \\ 0 \\ \vdots \\ 0 \end{matrix} & \begin{matrix} 0 & \cdots & 0 \\ 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{matrix} & \begin{pmatrix} & & \\ & M_{\phi_2} & \\ & & \end{pmatrix} \begin{matrix} 0 \\ \vdots \\ 0 \\ 1 \end{matrix} \end{pmatrix}$$

## Algorithmically Designed 1-bit Addition Circuit




## Ongoing and Future Work

- We have experimentally verified the 1-bit adder design.
- We have designed crossbar circuits that operate in the presence of stuck-at faults.
- We have algorithmically designed 2-bit adders and multipliers.
- Experimental validation of 2-bit arithmetic and logical circuits is ongoing.