Precompensation, BIST and Analogue Berger Codes for Multi-level Writing RRAM

Advisor: 黃宗柱 教授

Author: 電子碩一 黃靖

Outline

- Introduction
- Degradation & Defect Model
- Precompensate Analog Berger Code
- Sliding-Diagonal March Test BIST
- Stanford University RRAM VerilogA Model
- Multi-level Writing Simulation
- Conclusions & Future Works

Introduction

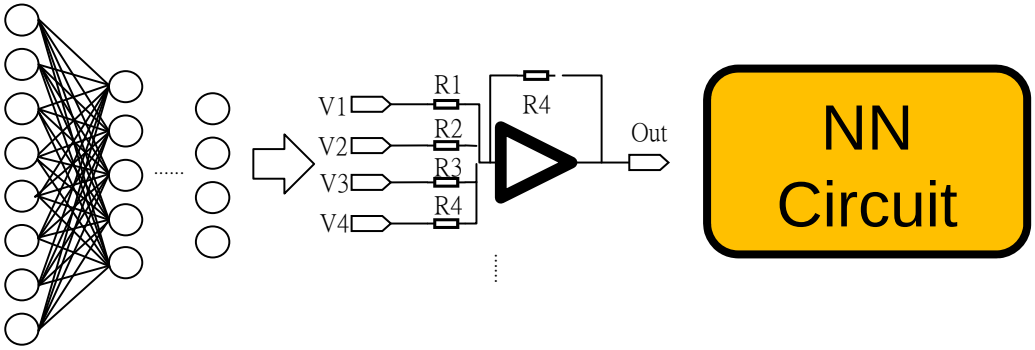
- **Resistive Random Access Memory (RRAM)**
  - High energy-efficient
  - Area-compact (Multi-level)
  - ➔ Good for AI & Automotive SoCs

- **Comparison of memories**

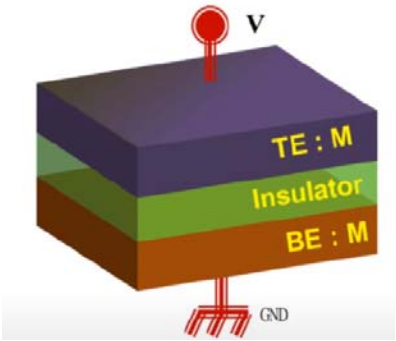
	SRAM	DRAM	Flash	PCRAM	RRAM	MRAM
Cell size	>100F^2	6-8F^2	4-5F^2	8-16F^2	>5F^2	37F^2
Read	<10ns	10-60ns	25us	48ns	<10ns	<10ns
Write	<10ns	10-60ns	200us	40-150ns	~10ns	12.5ns
Endurance	>10^{15}	>10^{15}	10^4	10^8	10^5	>10^{15}
Volatile	Yes	Yes	No	No	No	No

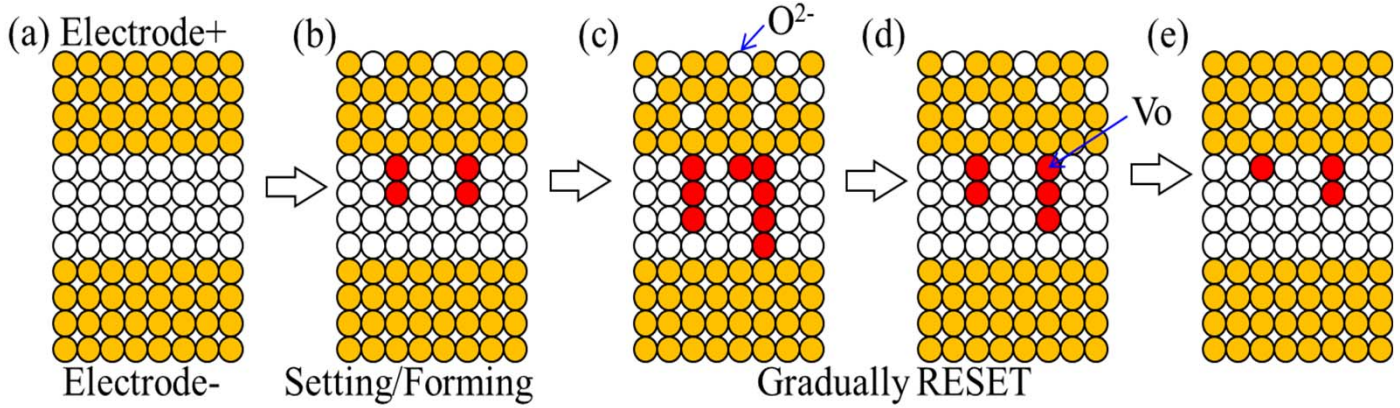
Introduction

- **Neuromorphic RRAM in NN**



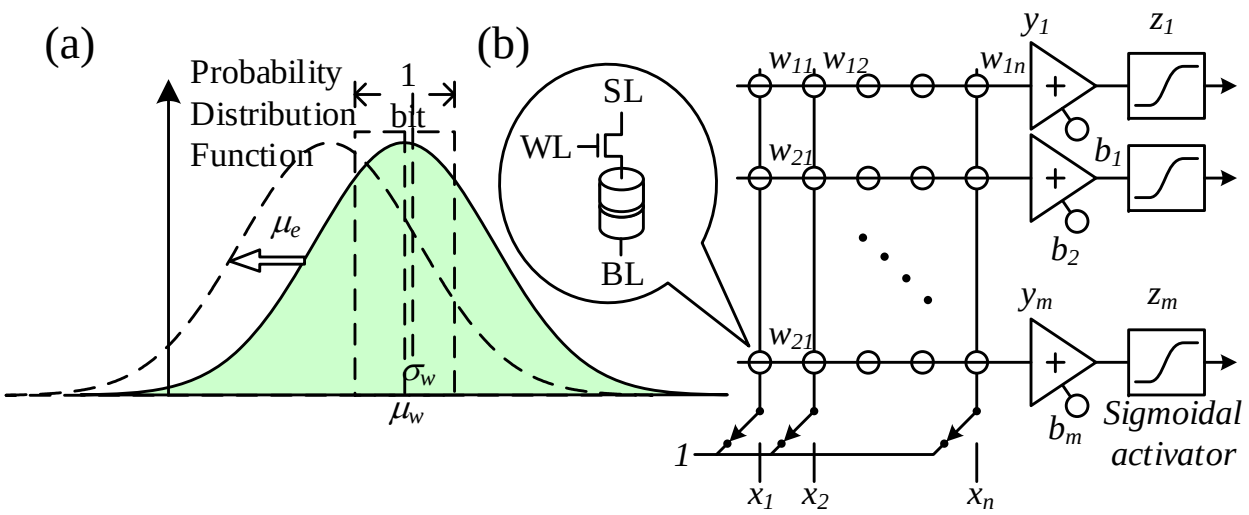
- **Structure :** O^{2-} Filament

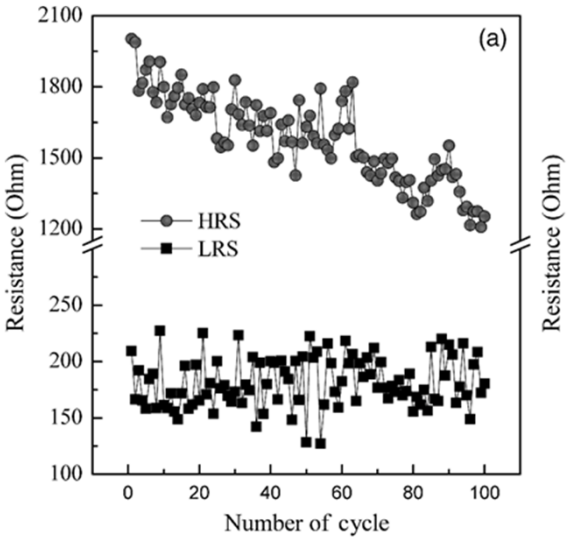




Degradation & Defect Model

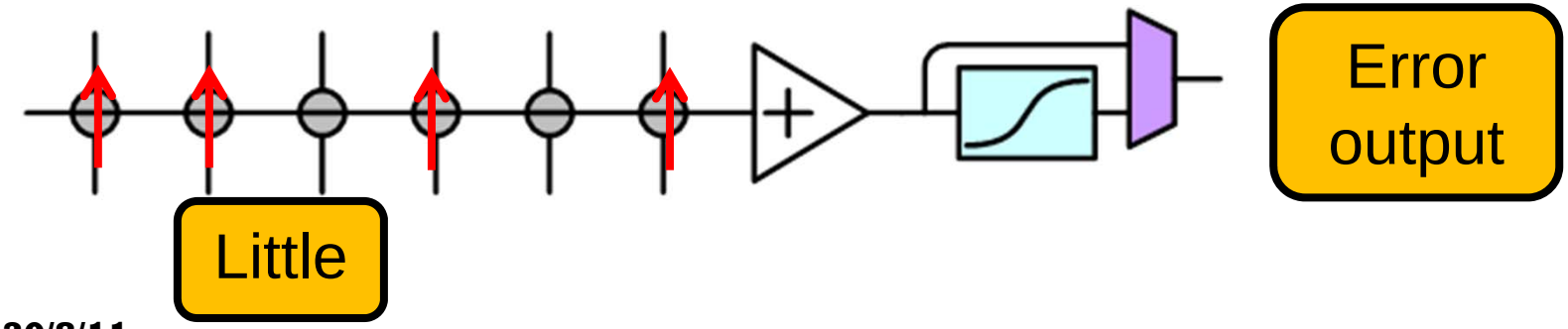
- **Degradation Model**
  - Asymmetric degradation: HRS ➔ LRS
  - Linear decrease in short time





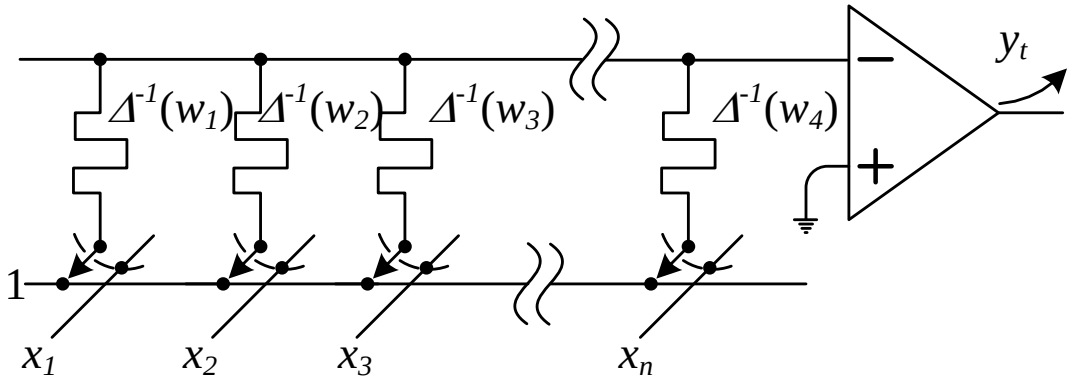
Degradation & Defect Model

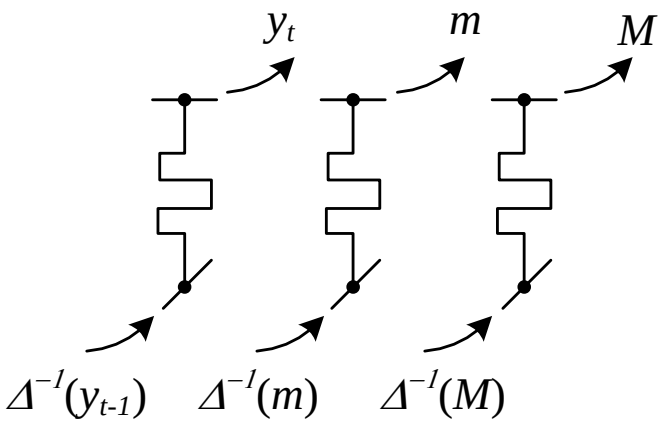
- **Fault Model**
  - Specific value fault: $W0R0W1R1W2R3W3R3$
  - Slow-write fault: $WiR0.99i$
  - Fast-write fault: $WiR1.01i$
- **Dynamic Range Inflate**
- $(\uparrow_{i=0k} Wi) Rsk$, $if((k-i_n)>0.5)$, R_S is the read output value



Precompensate ABC

- **Berger Code:** Detect Asymmetric error
- **Precompensate Analog Berger code**



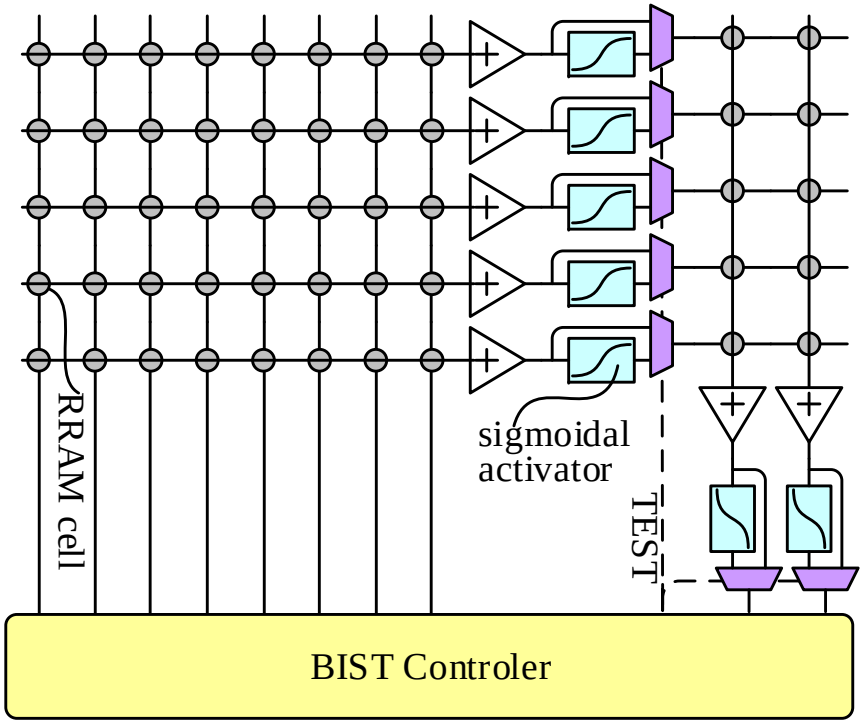


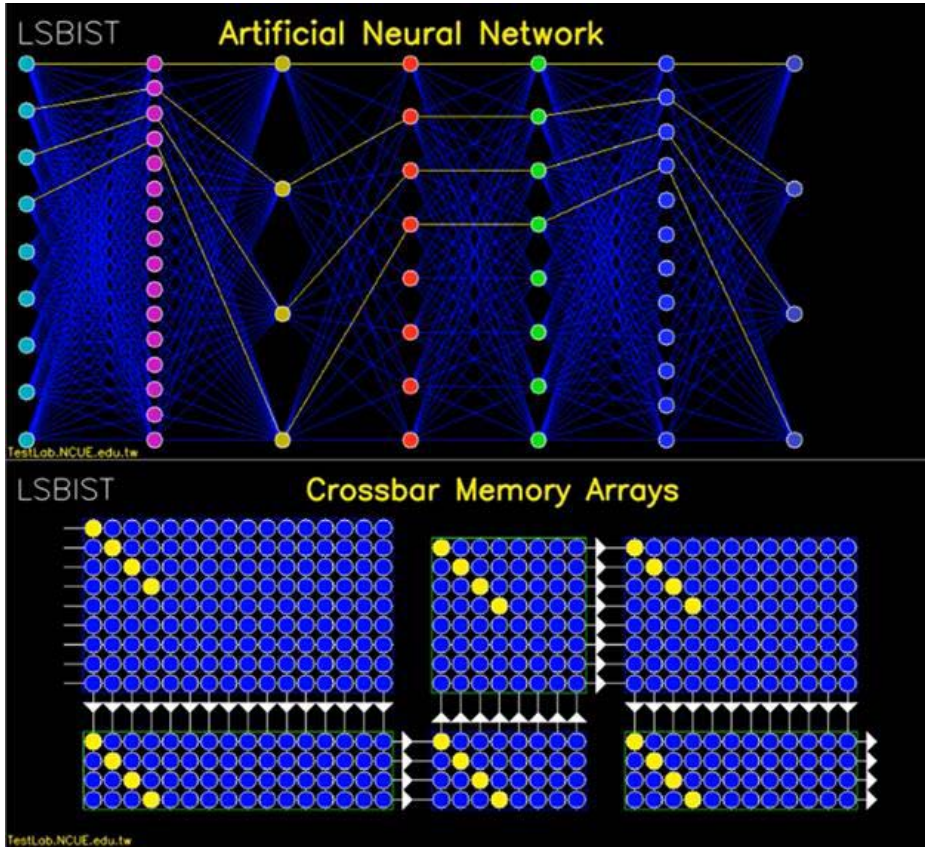
Error output ➔ Precompensate value at output

Compensate cells (Reverse value)

SDMT BIST

- **Sliding-Diagonal March Test**

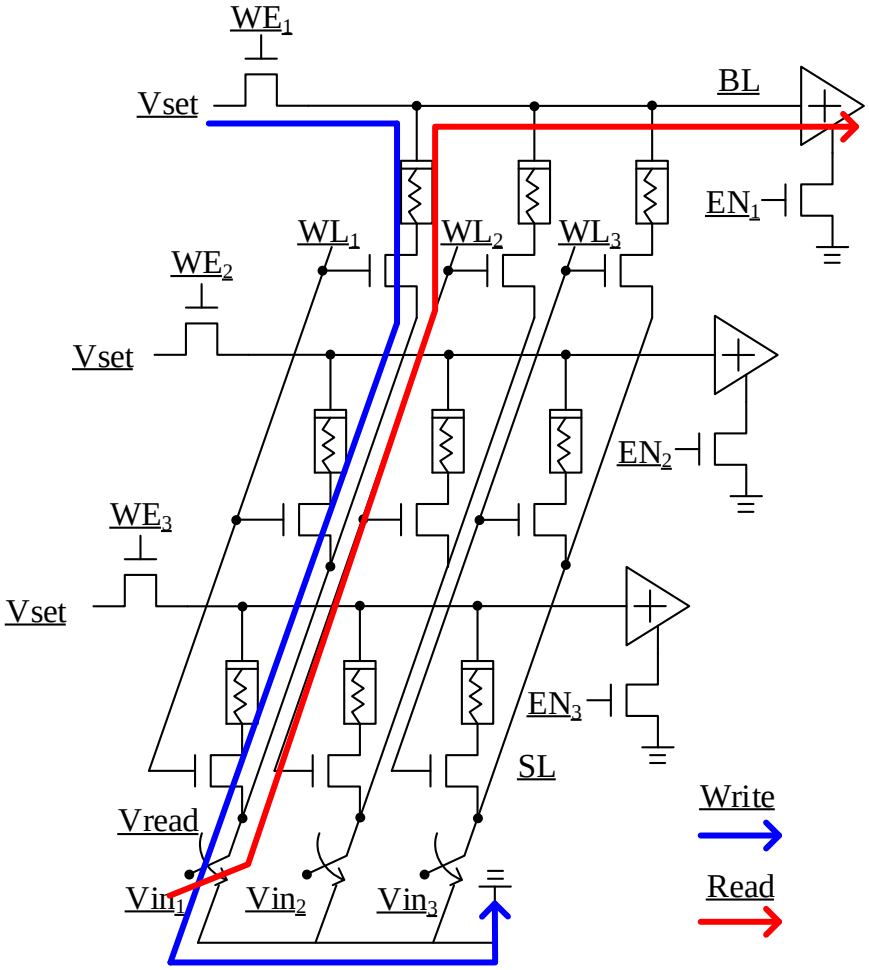




SDMT BIST

- **RRAM Network Circuit**
- **Sneak Path Issue**

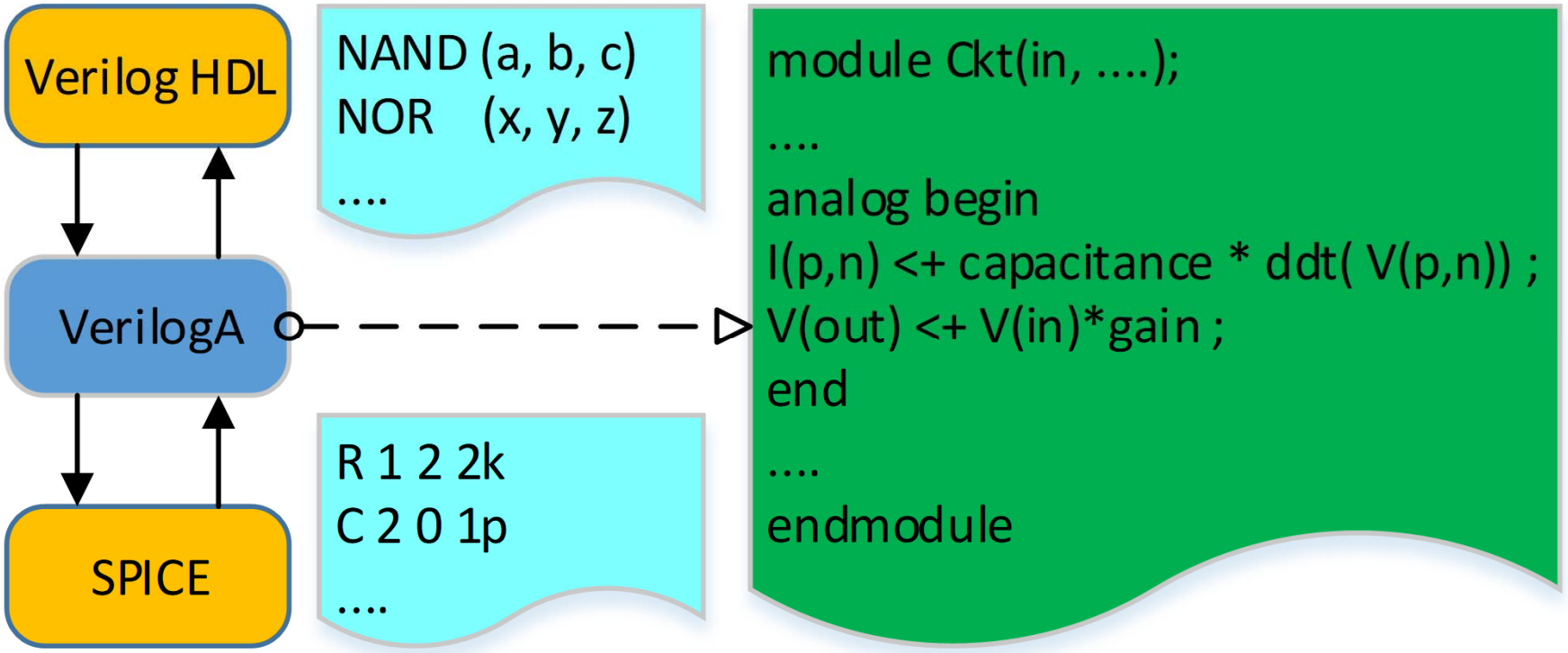
Sneak Path Issue		Set/Bit	Read
Pure Array	1R	**Yes**	**Yes**
	1T 1R	**No**	**No**
Neuromorphic	1R	**Yes**	**No**
	1T 1R	**No**	**No**



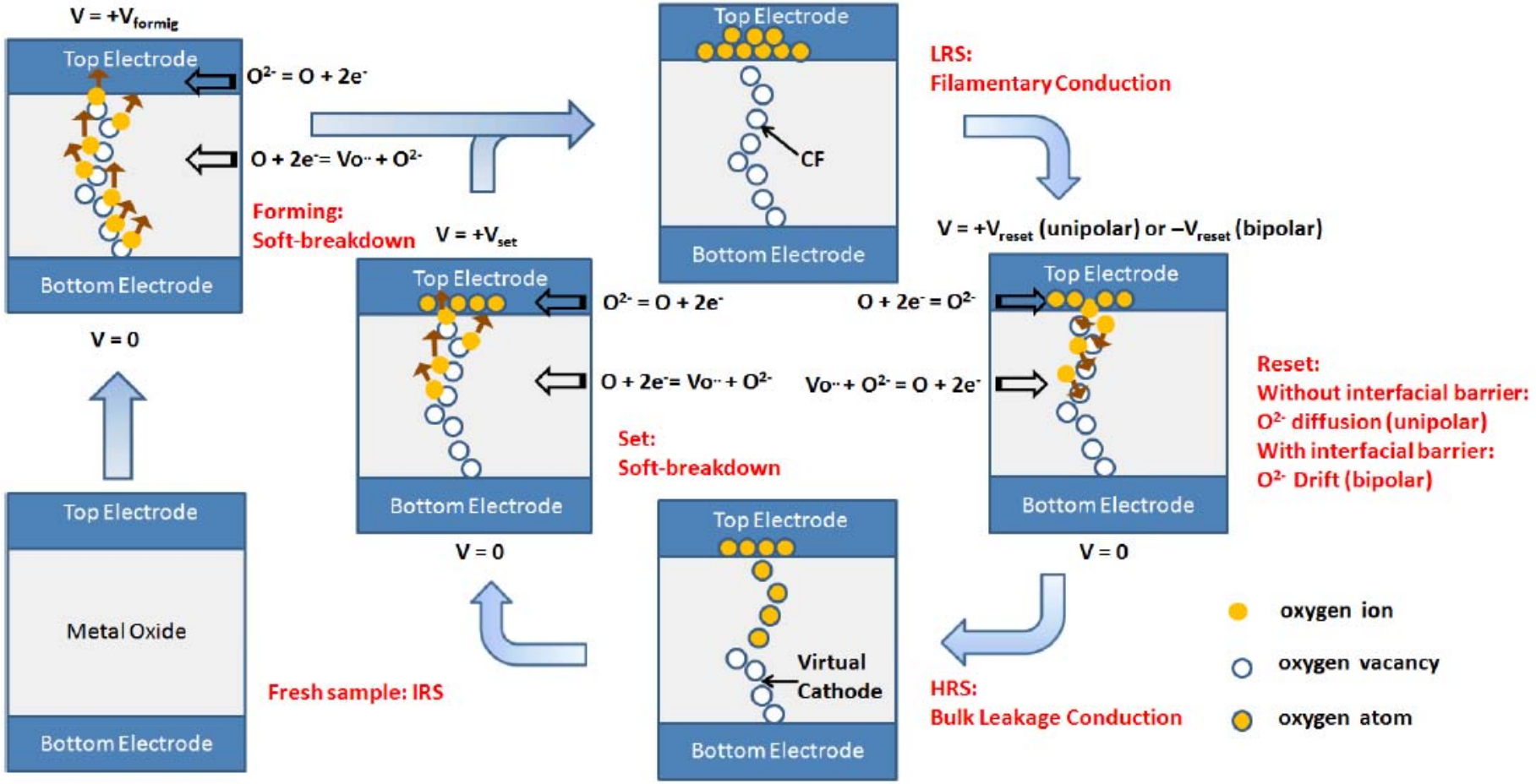
Stanford's RRAM Model

- **VerilogA model:**

HDL code for analog behavior description



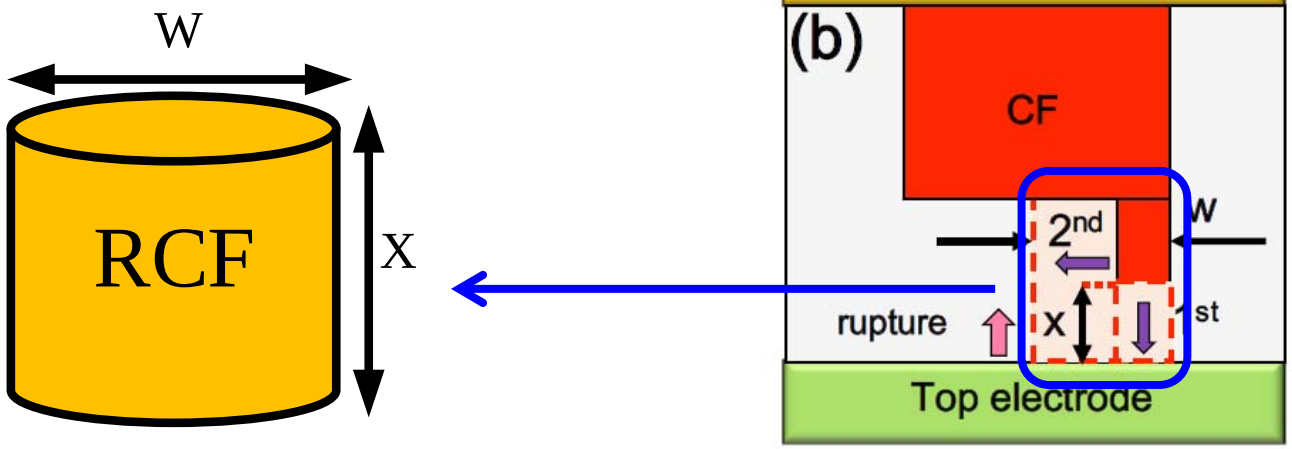
Stanford's RRAM Model



Stanford's RRAM Model

- **VerilogA Code**

```
analog
begin
    Temp = T0 + abs(I(t,b) * V(t,b)) * Rth ;
    I1 = I0*pi*(WCF*WCF/4-w*w/4)*exp(-L0/XT)*sinh(V(t,b)/VT) ;
    RCF = rou*(L0-x)/(pi*w*w/4) ;
    Vg = V(t,b) - (I(t,b)-I1)*RCF ;
```



Stanford's RRAM Model

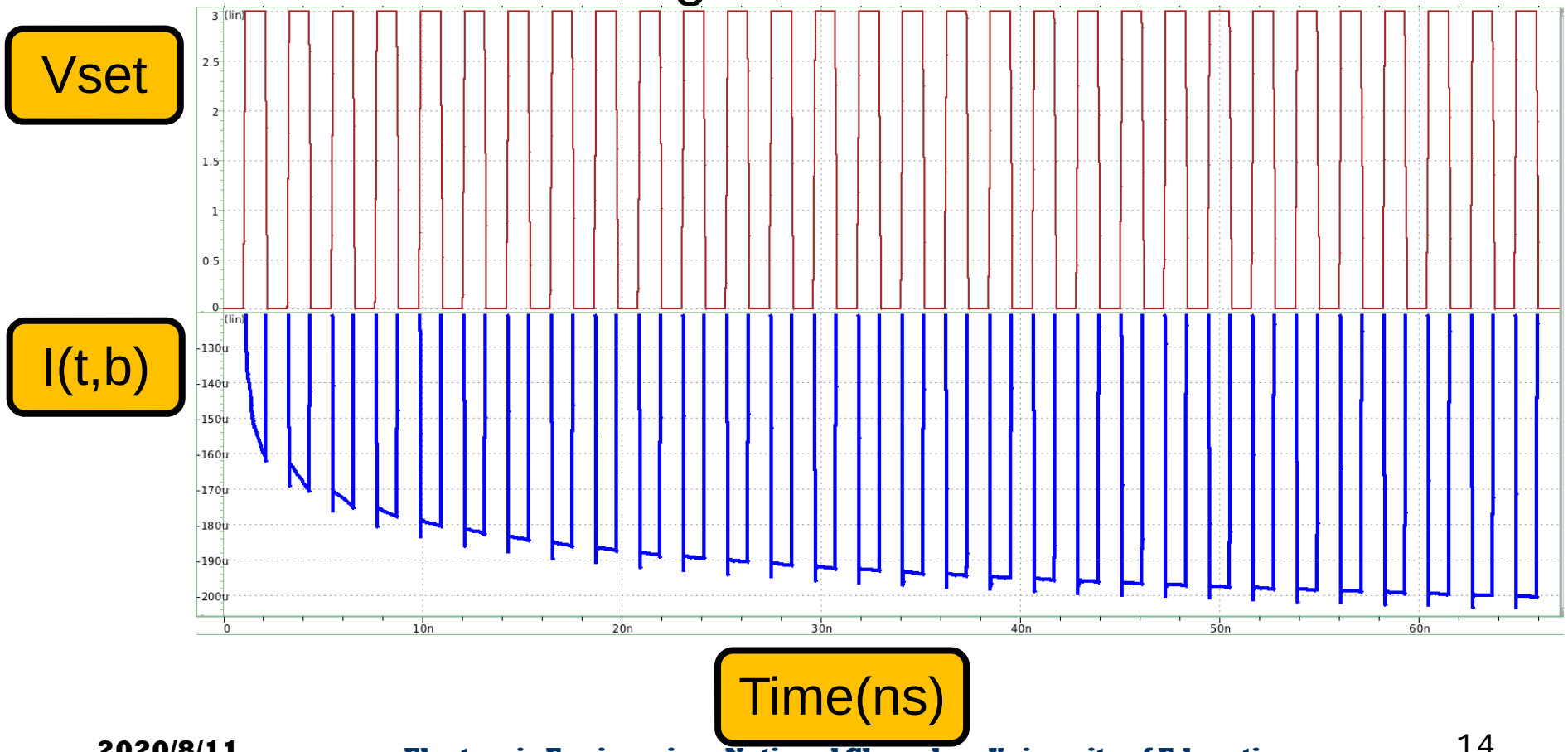
- **VerilogA Code**

```
if ( V(t,b) == 0 )
begin
    I(t,b) <+ 0 ;
end
else if ( V(t,b) > 0 )
begin
    if ( x > 0 )
    begin
        I(t,b) <+ I1 + I0*pi*(w*w/4)*exp(-x/XT)*sinh(Vg/VT) ;
    end
    else if ( x <= 0 )
    begin
        I(t,b) <+ I1 + V(t,b)/(rou*L0/(pi*w*w/4)) ;
    end
end
else if ( V(t,b) < 0 )
begin
    I(t,b) <+ I1 + I0*pi*(w*w/4)*exp(-x/XT)*sinh(Vg/VT) ;
end
```

Describe the amount of current I(t,b)

Stanford's RRAM Model

- **Simulation**: Forming

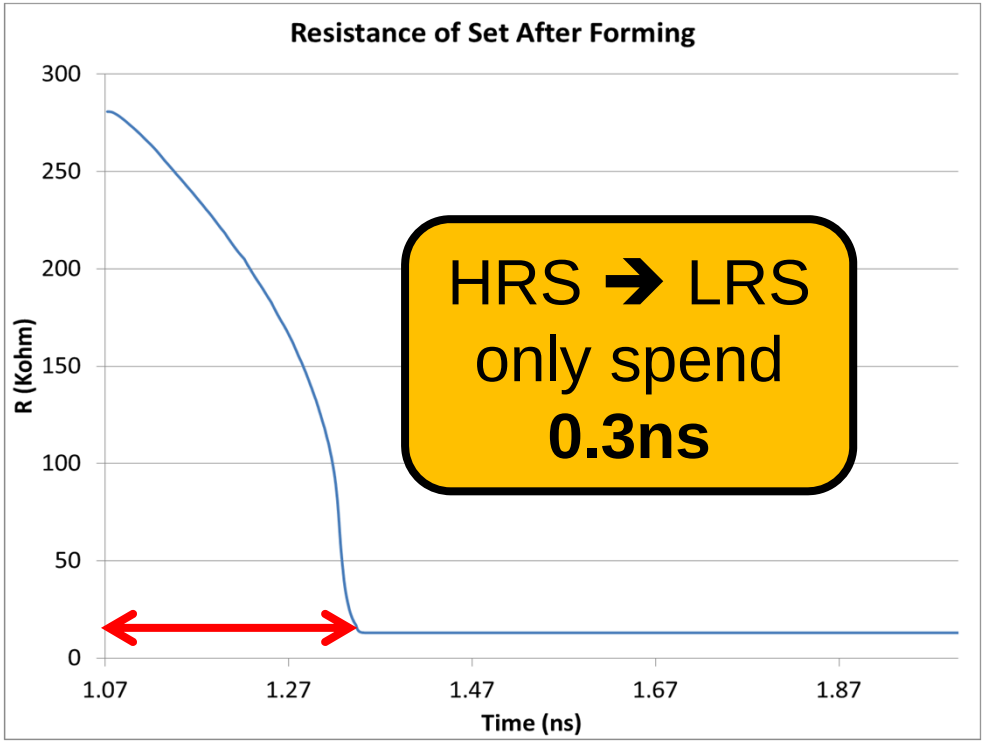


Multi-level Writing Simulation

- **Why Multi-level Writing?**
  - Better than traditional memory (Greater storage)
  - Neuromorphic RRAM for NN
  - Make precompensation effective
- **How to do Multi-level Writing?**
  - Change pulse high (Voltage)
  - Change pulse width (Time)

Multi-level Writing Simulation

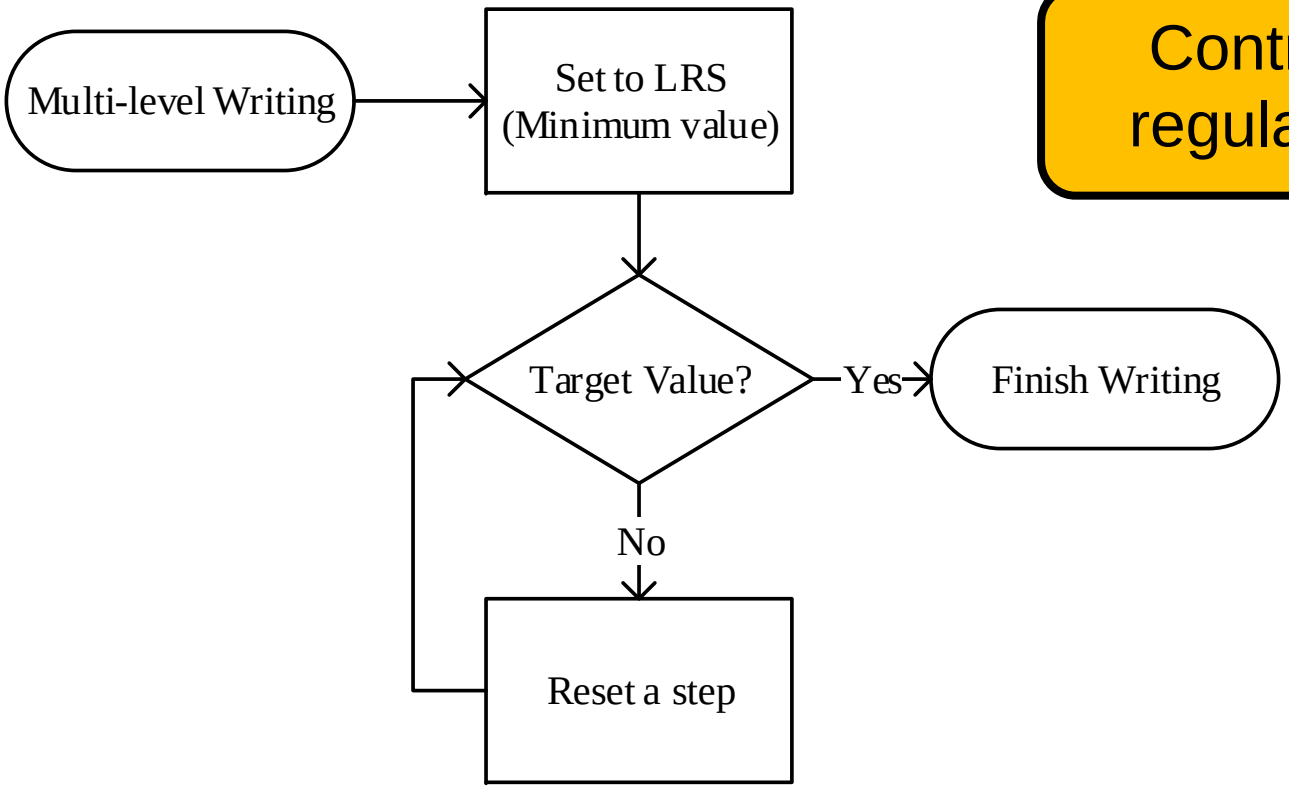
- After Forming, Set does not require much time and bias voltage



Set is EASY, But Reset takes more time and voltage

Multi-level Writing Simulation

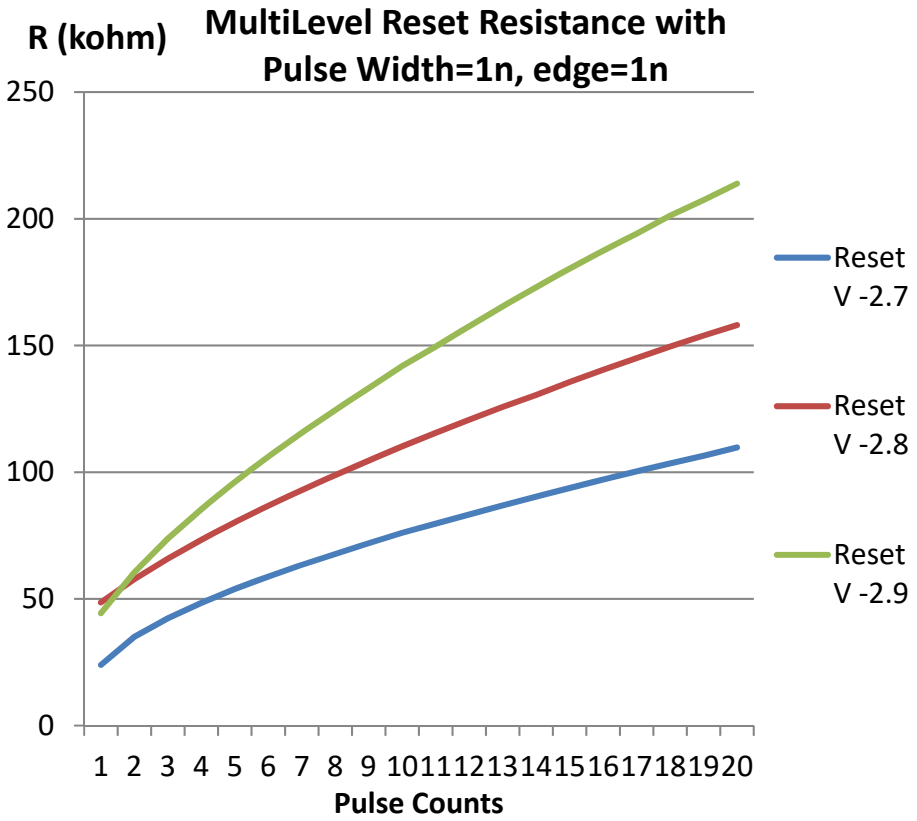
- **Multi-level writing process**

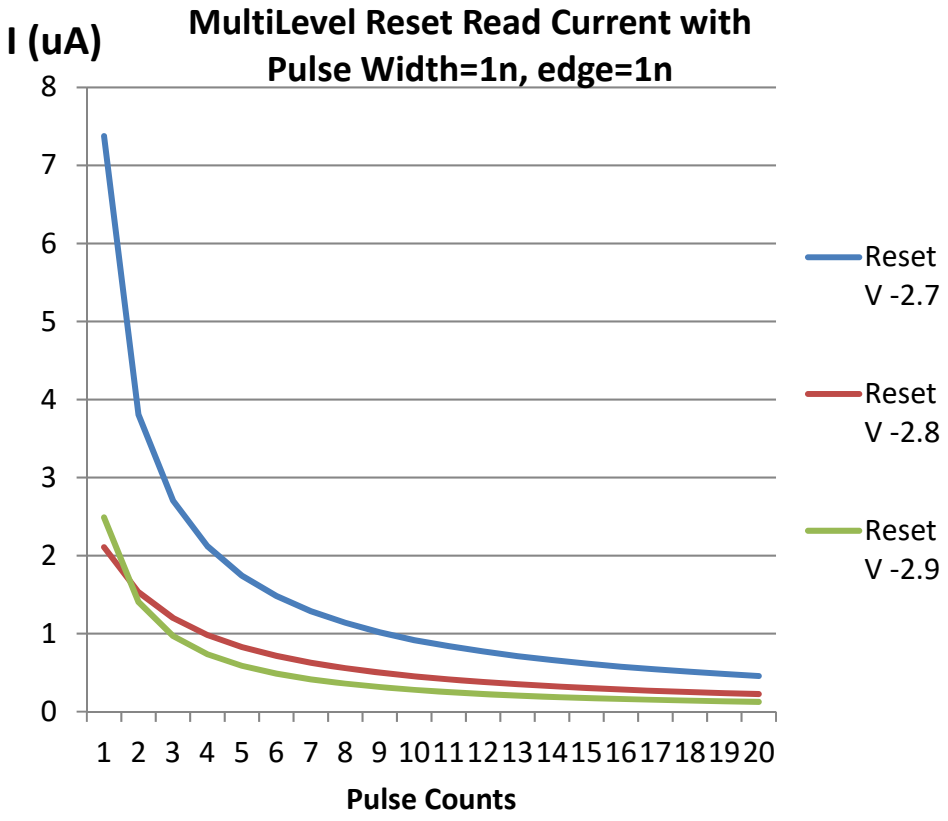


Control by switch, regulator, and FMS

Multi-level Writing Simulation

- **Multi-level writing with different Reset V**

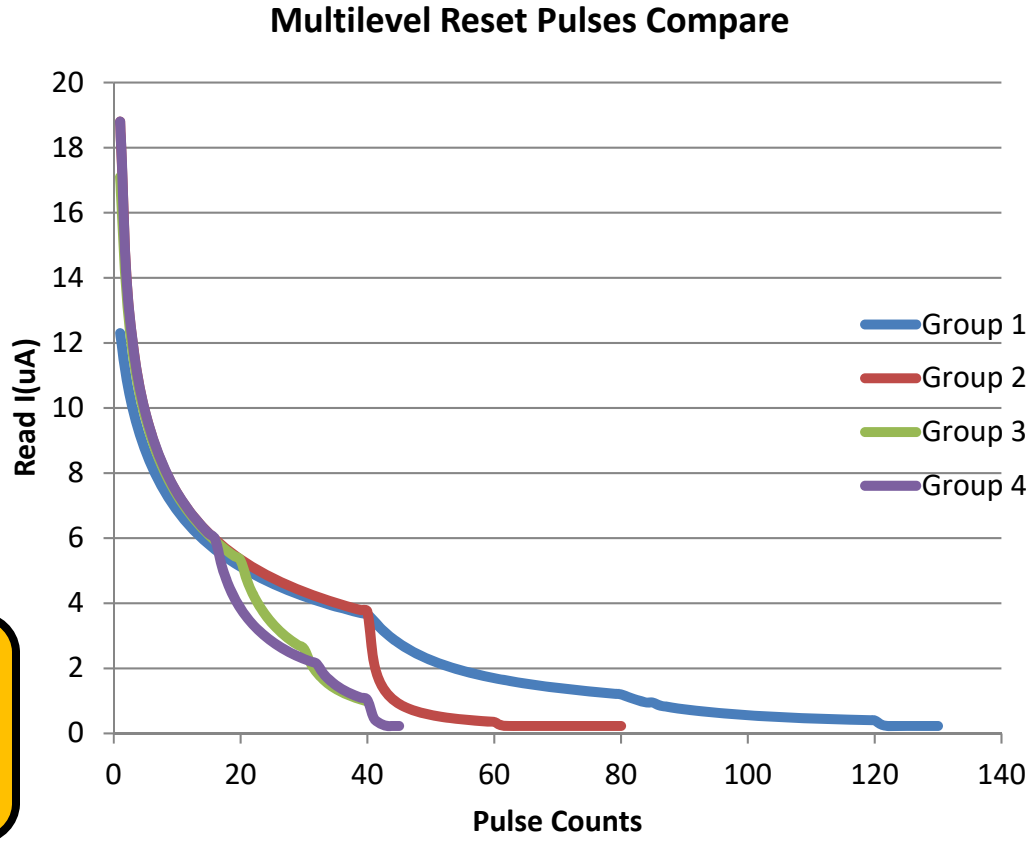




Multi-level Writing Simulation

- Multi-level writing condition with different Reset V

Group 1				
Reset V	-2.7	-3	-3.3	-4
Counts	40	40	40	10
Group 2				
Reset V	-2.7	-3.3	-4	
Counts	40	20	20	
Group 3				
Reset V	-2.7	-3	-3.3	
Counts	20	10	10	
Group 4				
Reset V	-2.7	-3	-3.3	-4
Counts	16	16	8	4

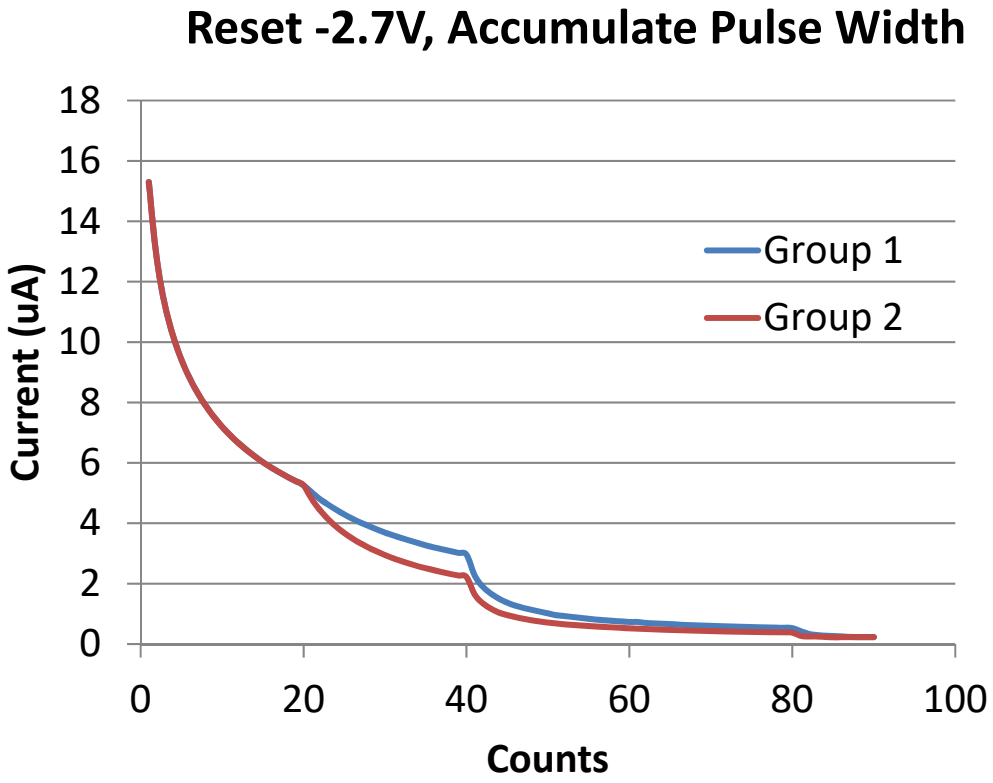


Good at speed, boundary
Difficult to design

Multi-level Writing Simulation

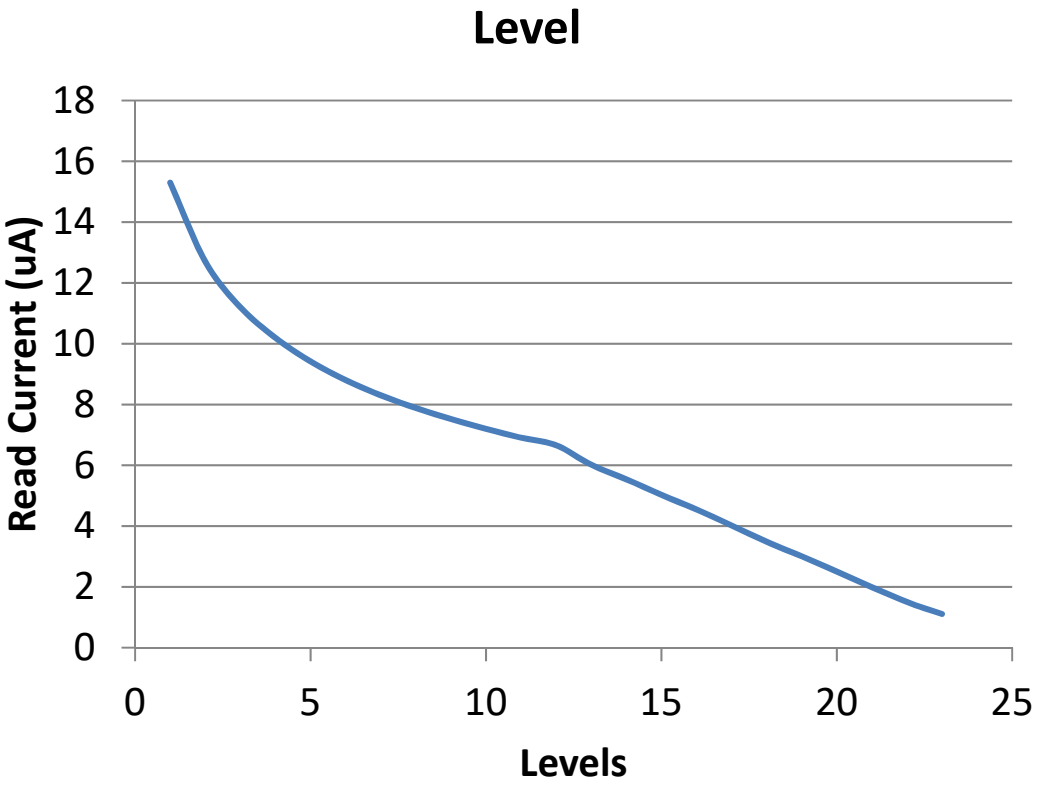
- **Change the Reset pulse width (times)**

Group 1		Group 2	
Pulse Width (ns)	Counts	Pulse Width (ns)	Counts
1	20	1	20
2	20	4	20
16	40	16	40
32	10	32	10



Multi-level Writing Simulation

- **Multi-level Writing Curve & LUT**

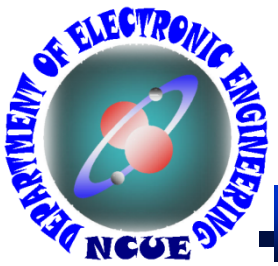


Counts	read I(uA)	Level	R (Kohm)
0		0	3.04
1	15.3	1	29.21569
2	12.7	2	37.24409
3	11.2	3	43.57143
4	10.2	4	48.82353
5	9.41	5	53.76196
6	8.81	6	58.10443
7	8.31	7	62.20217
8	7.88	8	66.14213
9	7.52	9	69.78723
10	7.2	10	73.33333
11	6.91	11	76.83068
12	6.66	12	80.09009
15	6.02	13	89.66777
18	5.54	14	98.30325
22	5.03	15	109.2843
27	4.54	16	122.1586
34	4.03	17	138.8834
45	3.48	18	162.4138
60	3.01	19	189.3355
83	2.5	20	230
123	2	21	290
202	1.5	22	390
350	1.11	23	530.5405

Conclusions & Future Works

- Innovation and contribution
  - Propose a kind of multi-level writing method
  - Find out the suitable condition for RRAM model
- What can be done **next**:
  - Simulate and modify our test method
  - Insert aging and defect models into the RRAM
  - Design multi-level writing circuit for RRAM

  (Each work can simulate on low level simulator)



Thank you for your attention