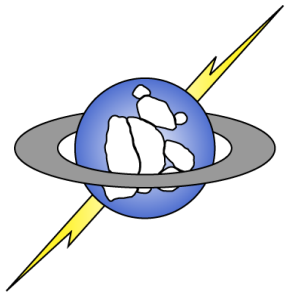
Extracting Unique Fingerprints From Flash Memory Devices

Pravin Prabhu[1], Ameen Akel[1], Laura M. Grupp[1], Wing-Kei S. Yu[2], G. Edward Suh[2], Edwin Kan[2], and Steven Swanson[1]

[1]Non-volatile Systems Laboratory, UCSD CSE
[2]Cornell University



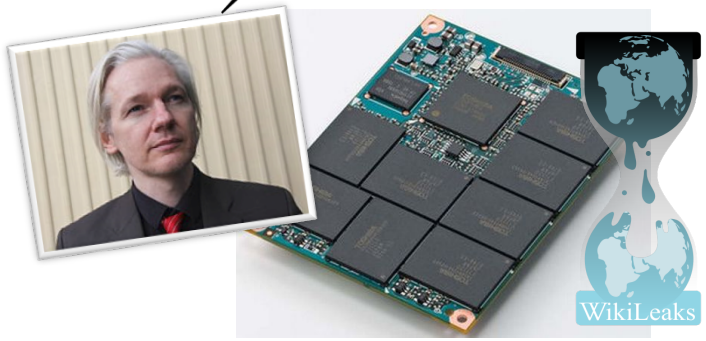
NVSL
Non-volatile Systems Laboratory

UCSD

Cornell University

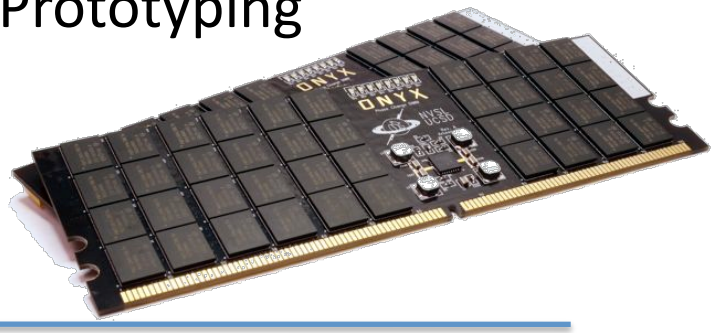
NVSL

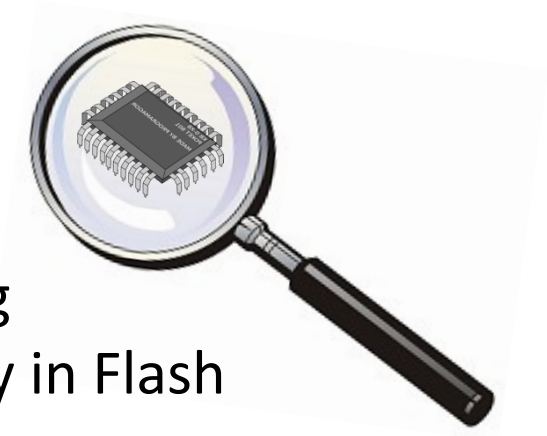
Non-volatile Systems Laboratory



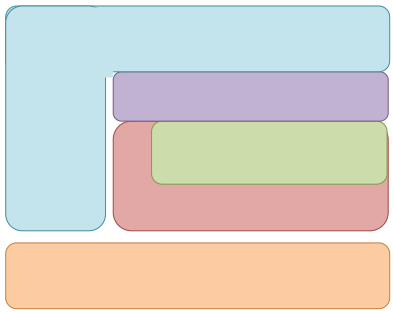
Trust and Security

Hardware/Software Prototyping





Exploiting Variability in Flash



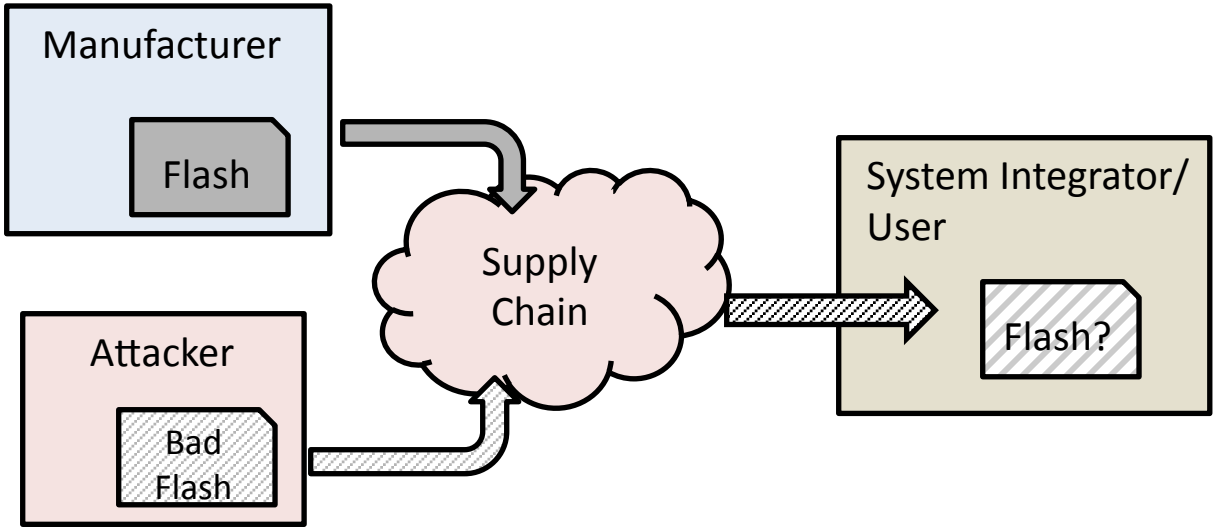
Programming interfaces

The Flash Juggernaut

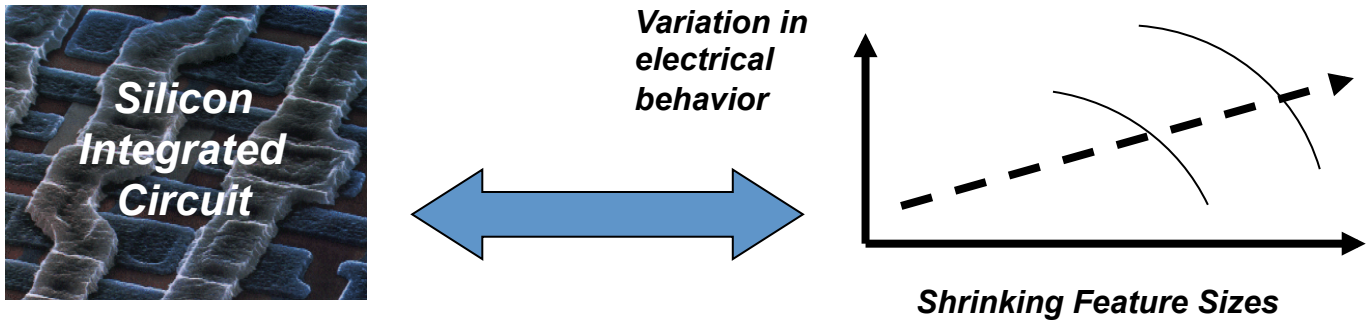
NVSL
Non-volatile Systems Laboratory

Flash Device Authentication

- Can we authenticate each flash chip?
  - Distinguish genuine flash chips from counterfeits
  - Authenticate a device with a flash chip



Physical Unclonable Functions (PUFs)

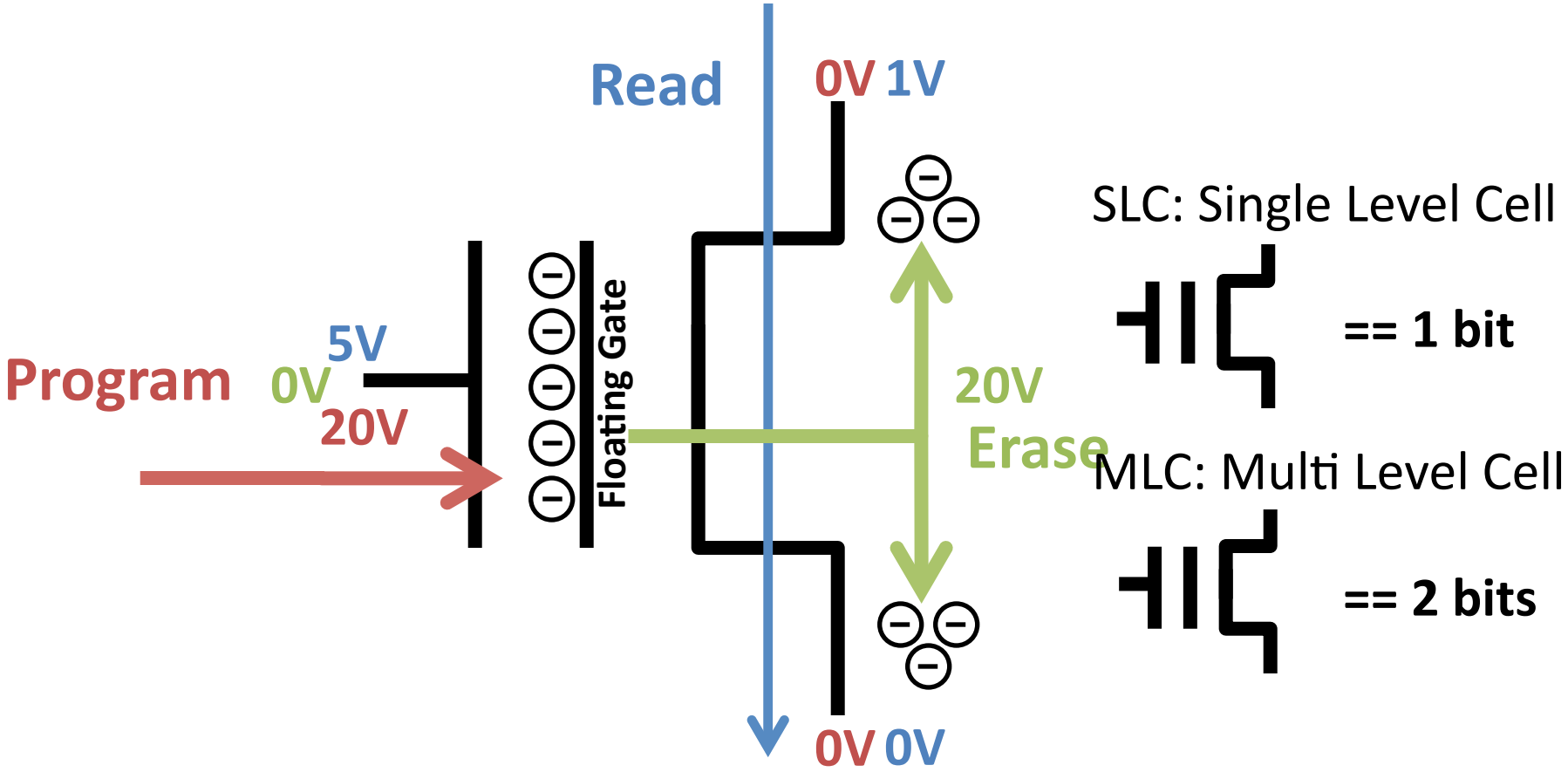


- Because of random manufacturing variations, no two Integrated Circuits are identical - even those using same mask
  - Hard to remove or predict in advance
  - Relative variation increases as feature sizes shrink
  - Variation persists, despite $ billions spent to control it
- We can generate fingerprints from unique analog characteristics of each IC: Response = PUF(Challenge)
  - Inexpensive; intrinsic to each device; effectively unclonable
- This work introduces a PUF based on flash chips

Outline

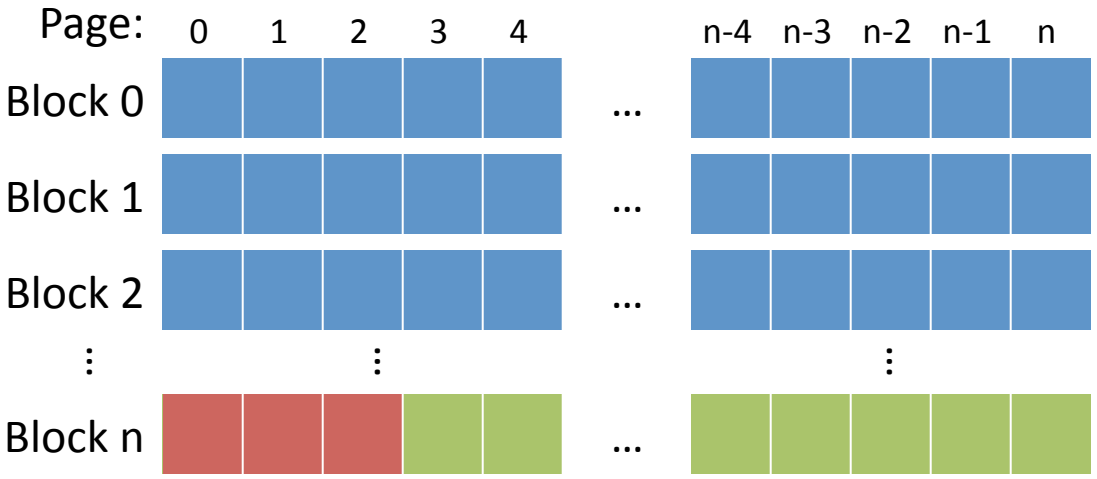
- Flash memory overview
- Experimental infrastructure
- Flash-based Physically Unclonable Functions (FPUFs)
  - Usage Model
  - Desiderata
  - Our FPUFs
- Conclusions

Flash Operations



NVSL
Non-volatile Systems Laboratory

NAND Flash Basics



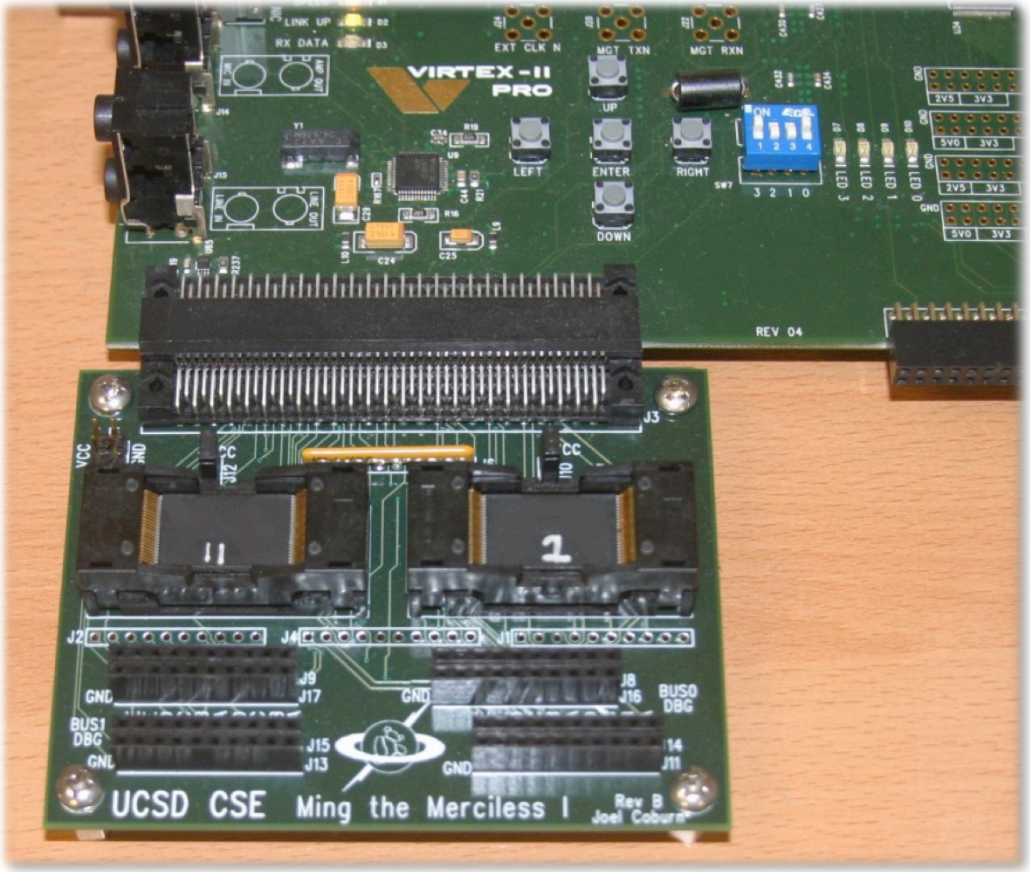
NVSL
Non-volatile Systems Laboratory

Flash Failure Mechanisms

- Program/Erase (PE) Wear
  - Permanent damaged to the gate oxide at each flash cell
  - After 3000 (MLC) – 100,000 (SLC) PE cycles, a cell becomes unreliable
- Program disturb
  - Data corruption caused by interference from programming adjacent cells.
  - No permanent damage

Experimental Setup

- Custom-Built Daughter Board
- Xilinx XUP Board
- EZ to integrate similar capabilities into existing systems



NVSL
Non-volatile Systems Laboratory

The Test Subjects

Chip Name	Node (nm)	Bytes Page	Pages Block	Planes Die	Dies
A-SLC2		2048	64	2	1
A-SLC4		2048	64	1	1
A-SLC8		2048	64	2	1
B-SLC2	50	2048	64	1	1
B-SLC4	72	2048	64	2	1
E-SLC8		2048	64	1	2

Chip Name	Node (nm)	Bytes Page	Pages Block	Planes Die	Dies
A-MLC16		4096	128	2	1
B-MLC8	72	2048	128	1	1
B-MLC32	50	4096	128	2	2
B-MLC32-2	34	4096	256	2	1
B-MLC128	34	4096	128	2	4
B-MLC128-2	34	4096	256	2	4
C-MLC64	43	8192	128	1	2
D-MLC32		4096	128	1	2
E-MLC8		4096	128	1	2
F-MLC16	41	4096	128	2	1

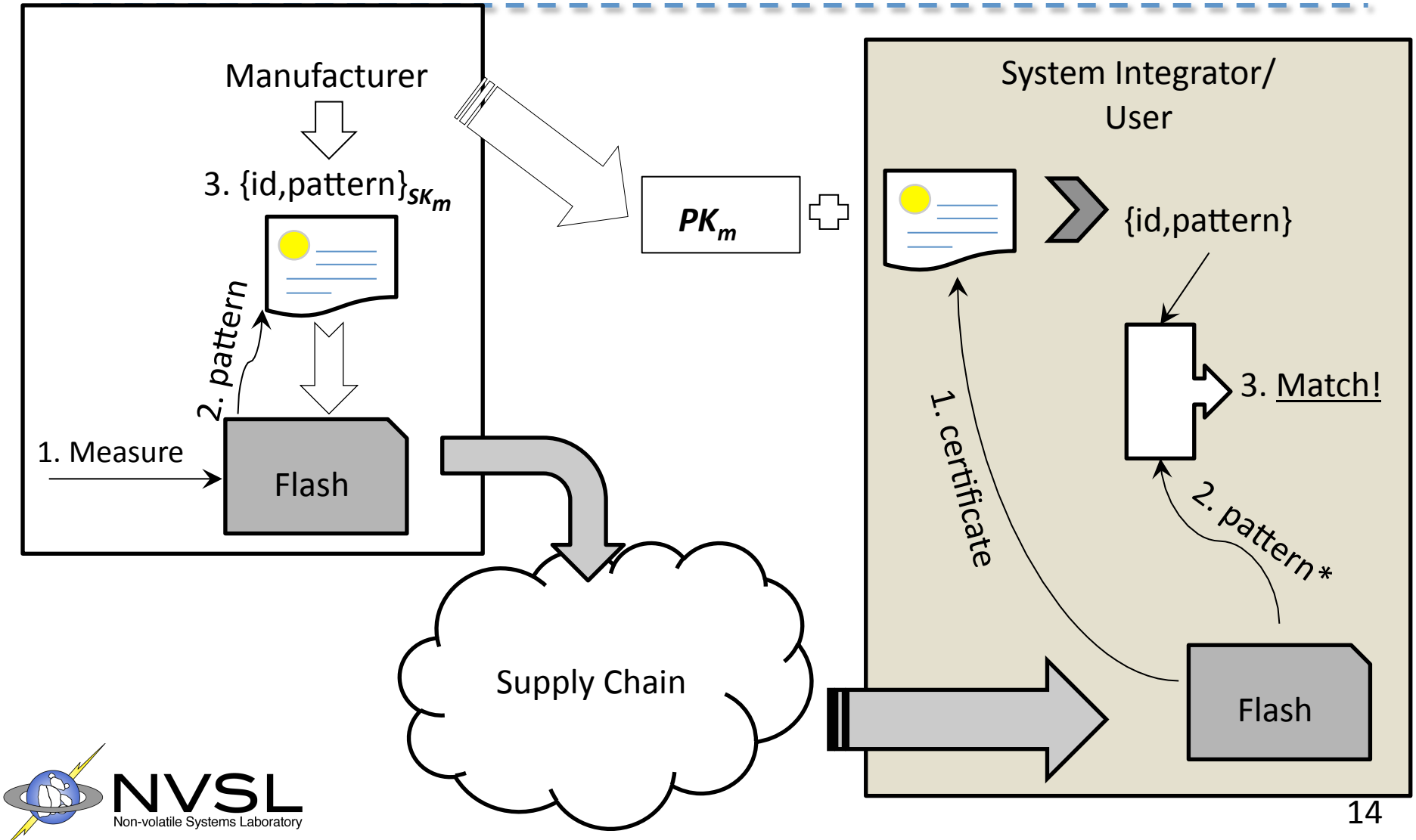
Outline

- Flash memory overview
- Experimental infrastructure
- Flash-based Physically Unclonable Functions (FPUFs)
  - Usage Model
  - Desiderata
  - Our FPUFs
- Conclusions

Flash-based Signatures

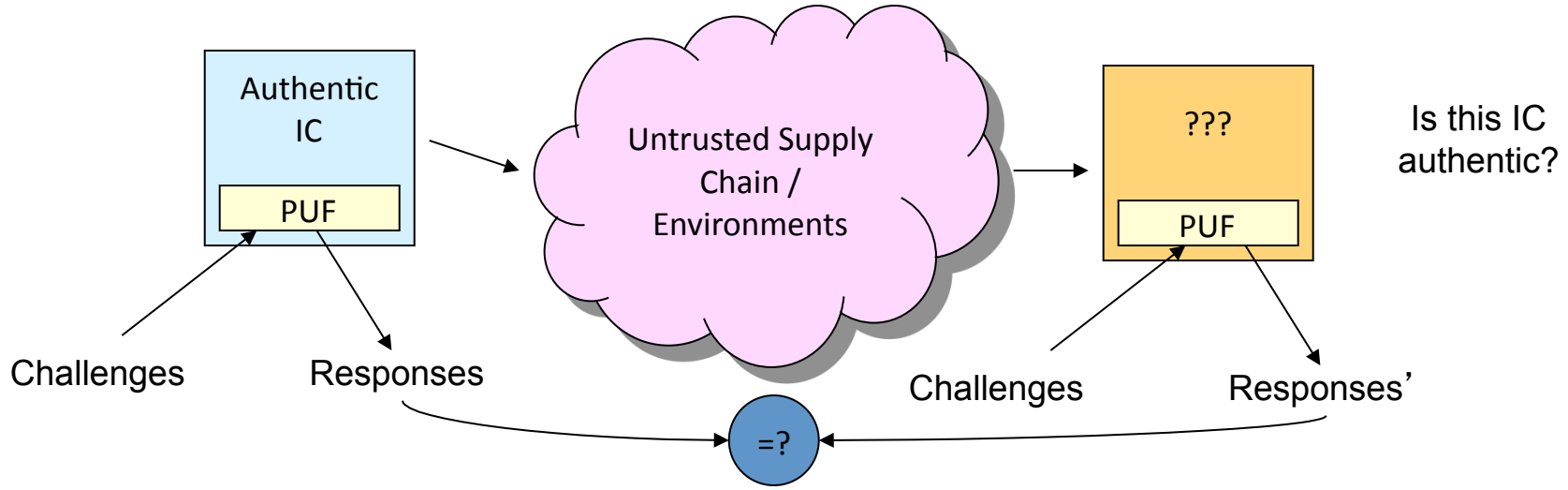
- Cell-level variation in flash devices makes each chip unique.
- Unique, unforgeable flash chip signatures have several uses
  - Device identification
  - Supply chain integrity

Authentication Model



NVSL
Non-volatile Systems Laboratory

Challenge-Response Based Authentication



- Create CRPs for IC with PUF when IC is in your possession
- Use CRPs to subsequently authenticate IC throughout the supply-chain and post-deployment
  - Use each CRP only once → prevent "replay"

Signature Characteristics

- Selectivity – an FPUF should be able to reliably distinguish between flash devices
- Speed – Computing an FPUF should be fast
- "Unforgeable" – It should be prohibitively difficult to forge the FPUF
- Non-Destructive – Extracting an FPUF should not wear out the flash device.

Basic Recipe for an FPUF

1. Identify an aspect flash chip behavior that varies based on manufacturing inconsistencies
2. Measure the variation at a bit, page, or block level
3. Use the sequence of measured values as a signature
4. Use statistical correlation to determine whether two signatures are for the same device.

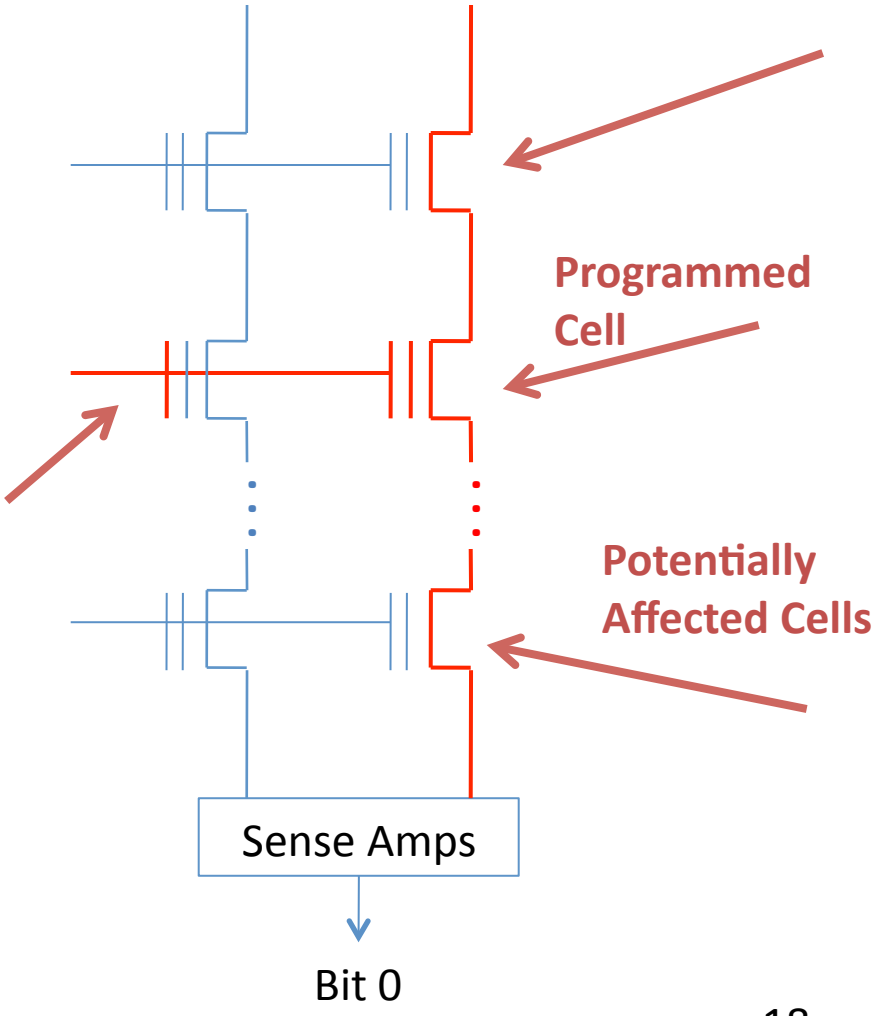
Chip #1	Chip #2
4	3
3	8
8	3
…	…

Signatures Correlated?

NVSL
Non-volatile Systems Laboratory

Program Disturb FPUF

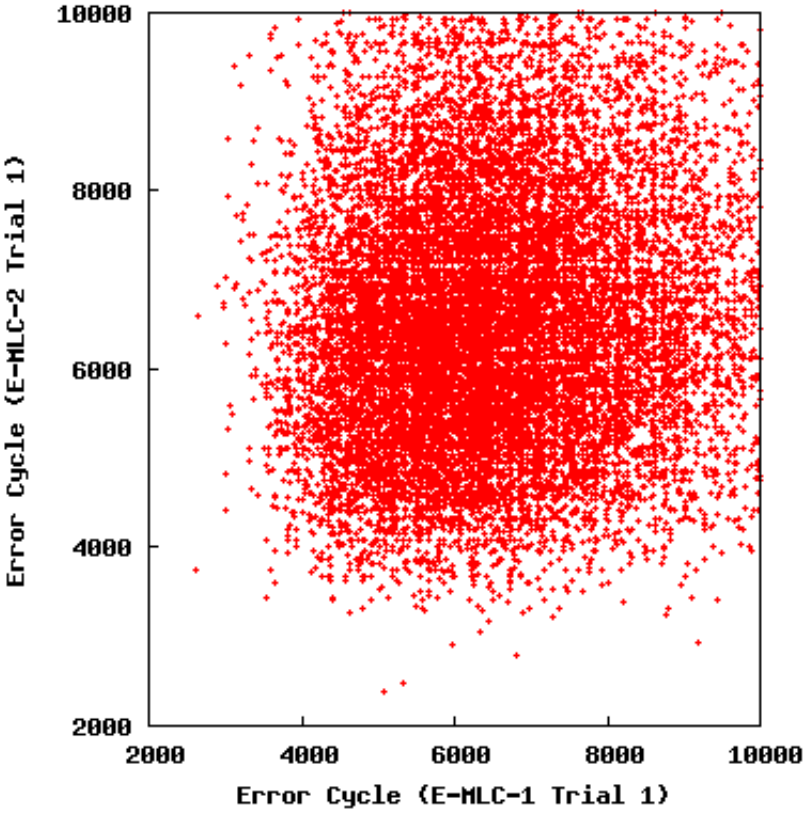
- Program one page repeatedly
- For each bit in the adjacent page
  - How many programs before the bit flips?
- The sequence of counts is a signature.



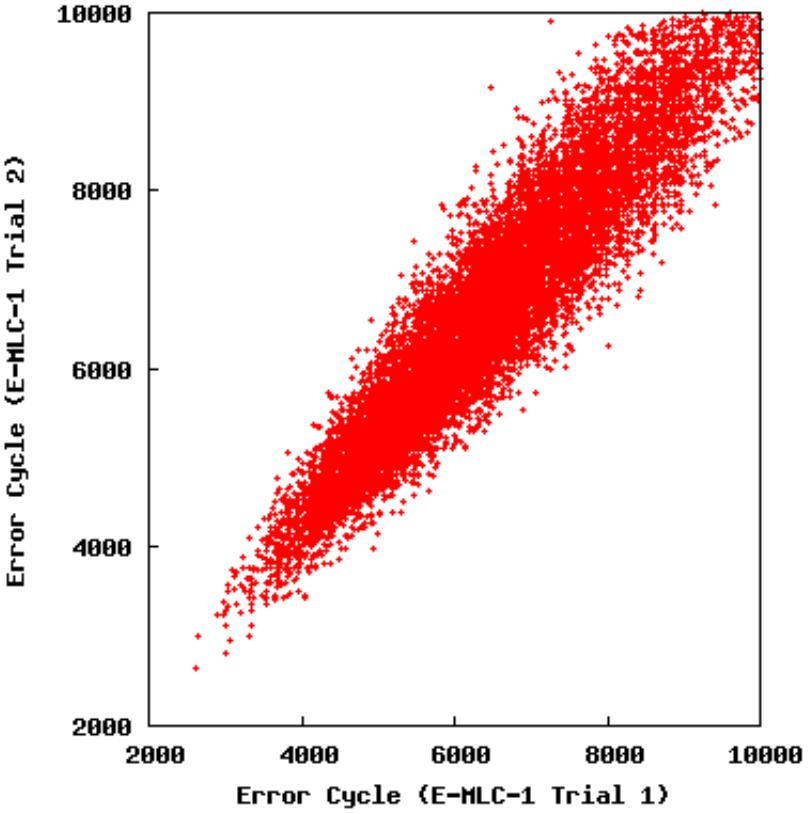
NVSL
Non-volatile Systems Laboratory

Signature Selectivity

Signatures from Different Pages

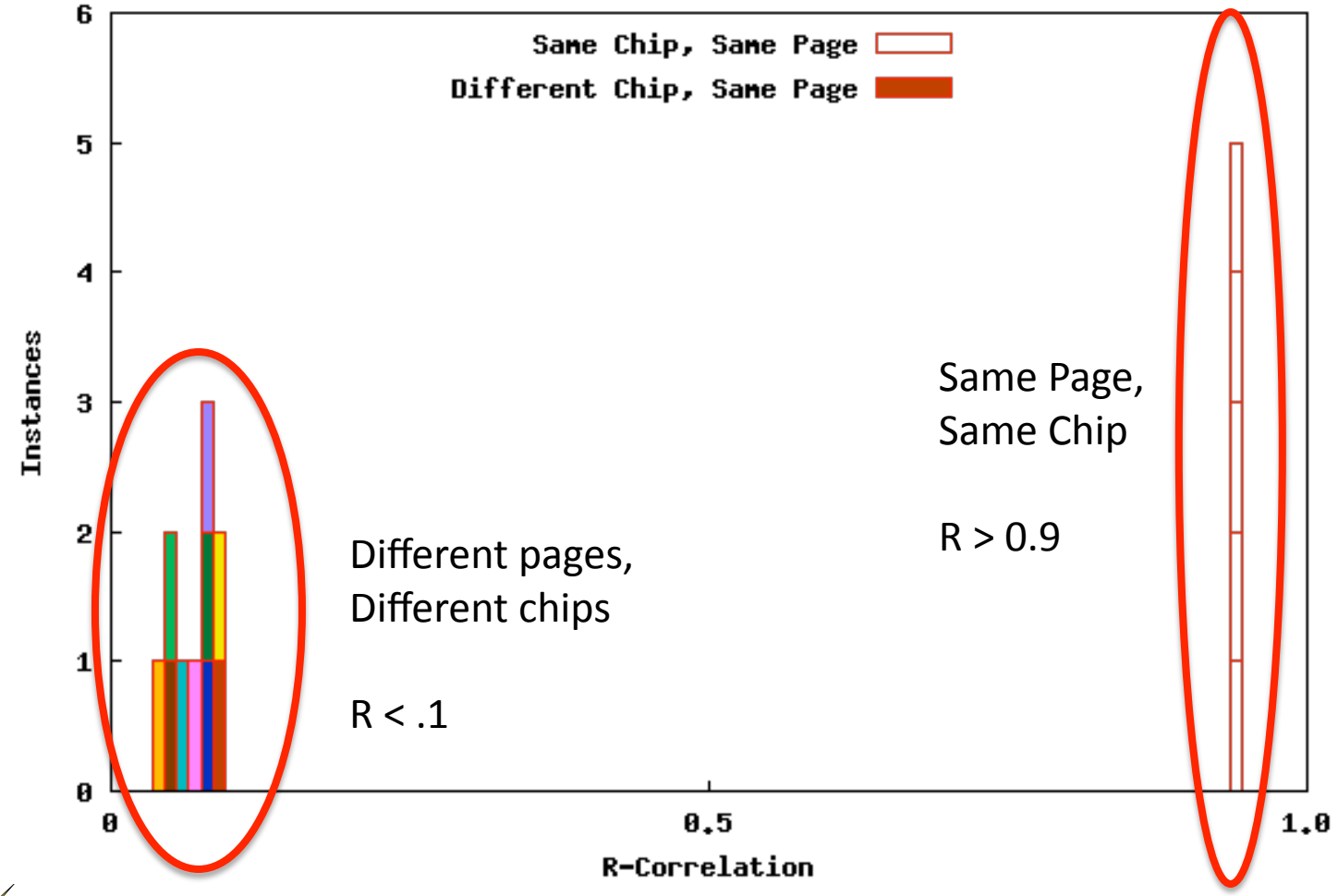


Signatures from the Same Page



NVSL
Non-volatile Systems Laboratory

Selectivity for Program Disturb

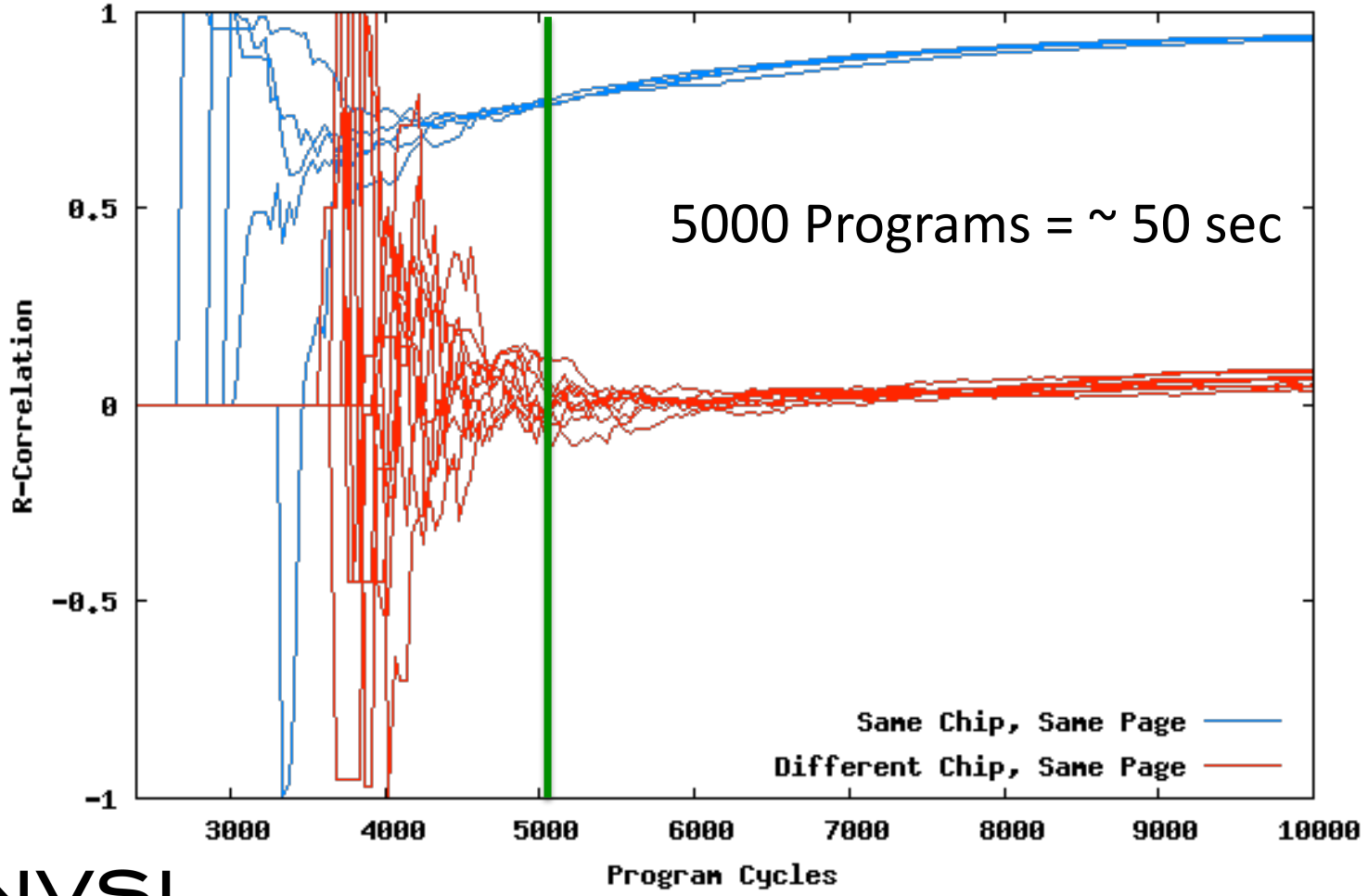


NVSL
Non-volatile Systems Laboratory

Program Disturb Latency

- Extract an program-disturb signature takes up to 5 minutes
  - Some usage models require many signatures from each chip
  - 5 minutes is prohibitively slow in these cases.
- Can we extract a useful signature with fewer program operations?

Reducing Programs/Signature



NVSL
Non-volatile Systems Laboratory

Forging Program Disturb FPUFs

- Forging an FPUF would require storing the signature in the flash device
- If the signature contains more than one bit of information per flash cell, storing the signature in the chip is not possible.
- However, our signatures are noisy, so precise forgery is not required.
  - It is possible to lossily compress signatures

Compressing the Signatures

- Raw signatures need 10 bits of program count information per flash cell
- We can quantizing program counts in to 4 values (i.e., the top two bits)
  - Quantized signatures correlate well (R = 0.8) with raw signatures
- The quantized signatures are not very compressible (entropy/bit is near 1)
  - It is still impossible to store the signature for every page in a flash chip

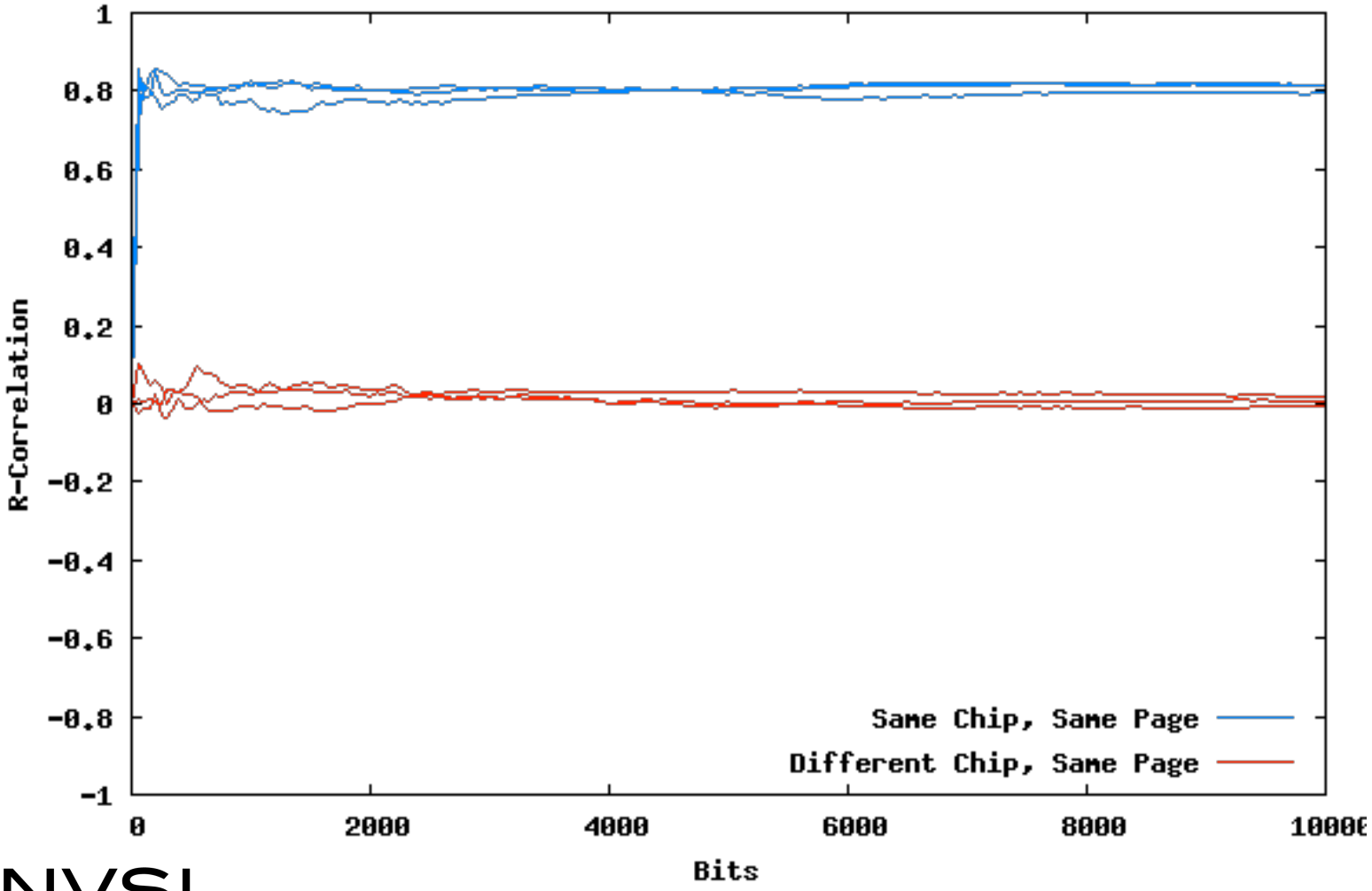
Program Disturb FPUF

- Selectivity: Very Good
- Speed: 1-5 Minutes per page
- Wear: 10,000 programs of the target page
- Forgeability: low

Per-bit Program Latency FPUF

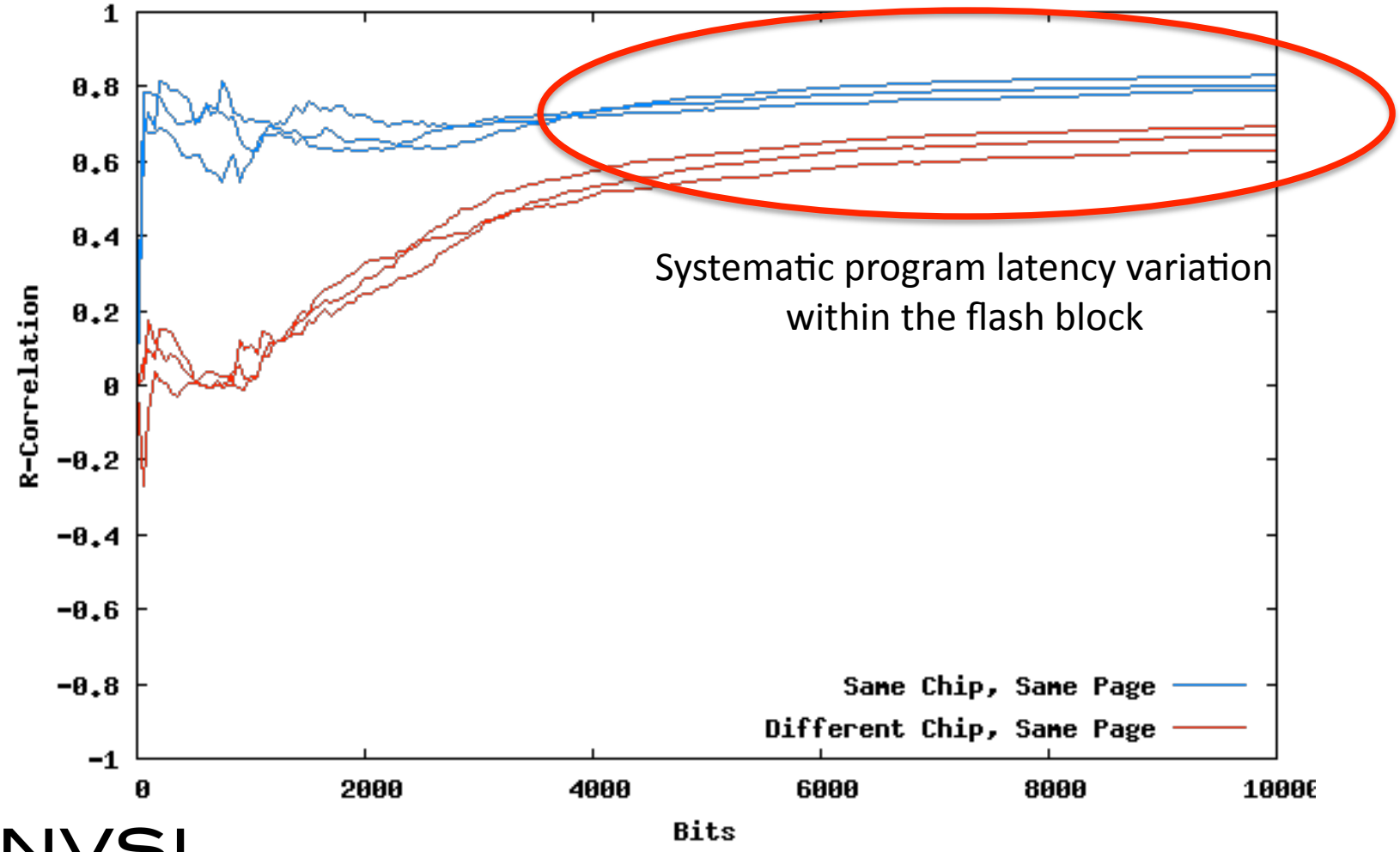
- Individual cells have different programming characteristics
- The chips only program bits that change
- We can measure per-bit program latency by programming one bit at a time.
  - Program bit 0 in page 0, bit 1 in page 1, etc.
  - The sequence of program latencies is the FPUF result

Program Latency FPUF Correlations (SLC)



NVSL
Non-volatile Systems Laboratory

Program Latency FPUF Correlations (MLC)



NVSL
Non-volatile Systems Laboratory

Per-bit Program Latency FPUF

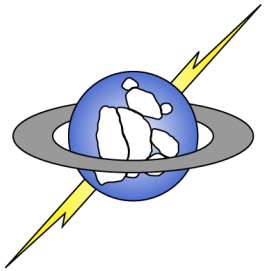
- Selectivity: Good
- Speed: 1-20s
- Wear: 1 PE cycle
- Forgeability: High

Other FPUFs

- Usable FPUFs
  - Per-bit program latency
  - Read disturb
- Unusable FPUFs
  - Per-block erase latency
  - Per-page read latency
  - Full page program latency (rather than bit-by-bit)

Conclusions

- FPUFs can provide a robust mechanism for identifying individual flash devices.
- Flash's ubiquity makes them an attractive method for device identification
  - Inexpensive
  - Easy to implement
- FPUFs will become even more useful as flash manufacturing variation grows



NVSL
Non-volatile Systems Laboratory

UCSD

Cornell University

Questions?