Concurrent Fault Detection for Secure QDI Asynchronous Circuits

Konrad J. Kulikowski, Mark G. Karpovsky, Alexander Taubin, Zhen Wang, Adrian Kulikowski

Boston University

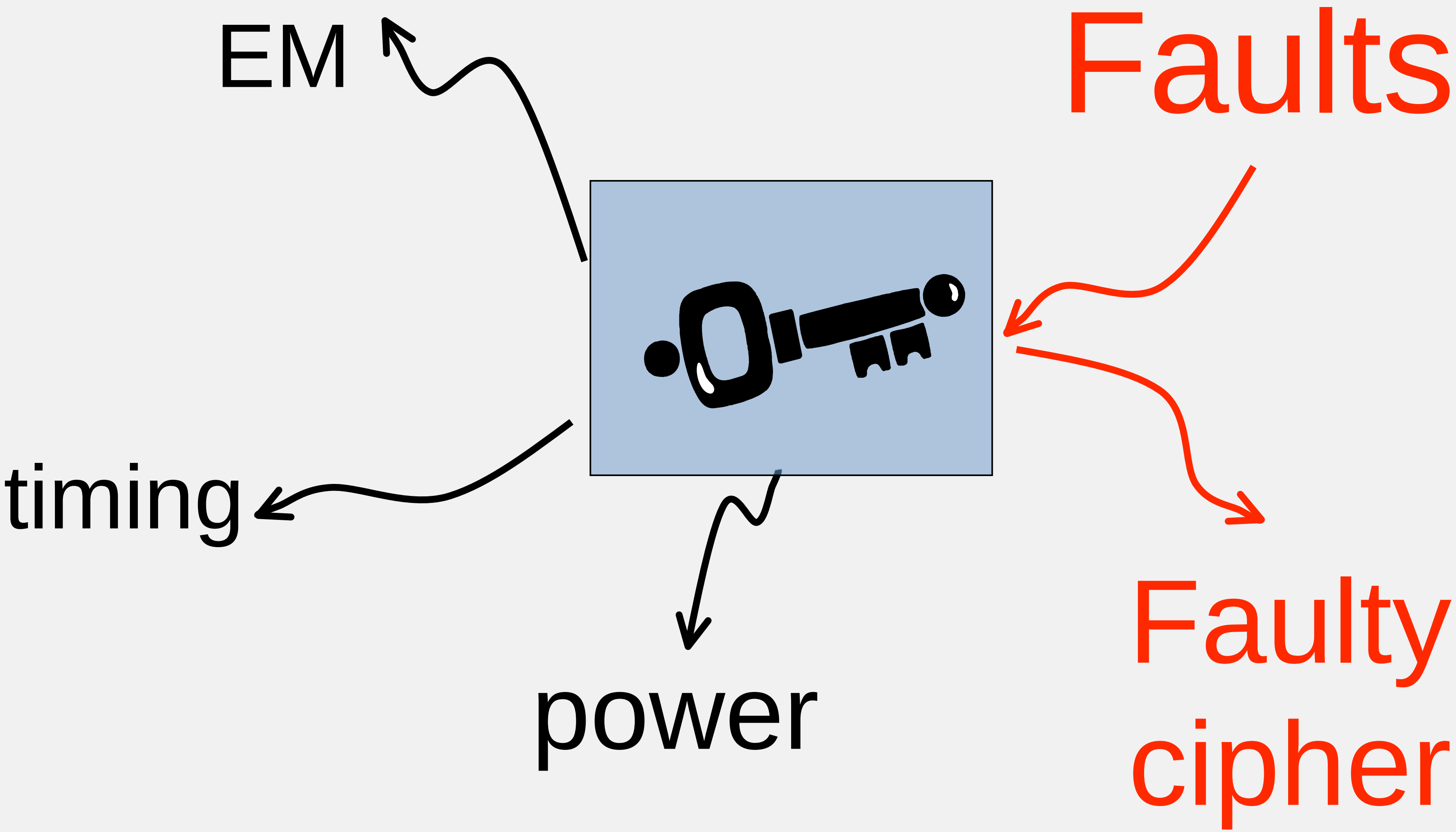
Reliable Computing Laboratory

6/27/2008

Outline

- Side Channel Attacks
- Asynchronous nanocircuits for security
- Faults in asynchronous fine grained pipelines
- Robust Codes
  - Basic properties and design purpose
  - Minimum distance robust codes
- Application to AES
- Fault Simulation

Side Channel Attacks



Nanocircuits and Async in Security

Nanocircuits

•Lower signal to noise ratio
•Harder to probe or reverse engineer
•Higher variability allows design of novel features like physically unclonable functions (PUF)
•Higher fault rates
•Higher variability

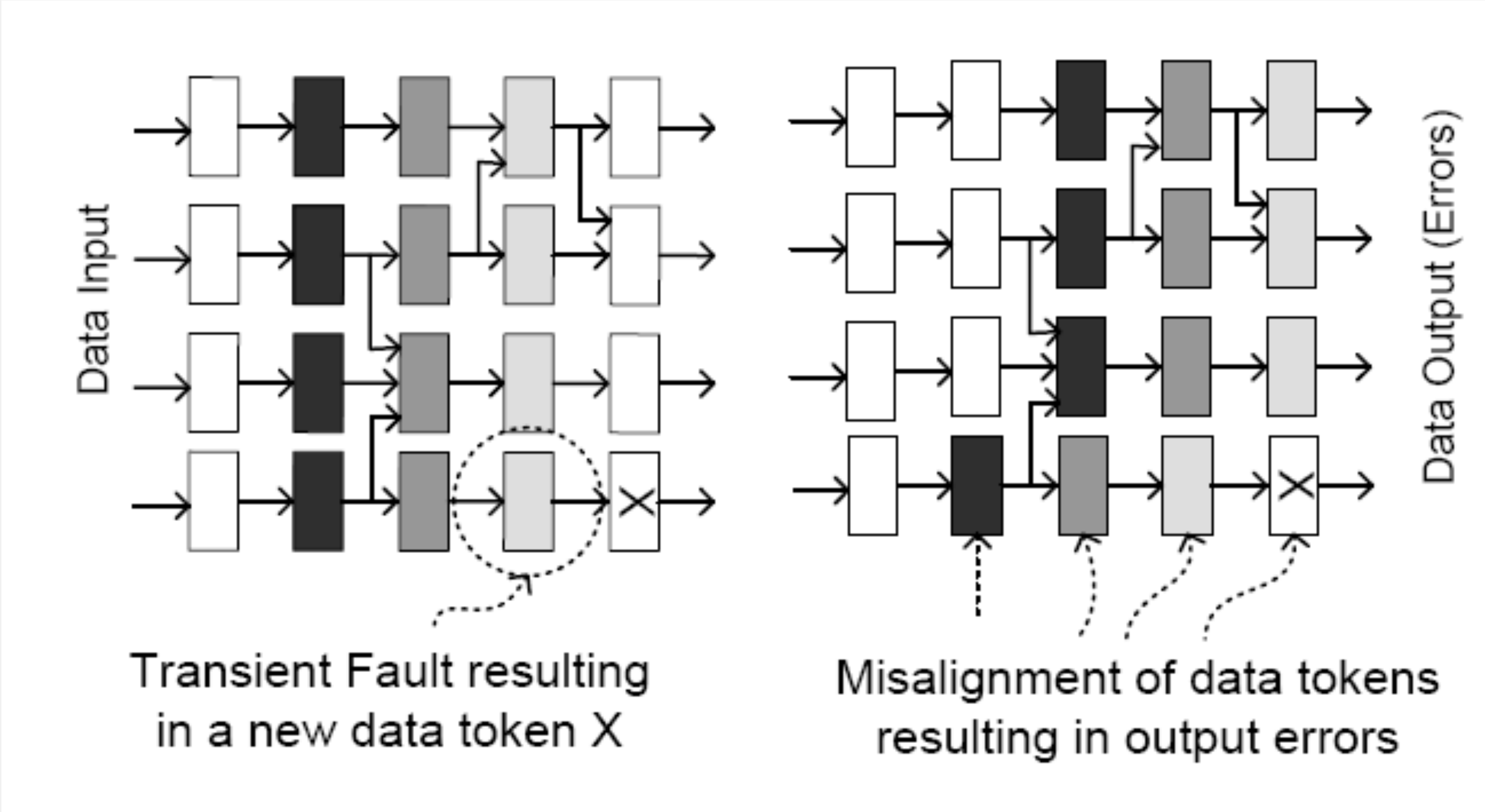
Asynchronous QDI

•Clockless designs have been shown to have natural benefits against power and EMI attacks
•**Tolerant to variability**
•**Natural fault tolerance**

Faults in Asynchronous QDI Design

1. Deadlock
2. Invalid data token (‘11’)
3. Data modification (flipping a value of a data token)
4. Data generation (creation of a data token)
5. Data deletion (deletion of a data token)

Data Insertion / Deletion



Data Creation / Deletion

Main Characteristics

- A single transient fault can create a stream of erroneous data
- Error at output can repeat indefinitely

Solution Criteria

- Detect token insertions, not just prevent the effect
    - Detection allows reaction/prevention to an attack
    - Concurrent error detection using error control codes
- Detect all possible token insertions
- Reduce the worst detection probability

Can we exploit the repeating nature of errors to improve error detection?

Robust Error Detecting Codes

- Nonlinear
- ALL errors are detectable with a high probability
- Provide a guaranteed level of protection for all errors

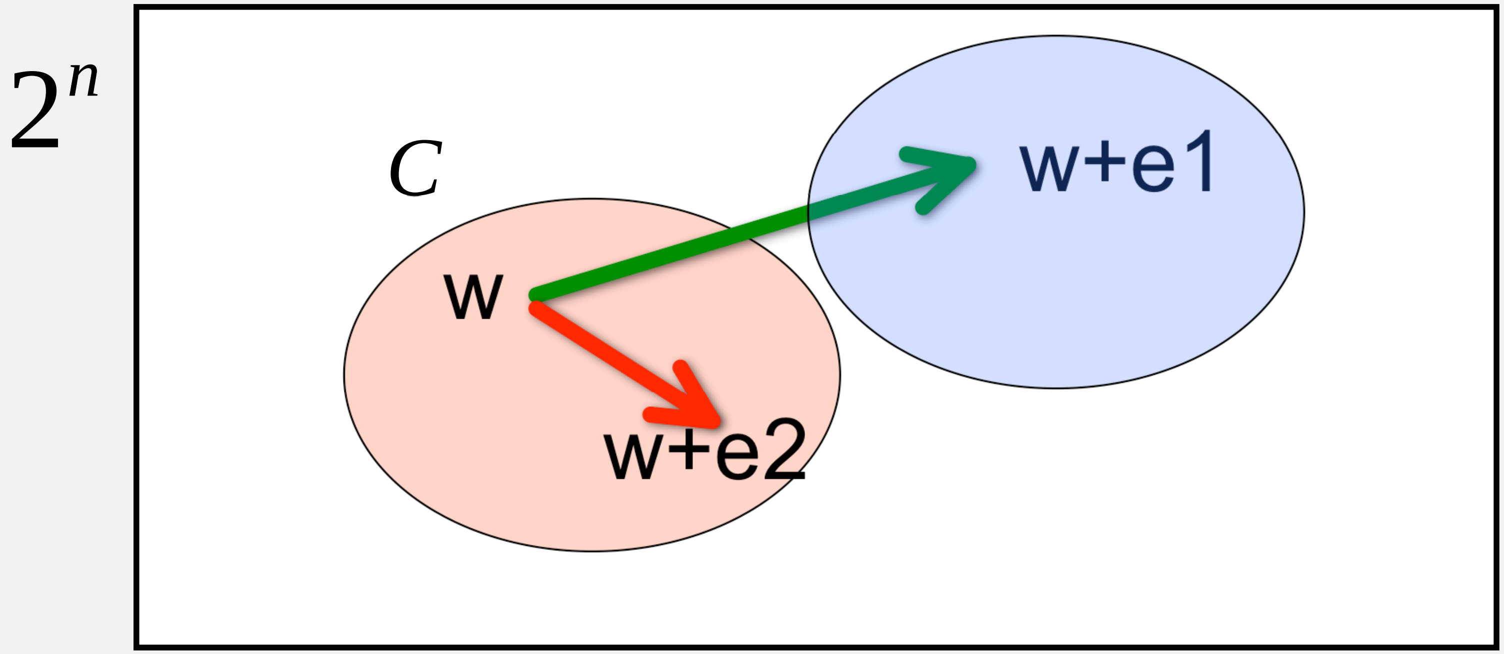
$$Q(e) = \frac{|\{w|w \in C, w+e \in C\}|}{|C|}$$

Definition 3.2 *A robust code C where*

$$R = \max_{e \neq 0, \in GF(q^n)} Q(e)|C|$$

*is called **R-robust**.*

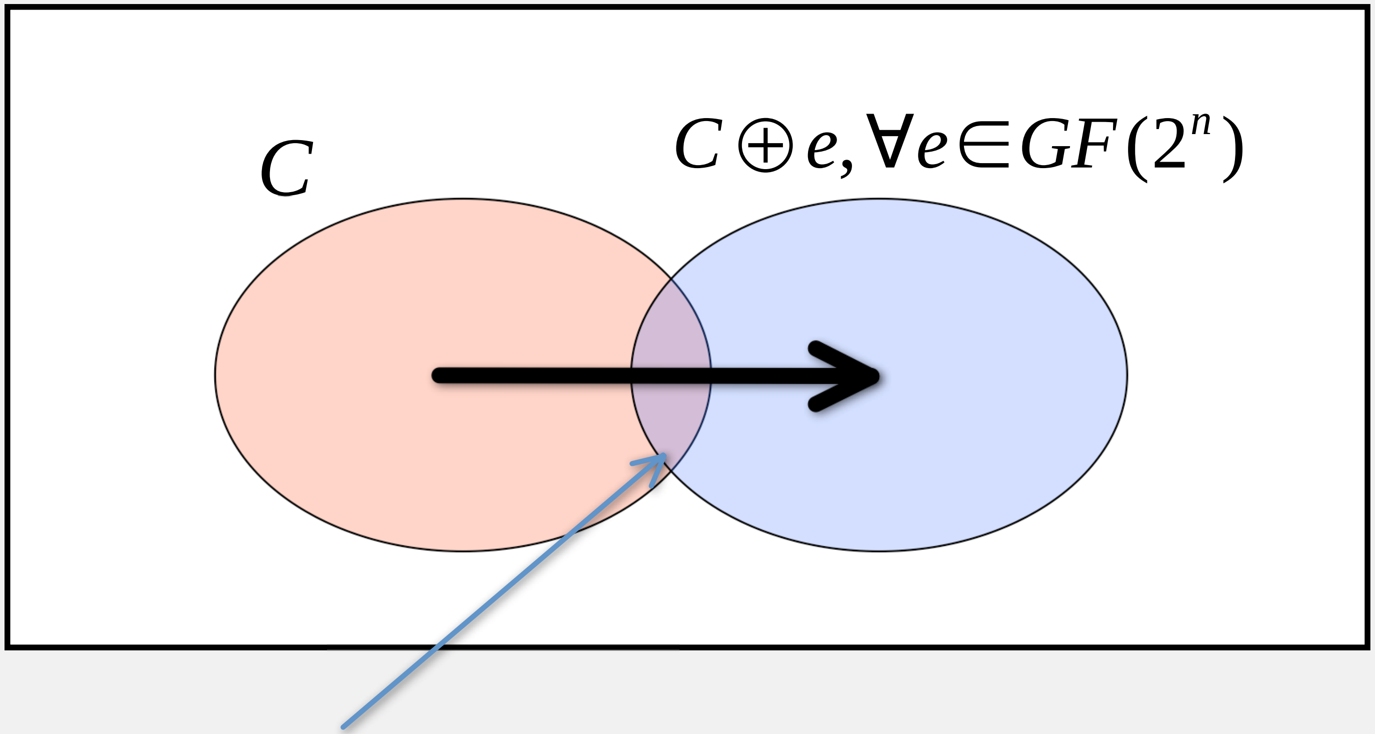
Error Detecting Codes



- Linear codes have |C| errors which are undetectable
- Repeating errors do not improve error detection

Robust Error Detecting Codes

2^n



$$R = \max |C \cap (C \oplus e)| < |C|$$

Every error is missed for at most R messages (max Q(e)=R/|C|)
Detection probability increases as more erroneous messages are observed

Systematic Robust Codes

$$C_1 = \{(x, f(x)) | x \in GF(2^k)\}$$

f(x) “highly nonlinear function”
optimum when *f(x)* is a “perfect nonlinear function”

$$f(x = (x_0, x_1, ...x_k)) = x_0x_1 + x_2x_3 + ... + x_{k-1}x_k$$

(k+1,k,1) code with $R=2^{k-1}$

Minimum Distance Robust Codes

$$C_2 = \{(x, p(x), f(x)) | x \in GF(2^k)\}$$

{(x,p(x)) } is a linear code with distance *d*

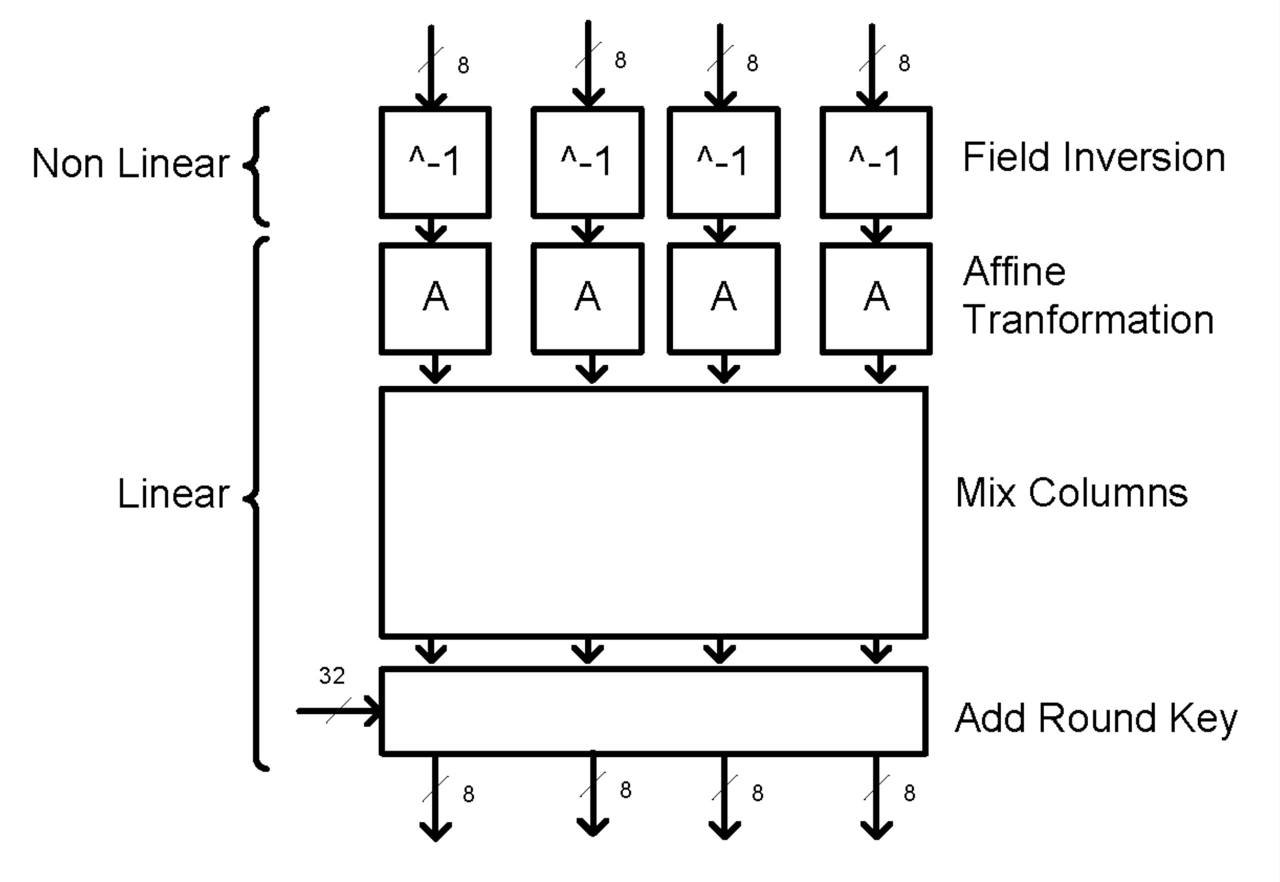
f(x) is a perfect nonlinear function

p(x) parity

$$f(x = (x_0, x_1, ...x_k)) = x_0x_1 + x_2x_3 + ... + x_{k-1}x_k$$

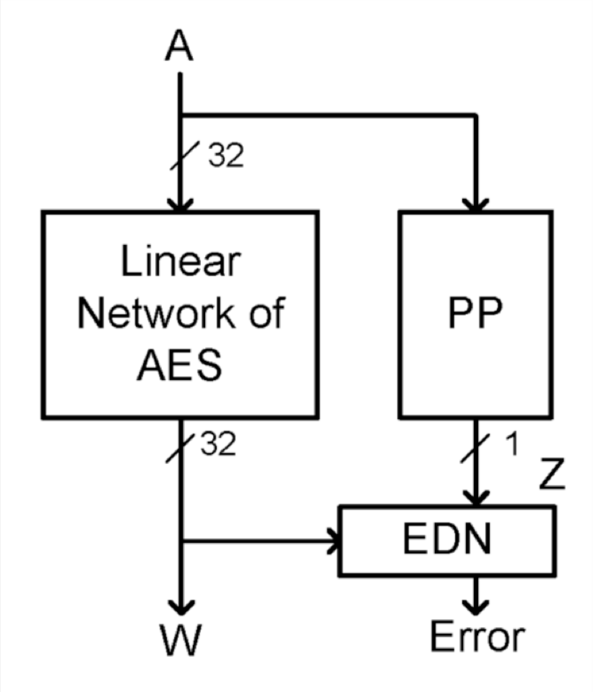
(k+2,k,2) code with $R=2^{k-1}$

Application to Asynchronous AES



M. Karpovsky, K. J. Kulikowski, and A. Taubin. “Differential Fault Analysis Attack Resistant Architectures for the Advanced Encryption Standard”. In CARDIS, 2004.

Concurrent Error Detection



Linear parity: 35%

(x,p(x))

Robust: 100%

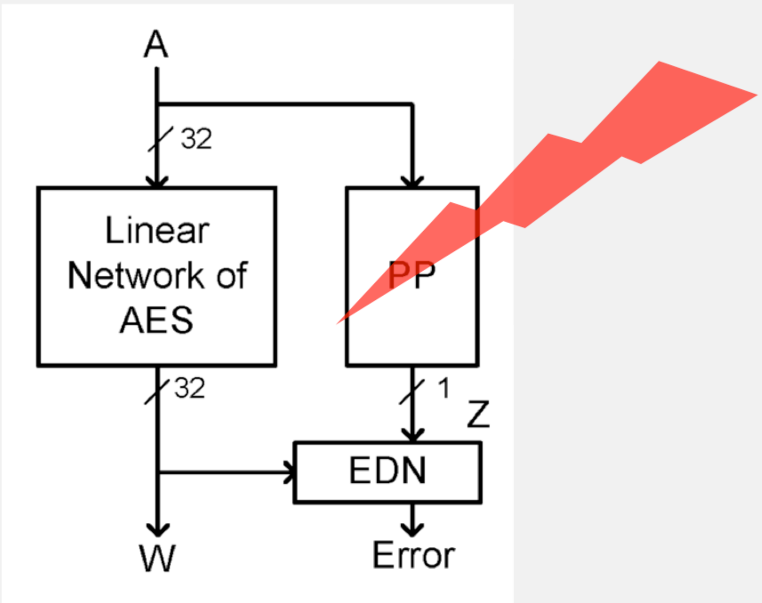
(x,f(x))

Robust and parity: 120%

(x,p(x),f(x))

Evaluation

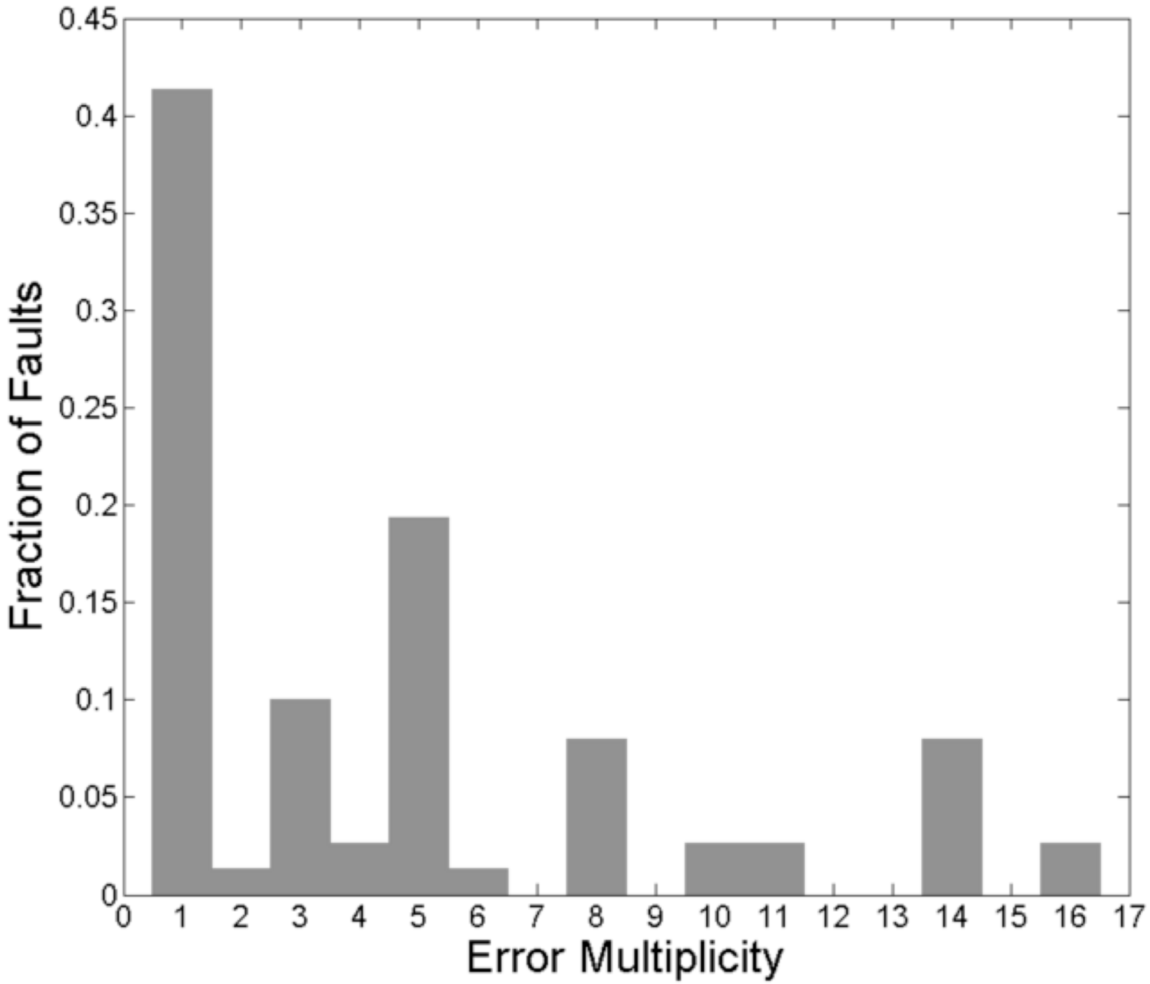
Random Inputs



Faults causing single token creations/deletions

How long does it take to detect the erroneous behavior?

Histogram of Manifestations



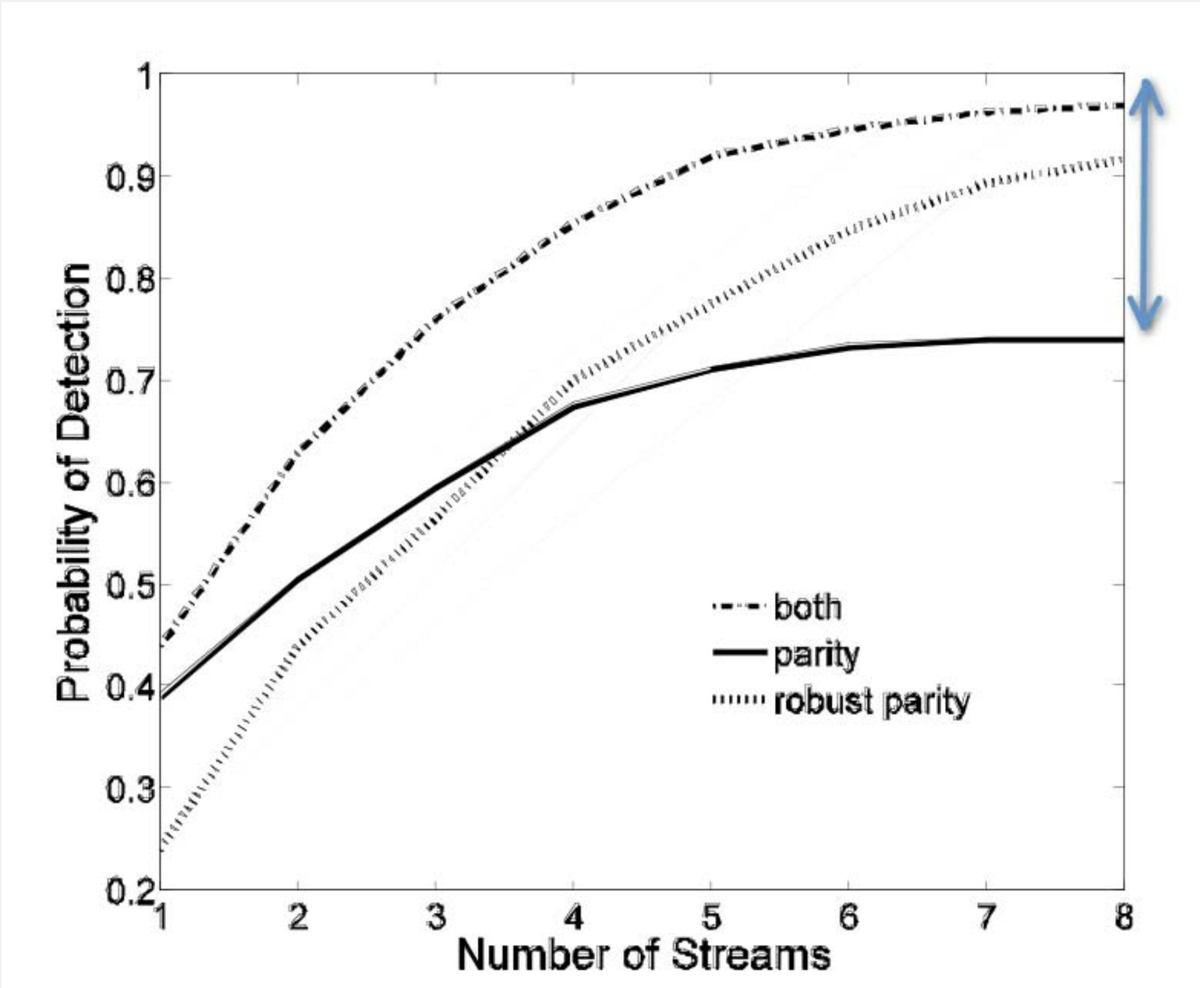
Synthesized using Desing Compiler

216 two input XOR gates

Multiplicity of Errors resulting from single faults

- 27% of errors are even
- Many Errors are of a high multiplicity

Simulation Results



27% of token creations/deletions missed

Summary

- Token creation/deletion can lead to a long stream of erroneous data
- Repeating nature of the errors can be used to enhance the error detection
- Beneficial for security
- Detect other failures (data modification)
- Adds another level of security