

**Non-Concurrent Error Detection and Correction in  
Fault-Tolerant Linear Finite-State Machines**

**Christoforos Hadjicostis**

Coordinated Science Laboratory and  
Department of Electrical and Computer Engineering

University of Illinois at Urbana-Champaign

## MOTIVATION FOR FAULT TOLERANCE

### **Fault tolerance describes ability to:**

- Withstand internal faults
- Produce desirable overall “behavior” (e.g., correct or acceptable output)

### **Necessary or desirable in:**

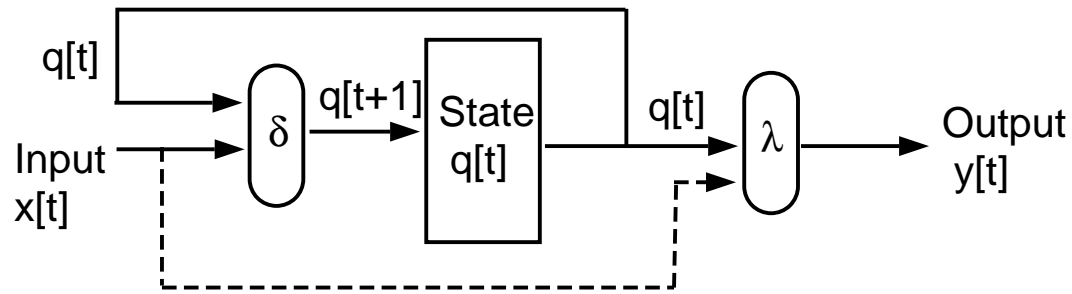
- Life-threatening circumstances (military, transportation, medical)
- Systems in inaccessible environments (space missions)
- Reliable systems from unreliable components, noise tolerance  
(faster, less expensive, less power)

**Previous work includes:** Communication systems, computational circuits, special-purpose systems, networked systems

## RELATED RESEARCH ON FAULT TOLERANCE

- Communication systems (channel noise, error-correcting codes)
- Computational circuits (hardware faults, modular redundancy)
  - Each component fails with *constant* probability
  - Earlier work by von Neumann, Shannon, Winograd, Elias and others; later work by Pippenger, Gács, Feder, Hajek, Reischuk, Hadjicostis, ...
- Special-purpose computational systems (Algorithm Based Fault Tolerance)
  - Protect against *fixed* number of faults
  - Recent work by Abraham, Chatterjee, Redinbo, Beckmann, Hadjicostis, ...
- Networked discrete-event systems (link or node failures, reconfiguration / rerouting / multiple paths)

## FAULT-TOLERANT DISCRETE-TIME DYNAMIC SYSTEMS



State Evolution:  $q[t + 1] = \delta(q[t], x[t])$

Output:  $y[t] = \lambda(q[t], x[t])$

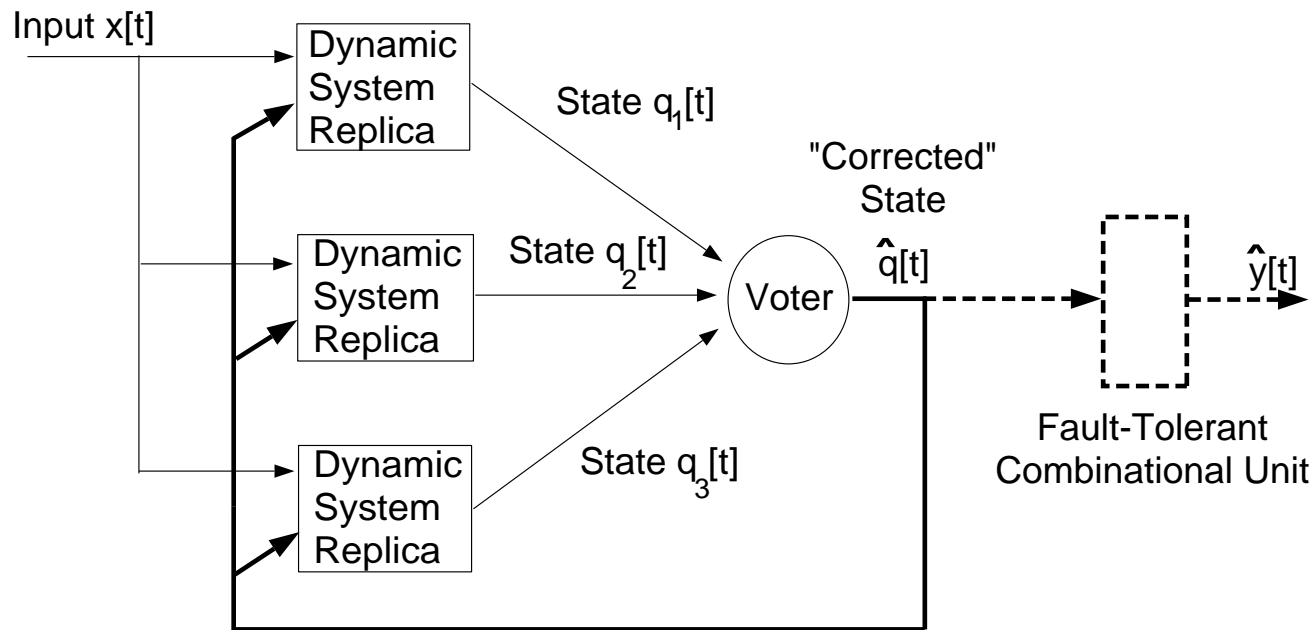
**Examples:** Digital filters, encoders/decoders, computer simulations

**Faults:** Affect state transition mechanism and/or output mechanism

**Research goals:**

- Resource-efficient monitoring/testing of dynamic systems/networks
- Tradeoffs between detection delay, redundant hardware and monitor complexity
- Fundamental limitations (coding- and information-theoretic techniques)

## UNIVERSAL APPROACH: MODULAR REDUNDANCY



### Problems with modular redundancy:

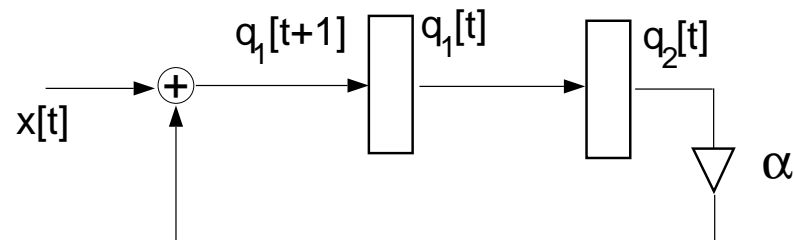
- Replication
  - Checking overhead, voter reliability
- } Will address both!

## LINEAR FINITE-STATE MACHINES (LFSMS)

**State evolution:**  $\mathbf{q}[t + 1] = \mathbf{A}\mathbf{q}[t] \oplus \mathbf{B}\mathbf{x}[t]$

- $\mathbf{A}$ ,  $\mathbf{b}$ ,  $\mathbf{q}[\cdot]$  and  $\mathbf{x}[\cdot]$  have entries in  $GF(p^m)$  ( $p$  prime,  $m \geq 1$ )
- Addition and multiplication in  $GF(p^m)$

**Implementation:** Uses adders, multipliers and memory elements

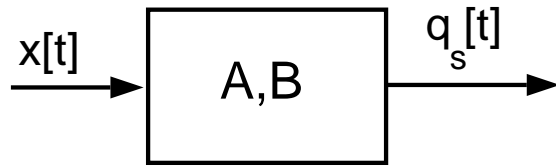


$$\mathbf{q}[t] \equiv \begin{bmatrix} q_1[t] \\ q_2[t] \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} 0 & \alpha \\ 1 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

**Examples:** Sequence enumerators, random number generators, encoders/decoders, linear feedback shift registers, linear cellular automata

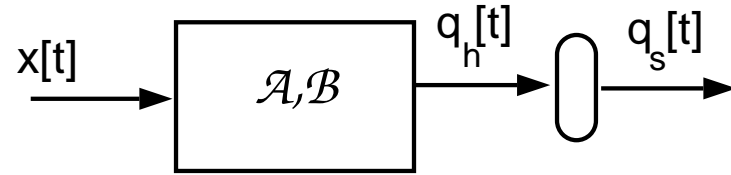
## FAULT DETECTION AND CORRECTION IN LFSMS

### Original LFSM



$$\mathbf{q}_s[t + 1] = \mathbf{A}\mathbf{q}_s[t] \oplus \mathbf{B}\mathbf{x}[t]$$

### Redundant LFSM



$$\mathbf{q}_h[t + 1] = \mathcal{A}\mathbf{q}_h[t] \oplus \mathcal{B}\mathbf{x}[t]$$

- **Concurrent simulation:**  $\mathbf{q}_s[t] = \mathbf{L}\mathbf{q}_h[t]$  } **Linear in GF(p<sup>m</sup>) (not necessary)**
- **Encoding constraints:**  $\mathbf{q}_h[t] = \mathbf{G}\mathbf{q}_s[t]$  }

- **Fault detection:** If  $\mathbf{q}_h[t]$  is *not* in the column space of  $\mathbf{G}$ , or

$$\mathbf{P}\mathbf{q}_h[t] \neq \mathbf{0}, \quad \mathbf{P}\mathbf{G} = \mathbf{0}$$

## CHARACTERIZATION OF REDUNDANT LFSM IMPLEMENTATIONS

<p><u>Original LFSM</u></p> $\mathbf{q}_s[t + 1] = \mathbf{A}\mathbf{q}_s[t] \oplus \mathbf{B}\mathbf{x}[t]$ <p><math>\mathbf{q}_s</math> is <math>d</math>-dimensional</p>	$\left. \begin{array}{l} \mathbf{q}_h[t] \xrightarrow{=\mathbf{G}} \mathbf{q}_s[t] \\ \mathbf{q}_s[t] \xleftarrow{=\mathbf{L}} \mathbf{q}_h[t] \end{array} \right\}$	<p><u>Redundant LFSM</u></p> $\mathbf{q}_h[t + 1] = \mathcal{A}\mathbf{q}_h[t] \oplus \mathcal{B}\mathbf{x}[t]$ <p><math>\mathbf{q}_h</math> is <math>\eta</math>-dimensional (<math>\eta = d + s</math>)</p>
---	--	---

**Standard redundant implementations (Hadjicostis & Verghese 2002):**

$(\mathcal{A}, \mathcal{B})$  is a redundant implementation for  $(\mathbf{A}, \mathbf{B})$  iff  $(\mathcal{A}, \mathcal{B})$  is similar to the following standard form:

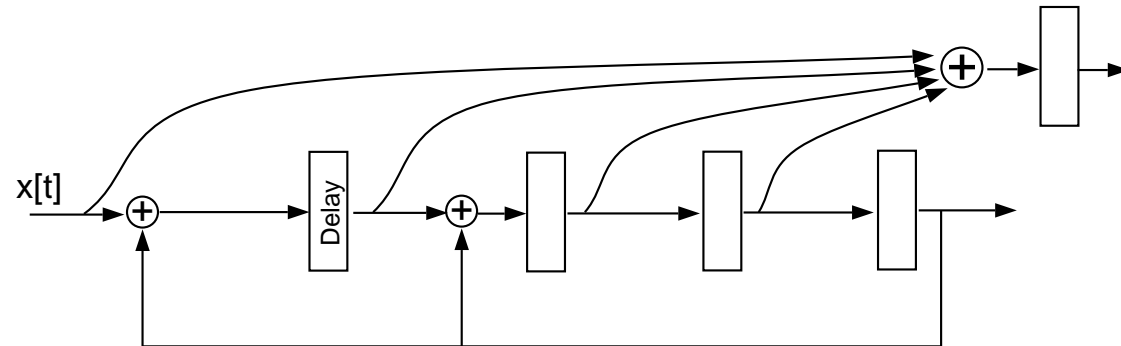
$$\mathbf{q}_\sigma[t + 1] = \underbrace{\begin{bmatrix} \mathbf{A} & \mathbf{A}_{12} \\ \mathbf{0} & \mathbf{A}_{22} \end{bmatrix}}_{\mathcal{A}_\sigma} \mathbf{q}_\sigma[t] \oplus \underbrace{\begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix}}_{\mathcal{B}_\sigma} \mathbf{x}[t]$$

for some matrices  $\mathbf{A}_{12}, \mathbf{A}_{22}$

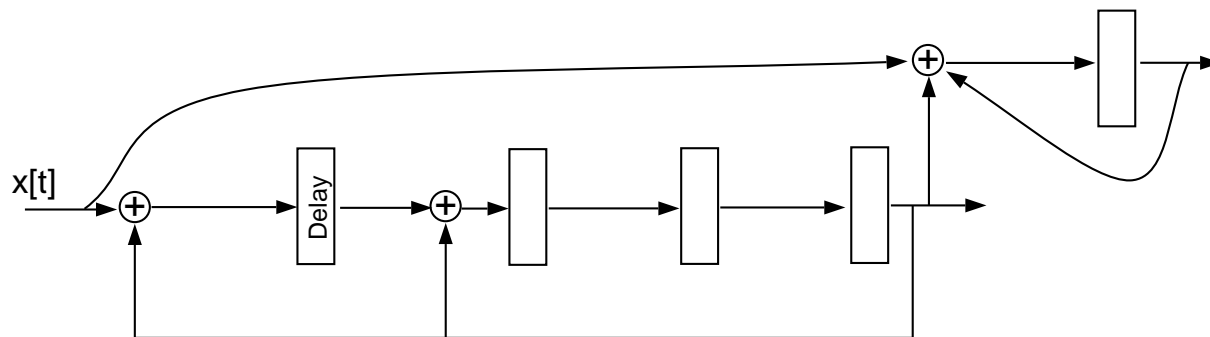
**Specifically:** Invertible  $\mathcal{T}$  such that  $\mathcal{A}_\sigma = \mathcal{T}^{-1}\mathcal{A}\mathcal{T}$  ,  $\mathcal{B}_\sigma = \mathcal{T}^{-1}\mathcal{B}$

# DIFFERENT REDUNDANT IMPLEMENTATIONS FOR CHECKSUM SCHEME IN $GF(2)$

**Traditionally (Chatterjee, Abraham, Reed):**



**Using previous theorem:**



## CONCURRENT FAULT DETECTION AND IDENTIFICATION

**Fault model:** Single fault corrupts  $i$ th state variable

$$\mathbf{q}_f[t] = \underbrace{\mathbf{q}_h[t]}_{\text{fault-free}} \oplus v \mathbf{e}_i$$

**Justification:** Interconnections of adders, multipliers and memory elements  
(constrained so that a single fault corrupts a single state variable)

$\Rightarrow$  Class of signal flow graphs, Hadjicostis & Verghese 1999

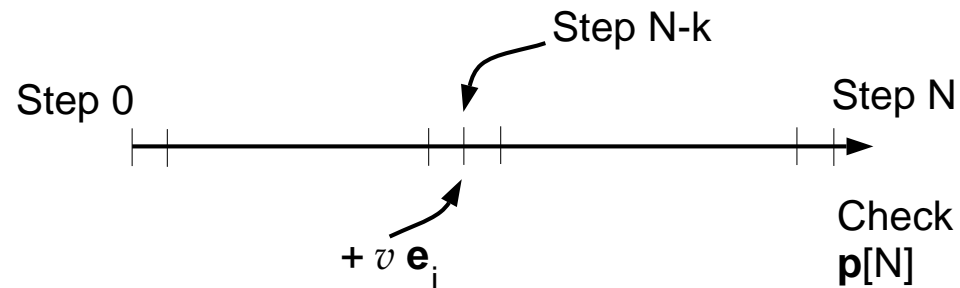
**Concurrent error detection (Abraham, Chatterjee, Hadjicostis, ...):**

At end of *each* time step, perform the *parity check*

$$\mathbf{p}[t] \equiv \mathbf{P} \mathbf{q}_f[t] = \mathbf{P} v \mathbf{e}_i \stackrel{?}{=} \mathbf{0}$$

**Error detection/correction capabilities:** Constraints on matrix  $\mathbf{P}$

## NON-CONCURRENT CHECKING (1)



**Goal:** Design redundant implementation so that knowledge of  $p[N]$  allows detection and identification of error(s) in interval  $[0, N]$

**Motivation:** Relax checking requirements (e.g., periodic checking)

**Need:** For each fault ( $j$ ), identify

- Value ( $v_j$ )
  - State variable ( $e_{i_j}$ )
  - Step ( $N - k_j$ )
- } For error correction, one may have to reset past states/outputs

## NON-CONCURRENT CHECKING (2)

**Error model:** Initially, fault  $j$  at step  $N - k_j$  causes

$$\mathbf{q}_f[N - k_j] = \mathbf{q}_h[N - k_j] \oplus v_j \mathbf{e}_{i_j}$$

**Error propagation:** At step  $N$ ,

$$\mathbf{q}_f[N] = \mathbf{q}_h[N] \oplus \mathcal{A}^{k_j} v_j \mathbf{e}_{i_j}$$

**Parity check:** At step  $N$ ,

$$\mathbf{p}[N] = \mathbf{P} \mathbf{q}_f[N] = v_j \mathbf{P} \mathcal{A}^{k_j} \mathbf{e}_{i_j}$$

**Multiple errors result in:**

$$\mathbf{p}[N] = \sum_{j=1}^D v_j \mathbf{P} \mathcal{A}^{k_j} \mathbf{e}_{i_j}$$

**Task:** Ensure that  $D$  errors can be detected/identified

## SYNDROME GENERATION

**Observation:** Syndrome  $\mathbf{p}[N]$  is a linear combination of  $D$  columns of

$$\mathbf{S} = \left[ \mathbf{P} \quad \mathbf{P}\mathcal{A} \quad \mathbf{P}\mathcal{A}^2 \quad \cdots \quad \mathbf{P}\mathcal{A}^N \right]$$

**Lemma 1:** Detection of  $D$  errors *if and only if*  
all sets of  $D$  columns of  $\mathbf{S}$  are linearly independent  
 $\Rightarrow$  Need at least  $D$  additional variables ( $s \geq D$ )

**Lemma 2:** Identification of  $D$  errors *if and only if*  
all sets of  $2D$  columns of  $\mathbf{S}$  are linearly independent  
 $\Rightarrow$  Need at least  $2D$  additional variables ( $s \geq 2D$ )

**Theorem:** The syndrome matrix  $\mathbf{S}$  can be expressed as

$$\mathbf{S} = \left[ \mathbf{P} \quad \mathbf{A}_{22}\mathbf{P} \quad \mathbf{A}_{22}^2\mathbf{P} \quad \cdots \quad \mathbf{A}_{22}^N\mathbf{P} \right]$$

## OPTIMAL NON-CONCURRENT IDENTIFICATION OF $D$ ERRORS

**Optimal:** Uses minimal number of additional state variables ( $s = 2D$ )

**Fact:** Any  $2D$  columns of  $\mathbf{V}$  are linearly independent if  $x_i \neq x_j$

$$\mathbf{V}(x_1, x_2, \dots, x_r) = \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_r \\ x_1^2 & x_2^2 & \dots & x_r^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{2D-1} & x_2^{2D-1} & \dots & x_r^{2D-1} \end{bmatrix}$$

### Construction of redundant implementation:

1. Start with appropriate parameters  $x, x_1, x_2, \dots, x_\eta$
2. **Set**  $\Lambda = \text{diag}(1, x, x^2, x^3, \dots, x^{2D-1}), \quad \mathbf{M} = \mathbf{V}(x_{d+1}, x_{d+2}, \dots, x_\eta)$
3. **Set**  $\mathbf{A}_{22} = \mathbf{M}^{-1}\Lambda\mathbf{M}, \quad \mathbf{C} = -\mathbf{M}^{-1}\mathbf{V}(x_1, x_2, \dots, x_d)$
4. **Perform** similarity transformation with  $\mathcal{T} = \begin{bmatrix} \mathbf{I}_d & \mathbf{0} \\ \mathbf{C} & \mathbf{I}_{2D} \end{bmatrix}$

## THEOREM AND PROOF

**Theorem:** Resulting redundant implementation allows non-concurrent identification of  $D$  errors (detection of  $2D$  errors)

**Why?** The syndrome matrix  $\mathbf{S}$  can be written as

$$\mathbf{S} = \mathbf{M}^{-1} \underbrace{\mathbf{V}(x_1, \dots, x_\eta, x_1x, \dots, x_\eta x, x_1x^2, \dots, x_\eta x^2, \dots, x_1x^N, \dots, x_\eta x^N)}_{\mathbf{Q}}$$

$\mathbf{Q}$  is a *large* Vandermonde matrix ( $2D \times (\eta(N + 1))$ -dimensional )

**Requirement:** Parameters  $x, x_1, x_2, \dots, x_\eta$  need to be chosen so that

$x_i x^k$  are unique  $\Rightarrow$  Any  $2D$  columns of  $\mathbf{Q}$  are linearly independent

Requirement easily satisfied in field of complex numbers; care needed in  $GF(p^m)$

## FINITE FIELD CONSIDERATIONS

**Design requirement:**  $x_i x^k$  are unique,  $1 \leq i \leq \eta$ ,  $0 \leq k \leq N$

- **Necessary condition:** Finite field needs at least  $\eta(N + 1)$  *nonzero* entries

$$p^m - 1 \geq \eta(N + 1)$$

- **One possibility:** Let  $g$  be a primitive element of  $GF(p^m)$   
(i.e.,  $\{1, g, g^2, g^3, \dots\}$  generates all nonzero elements in  $GF(p^m)$ )

$$\text{Set: } x = g^\eta, \quad x_i = g^i, \quad 1 \leq i \leq \eta$$

- Construction related to BCH convolutional codes (Rosenthal & York 1999)
- Redundant structure for non-concurrent detection of a single error

Choose  $\mathbf{P}$  and  $\mathbf{A}_{22}$  so that all entries are nonzero

E.g., in  $GF(2)$  this implies  $\mathbf{A}_{22} = 1$  and  $\mathbf{P} = [1 \ 1 \ \dots \ 1]$

## SUMMARY OF DESIGN SO FAR

- **Jointly choose:**

- (i) Encoding constraints ( $\mathbf{P}$  or  $\mathbf{G}$ )
- (ii) Redundant dynamics ( $\mathbf{A}_{22}$ )

- **Perform one (non-concurrent) parity check:**

$$\mathbf{p}[N] = \mathbf{P} \mathbf{q}_f[N]$$

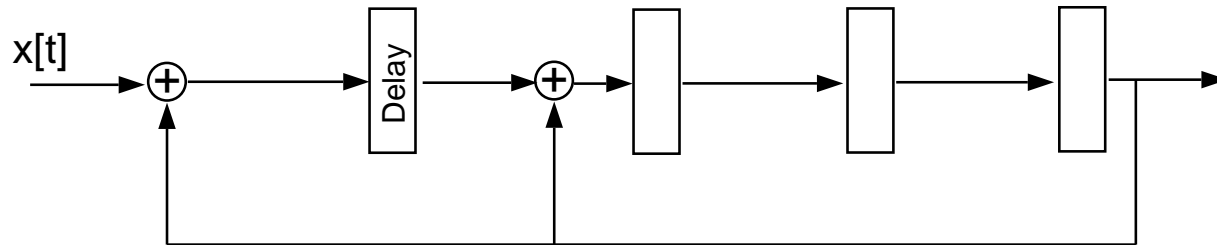
- Detect  $2D$  faults (on any variable, at any step in  $[0, N]$ )
- Identify  $D$  faults (on any variable, at any step in  $[0, N]$ )

- **Advantages:**

- Only  $2D$  additional state variables (optimal)
- Efficient identification (Peterson-Gorenstein-Ziegler decoding)

- **Limitation:** Size of the finite field

## EXAMPLE: NON-CONCURRENT ERROR IDENTIFICATION OF TWO ERRORS (1)



**State evolution (in  $\text{GF}(41)$ ):**

$$\begin{aligned} \mathbf{q}_s[t+1] &= \mathbf{A}\mathbf{q}_s[t] + \mathbf{b}x[t] \\ &= \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{q}_s[t] + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} x[t] \end{aligned}$$

**Goal:**  $N = 4$ , Detect and identify two errors in  $[0, 4] \Rightarrow$  Use 4 additional variables

**Note:**  $N = 4$ ,  $\eta = 8 \Rightarrow (N + 1)\eta = 40$

## EXAMPLE: NON-CONCURRENT ERROR IDENTIFICATION OF TWO ERRORS (2)

**Primitive element in GF(41):**  $g = 7$  generates all nonzero elements

$$\{7, 7^2, 7^3, 7^4, 7^5, 7^6, 7^7, 7^8, 7^9, 7^{10}, \dots\} = \{7, 8, 15, 23, 38, 20, 17, 37, 13, 9, \dots\}$$

**Choose:**

$$\mathbf{M} = \mathbf{V}(38, 20, 17, 37) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 38 & 20 & 17 & 37 \\ 9 & 31 & 2 & 16 \\ 14 & 5 & 34 & 18 \end{bmatrix}, \quad \mathbf{\Lambda} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & x & 0 & 0 \\ 0 & 0 & x^2 & 0 \\ 0 & 0 & 0 & x^3 \end{bmatrix}, \quad x = 37$$

**Set:**

$$\mathbf{C} = -\mathbf{M}^{-1}\mathbf{V}(7, 8, 15, 23) = \begin{bmatrix} 37 & 20 & 29 & 28 \\ 38 & 23 & 12 & 34 \\ 7 & 21 & 25 & 21 \\ 40 & 17 & 15 & 39 \end{bmatrix}$$

$$\mathbf{A}_{22} = \mathbf{M}^{-1}\mathbf{\Lambda}\mathbf{M} = \begin{bmatrix} 21 & 16 & 36 & 18 \\ 7 & 18 & 40 & 7 \\ 23 & 18 & 12 & 37 \\ 32 & 31 & 36 & 21 \end{bmatrix}$$

EXAMPLE: NON-CONCURRENT ERROR IDENTIFICATION OF TWO ERRORS (3)

**Redundant implementation after transformation:**

$$\mathbf{q}_h[t + 1] = \left[ \begin{array}{cccc|cccc} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 24 & 25 & 9 & 21 & 16 & 36 & 18 \\ 37 & 16 & 27 & 26 & 7 & 18 & 40 & 7 \\ 38 & 33 & 5 & 29 & 23 & 18 & 12 & 37 \\ 7 & 9 & 25 & 17 & 32 & 31 & 36 & 21 \end{array} \right] \mathbf{q}_h[t] + \frac{\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 37 \\ 38 \\ 7 \\ 40 \end{bmatrix}}{37} x[t]$$

**Syndrome matrix:**

$$\mathbf{S} = \mathbf{M}^{-1} \underbrace{\left[ \mathbf{S}_0 \quad \mathbf{S}_1 \quad \mathbf{S}_2 \quad \mathbf{S}_3 \quad \mathbf{S}_4 \right]}_{\mathbf{Q}}$$

where  $\mathbf{S}_k = \mathbf{V}(7x^k, 8x^k, 15x^k, 23x^k, 38x^k, 20x^k, 17x^k, 37x^k)$ ,  $x = 37$

**Decoding efficiency:** Slight modification allows the use of PGZ algorithm

## CONCLUSIONS AND FUTURE WORK

- Systematic, resource-efficient fault tolerance for LFSMs
- Reflection of hardware faults through appropriate error models
- Characterization of standard redundant LFSMs
  - (i) Generalizes modular redundancy and checksum schemes
  - (ii) Systematically develops non-concurrent error detection/correction (completely characterizes non-concurrent fault detection)
  - (iii) Optimal schemes (minimal number of additional state variables)
  - (iv) Connections with linear coding and linear system theory

### **Future work:**

- Other pairs of coding schemes and redundant dynamics
- General hardware descriptions
- Other types of systems (digital LTI filters, finite-state machines, analog systems)