# Fault-Tolerant Sequence Enumerators

**Christoforos Hadjicostis**

Decision and Control Laboratory
ECE and CSL
University of Illinois at Urbana-Champaign

# Fault Tolerance

**Fault tolerance describes ability to**

- Withstand internal failures
- Produce desirable overall "behavior"

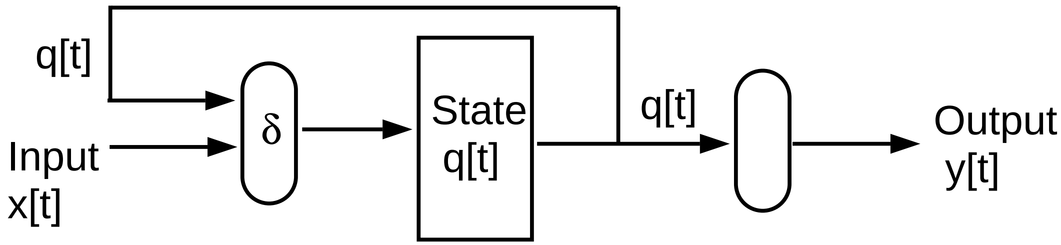
**Necessary or desirable in**

- Life-threatening circumstances
  (e.g., military, transportation, or medical systems)

- Systems in remote or inaccessible environments
  (e.g., space missions)

- Reliable systems from unreliable components
  (faster, less expensive, less power)

## Fault-Tolerant Discrete-Time Dynamic Systems




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

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

Examples include digital filters, encoders/decoders, computer simulations

Failures affect state transition mechanism and/or output mechanism

**Research goals:**

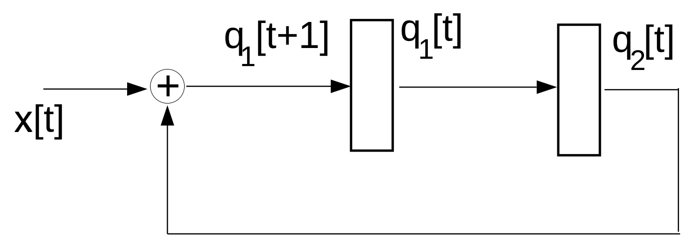
- Resource-efficient monitoring/testing of dynamic systems/networks
- Tradeoffs between detection delay, redundant hardware and monitor complexity
- Fundamental limitations (coding- and information-theoretic techniques)

# Linear Finite State Machines

**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 $x[\cdot]$ have entries in $GF(2)$ (i.e., "0" or "1")
- Modulo-2 addition and multiplication

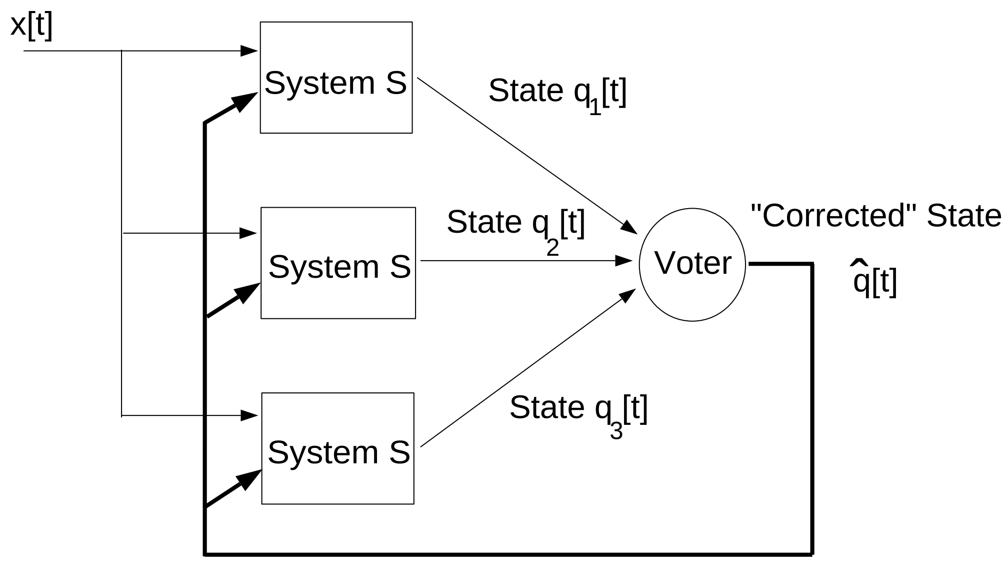
**Implementation:** Uses XOR gates and flip-flops




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

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

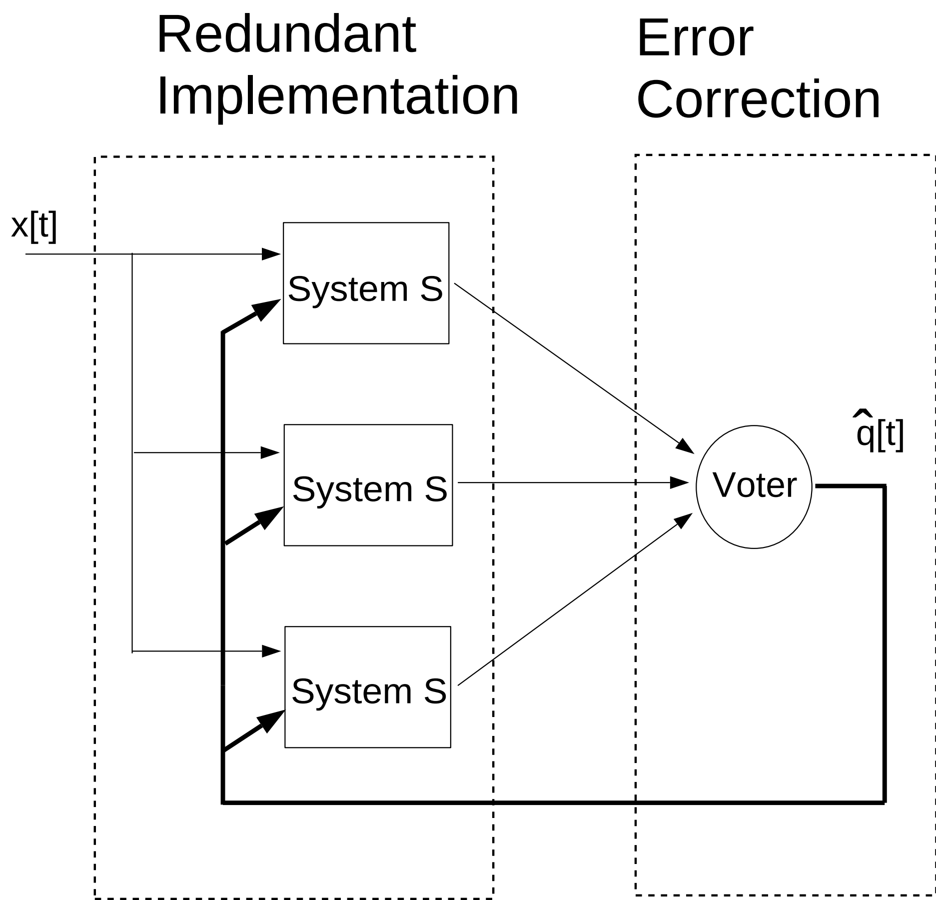
# Traditional Approach: Modular Redundancy (von Neumann, 1956)




**Problems:**

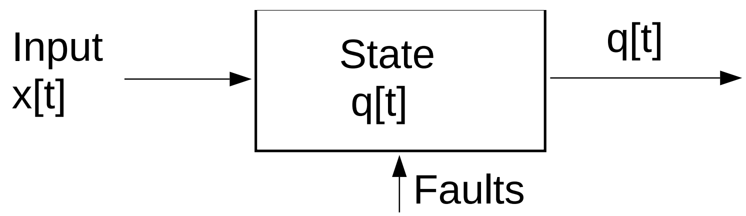
- Replication
- Voter failures

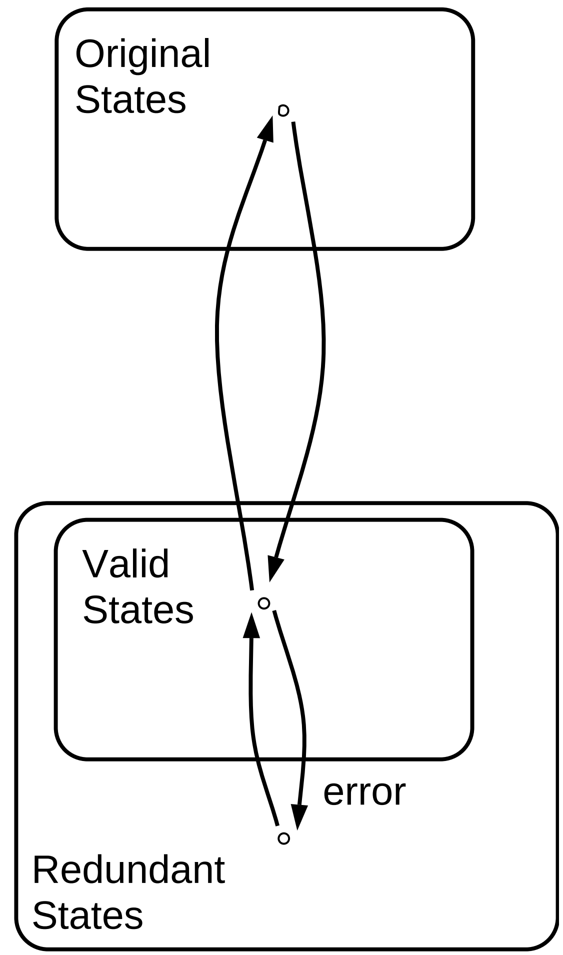
# Avoiding Replication



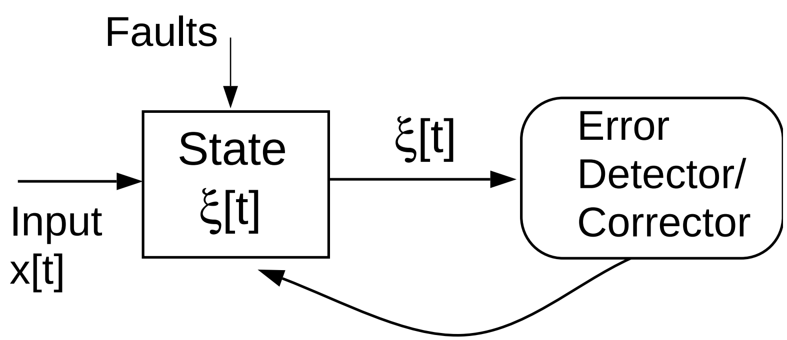

**How can we avoid replication and minimize redundant hardware?**

# Redundant Implementations




**Replace with larger dynamic system:**



## Redundant Embeddings of Linear Finite State Machines

| Original LFSM | | Redundant LFSM |
|---|---|---|
| $\mathbf{q}[t+1] \;=\; \mathbf{A}\mathbf{q}[t] \oplus \mathbf{B}\mathbf{x}[t]$ | $\xrightarrow{\xi[\cdot]=\mathbf{G}\mathbf{q}[\cdot]}$ | $\xi[t+1] \;=\; \mathcal{A}\xi[t] \oplus \mathcal{B}\mathbf{x}[t]$ |
| $(\;\mathbf{y}[t] \;=\; \cdots\;)$ | $\xleftarrow{\mathbf{q}[\cdot]=\mathbf{L}\xi[\cdot]}$ | $(\;\mathbf{y}[t] \;=\; \cdots\;)$ |
| $\mathbf{q}$ is $n$-dimensional | | $\xi$ is $\eta$-dimensional |

- Encoding: $\xi[t] = \mathbf{G}\mathbf{q}[t]$
- Decoding: $\mathbf{q}[t] = \mathbf{L}\xi[t]$
- Fault detected when:
  - Redundant state vector is *not* in the column space of $\mathbf{G}$
  - $\mathbf{P}\xi[\cdot] \neq 0$ (where $\mathbf{P}$ is chosen so that $\mathbf{P}\mathbf{G} = \mathbf{0}$)

## Characterization of Redundant LFSM's

**Theorem:** All redundant LFSM's for the LFSM with state evolution

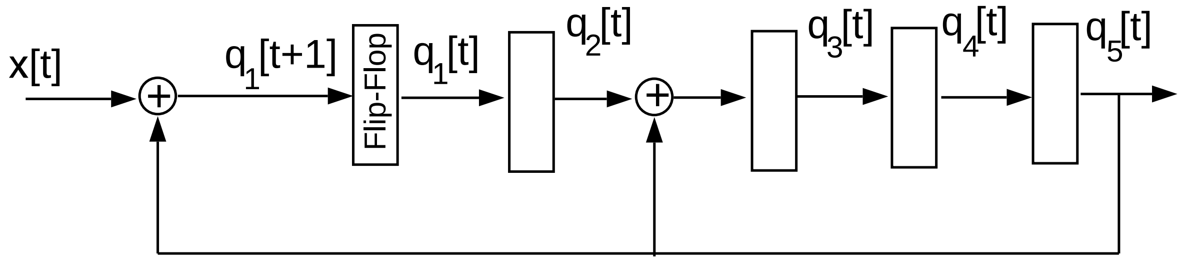
$$\mathbf{q}[t+1] = \mathbf{A}\mathbf{q}[t] \oplus \mathbf{B}\mathbf{x}[t]$$

are systems that are *similar* to

$$\xi_\sigma[t+1] \equiv \begin{bmatrix} \xi_{\sigma_1}[t+1] \\ \xi_{\sigma_2}[t+1] \end{bmatrix} = \left[\begin{array}{c|c} \mathbf{A} & \mathbf{A}_{12} \\ \hline \mathbf{0} & \mathbf{A}_{22} \end{array}\right] \xi_\sigma[t] \oplus \left[\begin{array}{c} \mathbf{B} \\ \hline \mathbf{0} \end{array}\right] \mathbf{x}[t]$$

- Decoding: $\mathbf{q}[t] = \mathbf{L}_\sigma \xi_\sigma[t]$, where $\mathbf{L}_\sigma = \begin{bmatrix} \mathbf{I}_n & \mathbf{0} \end{bmatrix}$
- Encoding: $\xi_\sigma[t] = \mathbf{G}_\sigma \mathbf{q}[t]$, where $\mathbf{G}_\sigma = \begin{bmatrix} \mathbf{I}_n \\ \mathbf{0} \end{bmatrix}$
- Parity Check: $\mathbf{P}_\sigma \, \xi_\sigma[t] = \begin{bmatrix} 0 & \mathbf{I}_d \end{bmatrix} \xi_\sigma[t] \stackrel{?}{=} \mathbf{0}$

# Example: Sequence Enumerator (1)



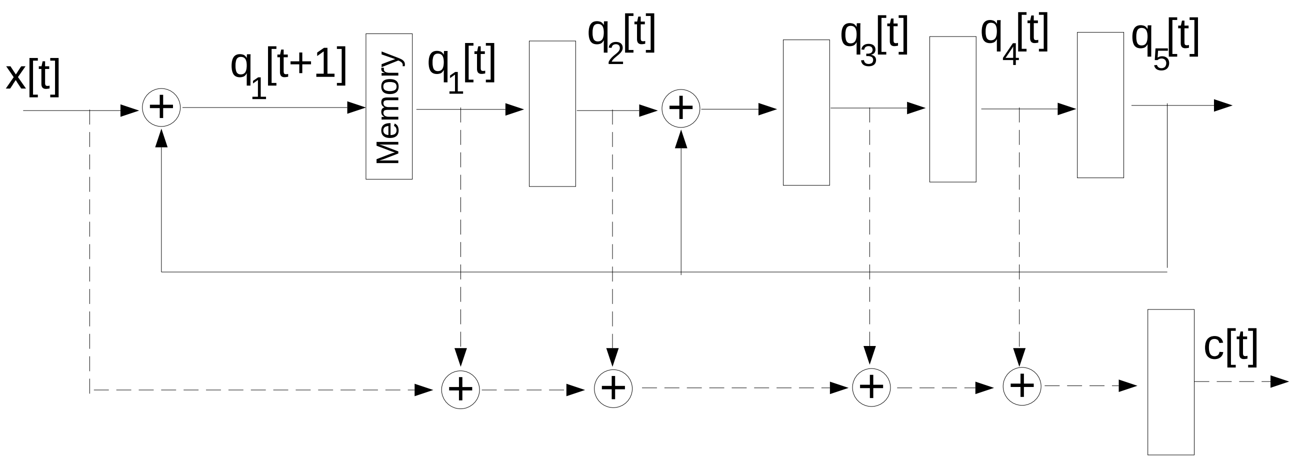

- State "evolution" $\mathbf{q}[t+1] = \mathbf{A}\mathbf{q}[t] \oplus \mathbf{b}x[t]$ where

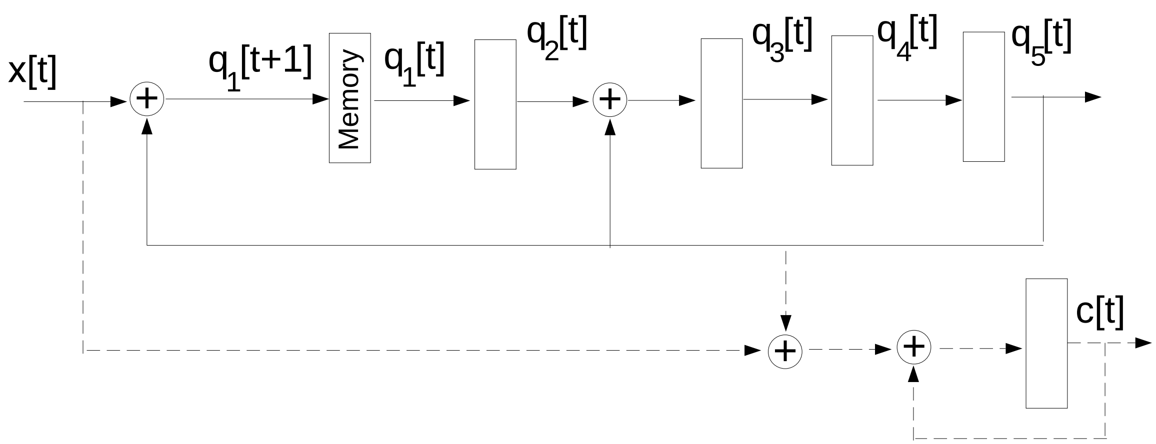
$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

- Would like to enforce

$$\mathbf{G} = \begin{bmatrix} \mathbf{I}_5 \\ \mathbf{c}^T \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \qquad \mathbf{L} = \begin{bmatrix} \mathbf{I}_5 & \mathbf{0} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

# Example: Sequence Enumerator (2)



## Resource-Efficient Redundant Implementations

- Given an LFSM, there is a class of embeddings that satisfy

$$\mathbf{G} = \begin{bmatrix} \mathbf{I}_n \\ \mathbf{C} \end{bmatrix}, \quad \mathbf{L} = \begin{bmatrix} \mathbf{I}_n & \mathbf{0} \end{bmatrix}, \quad \mathbf{P} = \begin{bmatrix} \mathbf{C} & \mathbf{I}_d \end{bmatrix},$$

  (where $\mathbf{C}$ is a $d \times n$ binary matrix)

- The redundant LFSM has $\xi[t+1] = \mathcal{A}\xi[t] \oplus \mathcal{B}\mathbf{x}[t]$ where

$$\mathcal{A} = \mathcal{T}^{-1} \begin{bmatrix} \mathbf{A} & \mathbf{A}_{12} \\ \mathbf{0} & \mathbf{A}_{12} \end{bmatrix} \mathcal{T} = \left[\begin{array}{c|c} \mathbf{A} \oplus \mathbf{A}_{12}\mathbf{C} & \mathbf{A}_{12} \\ \hline \mathbf{CA} \oplus \mathbf{CA}_{12}\mathbf{C} \oplus \mathbf{A}_{22}\mathbf{C} & \mathbf{CA}_{12} \oplus \mathbf{A}_{22} \end{array}\right],$$

$$\mathcal{B} = \mathcal{T}^{-1} \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{B} \\ \mathbf{CB} \end{bmatrix},$$

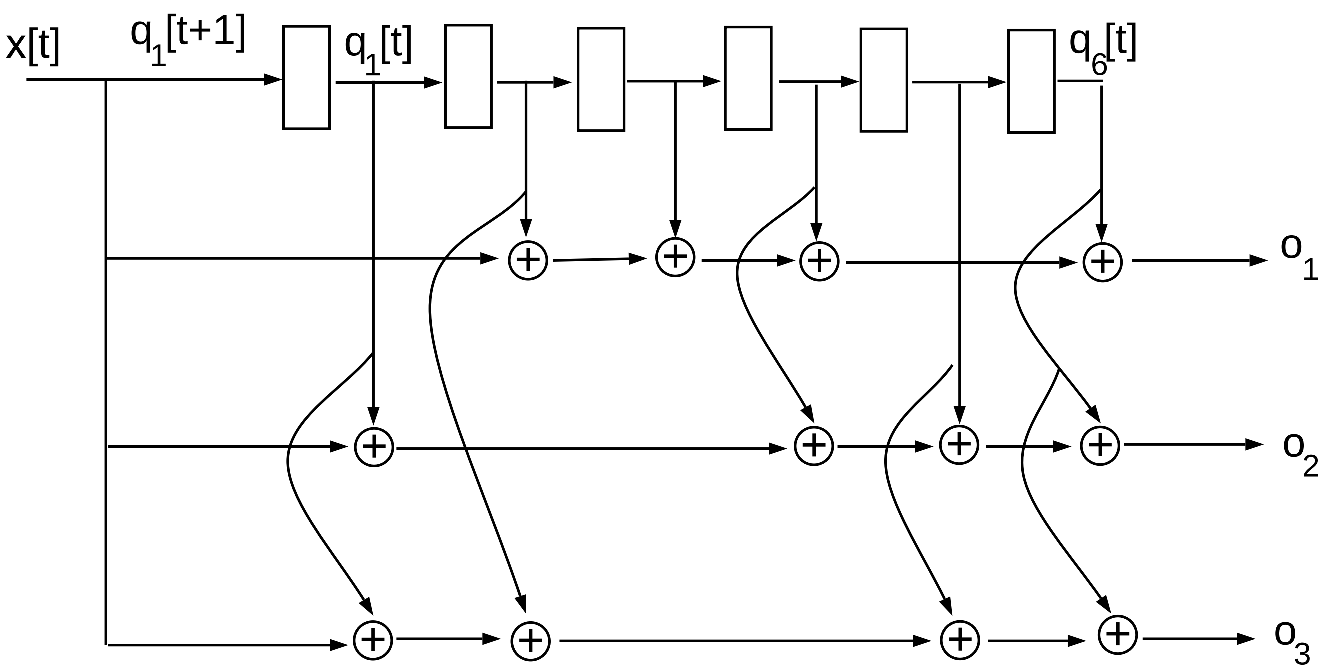
  where $\mathcal{T} = \begin{bmatrix} \mathbf{I}_n & \mathbf{0} \\ \mathbf{C} & \mathbf{I}_d \end{bmatrix}$.

- $2^{(n+d)\times d}$ different redundant versions (same $\mathbf{L}, \mathbf{G}, \mathbf{P}$)
- Efficient search algorithm to minimize computational hardware

# Convolutional Encoder (1)

A convolutional encoder

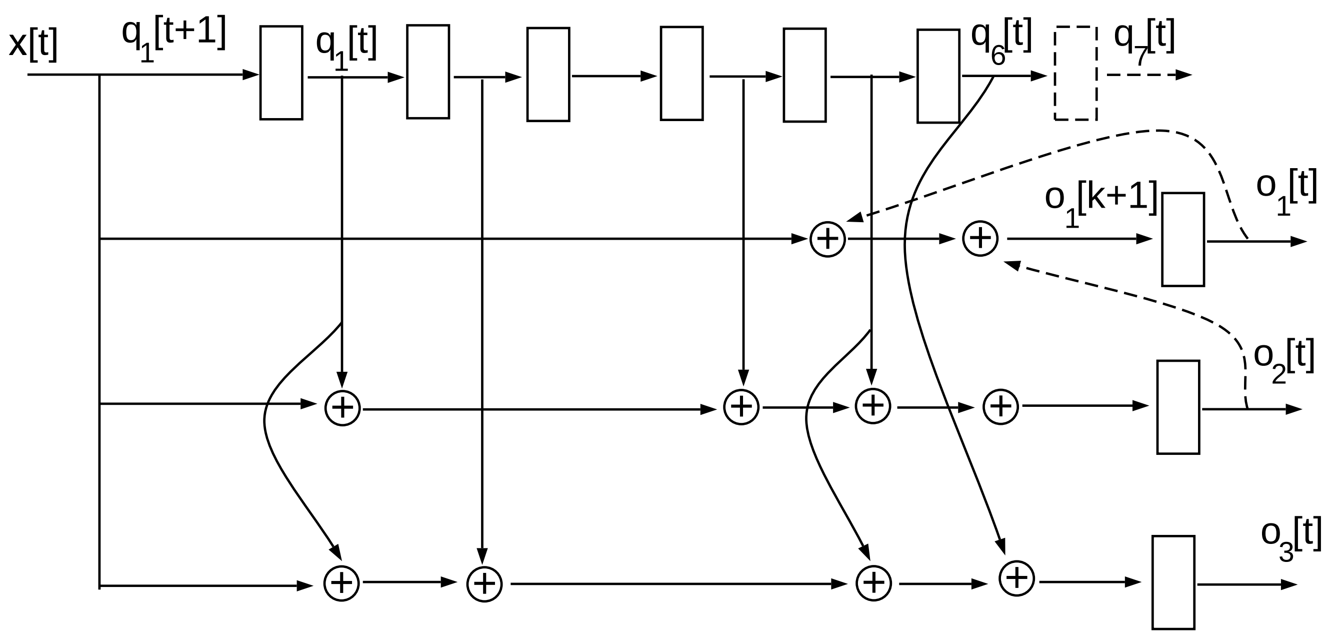
$$g_1(D) = 1 \oplus D^2 \oplus D^3 \oplus D^4 \oplus D^6 \equiv (1011101)$$
$$g_2(D) = 1 \oplus D \oplus D^4 \oplus D^5 \oplus D^6 \equiv (1100111)$$
$$g_3(D) = 1 \oplus D \oplus D^2 \oplus D^5 \oplus D^6 \equiv (1110011)$$



## Convolutional Encoder (2)

- The encoder (that memorizes its outputs) has state of the form

$$\xi[k] \equiv \begin{bmatrix} \mathbf{q}[k] \\ \mathbf{o}[k] \end{bmatrix} = \begin{bmatrix} \mathbf{q}[k] \\ \mathbf{C}\mathbf{q}[k] \end{bmatrix}, \quad \text{where} \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$



## Conclusions and Future Work

- **Conclusions:**
  - Resource-efficient fault tolerance for LFSM's
  - Reflection of hardware failures through appropriate error models
  - Characterization of standard redundant LFSM's
  - Connections to linear coding and linear system theory

- **Future Work:**
  - Joint choice of code and redundant system dynamics; non-separate codes
  - Hierarchical and/or distributed error detection and correction
  - Protection schemes for discrete event systems
  - Extensions to max-plus dynamic systems