ECE 5560 Project Example
Implementation Architectures of Linear Feedback Shift Registers

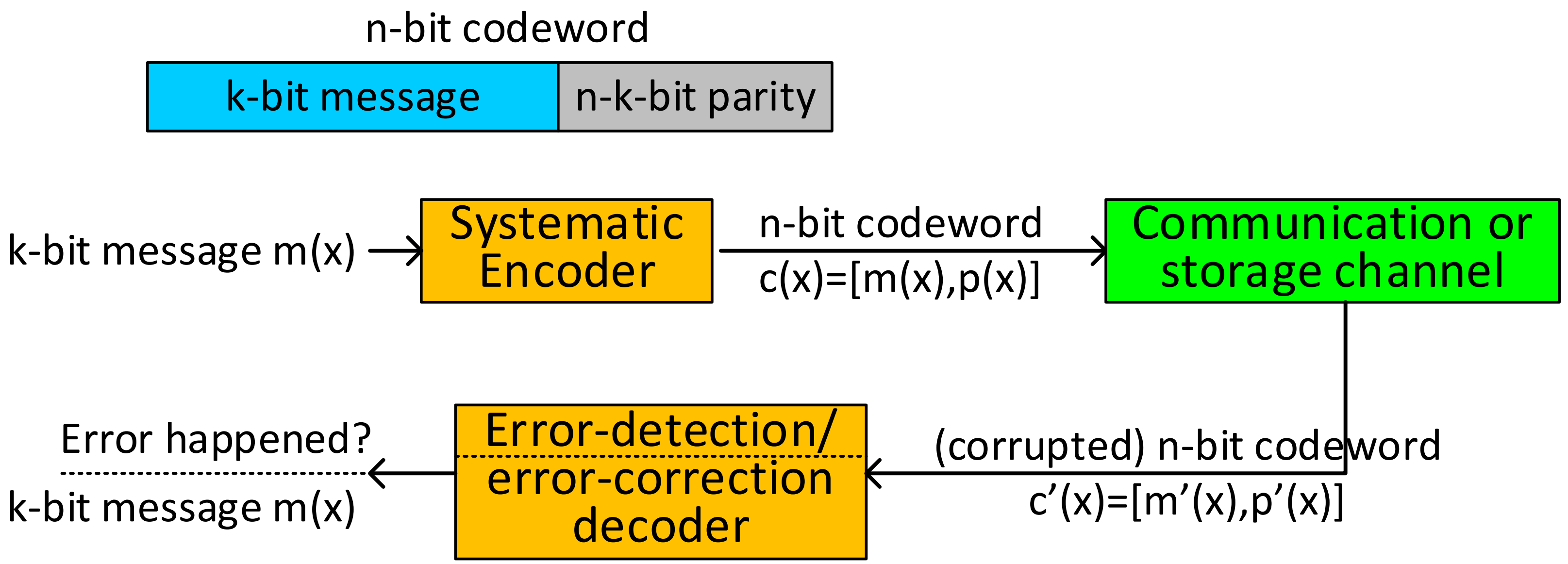
Prof. Xinmiao Zhang

Dept. of Electrical and Computer Engineering

The Ohio State University

Applications of Linear Feedback Shift Registers (LFSRs)

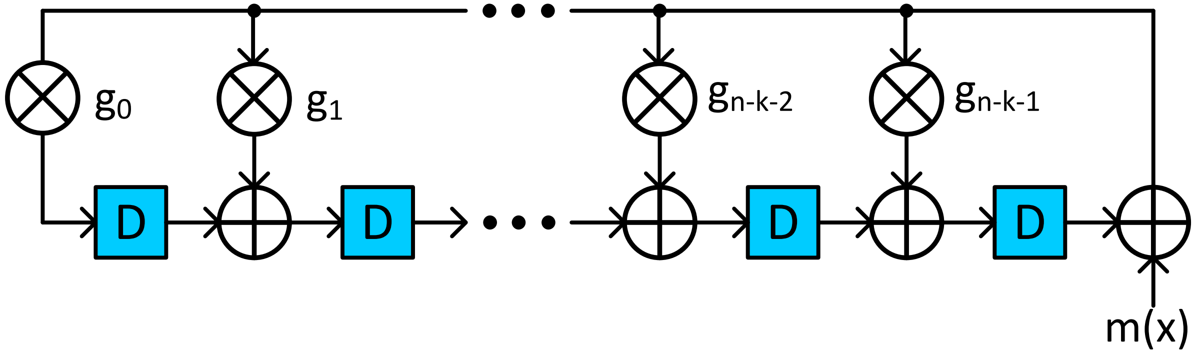
- Used in many digital communication and storage systems
  - Encoder and decoder of cyclic redundancy check (CRC) code for error detection
  - Encoder of Bose–Chaudhuri–Hocquenghem (BCH) codes for error correction



Architecture of Serial LFSRs

- Generator polynomial: $g(x)$, $\deg\big(g(x)\big) = n - k$
- BCH and CRC encoders: compute parity polynomial $p(x) = m(x)x^{n-k}\ mod\ g(x)$
- CRC decoder: compute $p''(x) = m'(x)x^{n-k} mod\ g(x)$; if $p''(x) = p'(x)$, then no error occurred

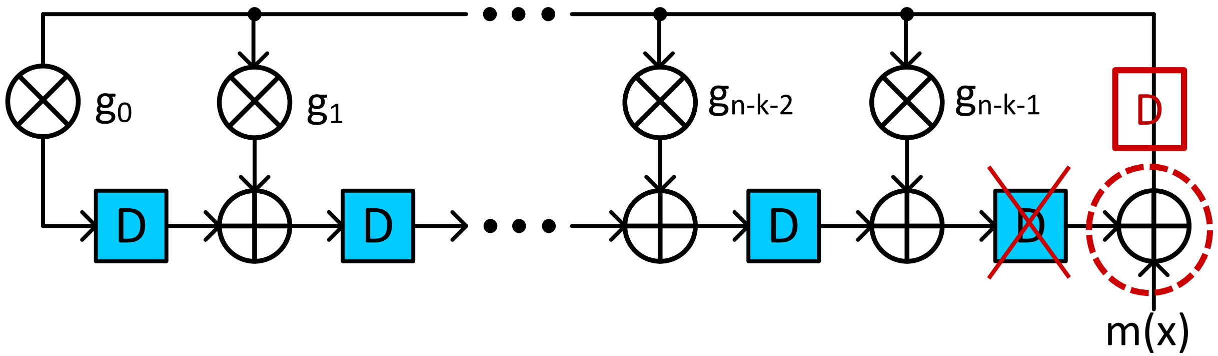
Serial LFSR



- $g(x) = g_0 + g_1 x + \cdots g_{n-k-1} x^{n-k-1} + x^{n-k}$
- Each multiplier is replaced by a wire or no connection when $g(x)$ is binary
- $m_{k-1}, m_{k-2,} \cdots, m_0$ are sent in serially
- $p(x)$ is located in the registers after the message bits are sent in

Parallel LFSRs for Long BCH Encoders

- $\deg(g(x))$ can be several hundreds for long BCH codes used in optimal communications and data storage
- A register is needed at the output of the right-most XOR gate to address the large fanout issue
- Parallel LFSRs are needed to achieve high speed in many systems



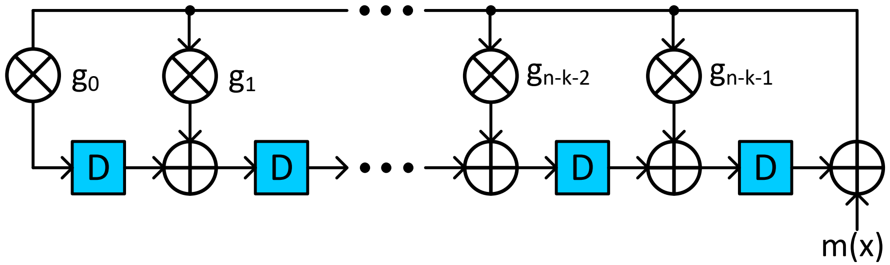
- How can I have enough registers between the two right-most XORs so that retiming can be applied in the unfolded architecture to move at least one register to the output of each copy of the right-most XOR?
- What is the iteration bound of the LFSR? Can I improve the iteration bound and hence reduce the achievable clock period by manipulating the generator polynomial?

References: [1-3]

Parallel LFSRs for CRC

- $\deg(g(x)) \leq 32$ for CRC
- Need high-speed, small-area, and/or low-power implementations

Serial LFSR



- Denote the register state at clock cycle t by

$$r(t) = [r_{n-k-1}(t), r_{n-k-2}(t), \cdots, r_0(t)]'$$

$$\downarrow$$

$$r(t+1) = A \times r(t) + b \times m(t)$$

$$\mathbf{A} = \begin{bmatrix} g_{n-k-1} & 1 & 0 & \cdots & 0 \\ g_{n-k-2} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ g_1 & 0 & 0 & \cdots & 1 \\ g_0 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

$$\rightarrow$$

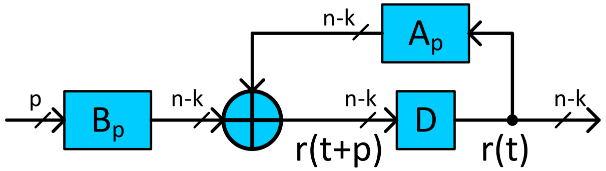
$$r(t+p) = A^p \times r(t) + \mathrm{B_p} \times m_p(t)$$

$$B_p = [A^{p-1}b, \cdots Ab, b]$$

$$m_p(t) = [m(t), \cdots, m(t+p-2), m(t+p-1)]'$$

$$b = [g_{n-k-1}, g_{n-k-2}, \cdots, g_0]'$$

p-parallel LFSR



Transformed Parallel LFSRs

$$r(t+p) = A^p \times r(t) + \mathrm{B_p} \times m_p(t)$$

$$\Big\downarrow \; r(t) = T \times r_T(t)$$

$$r_T(t+p) = A_{pT} \times r_T(t) + \mathrm{B_{pT}} \times m_p(t)$$

$$A_{pT} = T^{-1} \times A^p \times T$$

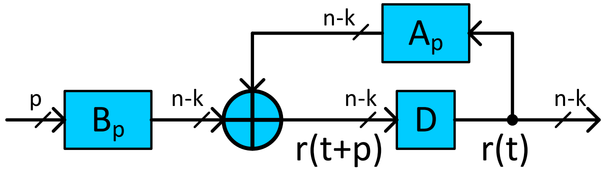
$$B_{pT} = T^{-1} \times B_p$$

- Transformation can be designed to reduce
  - critical path
  - gate count
  - power consumption

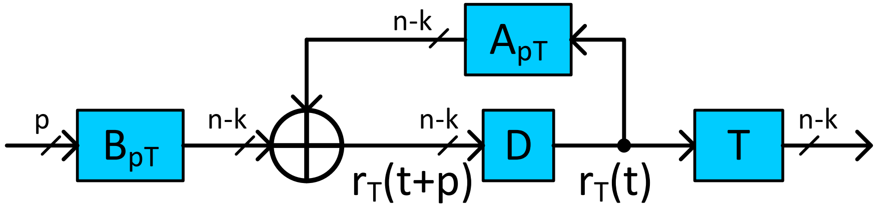
- Can we design a better transformation that leads to a faster, smaller, and/or lower-power design?

References: [4]-[10]

p-parallel LFSR



Transformed p-parallel LFSR



Other Variations of Parallel LFSR Architectures

- LFSR function is interpreted as recursive filtering
- Parallel LFSRs are derived by parallel processing techniques for recursive filters

References: [11][12]

- LFSRs with various generator polynomials $g_1(x), g_2(x), \cdots, g_r(x)$ need to be implemented in the same system
- The generator polynomials satisfy the constraints that $g_i(x)$ divides $g_j(x)$ if $\deg\left(g_i(x)\right) < \deg(g_j(x))$
- Multi-mode LFSRs share hardware units to implement all required polynomial divisions

References: [13]

References

- [1] K. K. Parhi, “Eliminating the fanout bottleneck in parallel long BCH encoders,” IEEE Trans. on Circuits and Syst.-I, vol. 51, no. 3, pp. 512 -516, Mar. 2004.
- [2] X. Zhang and K. K. Parhi, “High-speed architectures for parallel long BCH encoders,” IEEE Trans. on VLSI Syst., vol. 13, no. 7, pp. 872-877, Jul. 2005.
- [3] Y. J. Tang and X. Zhang, “Low-complexity architectures for parallel long BCH encoders,” Proc. of IEEE Workshop on Signal Processing Systems, Oct. 2020.
- [3]T.-B. Pei, and C. Zukowski, “High-speed parallel CRC circuits in VLSI,” IEEE Trans. on Commun., vol. 40, no. 4, pp. 653-657, Apr. 1992.
- [4] J. H. Derby, “High-speed CRC computation using state-space transformations,” Proc. IEEE Global Commun. Conf., pp. 166-170, Nov. 2001.
- [5] C. Kennedy and A. Reyhani-Masoleh, “High-speed CRC computations using improved state-space transformation,” Proc. IEEE Intl. Conf. Electro/Info. Tech., pp. 9-14, 2009.
- [6] G. Hu, J. Sha, and Z. Wang, “High-speed parallel LFSR architectures based on improved state-space transformations,” IEEE Trans. on VLSI Syst. vol. 25, no. 3, pp. 1159-1163, Mar. 2017.
- [7] X. Zhang, “A low-power parallel architecture for linear feedback shift registers,” IEEE Trans. on Circuits and Syst.-II, vol. 66, no. 3, pp. 412-416, Mar. 2019.
- [8] X. Zhang and Y. J. Tang, “Low-complexity parallel cyclic redundancy check,” Proc. of IEEE International Symposium on Circuits and Systems, May 2021.
- [9] X. Zhang, “High-speed and low-complexity parallel long BCH encoder,” Proc. of IEEE International Symposium on Circuits and Systems, virtual, Oct. 2020.
- [10] Y. J. Tang, J. Cai and X. Zhang, “Low-complexity linear feedback shift register architecture for CRC en/decoding,” Proc. of IEEE International Symposium on Circuits and Systems, London, UK, May 2025.
- [11] M. Ayinala and K. K. Parhi, “High-speed parallel architectures for linear feedback shift registers,” IEEE Trans. on Signal Process., vol. 59, no. 9, pp. 4459-4469, Sep. 2011.
- [12] J. Jung, et. al., “Efficient parallel architecture for linear feedback shift registers,” IEEE Trans. on Circuits and Syst.-II, vol. 62, no. 11, pp. 1068-1072, Nov. 2015.
- [13] H. Yoo, et. al., “Area-efficient multimode encoding architecture for long BCH codes,” IEEE Trans. on Circuits and Syst.-II, vol. 60, no. 12, pp. 872-876, Dec. 2013.