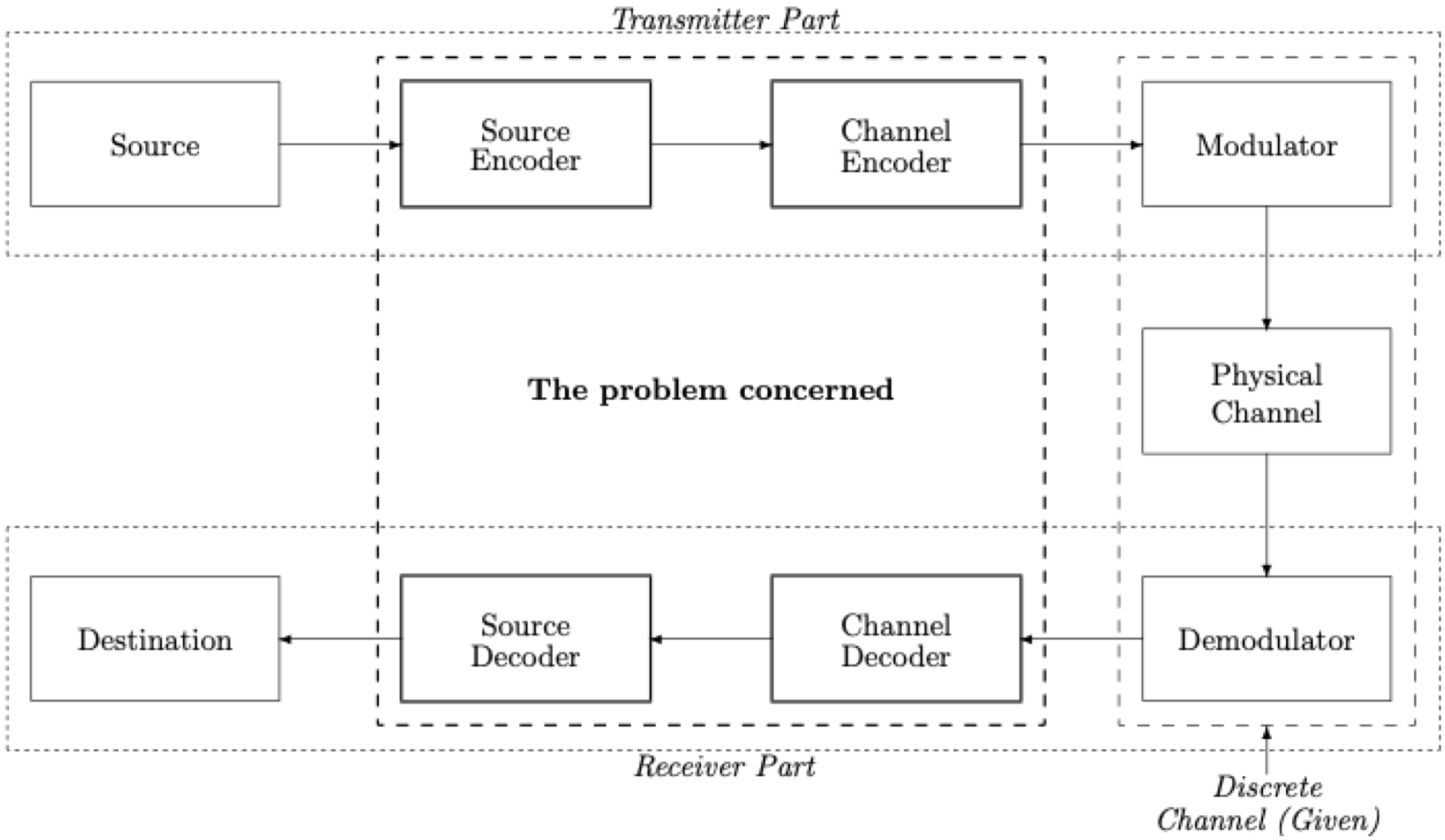
Part 7 Linear Block Codes, Polynomial Codes/Cyclic Codes, Convolutional Codes, and Viterbi Decoding

Introduction



Introduction

- Error correction versus error detection
  - There is an alternative system approach to achieve reliable transmission other than forward error correction (FEC).
  - By the system structure, it is named *automatic-repeat request* (ARQ), which is a combination of error detection and noiseless feedback.

Introduction

- Classifications of ARQ
  - ARQ with stop-and-wait strategy
    - After the transmission of a codeword, the transmitter stops and waits for the feedback before moving onto the next block of message bits.
  - Continuous ARQ with pullback
    - The transmitter continues its transmissions until a retransmission request is received, at which point it stops and pulls back to the incorrectly transmitted codeword.
  - Continuous ARQ with selective repeat
    - Only retransmit the codewords that are incorrectly transmitted.

Linear Block Codes

- Linear code
  - A code is *linear* if any two codewords in the code can be added in *modulo*-2 arithmetic to produce a third codeword in the code.
  - The codewords of a linear code can always be obtained through a "linear" operation in the sense of *modulo*-2 arithmetic.

Linear Block Codes

- For a linear code, there exists a k-by-n generator matrix $\mathbf{G}$ such that

$$\boldsymbol{c}_{1\times n} = \boldsymbol{m}_{1\times k}\mathbf{G}_{k\times n}$$

where code bits $\boldsymbol{c} = [c_0, c_1, \ldots, c_{n-1}]$
message bits $\boldsymbol{m} = [m_0, m_1, \ldots, m_{k-1}]$

(Here, we use *modulo*-2 addition and *modulo*-2 multiplication.)

- Generator matrix $\mathbf{G}$ is said to be in the *canonical form* if its k rows are linearly independent.

What will happen if one row is linearly dependent on other rows?

$$\begin{bmatrix} c_0 & \cdots & c_{n-1} \end{bmatrix} = \begin{bmatrix} m_0 & \cdots & m_{k-1} \end{bmatrix} \begin{bmatrix} g_{0,0} & g_{0,1} & g_{0,2} & \cdots & g_{0,n-1} \\ g_{1,0} & g_{1,1} & g_{1,2} & \cdots & g_{1,n-1} \\ g_{2,0} & g_{2,1} & g_{2,2} & \cdots & g_{2,n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_{k-1,0} & g_{k-1,1} & g_{k-1,2} & \cdots & g_{k-1,n-1} \end{bmatrix}$$

Suppose

$$a \begin{bmatrix} g_{0,0} & g_{0,1} & \cdots & g_{0,n-1} \end{bmatrix} + b \begin{bmatrix} g_{1,0} & g_{1,1} & \cdots & g_{1,n-1} \end{bmatrix} = \begin{bmatrix} g_{2,0} & g_{2,1} & \cdots & g_{2,n-1} \end{bmatrix}$$

Then

$$\begin{bmatrix} c_0 & \cdots & c_{n-1} \end{bmatrix} = \begin{bmatrix} m_0 & \cdots & m_{k-1} \end{bmatrix} \begin{bmatrix} g_{0,0} & g_{0,1} & \cdots & g_{0,n-1} \\ g_{1,0} & g_{1,1} & \cdots & g_{1,n-1} \\ ag_{0,0} + bg_{1,0} & ag_{0,1} + bg_{1,1} & \cdots & ag_{0,n-1} + bg_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ g_{k-1,0} & g_{k-1,1} & \cdots & g_{k-1,n-1} \end{bmatrix}$$

$$
\begin{aligned}
&\begin{bmatrix} c_0 & \cdots & c_{n-1} \end{bmatrix} \\
&= \begin{bmatrix} m_0 & m_1 & m_2 & m_3 & \cdots & m_{k-1} \end{bmatrix}
\begin{bmatrix}
g_{0,0} & g_{0,1} & \cdots & g_{0,n-1} \\
g_{1,0} & g_{1,1} & \cdots & g_{1,n-1} \\
ag_{0,0}+bg_{1,0} & ag_{0,1}+bg_{1,1} & \cdots & ag_{0,n-1}+bg_{1,n-1} \\
\vdots & \vdots & \ddots & \vdots \\
g_{k-1,0} & g_{k-1,1} & \cdots & g_{k-1,n-1}
\end{bmatrix}_{k\times n} \\
&= \begin{bmatrix} \tilde{m}_0 & \tilde{m}_1 & m_3 & m_4 & \cdots & m_{k-1} \end{bmatrix}
\begin{bmatrix}
g_{0,0} & g_{0,1} & \cdots & g_{0,n-1} \\
g_{1,0} & g_{1,1} & \cdots & g_{1,n-1} \\
g_{3,0} & g_{3,1} & \cdots & g_{3,n-1} \\
\vdots & \vdots & \ddots & \vdots \\
g_{k-1,0} & g_{k-1,1} & \cdots & g_{k-1,n-1}
\end{bmatrix}_{(k-1)\times n}
\end{aligned}
$$

where $\tilde{m}_0 = m_0 + m_2 a$ and $\tilde{m}_1 = m_1 + m_2 b$.

Hence, the number of distinct code words is at most 2^{k-1} (not the anticipated 2^k).

Linear Block Codes

- Parity-check matrix $\mathbf{H}$
  - The parity-check matrix of a canonical generator matrix is an (n–k)-by-n matrix satisfying

$$\mathbf{H}_{(n-k)\times n}\mathbf{G}^{T}_{n\times k} = \mathbf{0}_{(n-k)\times k}$$

    where the columns of $\mathbf{H}$ are linearly independent.
  - Then, the codewords (or error-free receptions) should satisfy (n–k) parity-check equations.

$$\Rightarrow \boldsymbol{c}_{1\times n}\mathbf{H}^{T}_{n\times(n-k)} = \boldsymbol{m}_{1\times k}\mathbf{G}_{k\times n}\mathbf{H}^{T}_{n\times(n-k)} = \mathbf{0}_{1\times(n-k)}$$

Linear Block Codes

- Syndrome ***s***
  - The receptions may be erroneous (with error pattern ***e***).

$$\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e}, \quad \text{“+” } \equiv \text{ exclusive or,}$$

$$e_i = \begin{cases} 1, & \text{if an error has occurred at the } i\text{th location;} \\ 0, & \text{otherwise} \end{cases}$$

  - With the help of parity-check matrix, we obtain

$$\begin{aligned} \boldsymbol{s}_{1\times(n-k)} &= \boldsymbol{r}_{1\times n}\mathbf{H}^T_{n\times(n-k)} = \boldsymbol{c}_{1\times n}\mathbf{H}^T_{n\times(n-k)} + \boldsymbol{e}_{1\times n}\mathbf{H}^T_{n\times(n-k)} \\ &= \boldsymbol{e}_{1\times n}\mathbf{H}^T_{n\times(n-k)} \end{aligned}$$

Linear Block Codes

- In short, syndromes are all possible symptoms that possibly happen at the output of parity-check examination.
  - Similar to the disease symptoms that can be observed and examined from outside the body.
  - Based on the symptoms, the doctors diagnose (possibly) what disease occurs inside the body.
- Based on the symptoms, the receiver "diagnoses" which bit is erroneous (i.e., ill) based on the symptoms (so that the receiver can correct the "ill" bit.)
  - Notably, the syndrome only depends on the error pattern, and is completely independent of the transmitted codeword.

Linear Block Codes

- Properties of syndromes
  - Syndrome depends only on the error pattern, and not on the transmitted code word.

$$\begin{aligned} \boldsymbol{s}_{1\times(n-k)} &= \boldsymbol{r}_{1\times n}\mathbf{H}^T_{n\times(n-k)} = \boldsymbol{c}_{1\times n}\mathbf{H}^T_{n\times(n-k)} + \boldsymbol{e}_{1\times n}\mathbf{H}^T_{n\times(n-k)} \\ &= \boldsymbol{e}_{1\times n}\mathbf{H}^T_{n\times(n-k)} \end{aligned}$$

  - All error patterns that differ by (at least) a code word have the same syndrome $\boldsymbol{s}$.

$$\begin{aligned} \text{coset}(\boldsymbol{s}_{1\times(n-k)}) = \{ & \boldsymbol{e}_{1\times n} + \boldsymbol{c}_{1\times n} : \text{ for some codeword } \boldsymbol{c}_{1\times n} \\ & \text{and for some error pattern } \boldsymbol{e}_{1\times n} \\ & \text{satisfying } \boldsymbol{s}_{1\times(n-k)} = \boldsymbol{e}_{1\times n}\mathbf{H}^T_{n\times(n-k)} \} \end{aligned}$$

0. It suffices to fix the error pattern $\boldsymbol{e}$ and vary the codewords when determining the coset.

1. All elements in a coset have the same syndrome since

$$\boldsymbol{c}\mathbf{H}^T = \mathbf{0}$$

2. There are 2^k elements in a coset since

$$\boldsymbol{e} + \boldsymbol{c}_i = \boldsymbol{e} + \boldsymbol{c}_j \Leftrightarrow \boldsymbol{c}_i = \boldsymbol{c}_j$$

   i.e., coset elements are the same iff two codewords are the same.

3. Cosets are disjoint.

4. The number of cosets is 2^{n-k}, i.e., the number of syndromes is 2^{n-k}.

   Thus, syndromes (with only $(n-k)$ unknowns) cannot uniquely determine the error pattern (with n unknowns).

Linear Block Codes

- Systematic code
  - A code is systematic if the message bits are a part of the codewords.
  - The remaining part of a systematic code is called *parity bits*.

$$c_i = \begin{cases} b_i, & i = 0, 1, \ldots, n-k-1 \quad \text{parity bits} \\ m_{i-(n-k)}, & i = n-k, n-k+1, \ldots, n-1 \quad \text{message bits} \end{cases}$$

  - Usually, message bits are transmitted first because the receiver can do "direct hard decision" when "necessary".

Linear Block Codes

- For a systematic code,

$$\mathbf{G} = \left[\mathbf{P}_{k\times(n-k)} \ \vdots \ \mathbf{I}_k\right] \text{ and } \mathbf{H} = \left[\mathbf{I}_{n-k} \ \vdots \ \mathbf{P}^T_{(n-k)\times k}\right]$$

where $\mathbf{I}_k$ is the k-by-k identity matrix.

  - Example. (n, 1) repetition codes

$$\mathbf{P}_{1\times(n-1)} = \underbrace{\left[1 \quad 1 \quad \cdots \quad 1\right]}_{\text{all ones}}$$

Linear Block Codes

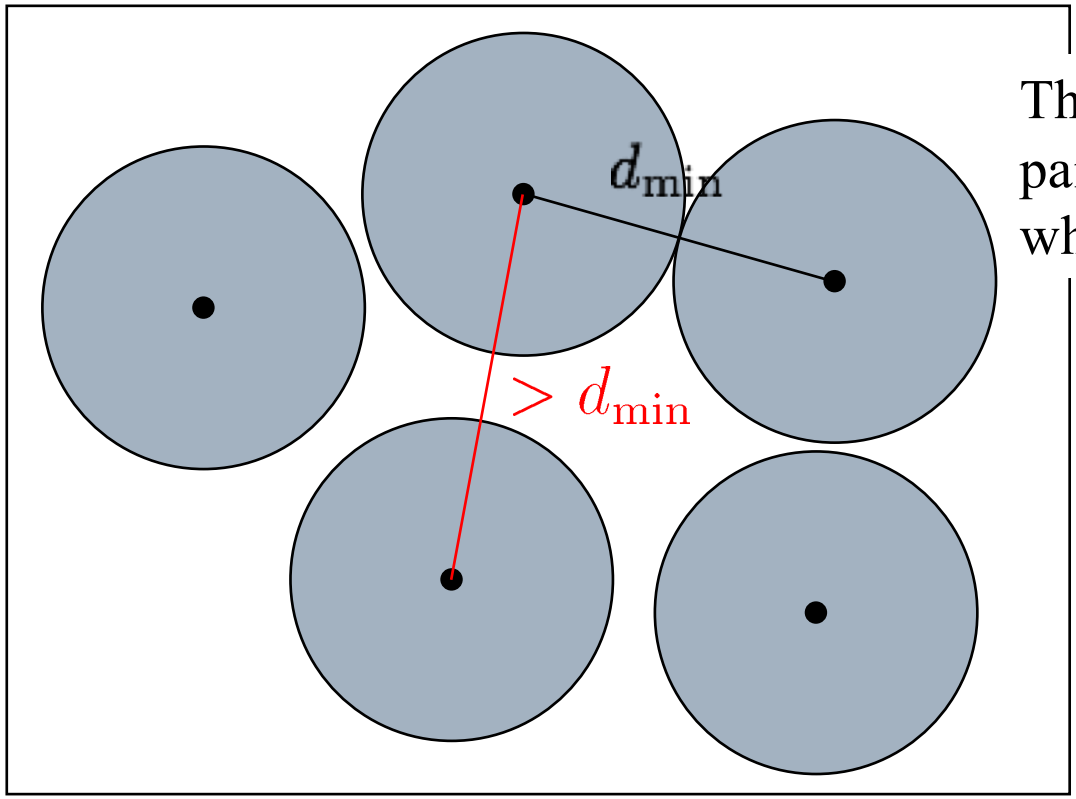
- Error correcting capability of a linear block code
  - Hamming distance (for "0"- "1" binary sequences)

$$d_{\mathrm{H}}(\boldsymbol{u},\boldsymbol{v}) = \sum_{i=1}^{n}(u_i + v_i), \quad \text{“+” } \equiv \text{exclusive or.}$$

  - Minimum (pair-wise) Hamming distance $d_{\min}$

$$d_{\min} = \min_{\boldsymbol{c}_i,\boldsymbol{c}_j \in \mathcal{C}, \boldsymbol{c}_i \neq \boldsymbol{c}_j} d_{\mathrm{H}}(\boldsymbol{c}_i,\boldsymbol{c}_j)$$

- Operational meaning of the minimum (pair-wise) Hamming distance



There exists at least a pair of codewords, of which the distance is $d_{\min}$.

- Minimum distance decoder

$$\text{decision} = \arg\min_{\boldsymbol{c}\in\mathcal{C}} d_{\mathrm{H}}(\boldsymbol{r}, \boldsymbol{c})$$

- Based on the minimum distance decoder, if the received vector and the transmitted codeword differ at most

$$\left\lfloor \frac{1}{2}(d_{\min}-1) \right\rfloor$$

namely, if the number of 1's in the (true) error pattern is at most this number, then no error in decoding decision is obtained.

- If the received vector and the transmitted codeword differ at t positions, where

$$t > \left\lfloor \frac{1}{2}(d_{\min}-1) \right\rfloor$$

then erroneous decision is possibly made (such as when the transmitted codeword is one of the pairs with distance $d_{\min}$.)

- We therefore name this quantity the *error correcting capability* of a code (not limited to linear block codes).

$$\left\lfloor \frac{1}{2}(d_{\min}-1) \right\rfloor$$

- The *error correcting capability* of a **linear** block code can be easily determined by the minimum *Hamming weight* of codewords. (This does not apply for **non-linear** codes!)

$$w_{\mathrm{H}}(\boldsymbol{u}) = d_{\mathrm{H}}(\boldsymbol{u}, \mathbf{0}) = \sum_{i=1}^{n}(u_i + 0) = \sum_{i=1}^{n} u_i$$

- By linearity,

$$(\forall\ \boldsymbol{c}_i \text{ and } \boldsymbol{c}_j \in \mathcal{C}) \text{ there exists } \boldsymbol{c}_k \in \mathcal{C} \text{ such that } w_{\mathrm{H}}(\boldsymbol{c}_k) = d_{\mathrm{H}}(\boldsymbol{c}_i, \boldsymbol{c}_j)$$

$$\Rightarrow d_{\min} = \min_{\boldsymbol{c}_i, \boldsymbol{c}_j \in \mathcal{C}, \boldsymbol{c}_i \neq \boldsymbol{c}_j} d_{\mathrm{H}}(\boldsymbol{c}_i, \boldsymbol{c}_j) = \min_{\boldsymbol{c} \in \mathcal{C}, \boldsymbol{c} \neq \mathbf{0}} w_{\mathrm{H}}(\boldsymbol{c})$$

Linear Block Codes

- Syndrome decoding for (n, k) linear block codes

$$\begin{aligned}
\text{decision} &= \arg\min_{\boldsymbol{c}\in\mathcal{C}} d_{\mathrm{H}}(\boldsymbol{r},\boldsymbol{c}) \\
&= \arg\min_{\boldsymbol{c}\in\mathcal{C}} d_{\mathrm{H}}(\boldsymbol{r}+\boldsymbol{c},\boldsymbol{0}) \left(= \arg\min_{\boldsymbol{e}+\boldsymbol{r}\in\mathcal{C}} d_{\mathrm{H}}(\boldsymbol{r}+\underbrace{\boldsymbol{e}+\boldsymbol{r}}_{\boldsymbol{e}+\boldsymbol{r}\in\mathcal{C}},\boldsymbol{0})\right) \\
&= \arg\min_{\boldsymbol{e}\in\boldsymbol{r}+\mathcal{C}} d_{\mathrm{H}}(\boldsymbol{e},\boldsymbol{0})+\boldsymbol{r} \\
&= \arg\min_{\boldsymbol{e}\in\mathrm{coset}(\boldsymbol{r}\mathbf{H}^T)} w_{\mathrm{H}}(\boldsymbol{e})+\boldsymbol{r}
\end{aligned}$$

$$\boldsymbol{r}=\boldsymbol{c}+\boldsymbol{e} \Rightarrow \begin{cases} \boldsymbol{e}=\boldsymbol{r}+\boldsymbol{c} \\ \boldsymbol{c}=\boldsymbol{e}+\boldsymbol{r} \end{cases}$$

Linear Block Codes

- Syndrome decoding using standard array for an (n, k) block code

	$\boldsymbol{c}_1 = \mathbf{0}$	$\boldsymbol{c}_2$	$\cdots$	$\boldsymbol{c}_i$	$\cdots$	$\boldsymbol{c}_{2^k}$
	$\boldsymbol{c}_1 + \boldsymbol{e}_2$	$\boldsymbol{c}_2 + \boldsymbol{e}_2$	$\cdots$	$\boldsymbol{c}_i + \boldsymbol{e}_2$	$\cdots$	$\boldsymbol{c}_{2^k} + \boldsymbol{e}_2$
	$\boldsymbol{c}_1 + \boldsymbol{e}_3$	$\boldsymbol{c}_2 + \boldsymbol{e}_3$	$\cdots$	$\boldsymbol{c}_i + \boldsymbol{e}_3$	$\cdots$	$\boldsymbol{c}_{2^k} + \boldsymbol{e}_3$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
syndrome $\boldsymbol{r}\mathbf{H}^T$	$\boldsymbol{c}_1 + \boldsymbol{e}_j$	$\boldsymbol{c}_2 + \boldsymbol{e}_j$	$\cdots$	$\boldsymbol{c}_i + \boldsymbol{e}_j$	$\cdots$	$\boldsymbol{c}_{2^k} + \boldsymbol{e}_j$
	$\boldsymbol{c}_1 + \boldsymbol{e}_{2^{n-k}}$	$\boldsymbol{c}_2 + \boldsymbol{e}_{2^{n-k}}$	$\cdots$	$\boldsymbol{c}_i + \boldsymbol{e}_{2^{n-k}}$	$\cdots$	$\boldsymbol{c}_{2^k} + \boldsymbol{e}_{2^{n-k}}$

Find the element in coset($\boldsymbol{r}\mathbf{H}^T$), who has the minimum weight.
This element is usually named the *coset leader*.
The *coset leader* in each coset is fixed and known before the reception of $\boldsymbol{r}$.

Linear Block Codes

- Example: (7,4) Hamming codes

$$\mathbf{G} = \underbrace{\begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{P}\qquad\qquad\mathbf{I}_k}$$

$$\mathbf{H} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}}_{\mathbf{I}_{n-k}\qquad\qquad\mathbf{P}^T}$$

Linear Block Codes

- Decoding table

syndrome	coset leader
000	0000000
100	1000000
010	0100000
001	0010000
110	0001000
011	0000100
111	0000010
101	0000001

$$\boldsymbol{r} = [1100010]$$

$$\Rightarrow \boldsymbol{r}\mathbf{H}^T = [001]$$

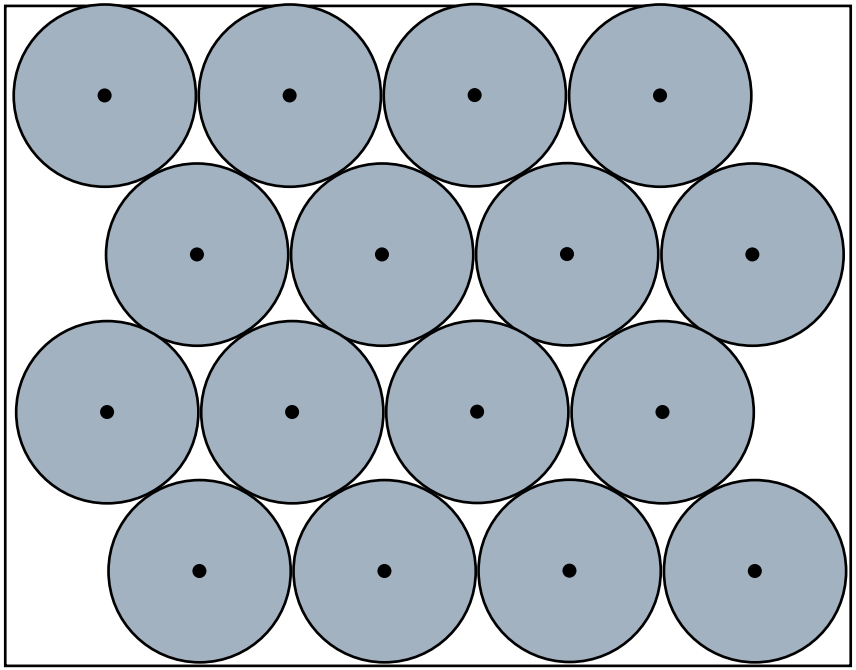
$$\Rightarrow \boldsymbol{e} = [0010000]$$

$$\begin{aligned} \Rightarrow \boldsymbol{c} &= \mathbf{r} + \mathbf{e} \\ &= [1100010] + [0010000] \\ &= [1110010] \end{aligned}$$

Linear Block Codes

For (7,4) Hamming code, the coset leaders form a perfect ball (of radius 1).

- Appendix: The notion of perfect code
  - (7, 4) Hamming code is a binary perfect code.



All of the $2^7 = 128$ binary sequences are confined with the $2^4 = 16$ non-overlapping balls of radius 1,

$$\text{i.e.,} \sum_{t=0}^{1} \binom{7}{t} = \frac{2^7}{2^4} = 2^3$$

Linear Block Codes

- Dual code

$$\mathbf{H}_{(n-k)\times n}\mathbf{G}_{n\times k}^{T}=\mathbf{0}_{(n-k)\times k}$$

$$\Rightarrow \mathbf{G}_{k\times n}\mathbf{H}_{n\times(n-k)}^{T}=\tilde{\mathbf{H}}_{k\times n}\tilde{\mathbf{G}}_{n\times(n-k)}^{T}=\mathbf{0}_{k\times(n-k)}$$

  - Every (n, k) linear block code with generator matrix $\mathbf{G}$ and parity-check matrix $\mathbf{H}$ has an (n, n–k) dual code with generator matrix $\mathbf{H}$ and parity-check matrix $\mathbf{G}$.

Polynomial Codes/Cyclic Codes

- Polynomial expression of a linear code
  - Polynomial code: A special type of linear codes.
    - Represent a code $[c_0, c_1, \ldots, c_{n-1}]$ as a code polynomial of degree $n-1$

$$c(X) = c_0 + c_1 X + c_2 X^2 + \cdots + c_{n-1} X^{n-1}$$

where X is called the indeterminate.

Polynomial Codes/Cyclic Codes

- Generator polynomial (of a polynomial code)

$$g(X) = 1 + g_1 X + g_2 X^2 + \cdots + g_{n-k-1} X^{n-k-1} + X^{n-k}$$

- The code polynomial (of a polynomial code) includes all polynomials of degree (n−1), which can be divided by $g(X)$, and hence can be expressed as

$$c(X) = a(X)g(X)$$

Polynomial Codes/Cyclic Codes

- Example of a (6, 3) polynomial code

$$c(X) = (a_0 + a_1 X + a_2 X^2)(1 + g_1 X + g_2 X^2 + X^3)$$

$$\begin{bmatrix} c_0 & c_1 & c_2 & c_3 & c_4 & c_5 \end{bmatrix} = \begin{bmatrix} a_0 & a_1 & a_2 \end{bmatrix} \begin{bmatrix} 1 & g_1 & g_2 & 1 & 0 & 0 \\ 0 & 1 & g_1 & g_2 & 1 & 0 \\ 0 & 0 & 1 & g_1 & g_2 & 1 \end{bmatrix}$$

So, the polynomial code is a special type of linear codes.

code for $g_1 = g_2 = 1$

$a_0 a_1 a_2$	$c_0 c_1 c_2 c_3 c_4 c_5$
000	000000
001	001111
010	011110
011	010001
100	111100
101	110011
110	100010
111	101101

Polynomial Codes/Cyclic Codes

- Property of a polynomial code

Let $c(X)$ be the code polynomial corresponding to $a(X)$, i.e., $c(X) = a(X)g(X)$.

$\Rightarrow X^{n-k}a(X) = q(X) \cdot g(X) + r(X)$,
where the degree of remainder $r(X)$ is less than $n - k$.

$\Rightarrow \bar{c}(X) = X^{n-k}a(X) - r(X) = q(X)g(X)$ is an alternative way to represent code polynomials.

$\Rightarrow$ The last k bits of $\bar{c}(X)$ should be exactly the same as $a(X)$. As a result, $\bar{c}(X)$ is a systematic code.

Polynomial Codes /Cyclic Codes

- Example of a (6, 3) polynomial code (Continue)

code for $g_1 = g_2 = 1$

$q_0q_1q_2$	$\bar{c}_0\bar{c}_1\bar{c}_2\bar{c}_3\bar{c}_4\bar{c}_5$		$a_0a_1a_2$		$c_0c_1c_2c_3c_4c_5$
000	000000		000		000000
011	010001		001		001111
110	100010	*encode*	010	*encode*	011110
101	110011	⇐	011	⇒	010001
100	111100		100		111100
111	101101		101		110011
010	011110		110		100010
001	001111		111		101101

$$\bar{c}(X) = X^3(a_0 + a_1X + a_2X^2) - X^3(a_0 + a_1X + a_2X^2) \bmod (1 + g_1X + g_2X^2 + X^3)$$

$$\begin{bmatrix}\bar{c}_0 & \bar{c}_1 & \bar{c}_2 & \bar{c}_3 & \bar{c}_4 & \bar{c}_5\end{bmatrix} = \begin{bmatrix}a_0 & a_1 & a_2\end{bmatrix} \begin{bmatrix} 1 & g_1 & g_2 & 1 & 0 & 0 \\ g_2 & 1+g_1g_2 & g_1+g_2 & 0 & 1 & 0 \\ g_1+g_2 & g_1+g_2+g_1g_2 & 1+g_2 & 0 & 0 & 1 \end{bmatrix}$$

So, the polynomial code can be made equivalent to a systematic linear code.

Notably, they are two "equivalent" polynomial coding systems
i) $c(X) = a(X)g(X)$: Not necessarily systematic
ii) $\bar{c}(X) = q(X)g(X)$: Systematic
Here we talk about the encoder for $\bar{c}(X)$.

Encoder for systematic polynomial codes

$$\begin{aligned}\bar{c}(X) &= X^{n-k}a(X) + r(X) \\ &= \underbrace{X^{n-k}a(X)}_{\text{degrees } n-k \text{ to } n-1} + \underbrace{[X^{n-k}a(X) \bmod g(X)]}_{\text{degrees } 0 \text{ to } n-k-1}\end{aligned}$$

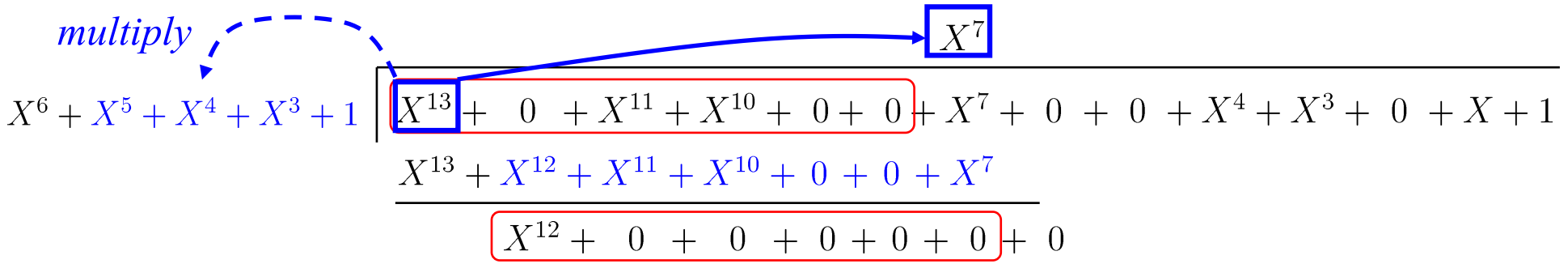
$$[\bar{c}_0 \ \cdots \ \bar{c}_{n-k-1} \ \bar{c}_{n-k} \ \cdots \ \bar{c}_{n-1}] = [u_0 \ \cdots u_{n-k-1} \ a_0 \ \cdots a_{k-1}]$$

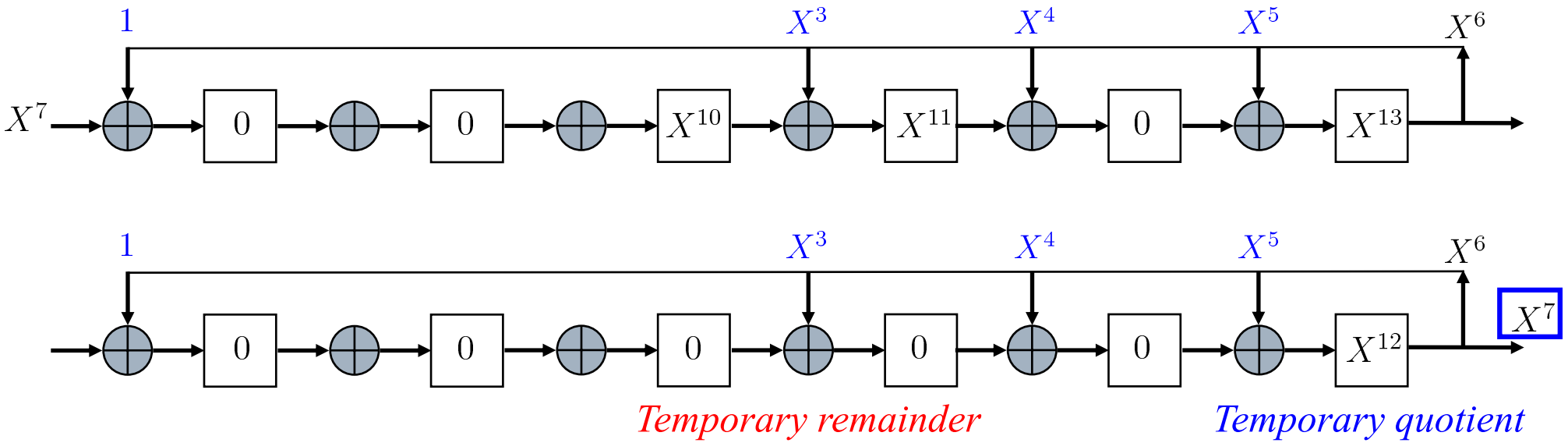
How to find the remainder $(u_0 \ u_1 \ \cdots \ u_{n-k-1})$ of $X^{n-k}a(X)$ divided by $g(X)$?

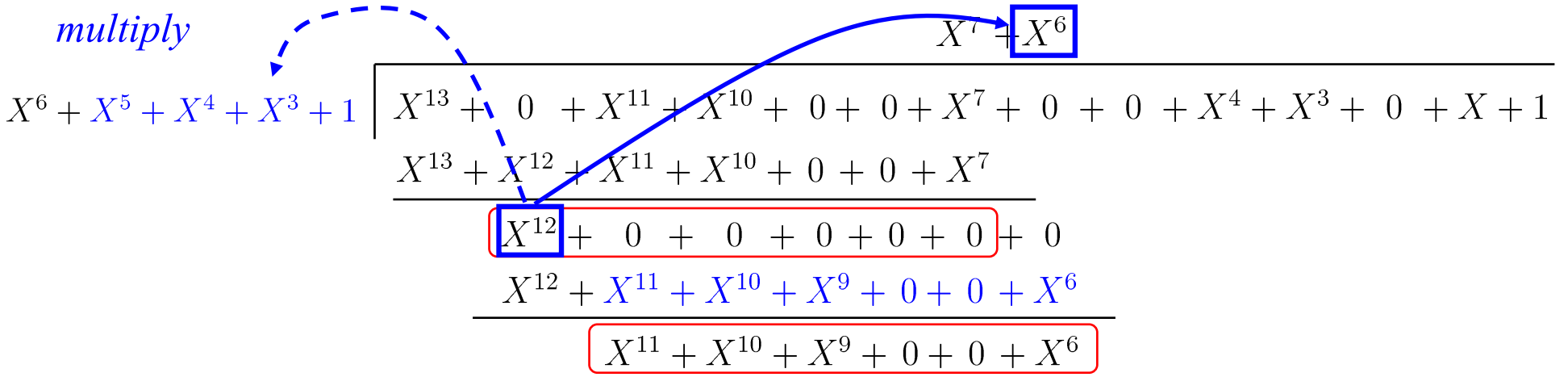
$$X^{n-k}a(X) = q(X)g(X) + u(X)$$

Example of the usual long division

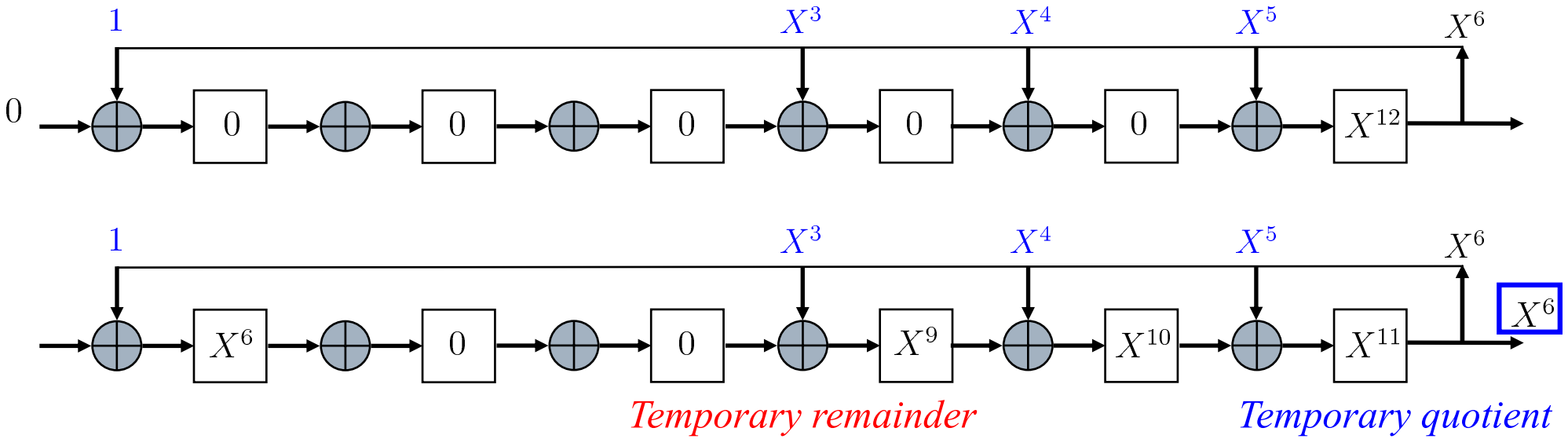
$X^{13}+X^{11}+X^{10}+X^{7}+X^{4}+X^{3}+X+1$ divided by $X^{6}+X^{5}+X^{4}+X^{3}+1$



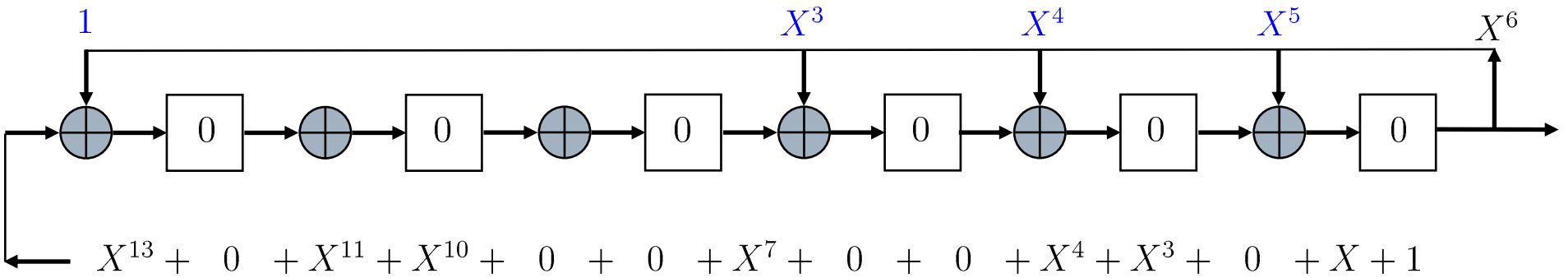




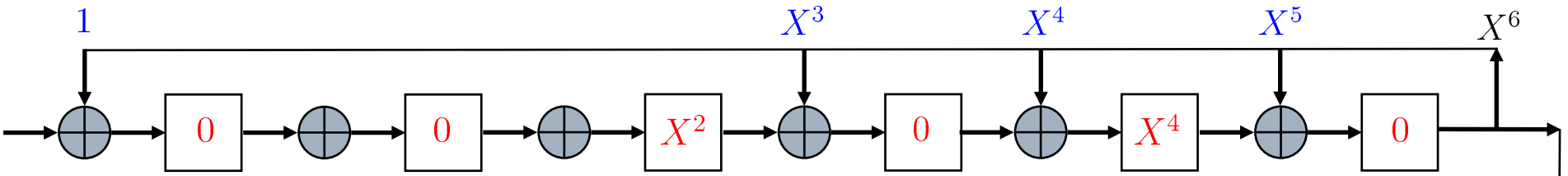
……. repeat this procedure until the last term is shifted into the shift register…..



$X^{13}+X^{11}+X^{10}+X^7+X^4+X^3+X+1$ divided by $X^6+X^5+X^4+X^3+1$



$X^{13}+\ 0\ +X^{11}+X^{10}+\ 0\ +\ 0\ +X^7+\ 0\ +\ 0\ +X^4+X^3+\ 0\ +X+1$



$0\ +\ 0\ +\ 0\ +\ 0\ +\ 0\ +\ 0\ +\underbrace{X^7+X^6+X^5+\ 0\ +\ 0\ +X^2+X+1}_{\text{quotient}}$

Remainder $= X^4+X^2$

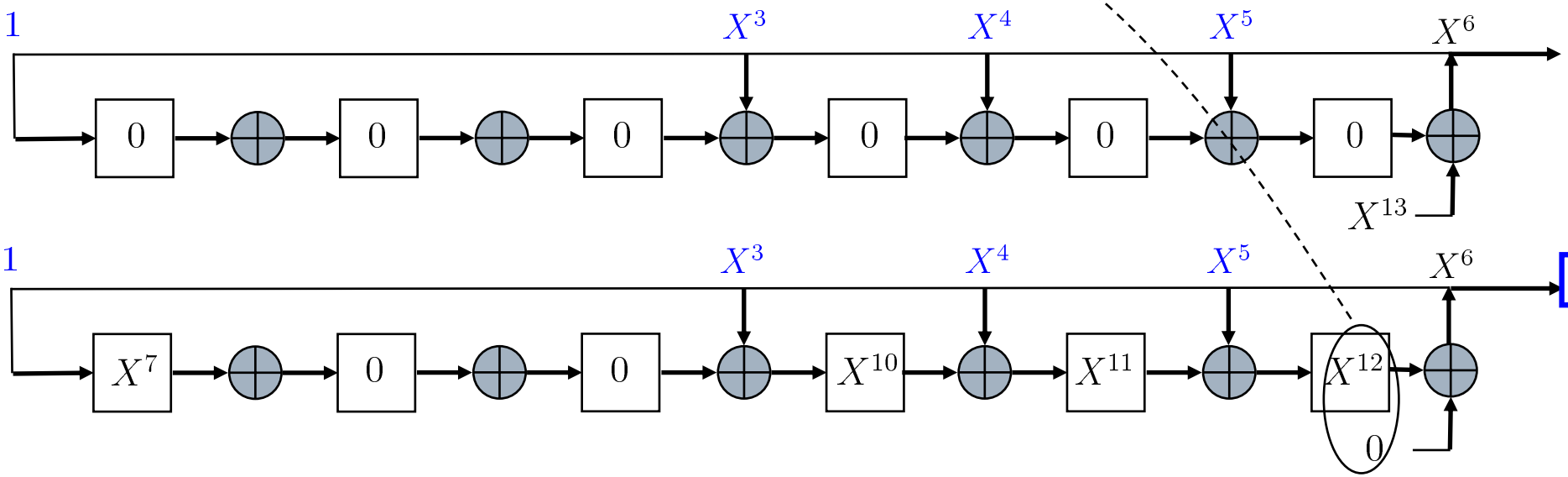
- An alternative realization of the long division (with the shift register containing not the remainder but the "lower power coefficients")

$$X^6+X^5+X^4+X^3+1 \;\Big|\; X^{13}+0+X^{11}+X^{10}+0+0+X^7+0+0+X^4+X^3+0+X+1$$

X^7

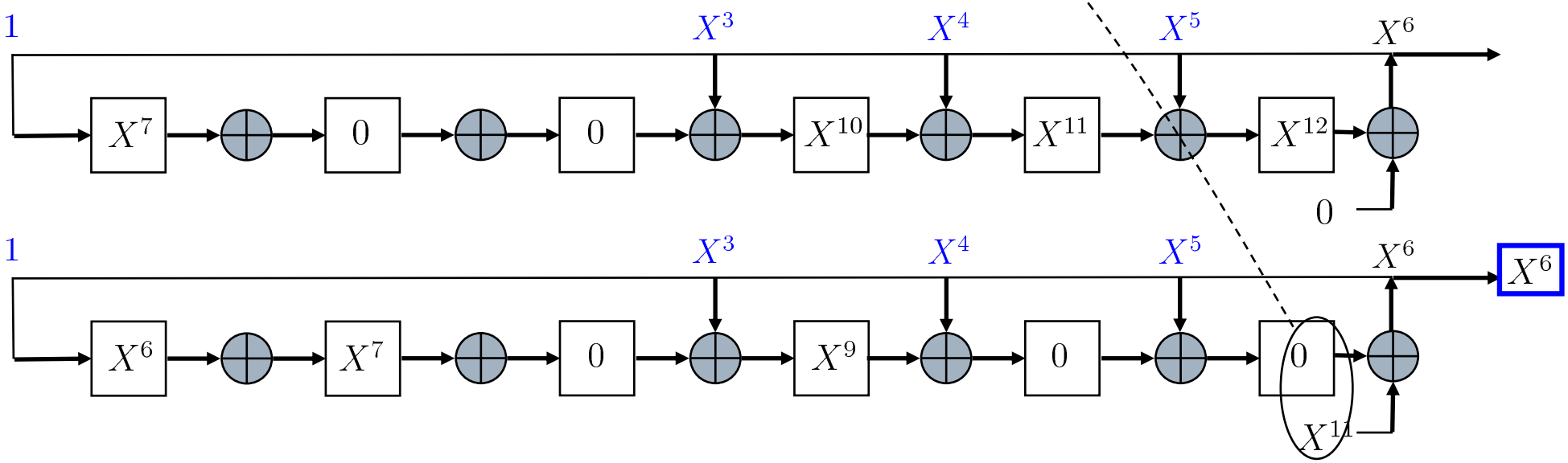
$$X^{13}+X^{12}+X^{11}+X^{10}+0+0+X^7$$

$$X^7(X^6+X^5+X^4+X^3+1)=X^{13}+X^{12}+X^{11}+X^{10}+0+0+X^7$$



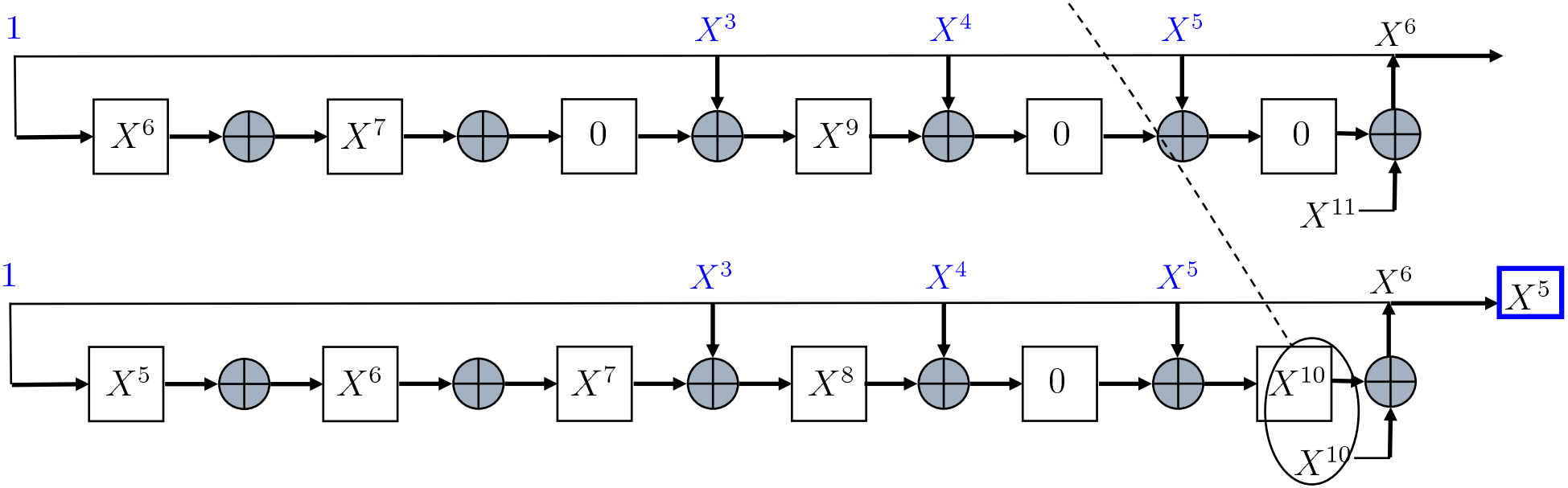
$$
\begin{array}{r|l}
 & X^7+X^6 \\
\hline
X^6+X^5+X^4+X^3+1 & X^{13}+0+X^{11}+X^{10}+0+0+X^7+0+0+X^4+X^3+0+X+1 \\
 & X^{13}+0+0+0+X^9+0+X^7+X^6
\end{array}
$$

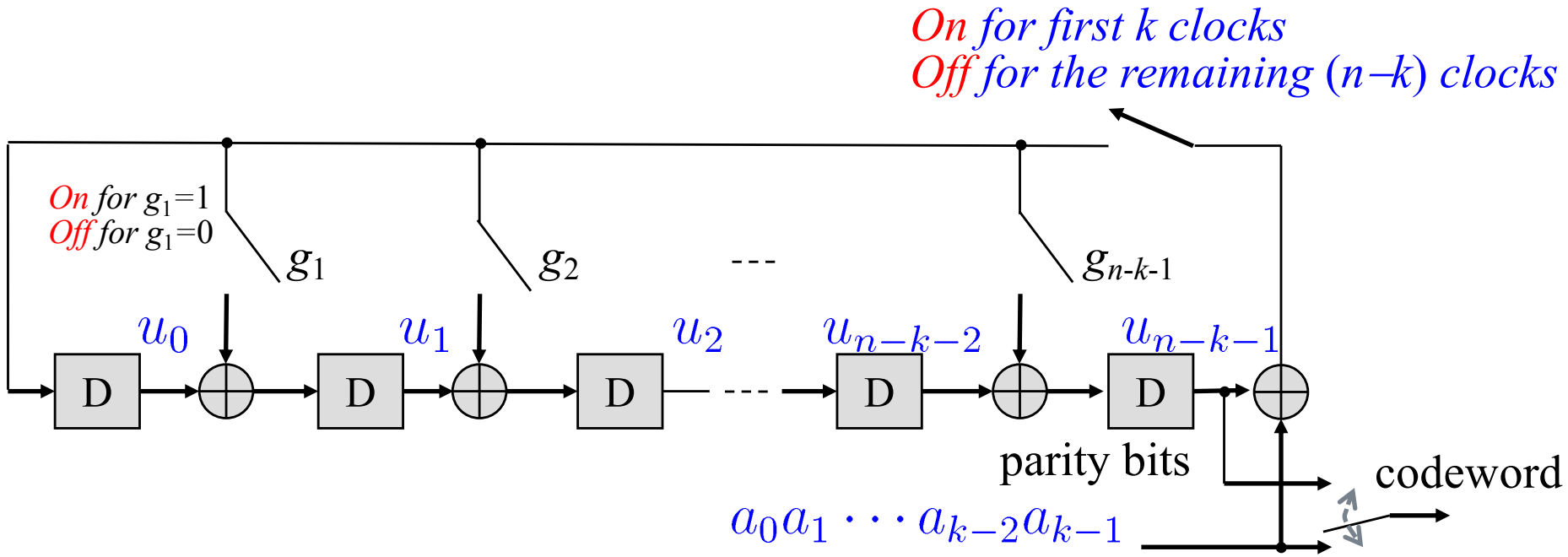
$$
(X^7+X^6)(X^6+X^5+X^4+X^3+1) = X^{13}+0+0+0+0+X^9+0+X^7+X^6
$$



$$
\begin{array}{r|l}
 & X^7+X^6+X^5 \\
\hline
X^6+X^5+X^4+X^3+1 & X^{13}+0+X^{11}+X^{10}+0+0+X^7+0+0+X^4+X^3+0+X+1 \\
 & X^{13}+0+X^{11}+X^{10}+0+X^8+X^7+X^6+X^5
\end{array}
$$

$$
(X^7+X^6+X^5)(X^6+X^5+X^4+X^3+1) = X^{13}+0+X^{11}+X^{10}+0+X^8+X^7+X^6+X^5
$$





$$\begin{aligned}\bar{c}(X) &= X^{n-k}a(X) + u(X) \\ &= \underbrace{X^{n-k}a(X)}_{\text{degrees } n-k \text{ to } n-1} + \underbrace{[X^{n-k}a(X) \bmod g(X)]}_{\text{degrees } 0 \text{ to } n-k-1}\end{aligned}$$

After k clocks,

$$\begin{aligned}q(X)g(X) &= \bar{c}(X) \\ &= a_{k-1}X^{n-1} + \cdots + a_0 X^{n-k} + \boxed{u_{n-k-1}X^{n-k-1} + \cdots + u_0}\end{aligned}$$

Notably, they are two "equivalent" polynomial coding systems
i) $c(X) = a(X)g(X)$: Not necessarily systematic
ii) $\bar{c}(X) = q(X)g(X)$: Systematic
Here we talk about the encoder for $\bar{c}(X)$.

Polynomial Codes/Cyclic Codes

- Decoder for polynomial codes
  - How to find the syndrome polynomial $s(X)$ of polynomial codes with respect to the received word polynomial $r(X)$?
    - Recall that syndromes are all symptoms that possibly happen at the output of parity-check examination.
    - If there is no error in transmission, $r(X)$ mod $g(X) = 0$. Indeed, $s(X) = r(X)$ mod $g(X)$.

$$X^{n-k}a(X) = q(X)g(X) + u(X)$$

Polynomial Codes/Cyclic Codes

- Relation of syndrome polynomial $s(X)$, error polynomial $e(X)$ and received vector polynomial $r(X)$ for **systematic** polynomial codes.

$$\begin{aligned} r(X) &= q_r(X)g(X) + s(X) \\ \text{Also, } r(X) &= \bar{c}(X) + e(X) \\ &\quad \text{where } e(X) \text{ is the error polynomial} \\ &= q(X)g(X) + e(X) \end{aligned}$$

$$\Rightarrow s(X) = r(X) \bmod g(X) = e(X) \bmod g(X)$$

$$\boxed{\Rightarrow e(X) = q_e(X)g(X) + s(X)}$$

Polynomial Codes/Cyclic Codes

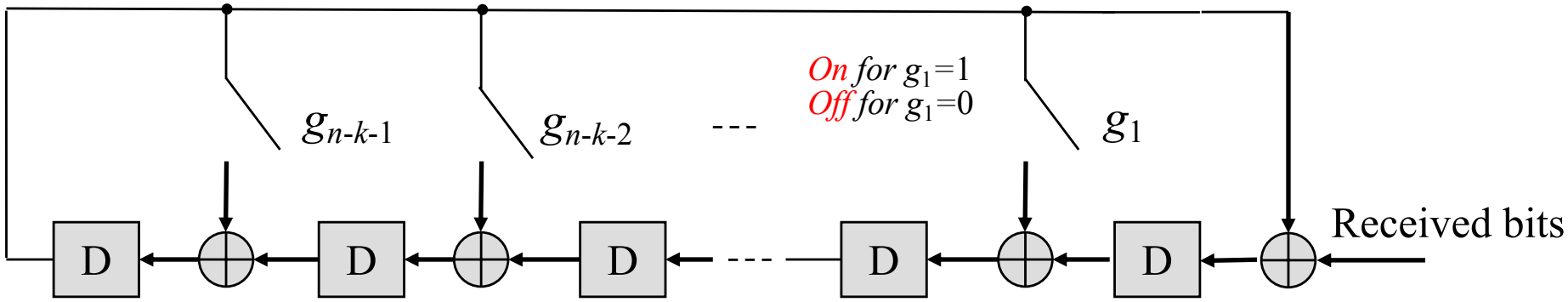
- Relation of syndrome polynomial $s(X)$, error polynomial $e(X)$ and received vector polynomial $r(X)$ for **general** polynomial codes.

$$\begin{aligned} r(X) &= q_r(X)g(X) + s(X) \\ \text{Also, } r(X) &= c(X) + e(X) \\ &\quad \text{where } e(X) \text{ is the error polynomial} \\ &= a(X)g(X) + e(X) \end{aligned}$$

$$\Rightarrow s(X) = r(X) \bmod g(X) = e(X) \bmod g(X)$$

$$\boxed{\Rightarrow e(X) = q_e(X)g(X) + s(X)}$$

Polynomial Codes/Cyclic Codes



Syndrome calculator

Polynomial Codes /Cyclic Codes

- Example of a (6, 3) polynomial code (Continue)

code for $g_1 = g_2 = 1$

$q_0q_1q_2$	$\bar{c}_0\bar{c}_1\bar{c}_2\bar{c}_3\bar{c}_4\bar{c}_5$	$a_0a_1a_2$	$c_0c_1c_2c_3c_4c_5$
000	000000	000	000000
011	010001	001	001111
110	100010	010	011110
101	110011	011	010001
100	111100	100	111100
111	101101	101	110011
010	011110	110	100010
001	001111	111	101101

encode *encode*

$$\bar{c}(X) = X^3(a_0 + a_1X + a_2X^2) - X^3(a_0 + a_1X + a_2X^2) \bmod (1 + g_1X + g_2X^2 + X^3)$$

$$\begin{bmatrix}\bar{c}_0 & \bar{c}_1 & \bar{c}_2 & \bar{c}_3 & \bar{c}_4 & \bar{c}_5\end{bmatrix} = \begin{bmatrix}a_0 & a_1 & a_2\end{bmatrix} \begin{bmatrix} 1 & g_1 & g_2 & 1 & 0 & 0 \\ g_2 & 1+g_1g_2 & g_1+g_2 & 0 & 1 & 0 \\ g_1+g_2 & g_1+g_2+g_1g_2 & 1+g_2 & 0 & 0 & 1 \end{bmatrix}$$

So, the polynomial code can be made equivalent to a systematic linear code.

$$\bar{\mathbf{G}}_{3\times 6}=\begin{bmatrix}1&1&1&1&0&0\\1&0&0&0&1&0\\0&1&0&0&0&1\end{bmatrix}\qquad \bar{\mathbf{H}}_{3\times 6}=\begin{bmatrix}1&0&0&1&1&0\\0&1&0&1&0&1\\0&0&1&1&0&0\end{bmatrix}\qquad \boldsymbol{s}_{1\times 3}=\boldsymbol{e}_{1\times 6}\bar{\mathbf{H}}_{6\times 3}^{T}$$

$e_0e_1e_2e_3e_4e_5$	$s_0s_1s_2$	$e_0e_1e_2e_3e_4e_5$	$s_0s_1s_2$	$e_0e_1e_2e_3e_4e_5$	$s_0s_1s_2$	$e_0e_1e_2e_3e_4e_5$	$s_0s_1s_2$
000000	000	010000	010	100000	100	110000	110
000001	010	010001	000	100001	110	110001	100
000010	100	010010	110	100010	000	110010	010
000011	110	010011	100	100011	010	110011	000
000100	111	010100	101	100100	011	110100	001
000101	101	010101	111	100101	001	110101	011
000110	011	010110	001	100110	111	110110	101
000111	001	010111	011	100111	101	110111	111
001000	001	011000	011	101000	101	111000	111
001001	011	011001	001	101001	111	111001	101
001010	101	011010	111	101010	001	111010	011
001011	111	011011	101	101011	011	111011	001
001100	110	011100	100	101100	010	111100	000
001101	100	011101	110	101101	000	111101	010
001110	010	011110	000	101110	110	111110	100
001111	000	011111	010	101111	100	111111	110

Polynomial Codes/Cyclic Codes

- The decoding of a systematic code is to simply add the coset leader, corresponding to the syndrome polynomial, to the received vector polynomial.

Polynomial Codes/Cyclic Codes

- Definition of cyclic codes
  - **Cyclic property**: Any cyclic shift of a codeword is also a codeword.
  - **Linearity property**: Any sum of two codewords is also a codeword.

- A cyclic code is also a polynomial code.
  - See Theorem 5.3 in Shu Lin and Daniel J. Costello, Jr, *Error Control Coding*, 2nd edition, Prentice Hall, 1983

Polynomial Codes/Cyclic Codes

- A cyclic code is a special type of polynomial codes with $g(X)$ dividing (X^n+1).
- Proof: Suppose $c(X)$ is a non-zero code polynomial.

Let i be the index satisfying

$$c_{n-1} = c_{n-2} = \cdots = c_{n-i+1} = 0 \text{ and } c_{n-i} = 1$$

Then

$$\begin{aligned} X^i c(X) &= X^i(c_0 + c_1 X + \cdots + c_{n-1}X^{n-1}) \\ &= c_0 X^i + c_1 X^{i+1} + \cdots + c_{n-1}X^{n+i-1} \\ &= c_{n-i} + c_{n-i+1}X + \cdots + c_{n-1}X^{i-1} \\ &\quad + c_0 X^i + c_1 X^{i+1} + \cdots + c_{n-i-1}X^{n-1} \\ &\quad + c_{n-i}(X^n+1) + c_{n-i+1}X(X^n+1) + \cdots + c_{n-1}X^{i-1}(X^n+1) \\ &= c^{(i)}(X) + (X^n+1) \end{aligned}$$

where $c^{(i)}(X)$ is a codeword due to cyclic property.

Because $c(X)$ and $c^{(i)}(X)$ are both code polynomials, there exist $a(X)$ and $a^{(i)}(X)$ satisfying

$$c(X) = a(X)g(X) \text{ and } c^{(i)}(X) = a^{(i)}(X)g(X).$$

Therefore,

$$\begin{aligned}(X^n + 1) &= X^i c(X) - c^{(i)}(X) \\ &= X^i a(X)g(x) - a^{(i)}(X)g(X) \\ &= [X^i a(X) - a^{(i)}(X)]g(X)\end{aligned}$$

$\Rightarrow g(X)$ must divide $(X^n + 1)$.

□

A cyclic codeword must contain at least two 1's, i.e., (100…000) cannot be a codeword of a cyclic code (except trivially $g(X) = 1$). Therefore, i in the above proof must be smaller than n.

A polynomial code is cyclic iff its generator polynomial divides X^n+1.

Polynomial Codes/Cyclic Codes

- Parity-check polynomial of cyclic codes
  - After proving that $g(X)$ must divide X^n+1, we can define the parity-check polynomial of a cyclic code as

$$g(X)h(X) \bmod (X^n + 1) = 0$$

where $h(X)$ is a polynomial of degree k with $h_0 = h_k = 1$.

  - Since the degrees of $g(X)$ and $h(X)$ are respectively $n-k$ and $k-1$, and $g_{n-k} = h_k = 1$, we may induce that

$$g(X)h(X) = X^n + 1$$

Polynomial Codes/Cyclic Codes

- Multiplying both sides by $a(X)$ yields:

$$\begin{aligned} a(X)g(X)h(X) &= c(X)h(X) \\ &= \underbrace{X^n a(X)}_{\text{degrees } n \text{ to } n+k-1} + \underbrace{a(X)}_{\text{degrees } 0 \text{ to } k-1} \end{aligned}$$

$\Rightarrow$ Coefficients of $c(X)h(X)$ equal zeros for degrees k to $n-1$

$$\Rightarrow \sum_{i=j}^{j+k} c_i h_{k+j-i} = 0 \text{ for } 0 \le j \le n-k-1$$

$$\Rightarrow \begin{bmatrix} c_0 & c_1 & \cdots & c_{n-1} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ h_{k-1} & 1 & 0 & \cdots & 0 \\ h_{k-2} & h_{k-1} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h_1 & h_2 & h_3 & \cdots & 0 \\ 1 & h_1 & h_2 & \cdots & 0 \\ 0 & 1 & h_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}_{n \times (n-k)}$$

$$= \quad \boldsymbol{c}_{1 \times n} \mathbf{H}^T_{n \times (n-k)} = \mathbf{0}_{1 \times (n-k)}$$

$$\Rightarrow [c_0\ c_1\ \cdots\ c_{n-1}] \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ h_{k-1} & 1 & 0 & \cdots & 0 \\ h_{k-2} & h_{k-1} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h_1 & h_2 & h_3 & \cdots & 0 \\ 1 & h_1 & h_2 & \cdots & 0 \\ 0 & 1 & h_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}_{n\times(n-k)}$$

$$= \quad \boldsymbol{c}_{1\times n}\mathbf{H}^T_{n\times(n-k)} = \mathbf{0}_{1\times(n-k)}$$

$$\mathbf{H}_{(n-k)\times n} = \begin{bmatrix} 1 & h_{k-1} & h_{k-2} & \cdots & h_1 & 1 & 0 & \cdots & 0 \\ 0 & 1 & h_{k-1} & \cdots & h_2 & h_1 & 1 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & h_3 & h_2 & h_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 1 \end{bmatrix}_{(n-k)\times n}$$

Recall that $\mathbf{G}_{k\times n} = \begin{bmatrix} 1 & g_1 & g_2 & \cdots & g_{n-k-1} & 1 & 0 & \cdots & 0 \\ 0 & 1 & g_1 & \cdots & g_{n-k-2} & g_{n-k-1} & 1 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & g_{n-k-3} & g_{n-k-2} & g_{n-k-1} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & \cdots & 0 & \cdots & g_{n-2k} & g_{n-2k+1} & \cdots & \cdots & 1 \end{bmatrix}_{k\times n}$

Also recall that $\mathbf{G}_{k\times n}\mathbf{H}^T_{n\times(n-k)} = \mathbf{0}_{k\times(n-k)}$

Observation.

The parity-check matrix arranges its entries according to the coefficients of the parity-check polynomial in ***reverse order*** as contrary to the generator matrix arranges its entries according to the coefficients of the generator polynomial.

Polynomial Codes/Cyclic Codes

- Remarks
  - The generator matrix and parity-check matrix derived previously are not for systematic codes.
  - We can manipulate these two matrices by adding their elements of selective rows so as to make them "systematic".

Polynomial Codes/Cyclic Codes

- Examples: Hamming code and Maximum-length code

$$X^7 + 1 = (X^3 + X^2 + 1)(X^3 + X + 1)(X + 1)$$

  - Irreducible polynomial: A polynomial that cannot be further factored.
    - All three of the above are irreducible.
  - Primitive polynomial: An irreducible polynomial of degree m, which divides $X^n + 1$ for $n = 2^m - 1$ but does not divides $X^n + 1$ for $n < 2^m - 1$.
    - All three irreducible polynomials are primitive.

Polynomial Codes/Cyclic Codes

- Example of cyclic codes: (7, 4, 3) Hamming code
  - Any cyclic code generated by a primitive polynomial is a Hamming code of minimum pairwise distance 3.
- Example of cyclic codes: $(2^m - 1, m, 2^{m-1})$ maximum-length code
  - The maximum-length code is a dual code of Hamming codes.
  - In other words, it is a cyclic code with primitive parity-check polynomial.
  - It is a code of minimum distance 2^{m-1} .

- It is named the *maximum-length code* because the codeword length for m information bits has been pushed to the *maximum (or equivalently, the number of codewords for code of length n has been pushed to the minimum)*. For example,

$$\text{if } (c_0, c_1, \ldots, c_6) \text{ is a codeword,}$$
$$\text{then } \begin{cases} (c_0, c_1, \ldots, c_6) \\ (c_1, c_2, \ldots, c_0) \\ (c_2, c_3, \ldots, c_1) \\ \vdots \\ (c_6, c_0, \ldots, c_5) \end{cases} \text{ are all codewords.}$$

So, there are in total 7 code words if $\boldsymbol{c}$ is originally nonzero. Adding the all-zero codeword gives $8 = 2^3$ codewords. This makes the (7, 3) maximum-length code.

- So, the nonzero codeword in a maximum-length code must be a circular shift of any other nonzero codeword.

- Example of cyclic codes: Cyclic redundancy check (CRC) codes
  - Cyclic codes are extremely well-suited for error detection owing to the *simplicity of its implementation*, and its superior *error-detection capability*.
  - Binary (n, k, $d_{\min}$) CRC codes can detect:
    - all contiguous error bursts of length $n-k$ or less.
    - $2^{n-k+1}-4$ of contiguous error bursts of length $n-k+1$
    - all combinations of $d_{\min}-1$ or fewer errors
    - all error patterns with an odd number of errors if the generator polynomial $g(X)$ has an even number of nonzero coefficients.

Code	*Generator Polynomial* $g(X)$	$n-k$
CRC-12 code	$1+X+X^2+X^3+X^{11}+X^{12}$	12
CRC-16 code	$1+X^2+X^{15}+X^{16}$	16
CRC-ITU code	$1+X^5+X^{12}+X^{16}$	16

- Example of cyclic codes: Bose-Chaudhuri-Hocquenghem (BCH) codes
  - A special type of BCH codes: Primitive (n, k, $d_{\min}$) BCH codes with parameters satisfying

$$\begin{cases} n & = & 2^m - 1 \\ k & \geq & n - mt \\ d_{\min} & \geq & 2t + 1 \\ m & \geq & 3 \\ t & \leq & (2^m - 1)/2 \end{cases}$$

  - (n, k, 3) Hamming code can be described as BCH codes.
  - The table in the next slide illustrates the generator polynomial $g(X)$ of some binary BCH codes.

n	k	t	Generator Polynomial								
7	4	1								1	011
15	11	1								10	011
15	7	2							111	010	001
15	5	3						10	100	110	111
31	26	1								100	101
31	21	2						11	101	101	001
31	16	3				1	000	111	110	101	111
31	11	5			101	100	010	011	011	010	101
31	6	7	11	001	011	011	110	101	000	100	111

n = block length

k = number of message bits

t = number of errors that are guaranteed correctable

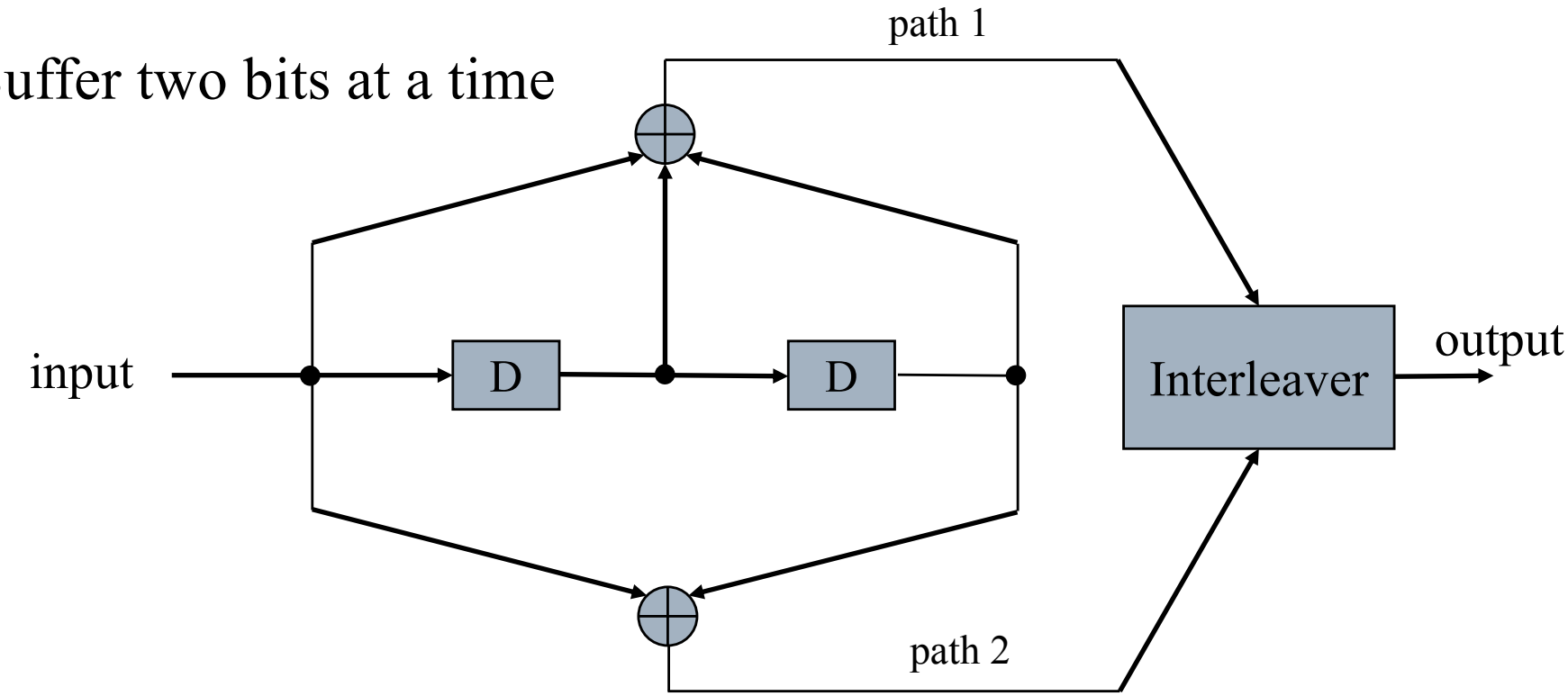
$$X^8 + X^7 + X^6 + X^4 + 1$$

Convolutional Codes

- For block codes, the encoding and decoding must perform in a block-by-block basis. Hence, a big buffer for the entire message block and codeword block is required.
- Instead, for the convolutional codes, since the inputs and outputs are governed through a convolution operation of "finite-order" filter, only a buffer of size equal to the filter order is demanded.
  - "Convolution" is defined based on modulo-2 arithmetic operations.

Convolutional Codes

Buffer two bits at a time



$(n, k, m) = (2, 1, 2)$ convolutional code

Note that here, n is not the codeword length, k is not the message length, and m is not the minimum distance between codewords. See the next slide for their definitions.

Convolutional Codes

- (n, k, m) convolutional codes
  - k input bits will produce n output bits.
  - The most recent m "k-message-bit input blocks" will be recorded (buffered).
  - The n output bits will be given by a linear combination of the buffered input bits.
  - Constraint length K of a convolutional code
    - It is the number of k-message-bit shifts, over which a single k-message-bit block influences the encoder output.
    - In other words, the encoder output depends on the current input message block and the previous $K - 1$ input message blocks.
    - Based on the definition, constraint length = m+1.

Convolutional Codes

- Effective code rate of a convolutional code
  - In practice, *km* zeros are often appended at the end of an information sequence to clear the shift register contents.
  - Hence, *kL* message bits will produce *n*(*L*+*m*) output bits.
  - The *effective code rate* is therefore given by:

$$\tilde{R} = \frac{kL}{n(L+m)}$$

  - Since L is often much larger than m, $\tilde{R} \approx k/n$.
    - k/n is named the *code rate* of a convolutional code.

Convolutional Codes

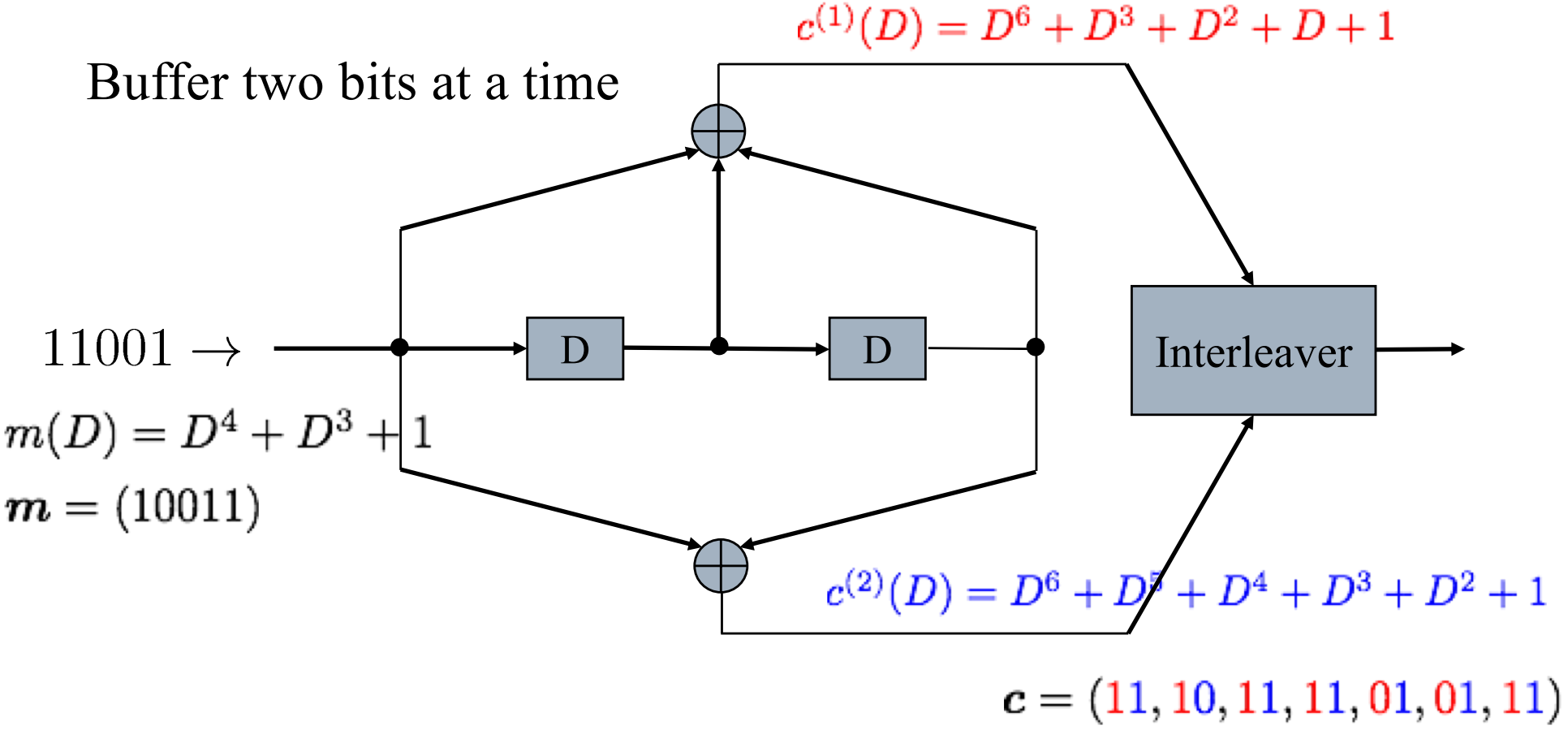
- Polynomial expression of convolutional codes
  - Example (Slide IDC 7-63)
    - D is used instead of X because the flip-flop (i.e., one time-unit delay) is often denoted by D.

$$g^{(1)}(D) = 1 + D + D^2$$
$$g^{(2)}(D) = 1 + D^2$$
$$m(D) = 1 + D^3 + D^4$$

$$\Rightarrow c^{(1)}(D) = g^{(1)}(D)m(D) = 1 + D + D^2 + D^3 + D^6$$
$$\Rightarrow c^{(2)}(D) = g^{(2)}(D)m(D) = 1 + D^2 + D^3 + D^4 + D^5 + D^6$$

Convolutional Codes

Buffer two bits at a time



Convolutional Codes

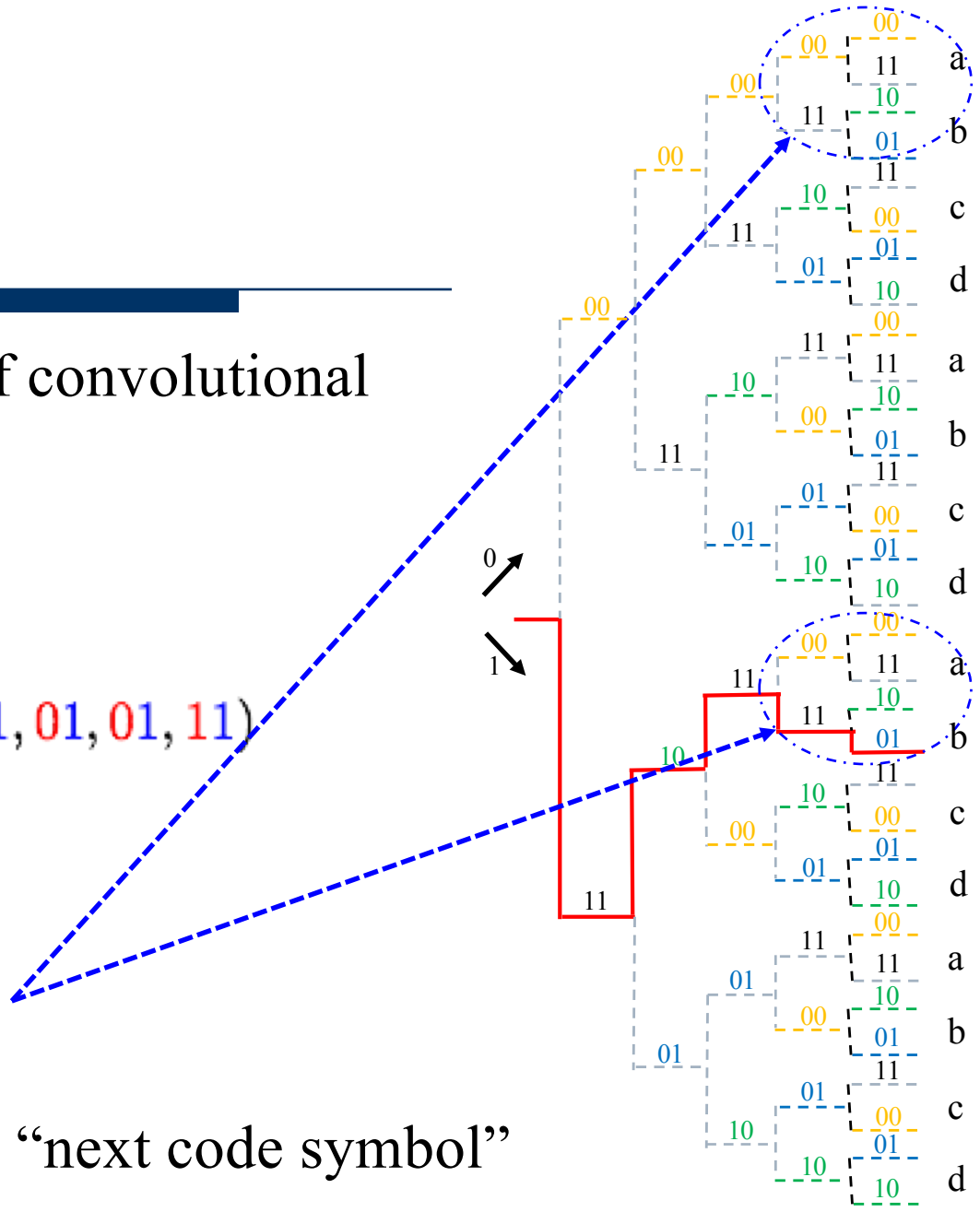
- Graphical expressions of convolutional codes
  - Code tree

$$\boldsymbol{m} = (10011)$$
$$\boldsymbol{c} = (11, 10, 11, 11, 01, 01, 11)$$

  - Code trellis

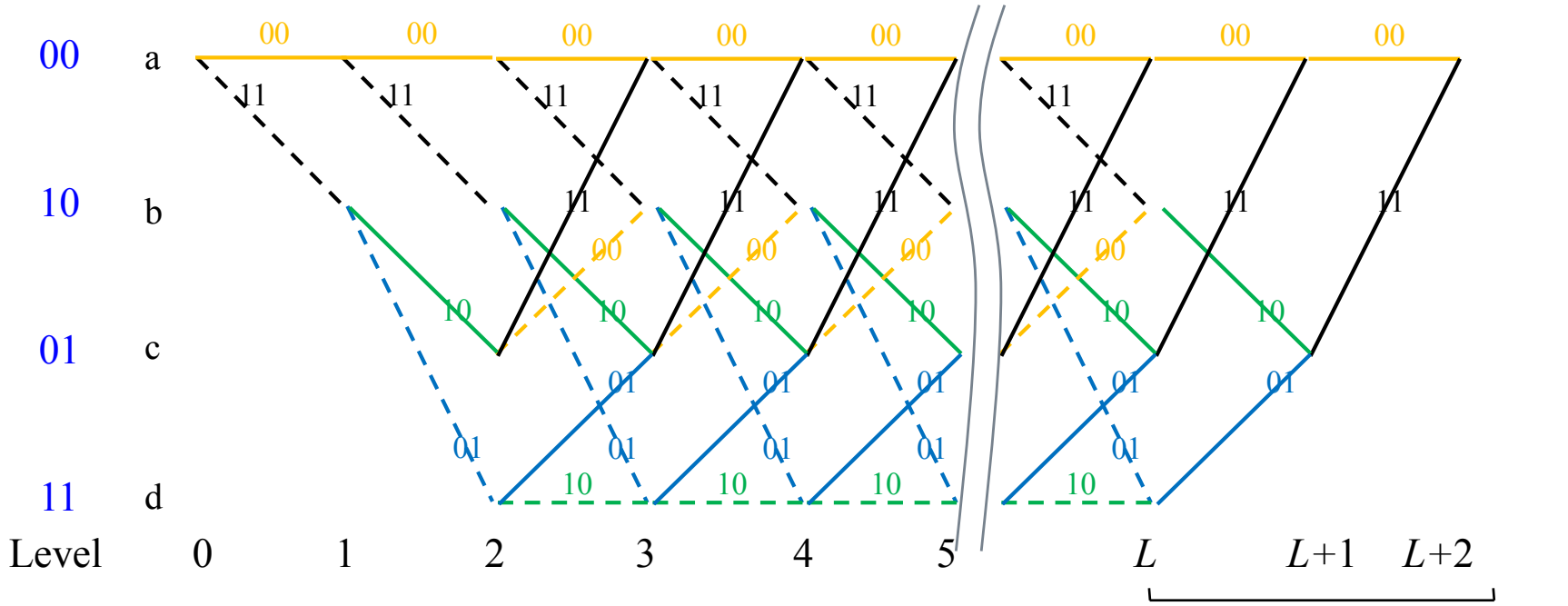
$$\begin{cases} (100m_3m_4\ldots) \\ (000m_3m_4\ldots) \end{cases}$$

generate the same "next code symbol"



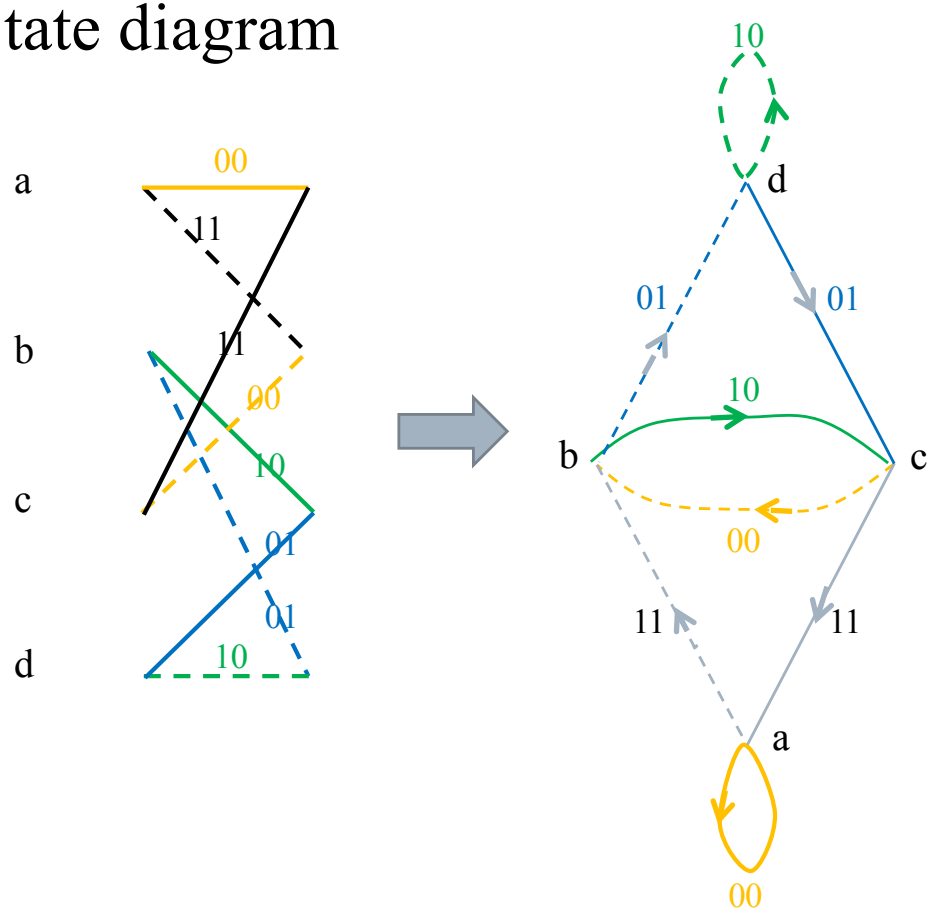
Convolutional Codes

- Code trellis (continue)



Convolutional Codes

- State diagram



Maximum Likelihood Decoding of Convolutional Codes

- Likelihood function (i.e., probability function)

$$p(\boldsymbol{r}|\boldsymbol{c}), \text{ where } \boldsymbol{r} = \boldsymbol{c} + \boldsymbol{n}.$$

- Maximum-likelihood decoding

$$\hat{\boldsymbol{c}} = \max_{\boldsymbol{c} \in \mathcal{C}} p(\boldsymbol{r}|\boldsymbol{c})$$

- For equal prior probability, ML decoding minimizes the error rate.

Maximum Likelihood Decoding of Convolutional Codes

- Minimum distance decoding
  - For an additive noise,

$$\begin{aligned}\hat{\boldsymbol{c}} &= \max_{\boldsymbol{c}\in\mathcal{C}} p(\boldsymbol{r}|\boldsymbol{c}) \\ &= \max_{\boldsymbol{c}\in\mathcal{C}} q(\boldsymbol{r}-\boldsymbol{c}), \text{ where } q \text{ is the distribution of } \boldsymbol{n} \\ &= \min_{\boldsymbol{c}\in\mathcal{C}} d(\boldsymbol{r},\boldsymbol{c})\end{aligned}$$

if $q(r-\boldsymbol{c})$ is a monotonely decreasing function of $d(\boldsymbol{r},\boldsymbol{c})$.

Maximum Likelihood Decoding of Convolutional Codes

- Example (of distance function): AWGN $\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{n}$

$$q(\boldsymbol{n}) = \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\left\{-\frac{||\boldsymbol{n}||^2}{2\sigma^2}\right\}$$

$$d(\boldsymbol{r}, \boldsymbol{c}) = ||\boldsymbol{r} - \boldsymbol{c}||^2$$

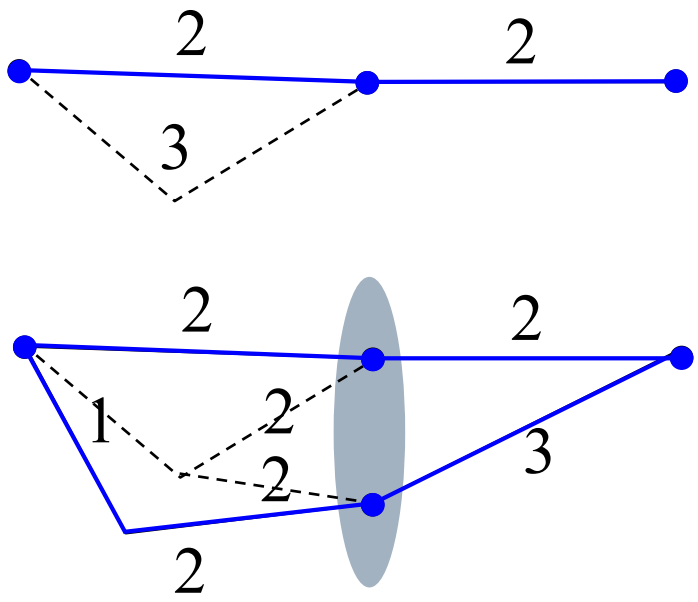
- Example (of distance function): BSC $\boldsymbol{r} = \boldsymbol{c} \oplus \boldsymbol{n}$

$$q(\boldsymbol{n}) = p^{w_H(\boldsymbol{n})}(1-p)^{N-w_H(\boldsymbol{n})}$$

$$d(\boldsymbol{r}, \boldsymbol{c}) = w_H(\boldsymbol{r} \oplus \boldsymbol{c})$$

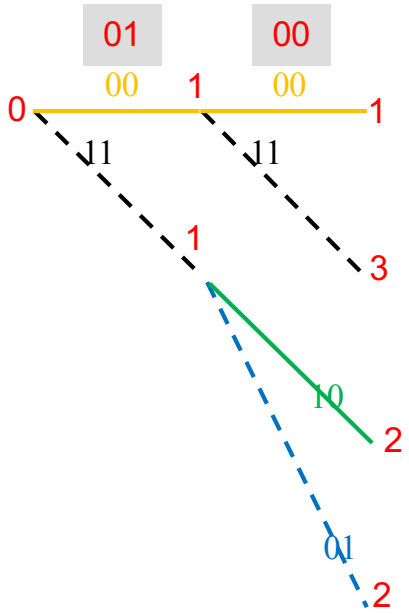
Maximum Likelihood Decoding of Convolutional Codes

- Viterbi algorithm (for minimum distance decoding over code trellis)
  - Optimality



One *survivor path (solid line)* for each intermediate node

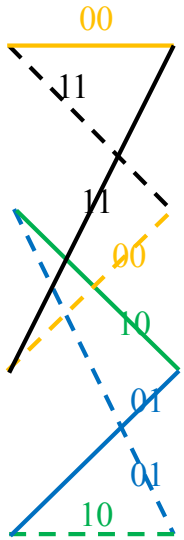
Received sequence



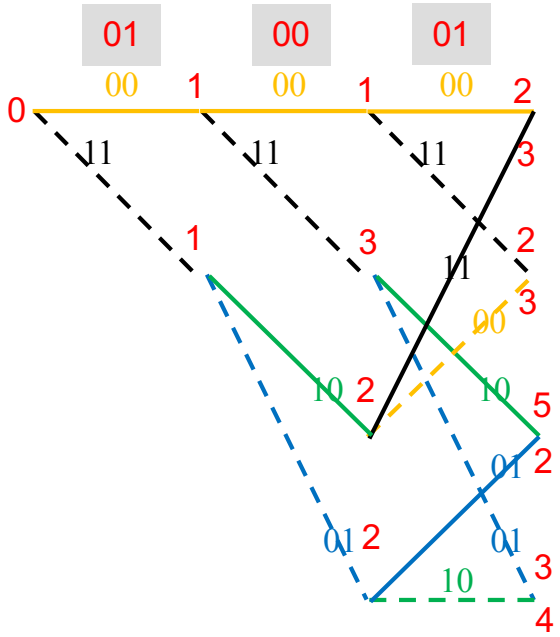
Assume that the all-zero sequence is transmitted.

$$d(\boldsymbol{r}, \boldsymbol{c}) = w_H(\boldsymbol{r} \oplus \boldsymbol{c})$$

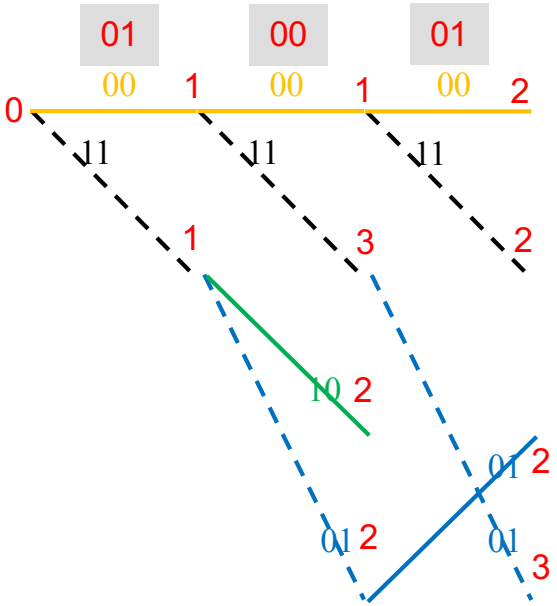
Solid line = information 0
Dashed line = information 1

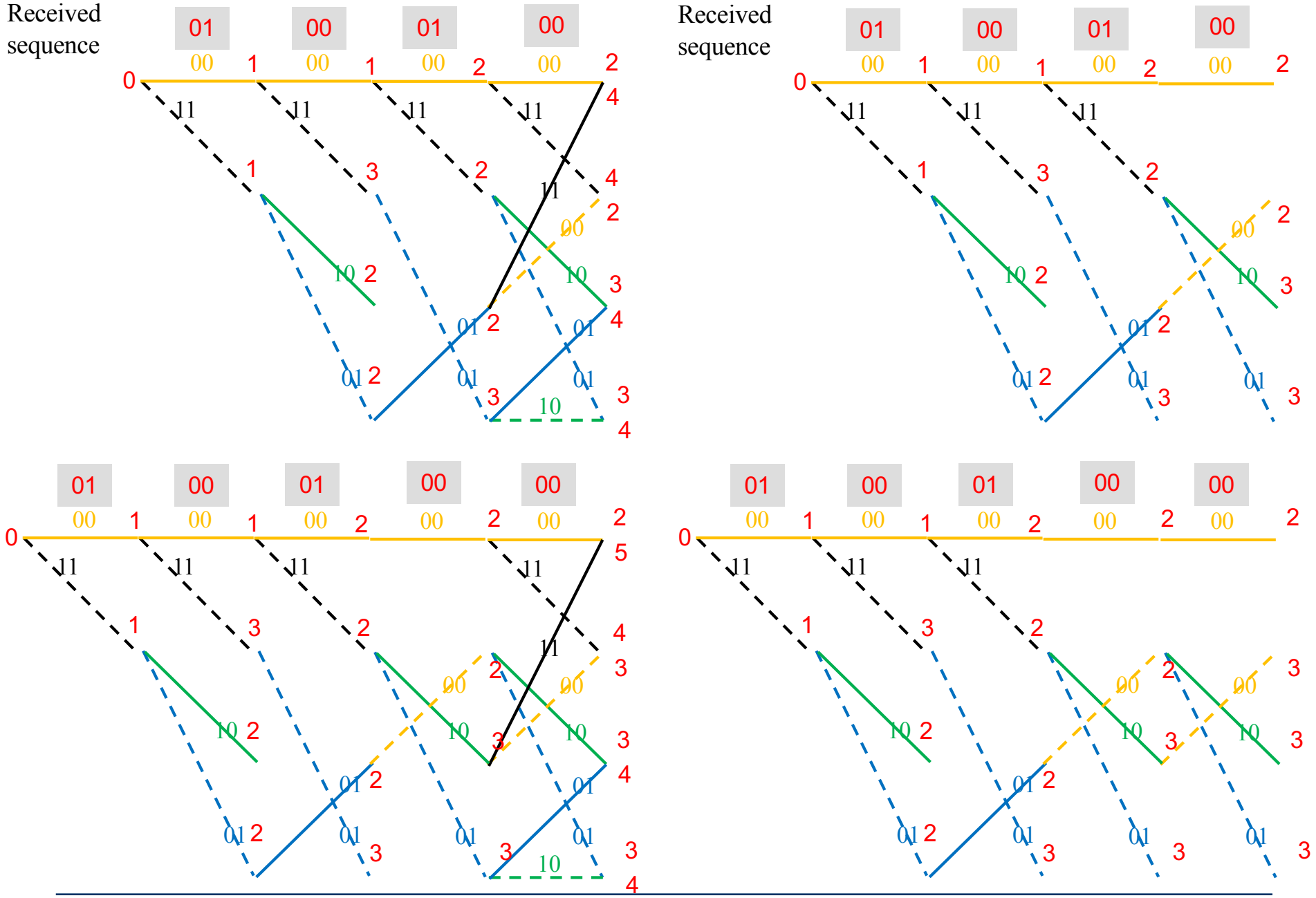


Received sequence



Received sequence



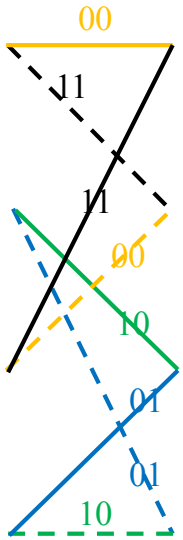


It is possible that the *ML codeword* or the *minimum distance codeword* is not equal to the *transmitted codeword*.

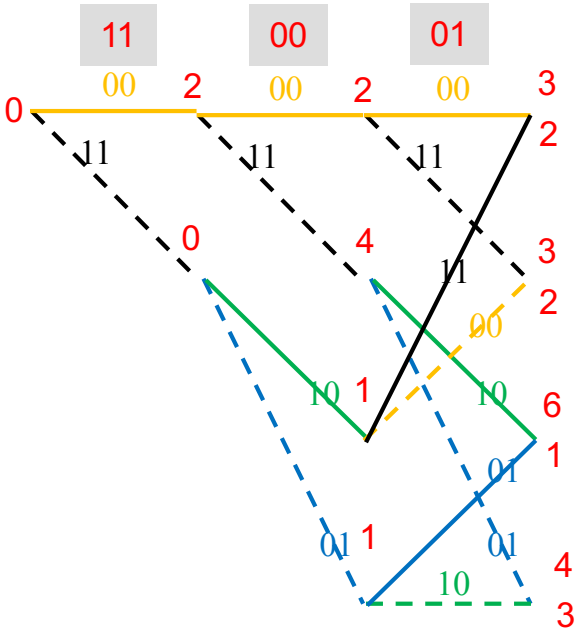
Assume that the all-zero sequence is transmitted.

$$d(\boldsymbol{r}, \boldsymbol{c}) = w_H(\boldsymbol{r} \oplus \boldsymbol{c})$$

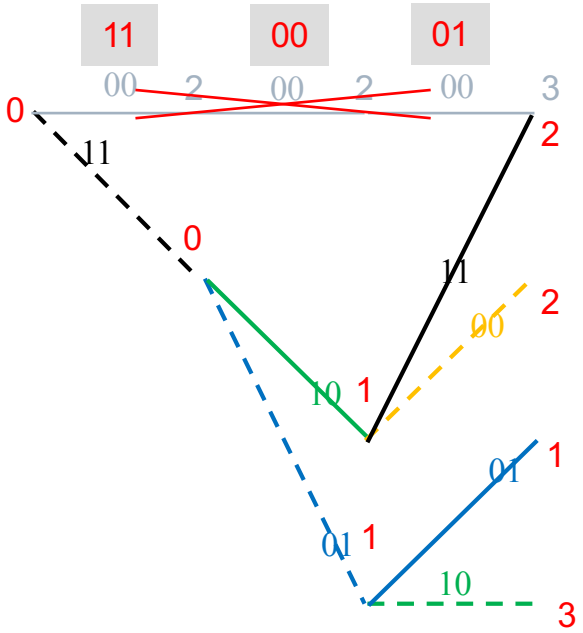
Solid line = information 0
Dashed line = information 1



Received sequence

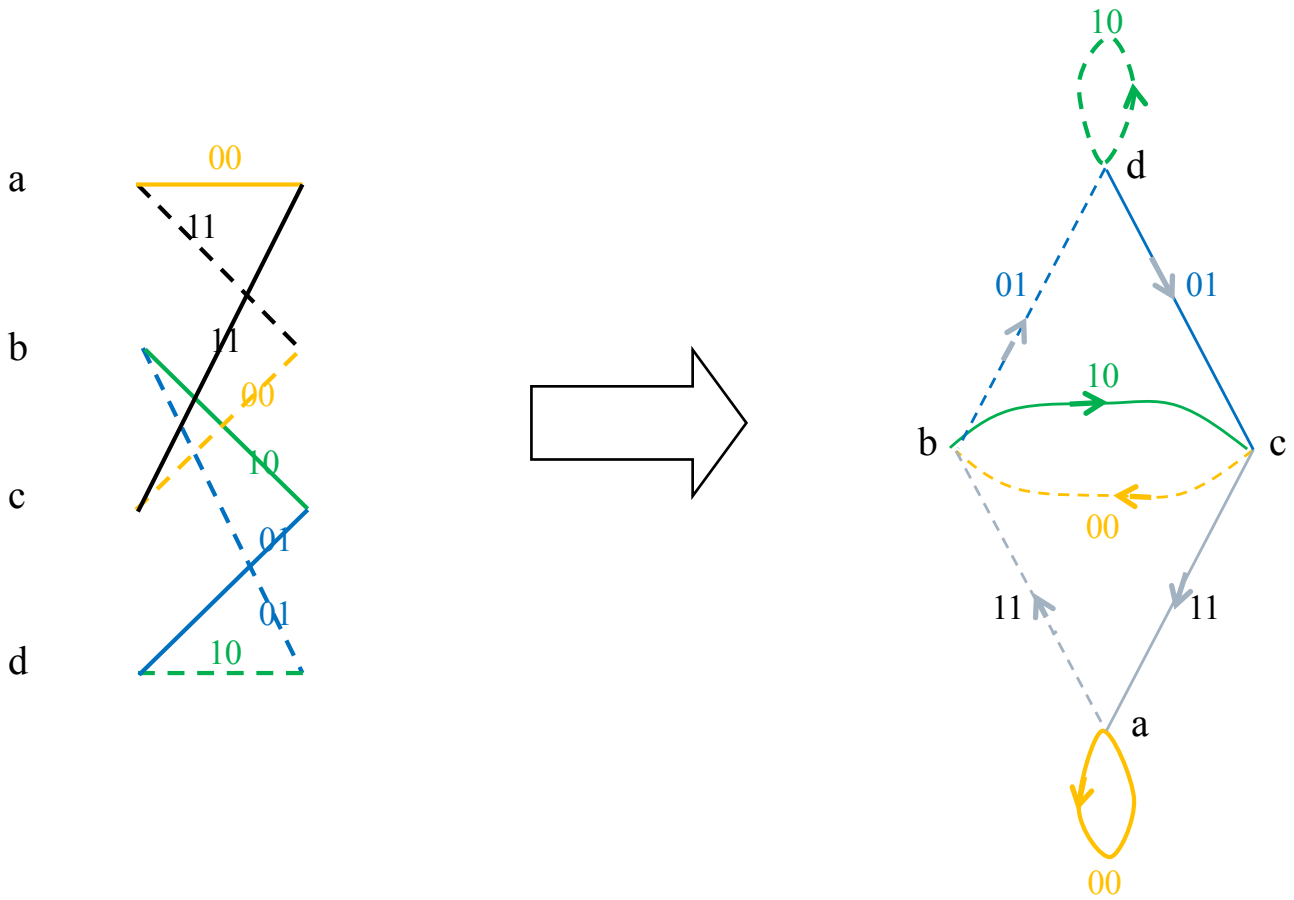


Received sequence



Maximum Likelihood Decoding of Convolutional Codes

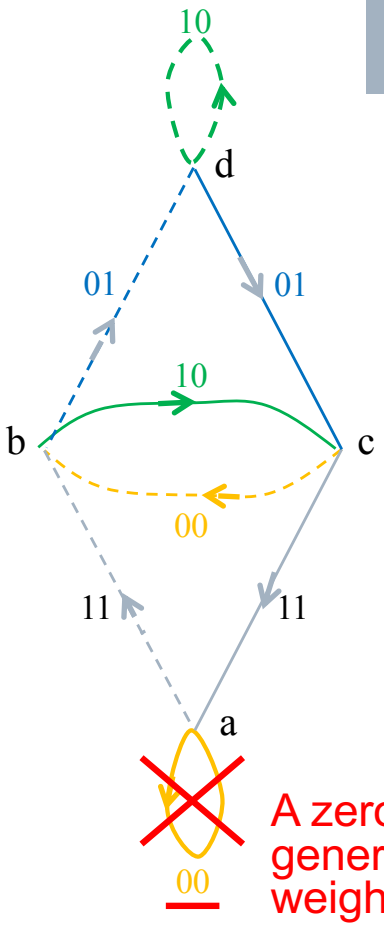
- Free distance of convolutional codes
  - Under binary symmetric channels (BSCs), a convolutional code with free distance d_{free} can correct t errors iff d_{free} is greater than $2t$.
  - Question: How to determine d_{free}?
  - Answer: By *signal-flow graph*.



State diagram

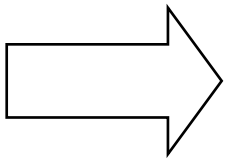
Example. $100 \rightarrow D^5L^3$ Input 100 generates a codeword of length 3 (branches) with weight 5.

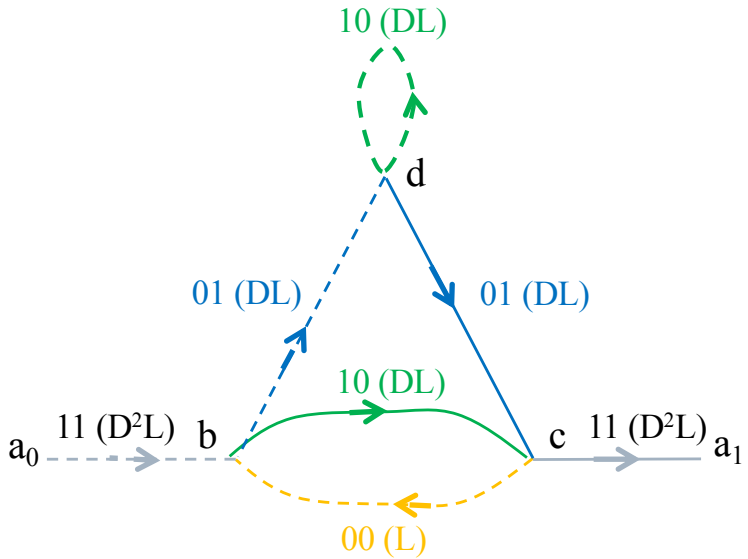
$10100 \rightarrow D^6L^5$ Input 10100 generates a codeword of length 5 (branches) with weight 6.



A zero-weight input generates a zero-weight code pattern.

State diagram





Signal graph

Exponent of D = Hamming weight on the branch

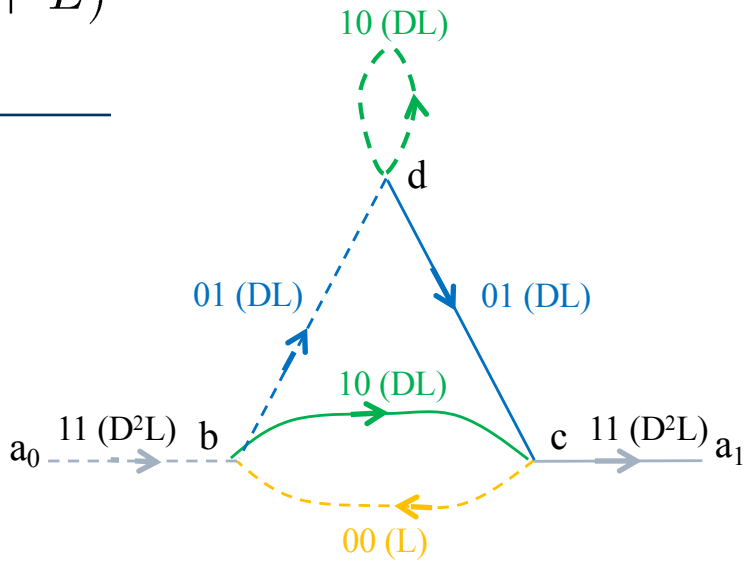
Exponent of L = Length of the branch

$$\left.\begin{aligned} b &= D^2L\cdot a_0 + L\cdot c \\ c &= DL\cdot b + DL\cdot d \\ d &= DL\cdot b + DL\cdot d \\ a_1 &= D^2L\cdot c \end{aligned}\right\} \Rightarrow \frac{a_1}{a_0} = \frac{D^5L^3}{1-DL(1+L)}$$ (See the next slide)

$$\frac{1}{1-x} = 1 + x + x^2 + x^3 + \cdots$$

$$\Rightarrow \frac{a_1}{a_0} = D^5L^3\sum_{i=0}^{\infty}[DL(1+L)]^i$$

$$= D^5L^3 + D^6(L^4 + L^5) + D^7(L^5 + 2L^6 + L^7) + \cdots$$



$100(a_0bca_1)(L^3) \rightarrow$ codeword of weight 5

$1100(a_0bdca_1)(L^4) \rightarrow$ codeword of weight 6

$10100(a_0bcbca_1)(L^5) \rightarrow$ codeword of weight 6

$$\left.\begin{aligned} b &= D^2L\cdot a_0 + L\cdot c \\ c &= DL\cdot b + DL\cdot d \\ d &= DL\cdot b + DL\cdot d \\ a_1 &= D^2L\cdot c \end{aligned}\right\} \Rightarrow \boxed{c=d} \Rightarrow \left\{\begin{aligned} b &= D^2L\cdot a_0 + L\cdot d \\ d &= DL\cdot b + DL\cdot d \\ a_1 &= D^2L\cdot d \end{aligned}\right.$$

$$\Rightarrow \left\{\begin{aligned} d &= DL\cdot(D^2L\cdot a_0 + L\cdot d) + DL\cdot d \\ a_1 &= D^2L\cdot d \end{aligned}\right.$$

$$\Rightarrow \left\{\begin{aligned} (1-DL-DL^2)d &= D^3L^2a_0 \quad (1) \\ a_1 &= D^2L\cdot d \quad (2) \end{aligned}\right.$$

$$\Rightarrow (1-DL-DL^2)a_1 d = D^5L^3a_0 d \qquad (1)\times(2)$$

$$\Rightarrow \frac{a_1}{a_0} = \frac{D^5L^3}{1-DL(1+L)}$$

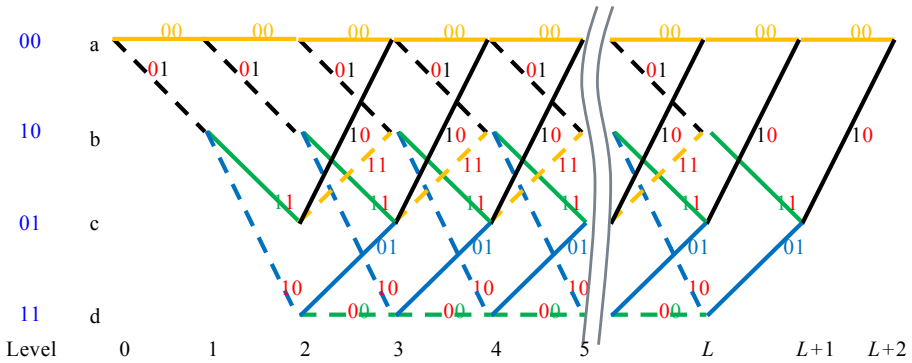
(Let $T(D,L) = \frac{a_1}{a_0}$.)

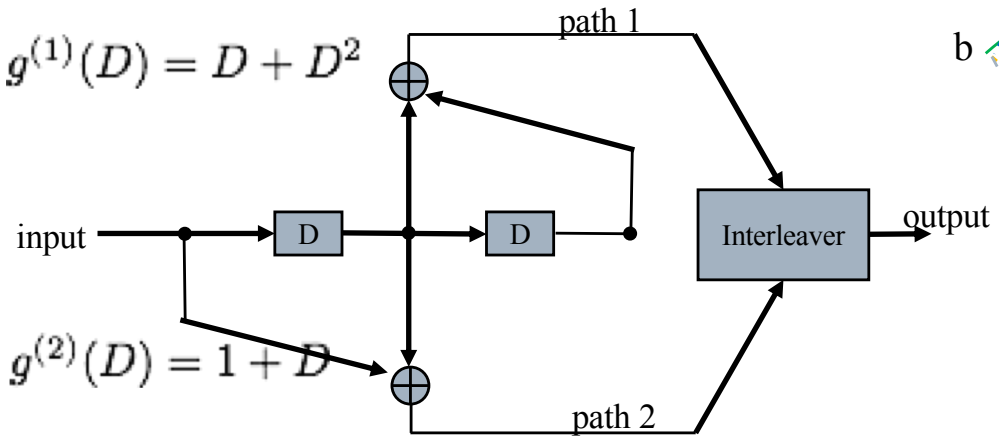
Maximum Likelihood Decoding of Convolutional Codes

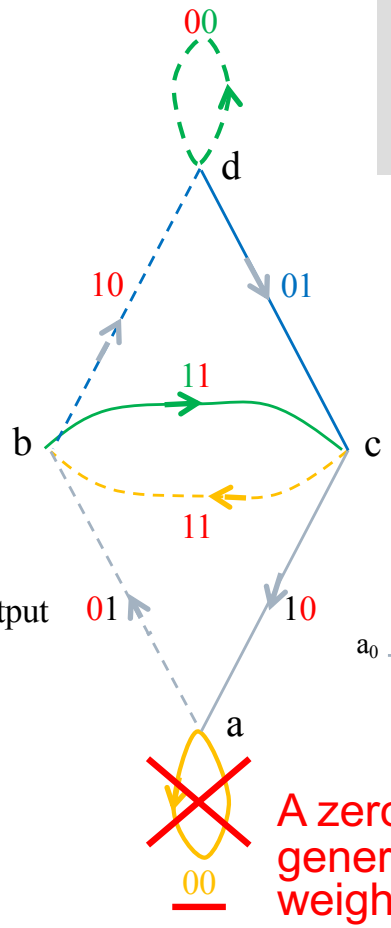
- Since the distance transfer function $T(D, 1) = D^5 + 2\,D^6 + 4\,D^7 + \ldots$ enumerates the number of codewords that are a given distance apart, it follows that

$$\Rightarrow d_{\text{free}} = 5 \text{ in the previous example.}$$

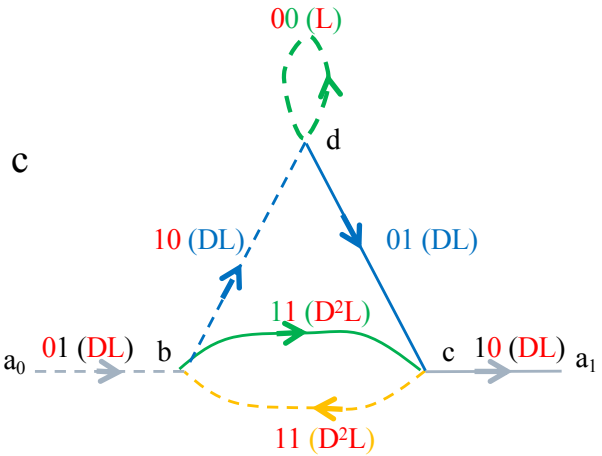
- A convolutional code may be subject to *catastrophic error propagation*.
  - *Catastrophic error propagation* = A finite number of transmission errors may cause infinite number of decoding errors.
  - A code with potential *catastrophic error propagation* is named a *catastrophic code*.







A non-zero-weight input generates a zero-weight code pattern.



A zero-weight input generates a zero-weight code pattern.

$$\boldsymbol{m} = (1111111\ldots)$$

$$m(D) = 1 + D + D^2 + D^3 + \cdots$$

$$\Rightarrow c^{(1)}(D) = g^{(1)}(D)m(D) = D$$

$$\Rightarrow c^{(2)}(D) = 1$$

$$\boldsymbol{c} = (01, 10, 00, 00, 00, 00, 00, \ldots)$$

The maximum-likelihood decoder will decode $\boldsymbol{r} = (00, 00, 00, 00, 00, 00, 00\ldots)$ as the all-zero codeword corresponding to $\boldsymbol{m} = (0000000\ldots)$ even if there are only two transmission errors!

$$\left.\begin{array}{rcl} b &=& DL\cdot a_0 + D^2L\cdot c \\ c &=& D^2L\cdot b + DL\cdot d \\ d &=& DL\cdot b + L\cdot d \\ a_1 &=& DL\cdot c \end{array}\right\} \Rightarrow \boxed{c = Dd} \Rightarrow \left\{\begin{array}{rcl} b &=& DL\cdot a_0 + D^3L\cdot d \\ d &=& DL\cdot b + L\cdot d \\ a_1 &=& D^2L\cdot d \end{array}\right.$$

$$\Rightarrow \left\{\begin{array}{rcl} DL\cdot b &=& D^2L^2\cdot a_0 + D^4L^2\cdot d \\ d &=& DL\cdot b + L\cdot d \\ a_1 &=& D^2L\cdot d \end{array}\right.$$

$$\Rightarrow \left\{\begin{array}{rcl} d &=& D^2L^2\cdot a_0 + D^4L^2\cdot d + L\cdot d \\ a_1 &=& D^2L\cdot d \end{array}\right.$$

$$\Rightarrow \left\{\begin{array}{rcll} (1-L-D^4L^2)d &=& D^2L^2\cdot a_0 & (1) \\ a_1 &=& D^2L\cdot d & (2) \end{array}\right.$$

$$\Rightarrow (1-L-D^4L^2)a_1 = D^4L^3a_0 \qquad (1)\times(2)$$

$$\Rightarrow T(D,L) = \frac{a_1}{a_0} = \frac{D^4L^3}{1-L-D^4L^2} \text{ and } T(D,1) = \frac{a_1}{a_0} = -1$$

- Alternative definition of catastrophic codes: A code for which an infinite weight input causes a finite weight output
  - In terms of the state diagram, a catastrophic code will have a loop corresponding to a nonzero input for which all the output bits are zeros.
- It can be proved that a systematic convolutional code cannot be catastrophic.
- Unfortunately, the free distance of systematic convolutional codes is usually smaller than that of nonsystematic convolutional codes.

Constraint length	Systematic	Nonsystematic
2	3	3
3	4	5
4	4	6
5	5	7
6	6	8
7	6	10
8	7	10

Maximum free distances attainable for convolutional codes of rate 1/2.

Maximum Likelihood Decoding of Convolutional Codes

- Asymptotic coding gain
  - Hard decision decoding
    - Section 6.3 (cf. Slide IDC 1-30) has established that for BPSK transmission, the hard-decision error for each code bit is given by:

$$p = P(\text{Error}) = \Phi\left(-\sqrt{2\frac{E}{N_0}}\right)$$

    - The error of convolutional codes (particularly at high SNR) is dominated by the "two codewords" whose pairwise Hamming distance is equal to d_{free}.
    - Thus, the (code)word error rate (WER) can be analyzed via an equivalent binary symmetric channel (BSC) with crossover probability p.

$\Rightarrow$ Equivalently $x_j = s_{j,m} \oplus w_j$ for $j = 1, \ldots, d_{\text{free}}$

where $s_{j,0} = 0, s_{j,1} = 1,$ and $x_j, w_j \in \{0, 1\}$

$\Rightarrow \quad \hat{m} = \arg\max \left\{ P\left(\boldsymbol{x} \,|\boldsymbol{s}_0\right), P\left(\boldsymbol{x} \,|\boldsymbol{s}_1\right) \right\}$

$\Rightarrow \quad \hat{m} = \arg\max \left\{ p^{w_H(\boldsymbol{x})}(1-p)^{d_{\text{free}} - w_H(\boldsymbol{x})}, (1-p)^{w_H(\boldsymbol{x})} p^{d_{\text{free}} - w_H(\boldsymbol{x})} \right\}$

$$\Rightarrow \quad w_H(\boldsymbol{x}) \underset{\boldsymbol{s}_1}{\overset{\boldsymbol{s}_0}{\lessgtr}} \frac{d_{\text{free}}}{2}, \text{ if } p < \frac{1}{2}$$

$\Rightarrow$ Dominant pairwise error

$$= \quad P\left(\boldsymbol{s}_0 \text{ transmitted}\right) P\left(w_H(\boldsymbol{x}) > \frac{d_{\text{free}}}{2} \middle| \boldsymbol{s}_0 \text{ transmitted} \right)$$

$$+P\left(\boldsymbol{s}_1 \text{ transmitted}\right) P\left(w_H(\boldsymbol{x}) < \frac{d_{\text{free}}}{2} \middle| \boldsymbol{s}_1 \text{ transmitted} \right)$$

(cf. the next slide)

$$\leq \quad [4p(1-p)]^{d_{\text{free}}/2} \leq (4p)^{d_{\text{free}}/2} = \left[4\Phi\left(-\sqrt{2\frac{E}{N_0}} \right) \right]^{d_{\text{free}}/2}$$

$$\leq \quad \exp\left\{ -d_{\text{free}} E/(2N_0) \right\} = \exp\left\{ -d_{\text{free}} R E_b/(2N_0) \right\}$$

$$4\Phi(-x) = 2\text{erfc}\left(\frac{x}{\sqrt{2}} \right) \leq \frac{4}{x\sqrt{2\pi}} e^{-x^2/2} \leq e^{-x^2/2} \text{ for } x > 4/\sqrt{2\pi}$$

For your information, assuming d_{free} is odd, we derive

$$
\begin{aligned}
&\Rightarrow \text{Dominant pairwise error} \\
&= P(\boldsymbol{s}_0 \text{ transmitted}) P\left(w_H(\boldsymbol{x}) > \frac{d_{\text{free}}}{2} \middle| \boldsymbol{s}_0 \text{ transmitted} \right) \\
&\quad + P(\boldsymbol{s}_1 \text{ transmitted}) P\left(w_H(\boldsymbol{x}) < \frac{d_{\text{free}}}{2} \middle| \boldsymbol{s}_1 \text{ transmitted} \right) \\
&= \Pr\left[W_1 + W_2 + \cdots + W_{d_{\text{free}}} > \frac{d_{\text{free}}}{2} \right], \\
&\quad \text{where } \{W_j\} \text{ i.i.d. with } P(W_j = 1) = 1 - P(W_j = 0) = p \\
&= \Pr\left[e^{\theta(W_1 + W_2 + \cdots + W_{d_{\text{free}}})} > e^{\theta d_{\text{free}}/2} \right], \text{ where } e^{\theta} = (1-p)/p \\
&\leq \frac{\left(E\left[e^{\theta W_1}\right]\right)^{d_{\text{free}}}}{e^{\theta d_{\text{free}}/2}} = \frac{\left(pe^{\theta} + 1 - p\right)^{d_{\text{free}}}}{e^{\theta d_{\text{free}}/2}} \\
&= \left[4p(1-p)\right]^{d_{\text{free}}/2}
\end{aligned}
$$

- Soft decision decoding (can be analyzed via an equivalent binary-input additive white Gaussian noise channel)
  - WER of convolutional codes (particularly at high SNR) is dominated by the "two codewords" whose pairwise Hamming distance is equal to d_{free}.

$\Rightarrow$ Equivalently $x_j = s_{j,m} + w_j$ for $j = 1, \ldots, d_{\text{free}}$ where $s_{j,0} = -\sqrt{E}$ and $s_{j,1} = \sqrt{E}$

$$\Rightarrow \quad \hat{m} = \arg\max \left\{ P\left(\boldsymbol{x} \,|\boldsymbol{s}_0\right), P\left(\boldsymbol{x} \,|\boldsymbol{s}_1\right) \right\}$$

$$\Rightarrow \quad \hat{m} = \arg\max \left\{ \prod_{j=1}^{d_{\text{free}}} e^{-(x_j+\sqrt{E})^2/2\sigma^2}, \prod_{j=1}^{d_{\text{free}}} e^{-(x_j-\sqrt{E})^2/2\sigma^2} \right\}$$

$$\Rightarrow \quad x = \sum_{j=1}^{d_{\text{free}}} x_j \underset{\boldsymbol{s}_1}{\overset{\boldsymbol{s}_0}{\lessgtr}} 0$$

$$x \begin{cases} \mathcal{N}(-d_{\text{free}}\sqrt{E}, d_{\text{free}}\sigma^2) & \boldsymbol{s}_0 \\ \mathcal{N}(d_{\text{free}}\sqrt{E}, d_{\text{free}}\sigma^2) & \boldsymbol{s}_1 \end{cases}$$

$\sigma^2 = N_0/2$ is the variance of w_j

- Based on the decision rule $x = \sum_{j=1}^{d_{\text{free}}} x_j \underset{\boldsymbol{s}_1}{\overset{\boldsymbol{s}_0}{\lessgtr}} 0$

$$
\begin{aligned}
\text{Dominant pairwise error} &= P\left(\boldsymbol{s}_0 \text{ transmitted}\right) P\left(x > 0 \,|\boldsymbol{s}_0 \text{ transmitted}\right) \\
&\quad + P\left(\boldsymbol{s}_1 \text{ transmitted}\right) P\left(x < 0 \,|\boldsymbol{s}_1 \text{ transmitted}\right) \\
&= \frac{1}{2}\int_0^{\infty} \frac{1}{\sqrt{2\pi d_{\text{free}}\sigma^2}} e^{-(x+d_{\text{free}}\sqrt{E})^2/2d_{\text{free}}\sigma^2} dx \\
&\quad + \frac{1}{2}\int_{-\infty}^{0} \frac{1}{\sqrt{2\pi d_{\text{free}}\sigma^2}} e^{-(x-d_{\text{free}}\sqrt{E})^2/2d_{\text{free}}\sigma^2} dx \\
&= \int_{-\infty}^{0} \frac{1}{\sqrt{2\pi d_{\text{free}}\sigma^2}} e^{-(x-d_{\text{free}}\sqrt{E})^2/2d_{\text{free}}\sigma^2} dx \\
&= \Phi\left(\frac{0 - d_{\text{free}}\sqrt{E}}{\sqrt{d_{\text{free}}\sigma^2}}\right) = \Phi\left(-\sqrt{2d_{\text{free}}\frac{E}{N_0}}\right) \\
&= \Phi\left(-\sqrt{2d_{\text{free}}R\frac{E_b}{N_0}}\right) \le \exp\left\{-d_{\text{free}}RE_b/N_0\right\}
\end{aligned}
$$

$$
\Phi(-x) = \frac{1}{2}\text{erfc}\left(\frac{x}{\sqrt{2}}\right) \le \frac{1}{x\sqrt{2\pi}} e^{-x^2/2} \le e^{-x^2/2} \text{ for } x > 1/\sqrt{2\pi}
$$

- Asymptotic coding gain (here, asymptotic = at high SNR) G_a
  - The performance gain due to coding (i.e., the performance gain of a coded system against an uncoded system)

$$\text{Uncoded BPSK } \Phi\left(-\sqrt{2\frac{E_b}{N_0}}\right) \leq \exp\left\{-\frac{E_b}{N_0}\right\}$$

Coded system	$\exp\left\{-G_a\frac{E_b}{N_0}\right\}$
Convolutional coded BPSK with hard-decision decoding	$\exp\left\{-\left(\frac{d_{\text{free}}R}{2}\right)\frac{E_b}{N_0}\right\}$
Convolutional coded BPSK with soft-decision decoding	$\exp\left\{-d_{\text{free}}R\frac{E_b}{N_0}\right\}$

Asymptotic coding gain (at high SNR)

Convolutional coded BPSK with hard-decision decoding

$$G_a = \frac{d_{\text{free}}R}{2} = 10\log_{10}\left(\frac{d_{\text{free}}R}{2}\right) \text{ dB}$$

Convolutional coded BPSK with soft-decision decoding

$$G_a = d_{\text{free}}R = 10\log_{10}(d_{\text{free}}R) \text{ dB}$$

$$\text{Asymptotic coding gain} = \frac{\left(\frac{E_b}{N_0}\right)_{\text{uncoded}}}{\left(\frac{E_b}{N_0}\right)_{\text{coded}}} \text{ under the same error rate}$$

$$\exp\left\{-\left(\frac{E_b}{N_0}\right)_{\text{uncoded}}\right\} \approx P_e \approx \exp\left\{-G_a\left(\frac{E_b}{N_0}\right)_{\text{coded}}\right\}$$

- Asymptotic coding gain (at high SNR)