การสื่อสารดิจิทัล

การเข้ารหัสช่องสัญญาณ (14 - 15)

Assoc.Prof.**Piya Kovintavewat**, Ph.D.

Data Storage Technology Research Center

Nakhon Pathom Rajabhat University

http://home.npru.ac.th/piya

"Procrastination is the thief of time"

Outline

- รหัสบล็อกเชิงเส้น
  - เมทริกซ์ตัวกำเนิด / เมทริกซ์พาริตีเช็ก
  - ระยะทางน้อยสุดของรหัส
  - การถอดรหัสบล็อกเชิงเส้น
  - การสร้างวงจรเข้ารหัสและถอดรหัส
- รหัสวน
  - การเข้ารหัส
  - เมทริกซ์ตัวกำเนิด / เมทริกซ์พาริตีเช็ก
  - การสร้างวงจรเข้ารหัส / การถอดรหัส
- รหัสแฮมมิง
- รหัสคอนโวลูชัน
  - การเข้ารหัส / การถอดรหัส
- รหัสช่องสัญญาณที่น่าสนใจ

บทนำ

- การเข้ารหัสแหล่งต้นทาง ⇒ กำจัดความซ้ำซ้อนของข้อมูลก่อนส่ง ซึ่งจะช่วยลดเวลาในการส่งข่าวสารและลดปริมาณการใช้แบนด์วิดท์ของตัวกลาง
- **การเข้ารหัสช่องสัญญาณ (channel coding)** ⇒ เพื่อป้องกันผลกระทบที่เกิดจากความบกพร่องของช่องสัญญาณ เช่น สัญญาณรบกวน การแทรกสอด และการจางหาย
- **รหัสช่องสัญญาณ** (channel code) จะนิยมนำมาใช้ในระบบสื่อสารดิจิทัลต่างๆ เช่น ระบบสื่อสารไร้สาย ระบบสื่อสารดาวเทียม และระบบการบันทึกข้อมูล เป็นต้น เพราะสามารถช่วยเพิ่มสมรรถนะรวมของระบบได้มากกว่าระบบที่ไม่ได้ใช้การเข้ารหัสช่องสัญญาณ

- การเข้ารหัสช่องสัญญาณแบ่งเป็น 2 แบบคือ
  - การเข้ารหัสรูปคลื่น (waveform coding) ⇒ การออกแบบสัญญาณโดยทำการเปลี่ยนเซตของรูปคลื่นสัญญาณที่ต้องการส่งไปยังแหล่งปลายทางให้เป็นเซตของรูปคลื่นสัญญาณใหม่ที่ง่ายต่อการตรวจหาที่วงจรภาครับและทนทานต่อสัญญาณรบกวน เช่น antipodal signaling, M-ary signaling, orthogonal signaling, และ trellis-coded modulation เป็นต้น
  - ลำดับเชิงโครงสร้าง (structured sequence) ⇒ เพิ่ม**บิตเกิน** (redundant bits) เข้าไปใน**ลำดับบิต**เดิมที่ต้องการส่งไปยังแหล่งปลายทางให้เป็นลำดับบิตใหม่ ⇒ ช่วยวงจรถอดรหัสในการตรวจหาและแก้ไขข้อผิดพลาด
- บทนี้อธิบายเฉพาะการเข้ารหัสช่องสัญญาณแบบลำดับเชิงโครงสร้างเท่านั้น และเรียกสั้นๆ ว่า
  - **รหัสแก้ไขข้อผิดพลาดข้างหน้า** (forward error correction code) หรือ
  - **รหัสแก้ไขข้อผิดพลาด** (ECC: error correction code)

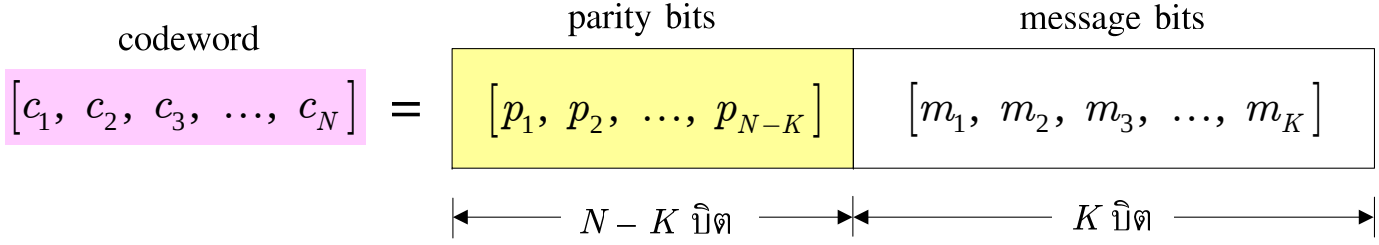
- รหัส ECC ⇒ รหัสบล็อก และรหัสคอนโวลูชัน
- **รหัสบล็อก** ⇒ เข้าและถอดรหัสทีละบล็อก บล็อกละหลายๆ บิต (ตามข้อกำหนดของแต่ละรหัส) โดยอาศัยความรู้ทางด้านฟิลด์จำกัด (finite field) และพีชคณิตนามธรรม (abstract algebra) ในการเข้าและถอดรหัส
  - เช่น รหัสบล็อกเชิงเส้น (linear block code), รหัสวน (cyclic code), รหัส RS, รหัส BCH (Bose-Chaudhuri-Hocquenghen), รหัสโกเลย์ (Golay code), และรหัสแฮมมิง (Hamming code) เป็นต้น
  - รหัสแฮมมิงแบบ (7,4) ถือเป็นรหัส ECC แบบแรกของโลกที่สร้างขึ้นโดยนาย Richard Hamming ในปี ค.ศ. 1950 ซึ่งจะเข้ารหัสข้อมูล 4 บิตและให้บิตพาริตี (parity bit) 3 บิต และสามารถตรวจหาข้อผิดพลาดได้สูงสุดจำนวน 2 บิตและแก้ไขข้อผิดพลาดได้ 1 บิต
- รหัสคอนโวลูชัน ⇒ เข้าและถอดรหัสทีละบิตต่อเนื่องกันไปเรื่อยๆ ในรูปของกระแสบิต (bit stream) จึงทำให้สามารถทำงานแบบเวลาจริงได้
  - สามารถถอดรหัสข้อมูลทีละบิตออกมาใช้งานได้ทันที
  - การถอดรหัสข้อมูลที่ผ่านการเข้ารหัสคอนโวลูชันจะใช้อัลกอริทึมวีเทอร์บิ (Viterbi algorithm)

รหัสบล็อกเชิงเส้น

- รหัสบล็อกเชิงเส้นแบบ (N, K) คือรหัสช่องสัญญาณที่แปลงบิตข้อความ (message bit) จำนวน K บิตให้เป็นคำรหัสขนาด N บิต ถ้าให้ $\mathbf{m} = [m_1, m_2, \ldots, m_K]$ คือเวกเตอร์ข้อมูลที่จะถูกเข้ารหัสเพื่อให้ได้เป็นคำรหัส $\mathbf{c} = [c_1, c_2, \ldots, c_N]$
- บิตเกินที่เพิ่มขึ้นมาจำนวน $N - K$ บิต เรียกว่า**บิตพาริตี** $\Rightarrow$ ช่วยให้วงจรภาครับสามารถตรวจหาข้อผิดพลาดได้ และถ้าบิตพาริตีมีจำนวนมากพอ ก็อาจทำให้สามารถแก้ไขข้อผิดพลาดของข้อมูลให้ถูกต้องได้ด้วย
- รหัสบล็อกเชิงเส้นจะทำการเข้าและถอดรหัสข้อมูลทีละบล็อก โดยอัตราส่วนของจำนวนบิตข้อความต่อจำนวนบิตของคำรหัสจะเรียกว่าอัตรารหัส (code rate) นิยามโดย

$$R = \frac{K}{N}$$

เมื่อ $0 < R \leq 1$ เสมอ



เมทริกซ์ตัวกำเนิด

พิจารณาบิตข้อความ $\mathbf{m} = [m_1, m_2, \ldots, m_K]$ ขนาด $1 \times K$ (นั่นคือ 1 แนวนอนและ K แนวตั้ง) รหัสบล็อกเชิงเส้นแบบ (N, K) สร้างได้โดยการนำบิตข้อความ $\mathbf{m}$ มาคูณกับเมทริกซ์ตัวกำเนิด (generator matrix) $\mathbf{G}$ ขนาด $K \times N$ ซึ่งอยู่ในรูป (ที่มีสมาชิกเป็นเลข 0 หรือเลข 1 เท่านั้น)

$$\mathbf{G}_{K\times N} = \left[\mathbf{P}_{K\times(N-K)} \,|\, \mathbf{I}_{K\times K}\right] = \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,(N-K)} & 1 & 0 & \cdots & 0 \\ p_{2,1} & p_{2,2} & \cdots & p_{2,(N-K)} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{K,1} & p_{K,2} & \cdots & p_{K,(N-K)} & 0 & 0 & \cdots & 1 \end{bmatrix}$$

เมื่อ $\mathbf{P}$ คือเมทริกซ์พาริตีขนาด $K \times (N-K)$ ที่สอดคล้องกับบิตพาริตีในคำรหัส และ $\mathbf{I}$ คือเมทริกซ์เอกลักษณ์ขนาด $K \times K$ ซึ่งจะได้ผลลัพธ์เป็นคำรหัส $\mathbf{c} = [c_1, c_2, \ldots, c_N]$ ขนาด $1 \times N$ นั่นคือ

$$\mathbf{c} = \mathbf{mG} = \left[p_1\ p_2\ \ldots\ p_{N-K}\ \ m_1\ m_2\ \ldots\ m_K\right] \Rightarrow \text{รหัสแบบมีระบบ (systematic code)}$$

Example 1

จงเข้ารหัสข้อมูล $\mathbf{m} = [100]$ และ $\mathbf{m} = [110]$ ถ้าให้ $\mathbf{G} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$

วิธีทำ เมทริกซ์ตัวกำเนิด **G** นี้จะใช้เข้ารหัสบิตข้อมูลครั้งละ 3 บิต นั่นคือ $\mathbf{m} = [m_1\ m_2\ m_3]$ เพราะฉะนั้นคำรหัสที่ได้จากการเข้ารหัสด้วยเมทริกซ์ **G** คือ

$$\mathbf{c} = \mathbf{mG} = \begin{bmatrix} m_1 & m_2 & m_3 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} m_2 \oplus m_3 & m_1 \oplus m_3 & m_1 \oplus m_2 & m_1 & m_2 & m_3 \end{bmatrix}$$

เมื่อ $\oplus$ คือตัวดำเนินการบวกแบบมอดุโลสอง หรือการทำ XOR ดังนั้นถ้า $\mathbf{m} = [100]$ จะได้ $\mathbf{c} = [011100]$ และถ้า $\mathbf{m} = [110]$ จะได้ $\mathbf{c} = [110110]$ เป็นต้น

เมทริกซ์พาริตีเช็ก

- รหัสบล็อกเชิงเส้นแบบ (N, K) ยังสามารถถูกกำหนดด้วยเมทริกซ์ตรวจสอบภาวะคู่หรือคี่ (parity-check matrix) หรือเรียกสั้นๆ ว่าเมทริกซ์พาริตีเช็ก $\mathbf{H}$ ขนาด $(N-K)\times N$ ได้ ซึ่งต้องสอดคล้องกับความสัมพันธ์ $\mathbf{GH}^{\mathrm{T}} = \mathbf{0}$ เมื่อ $(.)^{\mathrm{T}}$ คือเครื่องหมายเมทริกซ์สลับเปลี่ยน
- สำหรับคำรหัสใดๆ จะได้ $\mathbf{cH}^{\mathrm{T}} = \mathbf{mGH}^{\mathrm{T}} = \mathbf{0}$ เสมอ
- สมาชิกในแต่ละแนวนอนของเมทริกซ์ $\mathbf{H}$ ก็คือสมการพาริตีเช็ก (parity-check equation) ซึ่งเป็นตัวกำหนดความสัมพันธ์ของบิต c_i สำหรับ $i = 1, 2, \ldots, N$ ในคำรหัส
- ถ้าเมทริกซ์ตัวกำเนิด $\mathbf{G}$ อยู่ในรูปแบบมีระบบ $\Rightarrow$ เมทริกซ์พาริตีเช็กมีค่าเท่ากับ

$$\mathbf{H}_{(N-K)\times N} = \left[\mathbf{I}_{(N-K)\times(N-K)} \,|\, \mathbf{P}^{\mathrm{T}}\right]$$

เช่น เมทริกซ์ตัวกำเนิด $\mathbf{G}$ ในหน้า 8 $\Rightarrow$ จัดรูปได้เป็น $\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$

ระยะทางน้อยสุดของรหัส

- สมรรถนะของรหัสบล็อกเชิงเส้น $\Rightarrow$ **น้ำหนักแฮมมิง** (Hamming weight) ของคำรหัส นิยามโดย

$$w_H(\mathbf{c}) = \text{จำนวนบิตที่มีค่าเท่ากับ 1 ภายในคำรหัส } \mathbf{c}$$

เช่นถ้า $\mathbf{c} = 100100$ จะได้ $w_H([100100]) = 2$ เป็นต้น และ**ระยะทางแฮมมิง** (Hamming distance) ระหว่าง $\mathbf{c}_1$ และ $\mathbf{c}_2$ นิยามโดย

$$d_H(\mathbf{c}_1, \mathbf{c}_2) = w_H(\mathbf{c}_1 - \mathbf{c}_2) = \sum_{i=1}^{N} (c_{1,i} \neq c_{2,i})$$

ตัวอย่างเช่นถ้า $\mathbf{c}_1 = 110011$ และ $\mathbf{c}_2 = 000111$ จะได้ระยะทางแฮมมิง $d_H(\mathbf{c}_1, \mathbf{c}_2) = 3$

- ถ้าให้รหัส **c** มีทั้งหมด 2^K คำรหัส ระยะทางแฮมมิงที่น้อยสุดระหว่างคำรหัสจะเรียกกันทั่วไปว่า**ระยะทางน้อยสุด** (minimum distance) ของรหัส หรือ $d_{\min}$ ซึ่งนิยามโดย

$$d_{\min} = \min_{i \neq j} \left\{ d_H \left(\mathbf{c}_i, \mathbf{c}_j \right) \right\}$$

- รหัสบล็อกเชิงเส้นมีความสามารถในการ**แก้ไข**ข้อผิดพลาดเป็นจำนวน $t = \dfrac{\left| d_{\min} - 1 \right|}{2}$ บิต และมีความสามารถในการ**ตรวจหา**ข้อผิดพลาดได้จำนวน $e = d_{\min} - 1$ บิต
- ระยะทางน้อยสุดของรหัส $d_{\min}$ $\Rightarrow$ หาได้จากเมทริกซ์ตัวกำเนิด **G** และเมทริกซ์พาริตีเช็ก **H** นั่นคือระยะทางน้อยสุดของรหัสมีค่าเท่ากับ
  - น้ำหนักแฮมมิงน้อยสุดของสมาชิกในแนวนอนของเมทริกซ์ **G**
  - จำนวนแนวตั้งน้อยสุดของเมทริกซ์ **H** ที่บวกกันแบบมอดุโลสองแล้วได้ผลลัพธ์เท่ากับศูนย์

การถอดรหัสบล็อกเชิงเส้น

- การถอดรหัสบล็อกเชิงเส้น ⇒ **การถอดรหัสแบบซินโดรม** (syndrome decoding) เมื่อเวกเตอร์ซินโดรม **s** ขนาด $1\times(N-K)$ นิยามโดย

$$\mathbf{s}=\mathbf{r}\mathbf{H}^{\mathrm{T}}=\left[s_1,s_2,\ldots s_{N-K}\right]$$

เมื่อ $\mathbf{r}=\mathbf{c}\oplus\mathbf{e}=[r_1,\ r_2,\ \ldots,\ r_N]$ คือเวกเตอร์ข้อมูลที่ต้องการถอดรหัส, **c** คือเวกเตอร์คำรหัส, $\mathbf{e}=[e_1,\ e_2,\ \ldots,\ e_N]$ คือเวกเตอร์ข้อผิดพลาด โดยที่ $e_i\in\{0,\ 1\}$ และ $e_i=1$ หมายถึงคำรหัสบิตที่ i มีข้อผิดพลาด ($e_i=0$ หมายถึงคำรหัสบิตที่ i ไม่มีข้อผิดพลาด)

- แทนค่า $\mathbf{r}=\mathbf{c}\oplus\mathbf{e}$ จะได้

$$\mathbf{s}=(\mathbf{c}\oplus\mathbf{e})\mathbf{H}^{\mathrm{T}}=\mathbf{c}\mathbf{H}^{\mathrm{T}}\oplus\mathbf{e}\mathbf{H}^{\mathrm{T}}=\mathbf{e}\mathbf{H}^{\mathrm{T}}$$

นั่นคือค่าซินโดรมจะขึ้นอยู่กับเวกเตอร์ข้อผิดพลาด **e**

- ถ้าให้เมทริกซ์พาริตีเช็ก $\mathbf{H}=\left[\mathbf{h}_1\ \mathbf{h}_2\ \cdots\ \mathbf{h}_N\right]$ โดยที่ $\mathbf{h}_i$ สำหรับ $i=\{1, 2, ..., N\}$ คือเวกเตอร์แนวตั้งลำดับที่ i ในเมทริกซ์ $\mathbf{H}$ และให้ $\mathbf{e}=\left[0\cdots 0\,1\,0\cdots 0\right]$ โดยบิต 1 อยู่ตำแหน่งที่ p เมื่อ $1\le p\le N$ ในเวกเตอร์ $\mathbf{e}$ ดังนั้นจะได้

$$\mathbf{s}=\mathbf{e}\mathbf{H}^{\mathrm{T}}=\left[\mathbf{h}_p\right]^{\mathrm{T}}$$

นั่นคือถ้า $\mathbf{s}^{\mathrm{T}}=\mathbf{h}_p$ ก็แสดงว่าเวกเตอร์ข้อผิดพลาดมีค่าเท่ากับ $\mathbf{e}=\left[0\cdots 0\,1\,0\cdots 0\right]$ โดยบิต 1 อยู่ ณ ตำแหน่งที่ p ในเวกเตอร์ $\mathbf{e}$

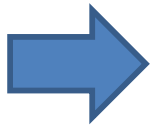
- ถ้าระบบมีข้อผิดพลาดเกิดขึ้นจำนวน x บิต ก็จะทำให้ได้เวกเตอร์ $\mathbf{e}$ ที่มีบิต 1 เป็นจำนวน x ตัว อยู่ ณ ตำแหน่งที่ $p_1, p_2, \ldots, p_x$ ในเวกเตอร์ $\mathbf{e}$ $\Rightarrow$ หาเวกเตอร์ซินโดรม $\mathbf{s}$ ได้จาก

$$\mathbf{s}=\mathbf{e}\mathbf{H}^{\mathrm{T}}=\left[\mathbf{h}_{p1}\oplus\mathbf{h}_{p2}\oplus\ldots\oplus\mathbf{h}_{px}\right]^{\mathrm{T}}$$

- ตารางการถอดรหัส (decoding table) สำหรับรหัสบล็อกเชิงเส้นที่มีเมทริกซ์ $\mathbf{H}$ (ในหน้า 9) ซึ่งมีจำนวนซินโดรมทั้งหมด $2^{N-K}=2^3=8$ แบบ

- ในทางปฏิบัติถ้า $s = 0$ แสดงว่า $\mathbf{r}$ เป็นคำรหัส (หรือตรวจหาข้อผิดพลาดไม่ได้) แต่ถ้า $s \neq 0$ แสดงว่า $\mathbf{r}$ มีข้อผิดพลาดอยู่อย่างน้อย 1 บิต

	ข้อผิดพลาด **e**	ซินโดรม **s**
	000000	000
ข้อผิดพลาด 1 บิต	100000	100
	010000	010
	001000	001
	000100	011
	000010	101
	000001	110
ข้อผิดพลาด 2 บิต →	100100	111



ตารางการถอดรหัส หรือ
แถวลำดับมาตรฐาน
(standard array)

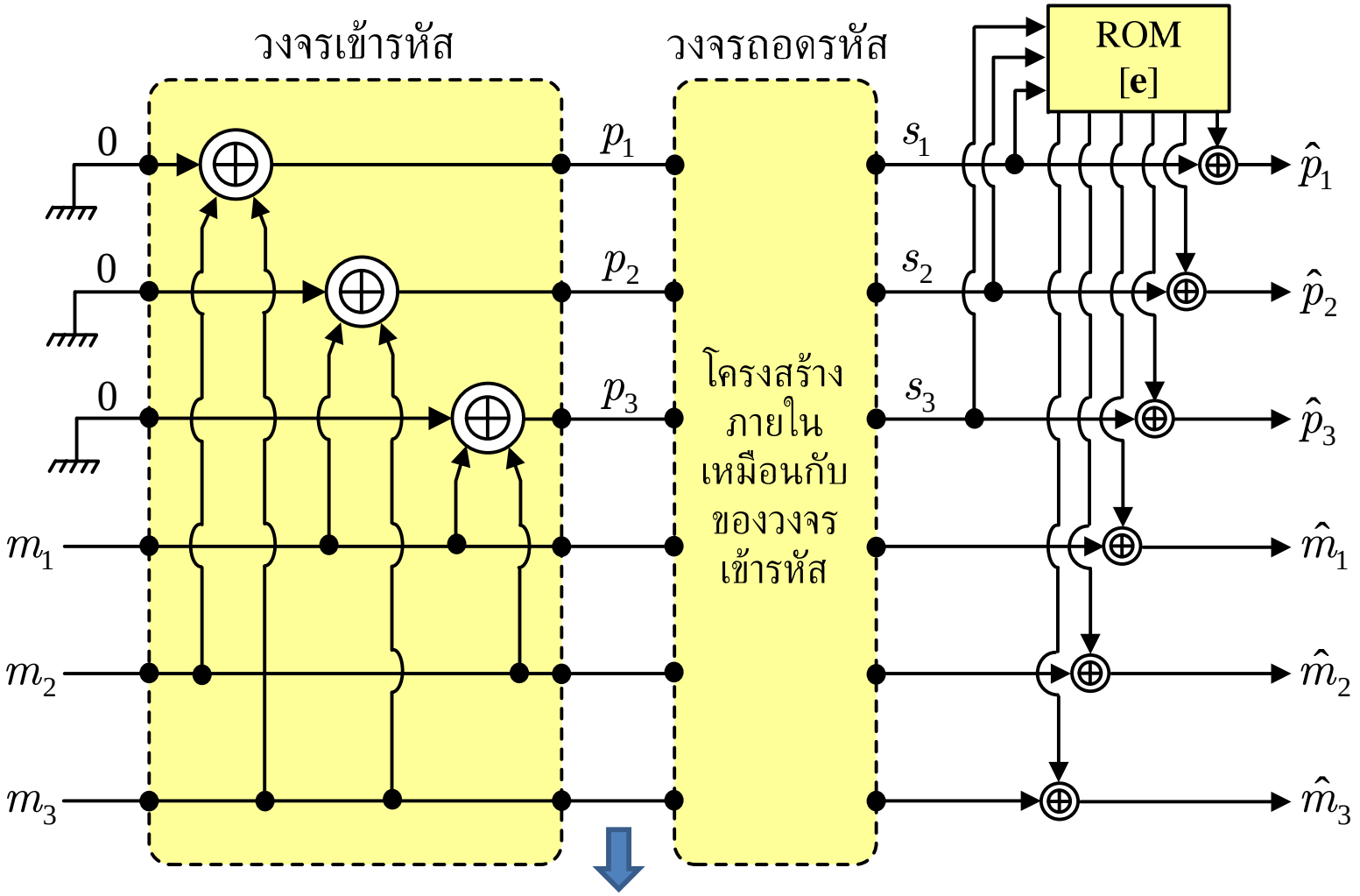
มีค่าเป็นอะไรก็ได้ที่ทำให้
ผลรวมของแนวตั้งใน **H**
(โดยใช้จำนวนแนวตั้งน้อยสุด)
มีค่าเท่ากับ 111

หมายเหตุ ค่าซินโดรม **s** = การนำเวกเตอร์แนวตั้งของเมทริกซ์ **H** หลายๆ แนวตั้งมาบวกกันแบบมอดุโลสอง (มี e หลายแบบ) ⇒ **e** ที่เป็นไปได้มากสุด ก็คือ**e** ที่**มีน้ำหนักแฮมมิงน้อยสุด** (หรือมีจำนวนเลข 1 น้อยสุด)

- เมื่อได้ข้อผิดพลาด **e** ที่ต้องการ ⇒ ถอดรหัสลำดับข้อมูล $\mathbf{r}$ ได้จาก $\hat{\mathbf{c}} = \mathbf{r} \oplus \mathbf{e}$
- การถอดรหัสแบบซินโดรมเหมาะกับระบบที่ใช้คำรหัสที่มีความยาวน้อยและข้อผิดพลาดที่เกิดขึ้นในแต่ละคำรหัสมีจำนวนน้อย

การสร้างวงจรเข้ารหัสและถอดรหัส

- พิจารณารหัสบล็อกเชิงเส้นแบบ (6, 3) ที่มีเมทริกซ์พาริตีเช็ก $\mathbf{H}$ ในหน้า 9 และสมมุติว่าข้อมูลที่ต้องการถอดรหัส**ไม่มี**ข้อผิดพลาด นั่นคือ $\mathbf{r} = [p_1\ p_2\ p_3\ m_1\ m_2\ m_3]$ คือคำรหัส
- ดังนั้นจาก $\mathbf{s} = \mathbf{r}\mathbf{H}^T = \mathbf{0} \Rightarrow p_1 = m_2 \oplus m_3$, $p_2 = m_1 \oplus m_3$ และ $p_3 = m_1 \oplus m_2$ ซึ่งสามารถนำมาใช้สร้างวงจรเข้าและถอดรหัสบล็อกเชิงเส้นแบบง่ายได้ตามรูปหน้า 16
- ภายในรอม (ROM) จะบันทึกตารางการถอดรหัส เพื่อให้ค่าเวกเตอร์ข้อผิดพลาด $\mathbf{e}$ สำหรับแต่ละเวกเตอร์ซินโดรม $\mathbf{s} = [s_1, s_2, s_3]$ ที่รับมา จากนั้นก็นำเวกเตอร์ $\mathbf{e}$ ไปบวกรวมแบบมอดุโลสองกับข้อมูลเอาต์พุตของวงจรถอดรหัส ก็จะได้ค่าประมาณของข้อความ $\hat{\mathbf{m}} = [\hat{m}_1, \hat{m}_2, \hat{m}_3]$



$$\mathbf{r} = \begin{bmatrix} p_1 & p_2 & p_3 & m_1 & m_2 & m_3 \end{bmatrix}$$

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

$$p_1 = m_2 \oplus m_3 \,,\quad p_2 = m_1 \oplus m_3 \quad \text{และ} \quad p_3 = m_1 \oplus m_2$$

Example 2

พิจารณารหัสบล็อกเชิงเส้นแบบ (6, 3) ที่มีเมทริกซ์ **G** ตามหน้า 8 และมีเมทริกซ์ **H** ตามหน้า 9 โดยคำรหัสจะอยู่ในรูป $\mathbf{c} = [p_1\ p_2\ p_3\ m_1\ m_2\ m_3]$

ก) จงหาสมการแสดงความสัมพันธ์ของบิตพาริตี p_i และหาคำรหัสที่เป็นได้ทั้งหมด

ข) จงหาความสามารถในการแก้ไขและตรวจหาข้อผิดพลาดของรหัสบล็อกเชิงเส้นนี้

ค) จงสร้างตารางการถอดรหัสที่แสดงความสัมพันธ์ระหว่างค่าซินโดรม **s** และเวกเตอร์ **e**

ง) จงถอดรหัสข้อมูลเมื่อข้อมูลที่ได้รับคือ $\mathbf{r}_1 = [111011]$, $\mathbf{r}_2 = [011010]$ และ $\mathbf{r}_3 = [000111]$

วิธีทำ

ก) เนื่องจาก $\mathbf{c} = \mathbf{mG}$ นั่นคือ

$$\underbrace{\begin{bmatrix} p_1 & p_2 & p_3 & m_1 & m_2 & m_3 \end{bmatrix}}_{\text{codeword } \mathbf{c}} = \begin{bmatrix} m_1 & m_2 & m_3 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

ซึ่งจะได้ว่า $p_1 = m_2 \oplus m_3$, $p_2 = m_1 \oplus m_3$ และ $p_3 = m_1 \oplus m_2$ ดังนั้นคำรหัสที่เป็นได้ทั้งหมดคือ

ข้อความ **m**	000	001	010	011	100	101	110	111
คำรหัส **c**	000000	110001	101010	011011	011100	101101	110110	000111

ข) ระยะทางน้อยสุด $d_{\min} = 3$ (น้ำหนักแฮมมิงน้อยสุดของสมาชิกในแนวนอนของเมทริกซ์ **G**) ดังนั้นรหัสบล็อกเชิงเส้นนี้มีความสามารถในการแก้ไขข้อผิดพลาด $t = (d_{\min} - 1)/2 = 1$ บิต และมีความสามารถในการตรวจหาข้อผิดพลาด $e = d_{\min} - 1 = 2$ บิต

ค) ซินโดรม $\mathbf{s}$ มีจำนวน $N - K = 6 - 3 = 3$ บิต แสดงว่ามีจำนวนรูปแบบที่เป็นไปได้ทั้งหมดเท่ากับ $2^3 = 8$ แบบ $\Rightarrow$ ตารางการถอดรหัสแสดงในหน้าที่ 14

ง) จาก $\mathbf{s} = \mathbf{r}\mathbf{H}^{\mathrm{T}} = (\mathbf{c}+\mathbf{e})\mathbf{H}^{\mathrm{T}} = \mathbf{e}\mathbf{H}^{\mathrm{T}}$ ทำให้ได้ว่า

- $\mathbf{s}_1 = \mathbf{r}_1\mathbf{H}^{\mathrm{T}} = [100]$ จากตารางการถอดรหัสจะได้ $\mathbf{e}_1 = [100000]$ (บิตที่หนึ่งมีข้อผิดพลาด) ดังนั้นวงจรถอดรหัสจะให้ข้อมูลเอาต์พุตเป็น $\hat{\mathbf{c}}_1 = \mathbf{r}_1 \oplus \mathbf{e}_1 = [011011]$
- $\mathbf{s}_2 = \mathbf{r}_2\mathbf{H}^{\mathrm{T}} = [110]$ จากตารางการถอดรหัสจะได้ $\mathbf{e}_2 = [000001]$ (บิตที่หกมีข้อผิดพลาด) ดังนั้นวงจรถอดรหัสจะให้ข้อมูลเอาต์พุตเป็น $\hat{\mathbf{c}}_2 = \mathbf{r}_2 \oplus \mathbf{e}_2 = [011011]$
- $\mathbf{s}_3 = \mathbf{r}_3\mathbf{H}^{\mathrm{T}} = [000]$ จากตารางการถอดรหัสจะได้ $\mathbf{e}_3 = [000000]$ แสดงว่าไม่มีข้อผิดพลาด (หรือตรวจหาข้อผิดพลาดไม่พบ) ซึ่งถ้าไม่มีข้อผิดพลาดก็จะได้ $\hat{\mathbf{c}}_3 = \mathbf{r}_3 \oplus \mathbf{e}_3 = [000111]$

Example 3

พิจารณารหัสบล็อกเชิงเส้นแบบ (6, 3) ที่มีเมทริกซ์พาริตีเช็ก $\mathbf{H}$ ดังนี้

$$\tilde{\mathbf{H}} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

จงหาคำรหัสทั้งหมดที่เป็นไปได้และสร้างตารางการถอดรหัสสำหรับถอดรหัสบล็อกเชิงเส้นนี้

วิธีทำ

เนื่องจากเมทริกซ์ $\tilde{\mathbf{H}}$ ไม่ได้อยู่ในรูปแบบมีระบบ $\Rightarrow$ อาศัยการดำเนินการตามแถวขั้นมูลฐาน (elementary row-operation) เพื่อจัดรูปเมทริกซ์ $\tilde{\mathbf{H}}$ ได้ดังนี้

$$\underbrace{\begin{bmatrix}1&0&0&1&1&1\\0&1&1&0&1&1\\1&0&1&0&0&1\end{bmatrix}}_{\tilde{\mathbf{H}}} \Rightarrow \underbrace{\begin{bmatrix}1&0&0&1&1&1\\0&1&1&0&1&1\\0&0&1&1&1&0\end{bmatrix}}_{\mathbf{H}_I} \Rightarrow \underbrace{\begin{bmatrix}1&0&0&1&1&1\\0&1&0&1&0&1\\0&0&1&1&1&0\end{bmatrix}}_{\mathbf{H}}$$

จากนั้นเมื่อได้เมทริกซ์ $\mathbf{H}$ ในรูปแบบมีระบบแล้ว $\Rightarrow$ เมทริกซ์ตัวกำเนิด $\mathbf{G}$ มีค่าเท่ากับ

$$\mathbf{H}=\begin{bmatrix}1&0&0&1&1&1\\0&1&0&1&0&1\\0&0&1&1&1&0\end{bmatrix} \Rightarrow \mathbf{G}=\begin{bmatrix}1&1&1&1&0&0\\1&0&1&0&1&0\\1&1&0&0&0&1\end{bmatrix}$$

เนื่องจากคำรหัส $\mathbf{c}=\mathbf{mG}=\begin{bmatrix}p_1 & p_2 & p_3 & m_1 & m_2 & m_3\end{bmatrix}$ โดยที่ $p_1=m_1\oplus m_2\oplus m_3$, $p_2=m_1\oplus m_3$ และ $p_3=m_1\oplus m_2$ เพราะฉะนั้นคำรหัสที่เป็นได้ทั้งหมดคือ

ข้อความ **m**	000	001	010	011	100	101	110	111
คำรหัส **c**	000000	110001	101010	011011	111100	001101	010110	100111

เนื่องจากซินโดรม **s** มีจำนวน $N - K = 3$ บิต $\Rightarrow$ มีจำนวนรูปแบบที่เป็นไปได้ทั้งหมดเท่ากับ $2^3 = 8$ แบบตามที่แสดงในตารางการถอดรหัสต่อไปนี้

	ข้อผิดพลาด **e**	ซินโดรม **s**
	000000	000
ข้อผิดพลาด 1 บิต	100000	100
	010000	010
	001000	001
	000100	111
	000010	101
	000001	110
ข้อผิดพลาด 2 บิต →	011000	011 →

ผลรวมแบบมอดุโลสอง ของข้อมูลในแนวตั้งที่ 2 และ 3 ของเมทริกซ์ **H**

รหัสวน

- รหัสวน (cyclic code) เป็นรหัสบล็อกเชิงเส้นประเภทหนึ่งที่มีความสามารถในการแก้ไขข้อผิดพลาดได้มากกว่ารหัสบล็อกเชิงเส้นที่กล่าวมาตอนต้น
  - การสร้างวงจรเข้ารหัสและวงจรถอดรหัสก็ทำได้ง่ายโดยใช้เรจิสเตอร์แบบเลื่อน (shift register)

- พิจารณารหัสวนแบบ (N, K) ซึ่งใช้เข้ารหัสข้อมูล $\mathbf{m} = [m_1, m_2, \ldots, m_K]$ จำนวน K บิต และให้คำรหัส $\mathbf{c} = [c_1, c_2, \ldots, c_N]$ จำนวน N บิต
- คุณสมบัติที่สำคัญคือ เมื่อนำคำรหัส $\mathbf{c}$ ใดๆ มาทำการ**เลื่อนแบบวน** (cyclic shift) ไปทางซ้ายหรือทางขวา ก็จะได้ผลลัพธ์เป็นคำรหัสใหม่

❑ ถ้านิยาม

$$\mathbf{c}^{(i)} = \left[c_{N-i+1}, \ldots, c_N, c_1, c_2, \ldots, c_{N-i}\right]$$

คือคำรหัส $\mathbf{c}$ ที่ถูกเลื่อนแบบวนไปทางขวาเป็นจำนวน i ครั้ง

ในทางปฏิบัตินิยมเขียนคำรหัส $\mathbf{c}$ ให้อยู่ในรูปของพหุนามของตัวแปร x ที่มีระดับขั้น $N-1$ ดังนี้

$$c(x) = c_1 + c_2 x + c_3 x^2 + \ldots + c_N x^{N-1}$$

เมื่อค่าสัมประสิทธิ์ $c_i \in \{0,1\}$ สำหรับ $i = \{1, 2, \ldots, N\}$ และมีคุณสมบัติการบวกและการคูณคือ

$$0+0=0 \qquad 0\times 0=0$$

$$0+1=1+0=1 \qquad 0\times 1=1\times 0=0$$

$$1+1=0 \qquad 1\times 1=1 \quad \text{(มอดุโลสอง)}$$

ดังนั้นพหุนามของ $\mathbf{c}^{(i)}$ มีค่าเท่ากับ

$$c^{(i)}(x) = c_{N-i+1} + c_{N-i+2} x + \ldots + c_N x^{i-1} + c_1 x^i + \ldots + c_{N-i} x^{N-i-1}$$

คุณสมบัติที่น่าสนใจของ**พหุนามรหัส** (code polynomial) คือถ้านำ $x^{*}c(x)$ มาหารด้วย $x^{N}+1$ ก็จะได้**เศษเหลือ**เท่ากับ $c^{(*)}(x)$ ซึ่งพิสูจน์ได้ดังนี้ พิจารณาการหาร $xc(x)$ ด้วย $x^{N}+1$ ซึ่งจะได้

$$
\begin{array}{r}
c_N \leftarrow \text{ผลลัพธ์} \\
x^N+1 \;\overline{)\; c_N x^N + c_{N-1}x^{N-1} + \ldots + c_2 x^2 + c_1 x} \\
\underline{c_N x^N + c_N} \\
c_{N-1}x^{N-1} + \ldots + c_2 x^2 + c_1 x + c_N \leftarrow \text{เศษเหลือ}
\end{array}
$$

หมายเหตุ เครื่องหมายบวก + และเครื่องหมายลบ – มีความหมายเหมือนกันสำหรับการ**มอดุโลสอง**

ซึ่งมีผลลัพธ์เท่ากับ $c^{(1)}(x)=c_N+c_1x+c_2x^2+\ldots+c_{N-1}x^{N-1}$ ในทำนองเดียวกันก็สามารถพิสูจน์ได้ว่าเศษเหลือของ $x^{*}c(x)$ หารด้วย $x^{N}+1$ ก็คือ $c^{(*)}(x)$

การเข้ารหัส

ถ้ากำหนดให้ข้อมูล $\mathbf{m} = [m_1, m_2, \ldots, m_K]$ ในรูปของ**พหุนามข้อมูล** (data polynomial) คือ

$$m(x) = m_1 + m_2 x + \ldots + m_K x^{K-1}$$

ที่มีระดับขั้น $K-1$ และ**พหุนามตัวกำเนิด** (generator polynomial) คือ

$$g(x) = g_1 + g_2 x + \ldots + g_{N-K+1} x^{N-K}$$

ซึ่งก็คือตัวประกอบใดๆ ของพหุนาม $x^N + 1$ ที่มีระดับขั้น $N - K$ และ $g_1 = g_{N-K+1} = 1$ เสมอ ดังนั้นพหุนามรหัส $c(x)$ มีค่าเท่ากับ

$$c(x) = m(x)g(x)$$

ที่มีระดับขั้นน้อยกว่าหรือเท่ากับ $N - 1$

Example 4

จงหาพหุนามตัวกำเนิด $g(x)$ ของรหัสวนแบบ $(7,4)$ พร้อมทั้งหาคำรหัสเมื่อข้อมูลที่ต้องการเข้ารหัสคือ 0101, 1111, 1000 และ 0001

วิธีทำ รหัสวนแบบ $(7,4)$ จะมี $N=7$ และ $N-K=3$ ดังนั้น x^N+1 แยกตัวประกอบได้เป็น

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

แสดงว่า $g(x)$ ที่ใช้ได้คือ x^3+x+1 หรือ x^3+x^2+1 สมมุติว่าเลือก $g(x)=x^3+x^2+1$

สำหรับ $\mathbf{m}=[0101] \Rightarrow m(x)=x+x^3$ ดังนั้นพหุนามรหัสหาได้จาก

$$c(x)=m(x)g(x)=(x+x^3)(x^3+x^2+1)=x+x^4+x^5+x^6$$

หรือเขียนในรูปของเวกเตอร์ได้คือ $\mathbf{c}=[0100111]$ ในทำนองเดียวกันสำหรับข้อมูล 1111, 1000 และ 0001 จะมีคำรหัสในรูปของเวกเตอร์คือ 1101001, 1011000 และ 0001011 ตามลำดับ

คำรหัสที่ได้ไม่อยู่ในรูปแบบมีระบบ (non-systematic form)

- โดยทั่วไปนิยมสร้างคำรหัสให้อยู่ในรูปแบบมีระบบเพื่อให้ง่ายต่อการทำงาน รหัสวนที่อยู่ในรูปแบบมีระบบสามารถสร้างได้จากความสัมพันธ์ดังนี้

$$c(x) = x^{N-K} m(x) + \rho(x)$$

เมื่อ

$$\rho(x) = \text{Rem}\left[\frac{x^{N-K} m(x)}{g(x)}\right]$$

คือเศษเหลือที่ได้จากการหาร $x^{N-K} m(x)$ ด้วย $g(x)$ และ $\text{Rem}[a/b]$ คือตัวดำเนินการหาค่าเศษเหลือที่ได้จากการหาร a ด้วย b

Example 5

จงสร้างรหัสวนแบบ $(7,4)$ ที่อยู่ในรูปแบบมีระบบโดยใช้พหุนามตัวกำเนิด $g(x)=x^3+x^2+1$ เมื่อข้อมูลที่ต้องการเข้ารหัสคือ 0101, 1111, 1000 และ 0001

วิธีทำ เนื่องจาก $N=7$ และ $N-K=3$ สำหรับ $\mathbf{m}=[0101]$ หรือ $m(x)=x+x^3$ ดังนั้น

$$x^{N-K}m(x)=x^3\left(x+x^3\right)=x^4+x^6$$

จากนั้นหาค่า $\rho(x)$ ซึ่งก็คือเศษเหลือที่ได้จากการหาร $x^{n-k}m(x)$ ด้วย $g(x)$ ซึ่งจะได้

$$
\begin{array}{r}
x^3+x^2+1 \\
\hline
x^3+x^2+1 \;\big)\; x^6+x^4 \\
\quad x^6+x^5+x^3 \\
\hline
\qquad x^5+x^4+x^3 \\
\qquad x^5+x^4+x^2 \\
\hline
\qquad\qquad x^3+x^2 \\
\qquad\qquad x^3+x^2+1 \\
\hline
\qquad\qquad\quad 1 \qquad \leftarrow \rho(x) \\
\qquad\qquad\quad =
\end{array}
$$

นั่นคือ $\rho(x)=1$ เพราะฉะนั้นคำรหัสที่อยู่ในรูปแบบมีระบบหาได้จาก

$$c(x)=x^{N-K}m(x)+\rho(x)=x^3\left(x+x^3\right)+1=1+x^4+x^6$$

หรือเขียนในรูปของเวกเตอร์ได้คือ **c** = [1000101] ในทำนองเดียวกันสำหรับข้อมูล 1111, 1000 และ 0001 จะมีคำรหัสในรูปของเวกเตอร์คือ 1111111, 1011000 และ 0110001 ตามลำดับ

เมทริกซ์ตัวกำเนิด

- ถ้าให้เมทริกซ์ตัวกำเนิด $\mathbf{G}_{cyc}$ ขนาด $K \times N$ คือ $\mathbf{G}_{cyc} = \left[g(x), xg(x), \cdots, x^{K-1}g(x) \right]^T$

  จากนั้นคำรหัสก็สามารถสร้างได้จาก $\mathbf{c} = \mathbf{m}\mathbf{G}_{cyc}$ $\Rightarrow$ คำรหัสที่ไม่อยู่ในรูปแบบมีระบบ

- ถ้าต้องการสร้างคำรหัสที่อยู่ในรูปแบบมีระบบ $\Rightarrow$ จัดรูปเมทริกซ์ $\mathbf{G}_{cyc}$ ให้อยู่ในรูปแบบมีระบบ โดยใช้เทคนิคการดำเนินการตามแถวขั้นมูลฐาน เช่น

$$\underbrace{\begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}}_{\mathbf{G}_{cyc}} \Rightarrow \underbrace{\begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}}_{\mathbf{G}_{I}} \Rightarrow \underbrace{\begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}}_{\mathbf{G}_{II}} \Rightarrow \underbrace{\begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{G}_{sys}}$$

- ถ้านำเมทริกซ์ $\mathbf{G}_{sys}$ มาใช้เข้ารหัสกับข้อมูล $\mathbf{m} = [0101]$ ตามความสัมพันธ์ $\mathbf{c} = \mathbf{m}\mathbf{G}_{sys}$ ก็จะได้คำรหัส $\mathbf{c} = [1000101] = [p_1\ p_2\ p_3\ m_1\ m_2\ m_3\ m_4]$ ในรูปแบบมีระบบ

เมทริกซ์พาริตีเช็ก

- รหัสวนแบบ (N, K) ยังสามารถถูกกำหนดด้วย**พหุนามพาริตีเช็ก** (parity-check polynomial) $h(x)$ ซึ่งมีความสัมพันธ์กับพหุนามตัวกำเนิด $g(x)$ ดังนี้

$$h(x)g(x) = 0 \quad \text{(มอดุโล } x^N + 1\text{)}$$

นั่นคือพหุนามพาริตีเช็ก $h(x) = \dfrac{x^N + 1}{g(x)} = h_1 + h_2 x + \ldots + h_{K+1} x^K$ ที่มีระดับขั้น K และ $h_1 = h_{K+1} = 1$ เสมอ หรือเขียนในรูปของเมทริกซ์พาริตีเช็ก $\mathbf{H}_{\text{cyc}}$ ที่มีขนาด $(N-K) \times N$

$$\mathbf{H}_{\text{cyc}} = \begin{bmatrix} h_{K+1} & h_K & \cdots & h_1 & 0 & 0 & \cdots & 0 \\ 0 & h_{K+1} & h_K & \cdots & h_1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & h_{K+1} & h_K & \cdots & h_1 \end{bmatrix}$$

ซึ่งยังคงมีความความสัมพันธ์ $\Rightarrow \mathbf{G}_{\text{cyc}}\mathbf{H}_{\text{cyc}}^{\text{T}} = \mathbf{0}$ และ $\mathbf{c}\mathbf{H}_{\text{cyc}}^{\text{T}} = \left(\mathbf{m}\mathbf{G}_{\text{cyc}}\right)\mathbf{H}_{\text{cyc}}^{\text{T}} = \mathbf{0}$

Example 6

พิจารณารหัสวนแบบ $(7,4)$ ที่มีพหุนามตัวกำเนิด $g(x) = x^3 + x^2 + 1$ จงคำนวณหาพหุนามพาริตีเช็ก $h(x)$ และเมทริกซ์พาริตีเช็ก $\mathbf{H}_{\text{cyc}}$

วิธีทำ เนื่องจาก $N = 7$ และ $K = 4$

พหุนามพาริตีเช็ก $h(x)$ หาได้จาก

$$
\begin{array}{r}
x^4 + x^3 + x^2 + 1 \\
x^3 + x^2 + 1 \overline{)\, x^7 + 1} \\
x^7 + x^6 + x^4 \\
\hline
x^6 + x^4 + 1 \\
x^6 + x^5 + x^3 \\
\hline
x^5 + x^4 + x^3 + 1 \\
x^5 + x^4 + x^2 \\
\hline
x^3 + x^2 + 1 \\
x^3 + x^2 + 1 \\
\hline
0
\end{array}
$$

$h(x)$

นั่นคือพหุนามพาริตีเช็ก $h(x) = x^4 + x^3 + x^2 + 1$ หรือเขียนในรูปของเมทริกซ์ $\mathbf{H}_{cyc}$ ได้คือ

$$\mathbf{H}_{cyc} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

ซึ่งเมื่อนำมาหาค่า $\mathbf{G}_{cyc}\mathbf{H}_{cyc}^{T}$ โดยที่ $\mathbf{G}_{cyc}$ เป็นไปตามหน้า 31 ก็จะได้

$$\mathbf{G}_{cyc}\mathbf{H}_{cyc}^{T} = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}^{T} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \mathbf{0}$$

- เนื่องจากเมทริกซ์ $\mathbf{H}_{cyc}$ ในหน้า 34 ใช้ตรวจสอบเฉพาะคำรหัสที่ไม่อยู่ในรูปแบบมีระบบ
- ถ้าต้องการตรวจสอบคำรหัสที่อยู่ในรูปแบบมีระบบ $\Rightarrow$ จัดรูปเมทริกซ์ $\mathbf{H}_{cyc}$ ให้อยู่ในรูปแบบมีระบบโดยใช้เทคนิคการดำเนินการตามแถวขั้นมูลฐาน เช่น

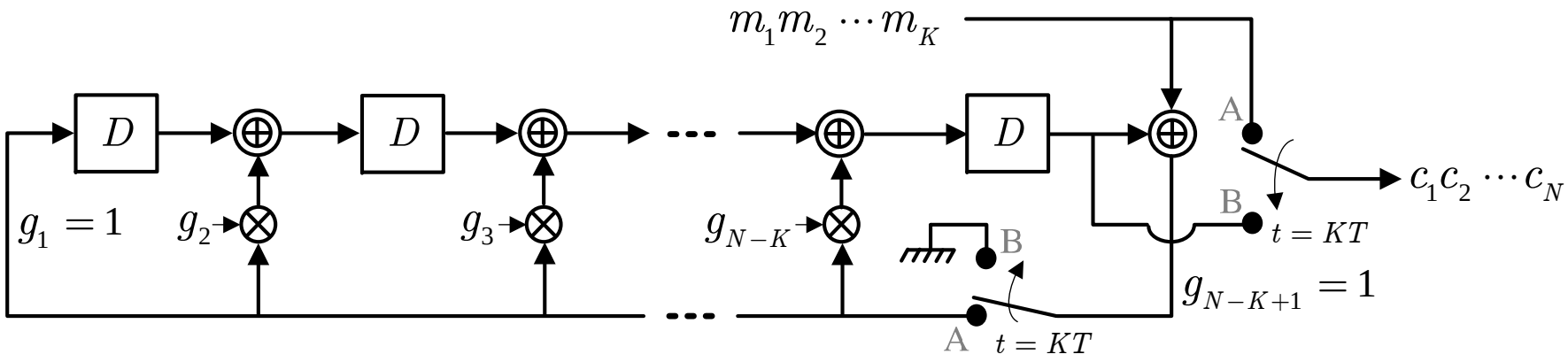
$$\underbrace{\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}}_{\mathbf{H}_{cyc}} \Rightarrow \underbrace{\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}}_{\mathbf{H}_{I}} \Rightarrow \underbrace{\begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}}_{\mathbf{H}_{sys}}$$

หรือหาได้ตรงจากเมทริกซ์ $\mathbf{G}_{sys}$ ตามความสัมพันธ์ในหน้า 7 และ 9

หมายเหตุ โดยทั่วไปค่าสัมประสิทธิ์ของพหุนามตัวกำเนิด $g(x)$ (เรียงจากระดับขั้นน้อยไปหามาก) จะปรากฏอยู่ในแนวนอนแรกของเมทริกซ์ $\mathbf{G}_{cyc}$ หรือ $\mathbf{G}_{sys}$ เสมอ ในขณะที่ค่าสัมประสิทธิ์ของพหุนามพาริตีเช็ก $h(x)$ (เรียงจากระดับขั้นมากไปหาน้อย) จะปรากฏอยู่ในแนวนอนสุดท้ายของเมทริกซ์ $\mathbf{H}_{cyc}$ หรือ $\mathbf{H}_{sys}$ เสมอเช่นกัน

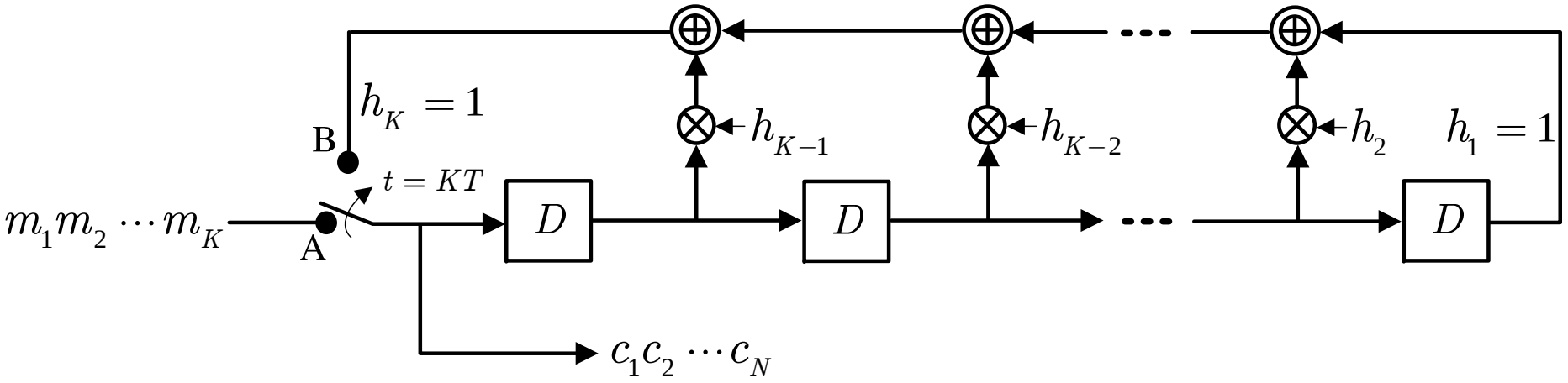
การสร้างวงจรเข้ารหัส

- วงจรเข้ารหัสวน (cyclic encoder) ทำได้ง่ายโดยใช้เพียงเรจิสเตอร์แบบเลื่อน
- วงจรเข้ารหัสที่ใช้วงจรการหารและใช้ $g(x) = g_1 + g_2 x + \ldots + g_{N-K+1} x^{N-K}$



โดยบิตข้อมูลด้านขวาสุดหรือ m_K จะถูกส่งเข้าไปในวงจรเข้ารหัสก่อน (และ m_1 จะถูกส่งเข้าไปในวงจรเข้ารหัสเป็นตัวสุดท้าย) โดยเมื่อเริ่มต้นการเข้ารหัสข้อมูล **m** สวิตช์จะอยู่ ณ ตำแหน่ง A และวงจรเข้ารหัสจะให้ข้อมูลเอาต์พุต (หรือคำรหัส) เป็น $c_{N-K+i} = m_i$ เมื่อ $i = \{K, K-1, \ldots, 1\}$ จากนั้นเมื่อข้อมูลถูกส่งเข้ามาในวงจรเข้ารหัสจนครบ ณ เวลา $t = kT$ สวิตช์ ก็จะเปลี่ยนไปอยู่ ณ ตำแหน่ง B เพื่อให้ข้อมูลเอาต์พุตตัวถัดไปจนกระทั่งได้ข้อมูลเอาต์พุตครบ ณ เวลา $t = NT$ ก็จะได้คำรหัสที่อยู่ในรูปแบบมีระบบ

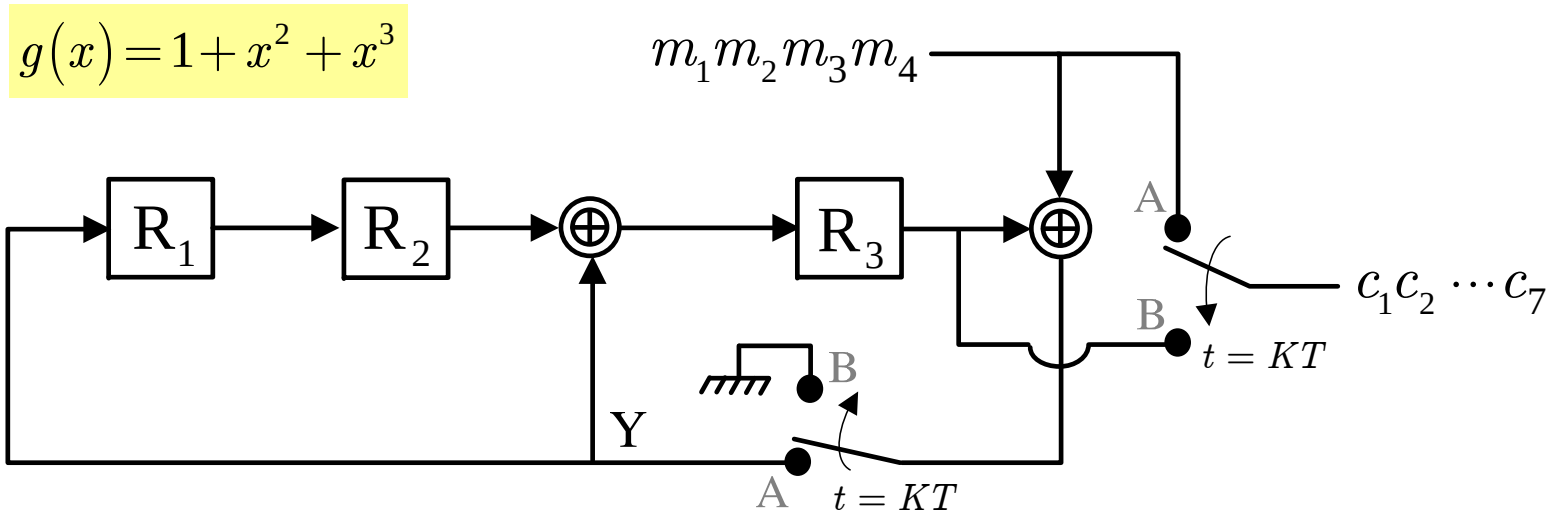
ในทางปฏิบัติยังสามารถนำพหุนามพาริตีเช็ก $h(x)$ มาสร้างเป็นวงจรเข้ารหัสได้ ซึ่งให้คำรหัสที่อยู่ในรูปแบบมีระบบเช่นกัน



$$h(x) = h_1 + h_2 x + \ldots + h_{K+1} x^K$$

Example 7

จงเข้ารหัสข้อมูล $\mathbf{m} = [m_1, m_2, m_3, m_4] = [0101]$ โดยใช้ $g(x) = x^3 + x^2 + 1$



วิธีทำ รายละเอียดของการเข้ารหัสมีดังนี้

i	m_{5-i}	บิต Y	เรจิสเตอร์แบบเลื่อน R_1	R_2	R_3	ตำแหน่งสวิตช์	c_{8-i}	
0	-	-	0	0	0	-	-	
1	1	1	1	0	1	A	1	c_7
2	0	1	1	1	1	A	0	c_6
3	1	0	0	1	1	A	1	c_5
4	0	1	1	0	0	A	0	c_4
5	-	0	0	1	0	B	0	c_3
6	-	0	0	0	1	B	0	c_2
7	-	0	0	0	0	B	1	c_1

นั่นคือ $\mathbf{c} = [c_1, c_2, c_3, c_4, c_5, c_6, c_7] = [1000101]$

การถอดรหัส

- เนื่องจากทุกพหุนามรหัส $c(x)$ เป็นพหุคูณของ $g(x)$ ถ้ามีข้อผิดพลาดเกิดขึ้นระหว่างการส่งผ่านข้อมูลก็จะทำให้ข้อมูลที่ได้รับ $r(x)$ ไม่เป็นพหุคูณของ $g(x)$ ดังนั้น

$$\frac{r(x)}{g(x)} = m(x) + \frac{s(x)}{g(x)}$$

โดยที่**พหุนามซินโดรม** (syndrome polynomial)

$$s(x) = \text{Rem}\left[\frac{r(x)}{g(x)}\right] = s_1 + s_2 x + \ldots + s_{N-K} x^{N-K-1}$$

คือเศษเหลือที่ได้จากการหาร $r(x)$ ด้วย $g(x)$ ที่มีระดับขั้นน้อยกว่าหรือเท่ากับ $N-K-1$ หรือเขียนในรูปของเวกเตอร์ซินโดรมได้คือ $\mathbf{s} = [s_1, s_2, \ldots, s_{N-K}]$

- ถ้าให้ $e(x)$ คือ**พหุนามข้อผิดพลาด** (error polynomial) ก็จะได้

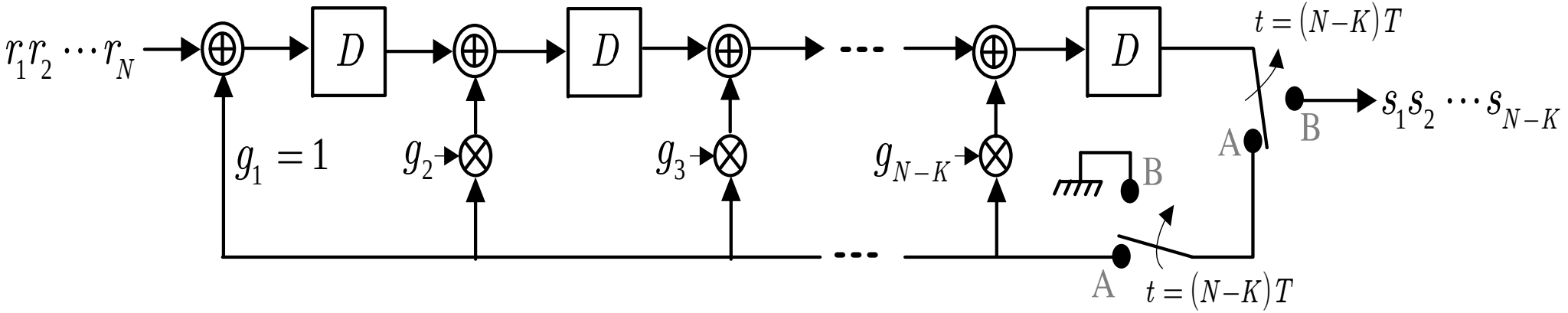
$$r(x) = c(x) + e(x)$$

จากนั้นแทนค่าลงในสมการ $s(x)$ จะได้

$$s(x) = \text{Rem}\left[\frac{r(x)}{g(x)}\right] = \text{Rem}\left[\frac{c(x)+e(x)}{g(x)}\right] = \text{Rem}\left[\frac{e(x)}{g(x)}\right]$$

- การถอดรหัสวนยังคงใช้**ตารางการถอดรหัส** ⇒ แสดงความสัมพันธ์ระหว่างเวกเตอร์ซินโดรม **s** และเวกเตอร์ข้อผิดพลาด **e** เหมือนที่ใช้ในการถอดรหัสบล็อก
- เมื่อวงจรถอดรหัสได้รับเวกเตอร์ข้อมูล **r** ⇒ หาค่าซินโดรม $s(x)$ ⇒ นำเวกเตอร์ซินโดรม **s** ไปหาเวกเตอร์ **e** จากตารางการถอดรหัส ⇒ ถอดรหัสข้อมูล **r** จาก $\hat{\mathbf{c}} = \mathbf{r} \oplus \mathbf{e}$

- การถอดรหัสวนยังคงใช้ตารางการถอดรหัส ⇒ แสดงความสัมพันธ์ระหว่างเวกเตอร์ซินโดรม **s** และเวกเตอร์ข้อผิดพลาด **e** เหมือนที่ใช้ในการถอดรหัสบล็อก
- เมื่อวงจรถอดรหัสได้รับเวกเตอร์ข้อมูล **r** ⇒ หาค่าซินโดรม $s(x)$ ⇒ นำเวกเตอร์ซินโดรม **s** ไปหาเวกเตอร์ **e** จากตารางการถอดรหัส ⇒ ถอดรหัสข้อมูล **r** จาก $\hat{\mathbf{c}} = \mathbf{r} \oplus \mathbf{e}$
- การคำนวณหาค่า $s(x)$ มีการทำงานเหมือนกับการหาค่าเศษเหลือ $\rho(x)$ ที่ได้จากเข้ารหัสวน ดังนั้นวงจรที่ใช้ในการหาค่าซินโดรมจึงมีโครงสร้างคล้ายกับวงจรเข้ารหัสดังนี้



วงจรหาค่าซินโดรมที่ใช้วงจรการหารและพหุนามตัวกำเนิด g(x)

รหัสแฮมมิง

- รหัสแฮมมิง (Hamming code) เป็นรหัสบล็อกที่ถูกกำหนดด้วยพารามิเตอร์

$$(N, K) = (2^r - 1, 2^r - 1 - r)$$

โดยที่ N คือจำนวนบิตของคำรหัส, K คือจำนวนบิตของข้อมูลที่ต้องการเข้ารหัส, และ $r \geq 2$ คือเลขจำนวนเต็ม โดยรหัสแฮมมิงจะมีระยะทางน้อยสุด $d_{min} = 3$ จึงสามารถตรวจหาข้อผิดพลาดได้สูงสุดจำนวน 2 บิต และแก้ไขข้อผิดพลาดได้ 1 บิต

- ค่าอัตราส่วนกำสังเฉลี่ยของสัญญาณต่อกำสังเฉลี่ยของสัญญาณรบกวน (SNR) ของระบบที่ถูกเข้ารหัส (coded system) มีค่าเท่ากับ

$$\frac{E_c}{N_0} = \frac{E_b}{N_0} R$$

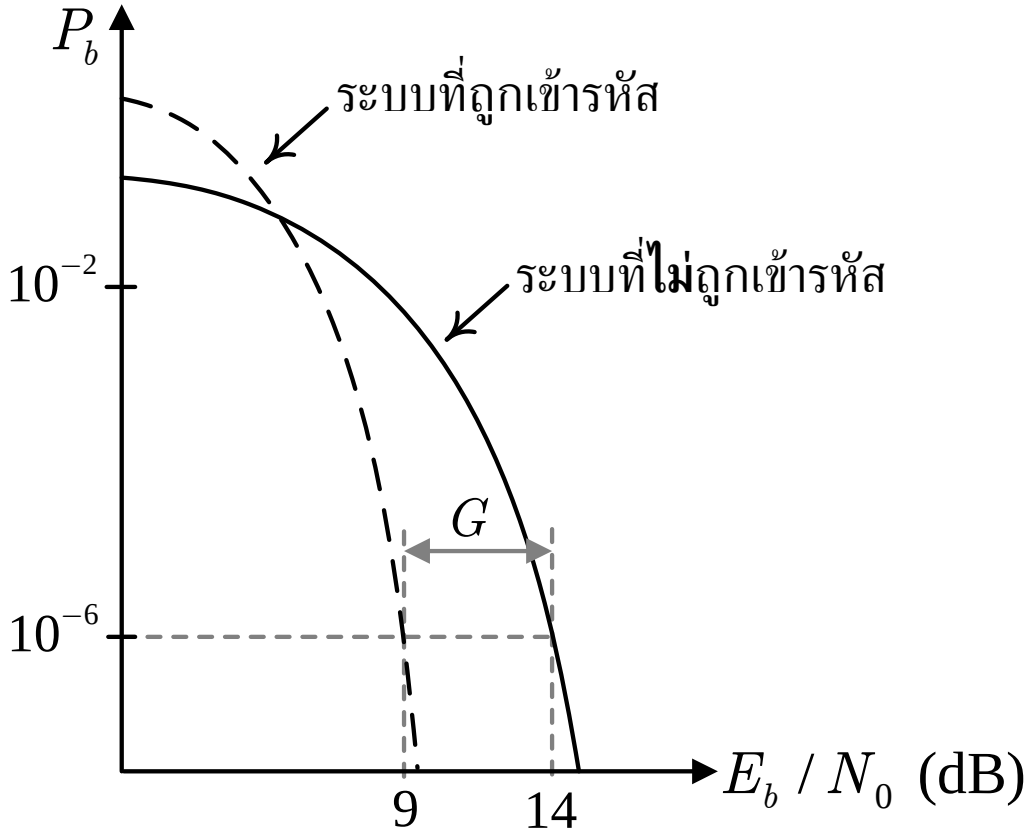
เมื่อ E_c คือพลังงานเฉลี่ยของบิตที่ถูกเข้ารหัส, E_b คือพลังงานเฉลี่ยของบิต, $R = K / N$ คืออัตรารหัส และ $N_0 / 2$ คือ PSD แบบสองด้านของสัญญาณรบกวนเกาส์สีขาว

- อัตราขยายการเข้ารหัส (coding gain) ⇒ ผลต่างระหว่างค่า SNR ของระบบที่ไม่ถูกเข้ารหัส และค่า SNR ของระบบที่ถูกเข้ารหัส ณ ระดับความน่าจะเป็นของข้อผิดพลาดที่สนใจ

$$G = \left(\frac{E_b}{N_0}\right)_u - \left(\frac{E_b}{N_0}\right)_c$$

โดยที่ $(E_b / N_0)_c = E_c / (RN_0)$

- ระบบที่ถูกเข้ารหัสจะใช้ค่า SNR **น้อยกว่า** ระบบที่ไม่ถูกเข้ารหัส (ทำให้ประหยัดพลังงาน)



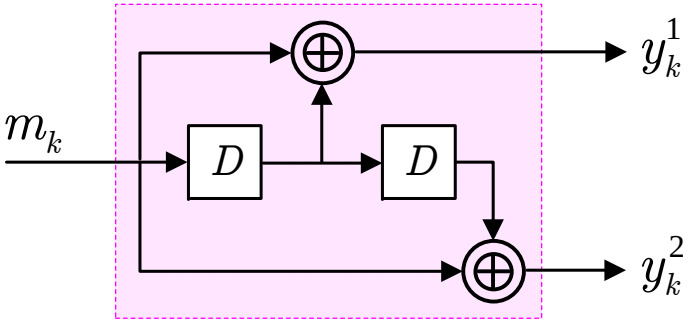
รหัสคอนโวลูชัน

- นิยมใช้ในหลายงานประยุกต์ เช่น วีดิทัศน์แบบดิจิทัล การสื่อสารแบบไร้สาย และการสื่อสารดาวเทียม เป็นต้น
  - เป็นส่วนประกอบที่สำคัญของรหัสเทอร์โบ (turbo code) ที่ถือว่าเป็นรหัส ECC แบบหนึ่งที่มีสมรรถนะเข้าใกล้ความจุช่องสัญญาณของแชนนอน

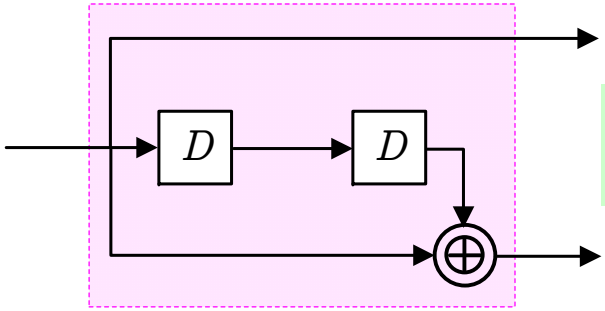
การเข้ารหัส

- วงจรเข้ารหัสคอนโวลูชันจะใช้เรจิสเตอร์แบบเลื่อนและวงจรบวกแบบมอดุโลสอง (modulo-2 adder) ในการเข้ารหัสข้อมูล
- เข้ารหัสลำดับข้อมูลอินพุตหนึ่งชุด และให้ลำดับข้อมูลเอาต์พุตจำนวนมากกว่าหรือเท่ากับหนึ่งชุด
- ถ้าวงจรเข้ารหัสคอนโวลูชันเข้ารหัสข้อมูลอินพุตจำนวน K บิต แล้วทำให้เกิดข้อมูลเอาต์พุตจำนวน N บิต $\Rightarrow$ อัตรารหัส $R = K/N$

❑ ตัวอย่างวงจรเข้ารหัสคอนโวลูชันที่มีอัตรารหัส R = 1/2



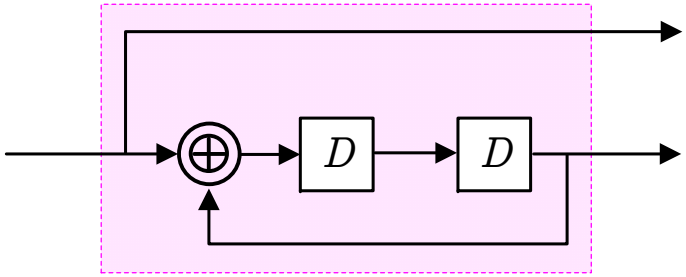
(ก)



$[1,\ 1\oplus D^2]$

(ข)

$$G(D)=[G_1(D),\ G_2(D)]$$
$$=[1\oplus D,\ 1\oplus D^2]$$



$[1,\ 1/(1\oplus D^2)]$

(ค)

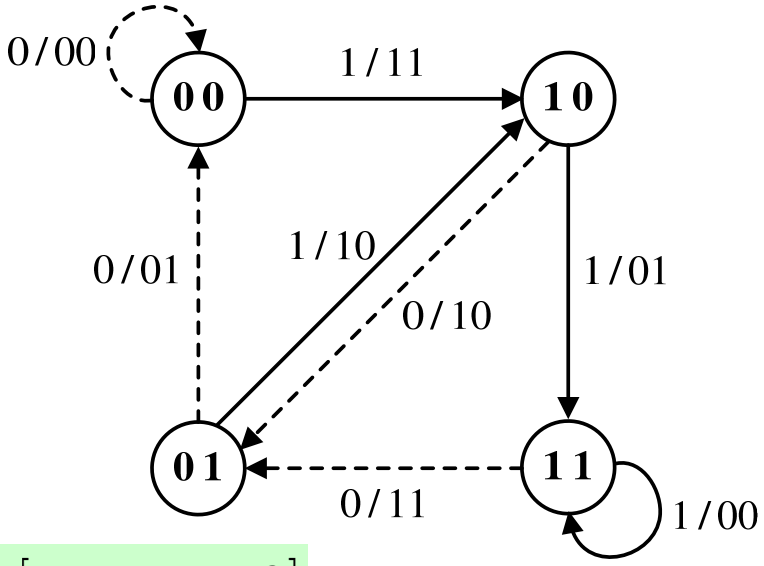
วงจรเข้ารหัสคอนโวลูชันแบบมีระบบเวียนเกิดจะนิยมใช้งานมากกว่าวงจรเข้ารหัสคอนโวลูชันแบบอื่นๆ

ในทางปฏิบัติวงจรเข้ารหัสคอนโวลูชันจะเขียนแทนด้วยพหุนามตัวกำเนิด $G(D)=\sum_{i=1}^{\mu} g_i D^i$

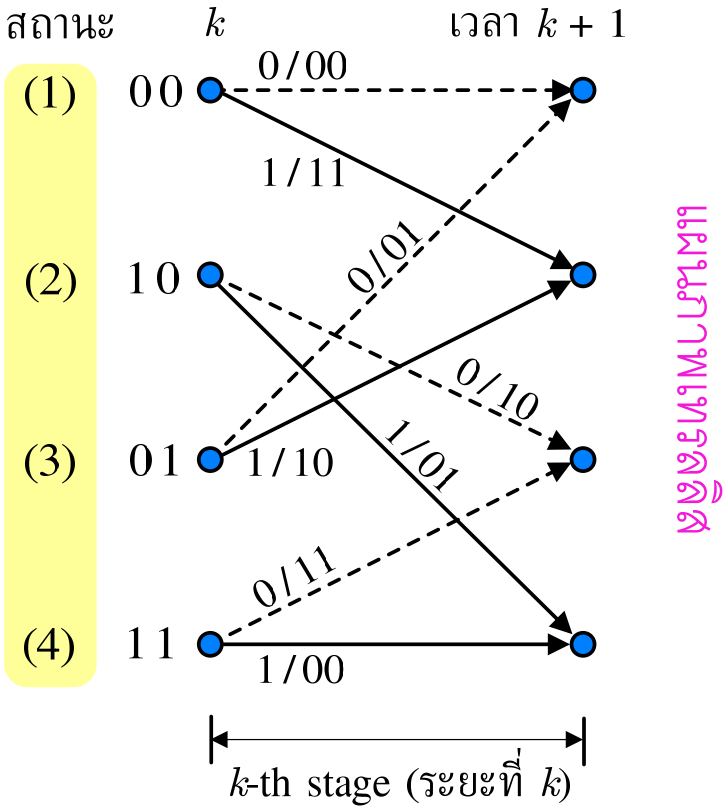
เมื่อ μ คือจำนวนหน่วยความจำของวงจรเข้ารหัสคอนโวลูชัน (หรือจำนวนเรจิสเตอร์แบบเลื่อน) และ $g_i = 1$ ถ้าบิตข้อมูลอินพุตที่ถูกหน่วงเวลาไป i หน่วย มีผลต่อการเกิดของบิตข้อมูลเอาต์พุต ณ เวลาปัจจุบัน

- การวิเคราะห์รหัสคอนโวลูชันจะอาศัย**เครื่องสถานะจำกัด** (FSM)
  - แบบจำลองที่แสดงให้เห็นถึงการเปลี่ยนแปลงของข้อมูลอินพุต, สถานะเริ่มต้น (start state), สถานะต่อไป (next state), และข้อมูลเอาต์พุต ของระบบ

FSM ของวงจรเข้ารหัสคอนโวลูชันในรูป (ก) หน้า 46



$$G(D) = \left[1 \oplus D,\ 1 \oplus D^2\right]$$



แผนภาพสถานะ

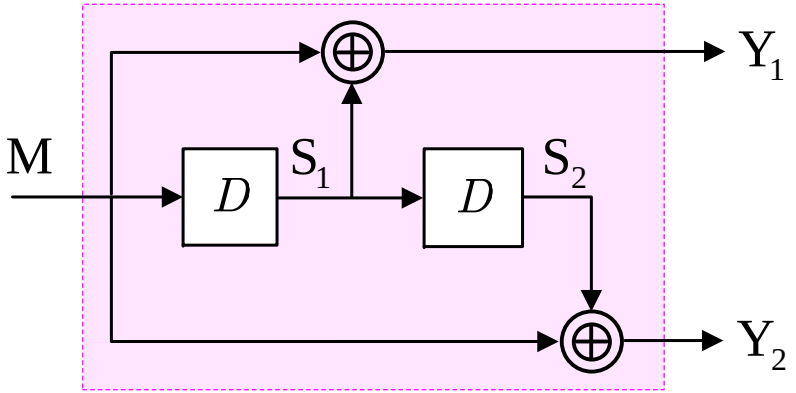
มีทั้งหมด $2^{\mu} = 4$ สถานะคือ 00, 01, 10 และ 11 หรือแสดงด้วยสัญลักษณ์ (1), (2), (3) และ (4) โดยที่เส้นลูกศรจะแสดงเส้นทางการเปลี่ยนสถานะ และค่า $m / y^1 y^2$ ที่อยู่ติดกับเส้นลูกศรจะใช้แทนค่าบิตอินพุต m และบิตเอาต์พุต y^1 และ y^2

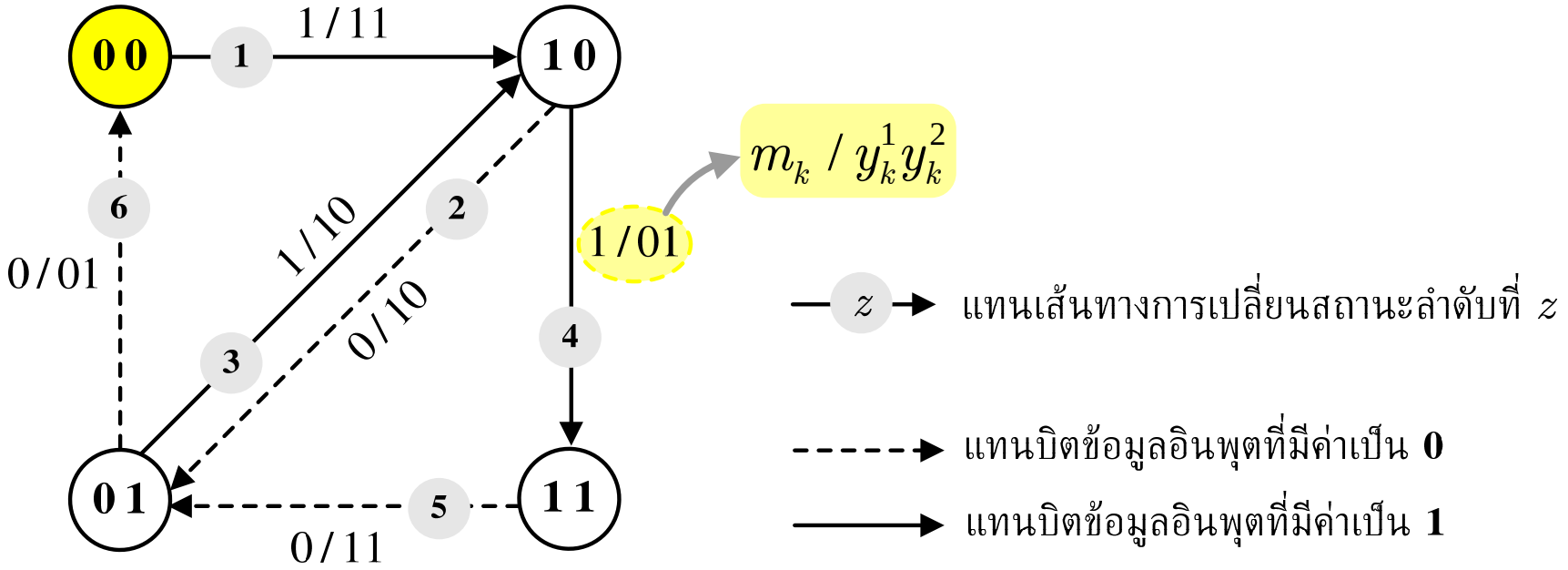
Example 8

จงแสดงขั้นตอนการเข้ารหัสของวงจรเข้ารหัสคอนโวลูชันต่อไปนี้ เมื่อบิตข้อมูลอินพุตคือ

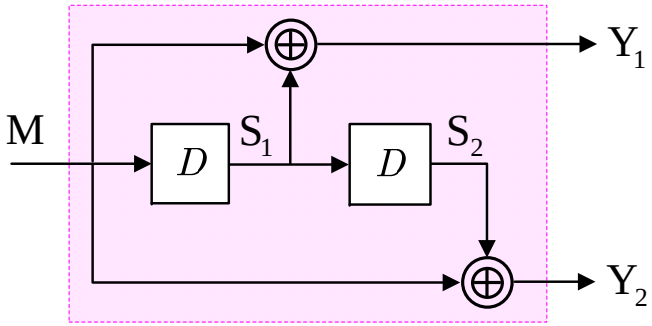
$\{m_1, m_2, m_3, m_4\} = \{1\ 0\ 1\ 1\}$

วิธีทำ



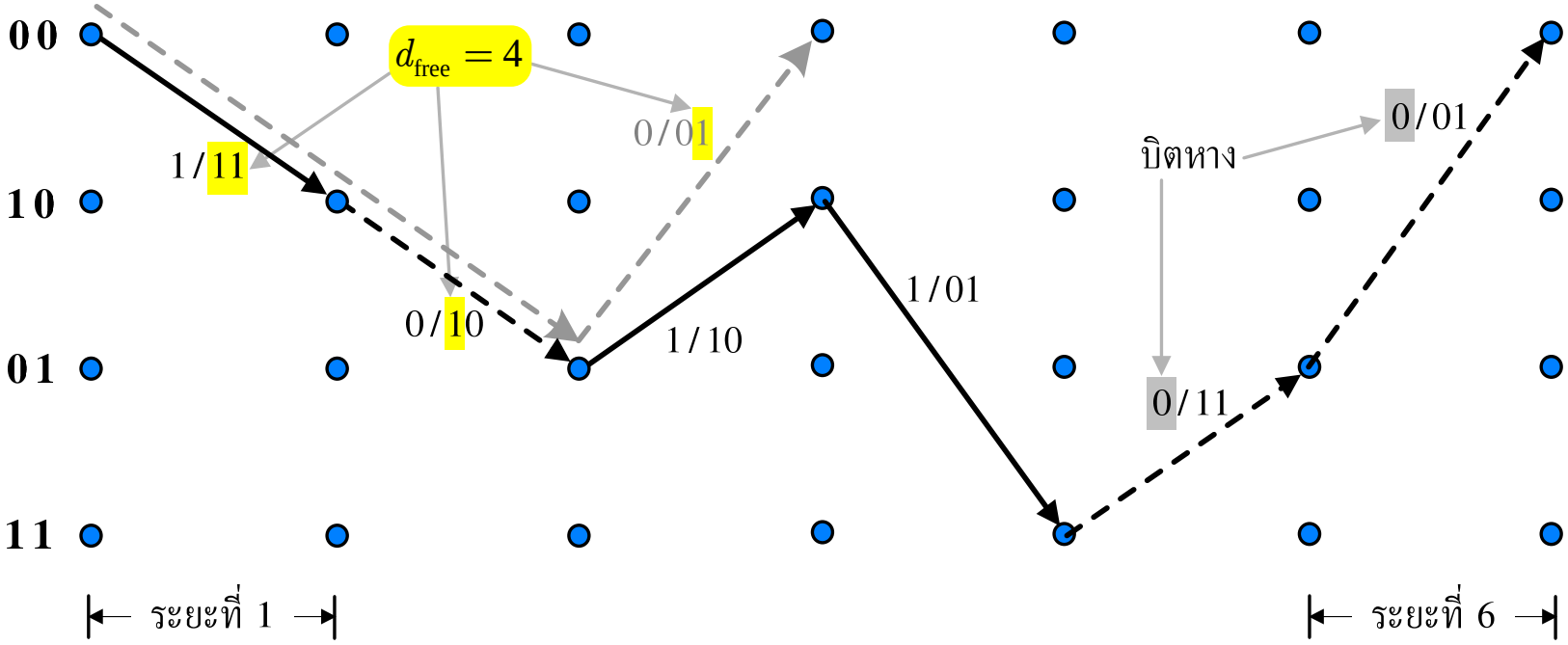


	M	S_1	S_2	Y_1	Y_2
กำหนดค่าเริ่มต้น	--	0	0	--	--
$k = 1$	1	0	0	1	1
$k = 2$	0	1	0	1	0
$k = 3$	1	0	1	1	0
$k = 4$	1	1	0	0	1
$k = 5$ บิตหาง	0	1	1	1	1
$k = 6$	0	0	1	0	1
เลื่อนอีกครั้ง	--	0	0	--	--



- **คำรหัส** (codeword) ทุกคำ (หรือข้อมูลเอาต์พุตของวงจรเข้ารหัสคอนโวลูชัน) ต้องสอดคล้องกับเส้นทางที่เป็นได้เพียงหนึ่งเดียว (unique) ในแผนภาพเทรลลิส

หมายเหตุ ในทางปฏิบัติสมรรถนะของรหัสคอนโวลูชันจะขึ้นกับ**ระยะทางอิสระ** (free distance) d_{free} ซึ่งหาได้แผนภาพเทรลลิส โดยการหา**เส้นทางสั้นสุด** (shortest path) ที่เริ่มจากสถานะ 00 แล้วเดินทางไปตามเส้นทางต่างๆ ในแผนภาพเทรลลิสจนกระทั่งย้อนกลับมาที่สถานะ 00 เหมือนเดิม ดังนั้น d_{free} ก็คือค่าน้ำหนักแฮมมิงรวมของเส้นทางที่สั้นสุด (หรือจำนวนบิต 1 ของข้อมูลเอาต์พุตที่สอดคล้องกับเส้นทางสั้นสุด) เมื่อเทียบกับเส้นทางที่มีข้อมูลเอาต์พุตเป็นศูนย์ทั้งหมด



- การเข้ารหัสคอนโวลูชันยังสามารถทำได้โดยใช้การแปลง D (D transform) เช่นกัน
- ข้อมูลเอาต์พุตที่ได้จากวงจรเข้ารหัสคอนโวลูชันมีค่าเท่ากับ $Y_i(D)=G_i(D)M(D)$ เมื่อ $Y_i(D)=\sum_k y_k^i D^k$ คือผลการแปลง D ของข้อมูลเอาต์พุต y_k^i สำหรับ $i\in\{1,2\}$, $G_i(D)$ คือพหุนามตัวกำเนิดของข้อมูลเอาต์พุต y_k^i, และ $M(D)=\sum_k m_k D^k$ คือผลการแปลง D ของข้อมูลอินพุต
- จากตัวอย่างที่ 8 จะได้ว่า $M(D)=1+D^2+D^3$ และ $G(D)=\left[1\oplus D,\ 1\oplus D^2\right]$ ดังนั้นข้อมูลเอาต์พุตที่ได้จากการเข้ารหัสทั้งสองชุด $\{y_k^1, y_k^2\}$ มีค่าเท่ากับ

$$Y_1(D)=G_1(D)M(D)=(1\oplus D)(1+D^2+D^3)=1+D+D^2+D^4$$

$$Y_2(D)=G_2(D)M(D)=(1\oplus D^2)(1+D^2+D^3)=1+D^3+D^4+D^5$$

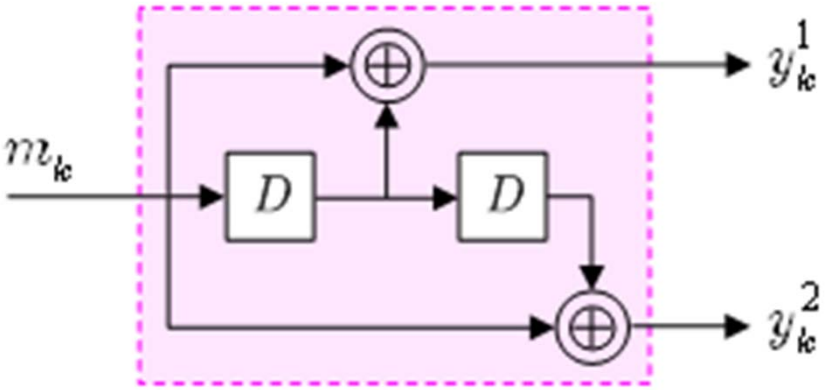
นั่นคือ $\{y_1^1, y_2^1, y_3^1, y_4^1, y_5^1, y_6^1\}=\{1\,1\,1\,0\,1\,0\}$ และ $\{y_1^2, y_2^2, y_3^2, y_4^2, y_5^2, y_6^2\}=\{1\,0\,0\,1\,1\,1\}$ ซึ่งตรงกับข้อมูลเอาต์พุตในตัวอย่างที่ 8

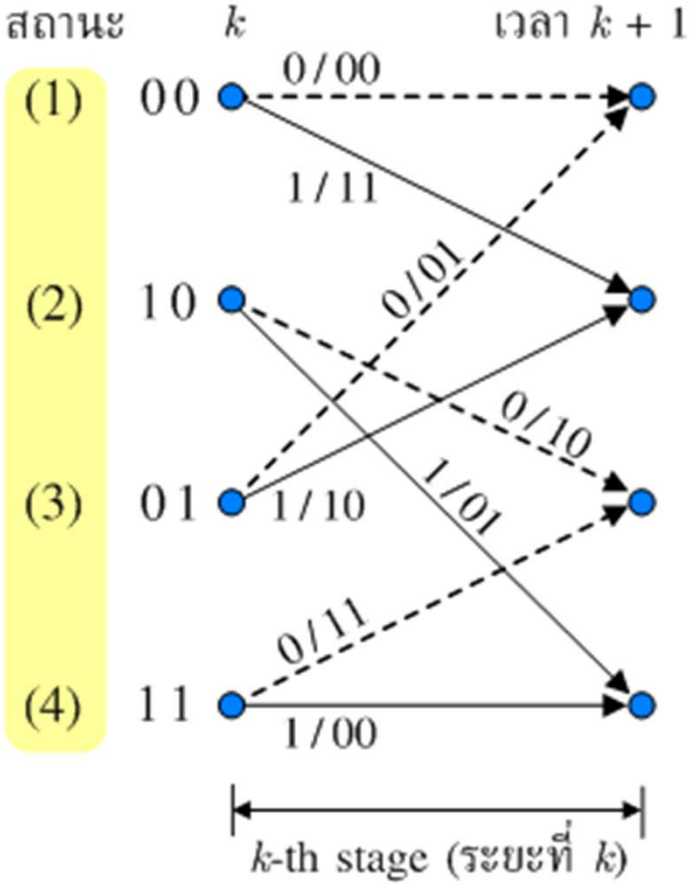
การถอดรหัส

- ในที่นี้อธิบายเฉพาะกระบวนการถอดรหัส**แบบฮาร์ด** (hard decoding) ของข้อมูลที่ถูกเข้ารหัสด้วยรหัสคอนโวลูชันที่มีอัตรารหัส R = 1/N โดยใช้ **อัลกอริทึมวีเทอร์บิ**

อัลกอริทึมวีเทอร์บิ

- พิจารณาวงจรเข้ารหัสคอนโวลูชันที่มี R = 1/2 และมีแผนภาพเทรลลิส





ค่าที่ประจำอยู่ในแต่ละเส้นสาขาคือ $m_k / y_k^1 y_k^2$

- ถ้าให้ (u,q) แทนการเปลี่ยนสถานะ (transition) จากสถานะ u ไปสถานะ q พารามิเตอร์ที่อัลกอริทึมวีเทอร์บิจะต้องคำนวณในทุกช่วงเวลาคือ
  - ค่าเมตริกสาขา (branch metric) ณ เวลา k ของการเปลี่ยนสถานะจากสถานะ u ไปยังสถานะ q, $\lambda_k(u,q)$
  - ค่าเมตริกเส้นทาง (path metric) ณ เวลา $k+1$ ที่สถานะ q, $\Phi_{k+1}(q)$
  - ตัวนำหน้า (predecessor) สำหรับสถานะ q ณ เวลา $k+1$, $\pi_{k+1}(q)$, ซึ่งจะเก็บค่าสถานะเริ่มต้นที่เป็นผลทำให้เกิดเส้นทางการเปลี่ยนสถานะที่ดีสุด (best transition)

- ถอดรหัสโดยใช้อัลกอริทึมวีเทอร์บิ $\Rightarrow$ **การถอดรหัสเหมาะที่สุด** (optimum decoding)
- วงจรถอดรหัสจะเลือกลำดับข้อมูลอินพุต $\{m_k\}$ ตามเส้นทางในแผนภาพเทรลลิสที่ทำให้**ระยะทางแฮมมิง**ระหว่างลำดับข้อมูลที่ได้รับ $\{z_k^*\}$ และคำรหัส $\{y_k^*\}$ มีค่าน้อยสุด

- วงจรตรวจหาวีเทอร์บิจะเลือกลำดับข้อมูลอินพุต $\{m_k\}$ ที่ทำให้ผลรวมของระยะทางแฮมมิงระหว่าง $\{z_k^i\}$ และ $\{y_k^i\}$ นั่นคือ

$$\sum_{k=1}^{L+\nu}\sum_{i=1}^{N}\left|z_k^i - y_k^i\right|$$

มีค่า**น้อยสุด** โดยที่ L คือความยาวของลำดับข้อมูล $\{m_k\}$ และ ν คือความยาวเงื่อนไขบังคับ (constraint length) หรือจำนวนเรจิสเตอร์แบบเลื่อนมากสุดที่ใช้ในวงจรเข้ารหัส

- ค้นหา**เส้นทาง** (path) ที่มีค่าเมตริกน้อยสุดตามแผนภาพเทรลลิส เมื่อเมตริกเส้นทางมีค่าเท่ากับผลรวมของเมตริกสาขา โดยเมตริกสาขาของการเปลี่ยนสถานะ (u,q) ในระยะที่ k นิยามโดย

$$\lambda_k(u,q) = \sum_{i=1}^{N}\left|z_k^i - y_k^i(u,q)\right|$$

เมื่อ $y_k^i(u,q)$ คือข้อมูลเอาต์พุตที่ได้จากการเข้ารหัสซึ่งสอดคล้องกับ (u,q) และเมตริกเส้นทางณ เวลา $k+1$ มีค่าเท่ากับ

$$\Phi_{k+1}(q) = \sum_{i=1}^{k}\lambda_i$$

ขั้นตอนการทำงานของอัลกอริทึมวีเทอร์บิในการถอดรหัสคอนโวลูชัน

(A-1) กำหนดค่าเริ่มต้นของเมตริกเส้นทาง $\Phi_0(p)=0$ สำหรับทุกๆ ค่า p

(A-2) For $k=1, 2,..., L+\nu$

(A-3) For $q=1, 2,..., Q$

(A-4) $\lambda_k(p,q)=\sum_{i=1}^{N}\left|z_k^i-y_k^i(p,q)\right|$ for $\forall p$

(A-5) $\pi_{k+1}(q)=\arg\min_p\left\{\Phi_k(p)+\lambda_k(p,q)\right\}$

(A-6) $\Phi_{k+1}(q)=\Phi_k\left(\pi_{k+1}(q)\right)+\lambda_k\left(\pi_{k+1}(q),q\right)$

(A-7) $\mathbf{S}_{k+1}(q)=\left[\mathbf{S}_k\left(\pi_{k+1}(q)\right)\middle|\pi_{k+1}(q)\right]$

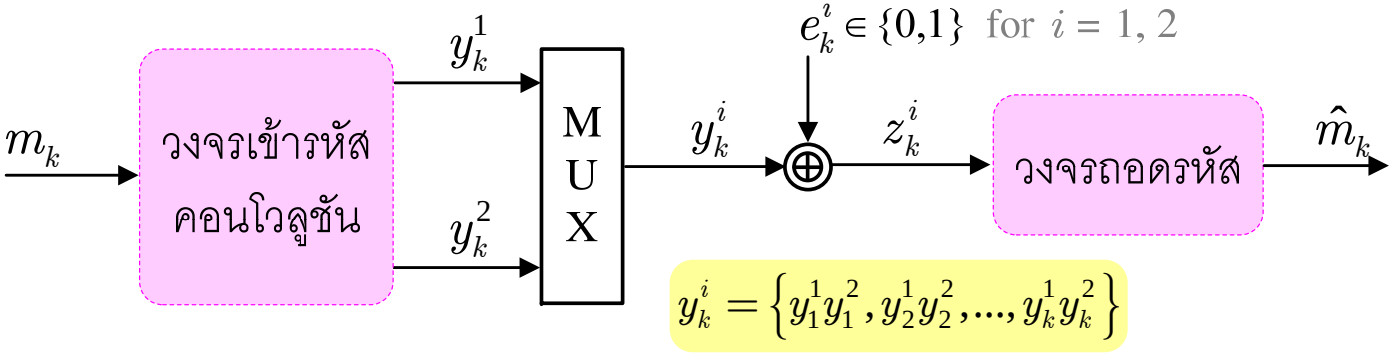
(A-8) End

(A-9) End

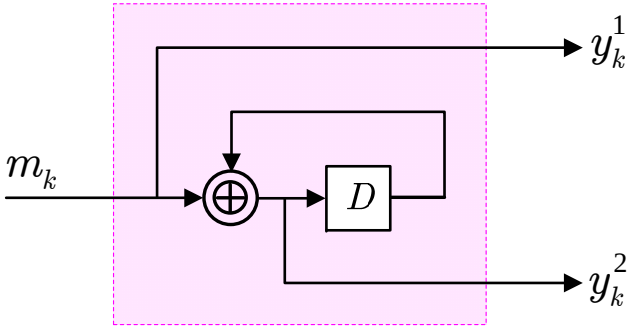
(A-10) ถอดรหัสลำดับข้อมูลอินพุต $\{\hat{m}_k\}$ จากเส้นทางที่ยังมีชีวิตอยู่ที่มีค่า $\Phi_{L+\nu}$ น้อยที่สุด

Example 9

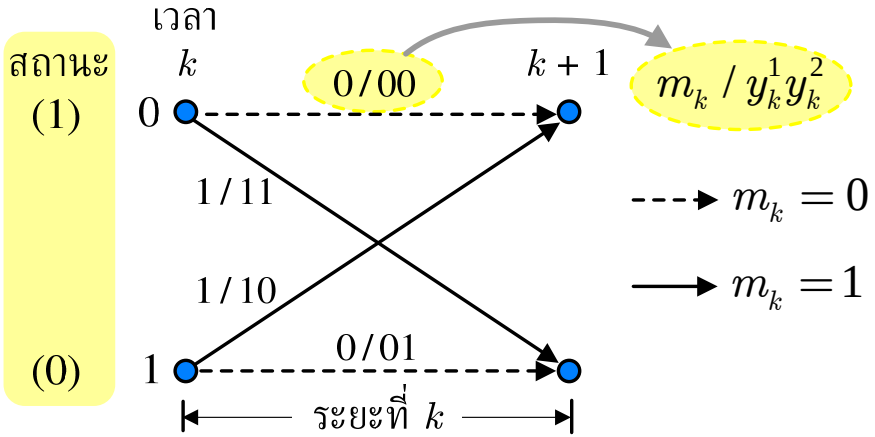
พิจารณาแบบจำลองช่องสัญญาณต่อไปนี้



เมื่อวงจรเข้ารหัสคอนโวลูชันคือ



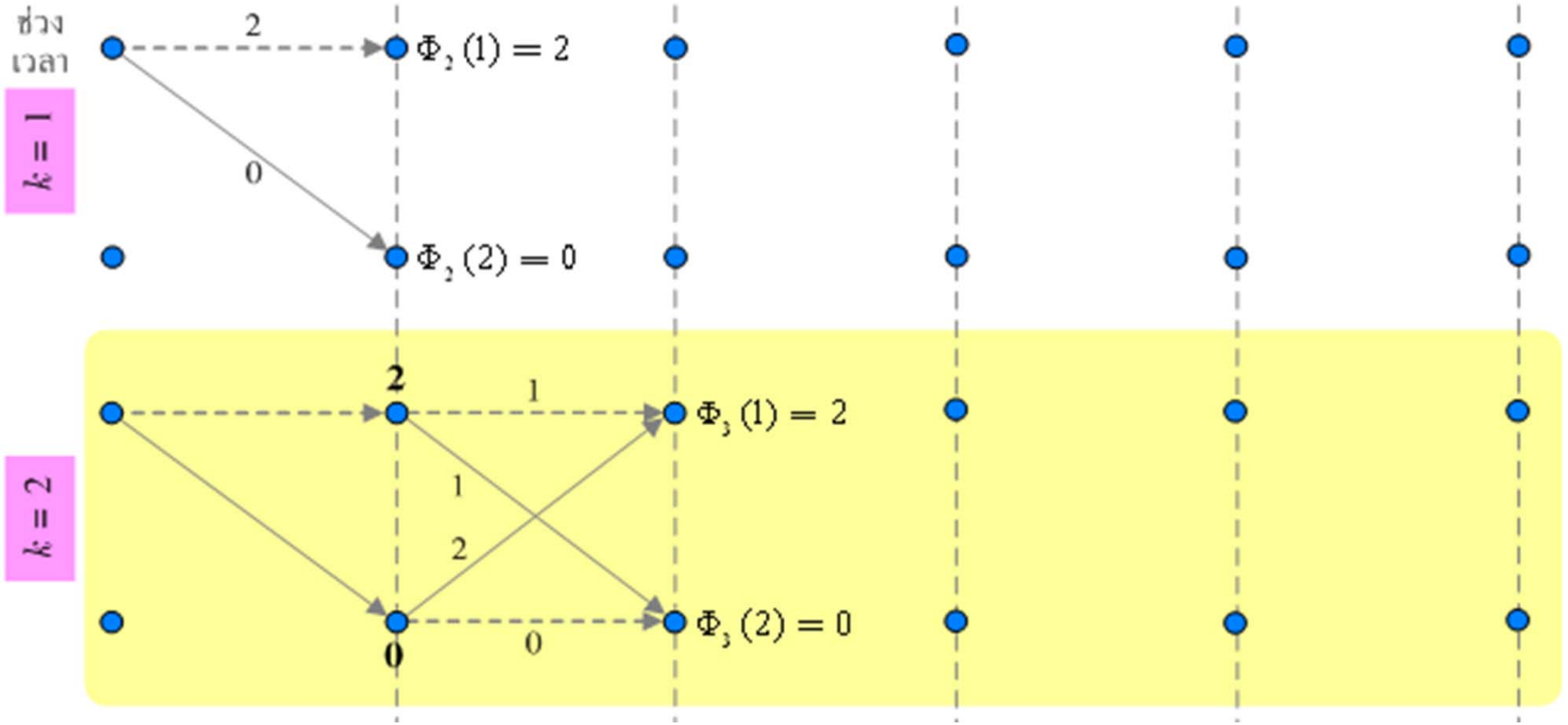
(ก) วงจรเข้ารหัสคอนโวลูชัน

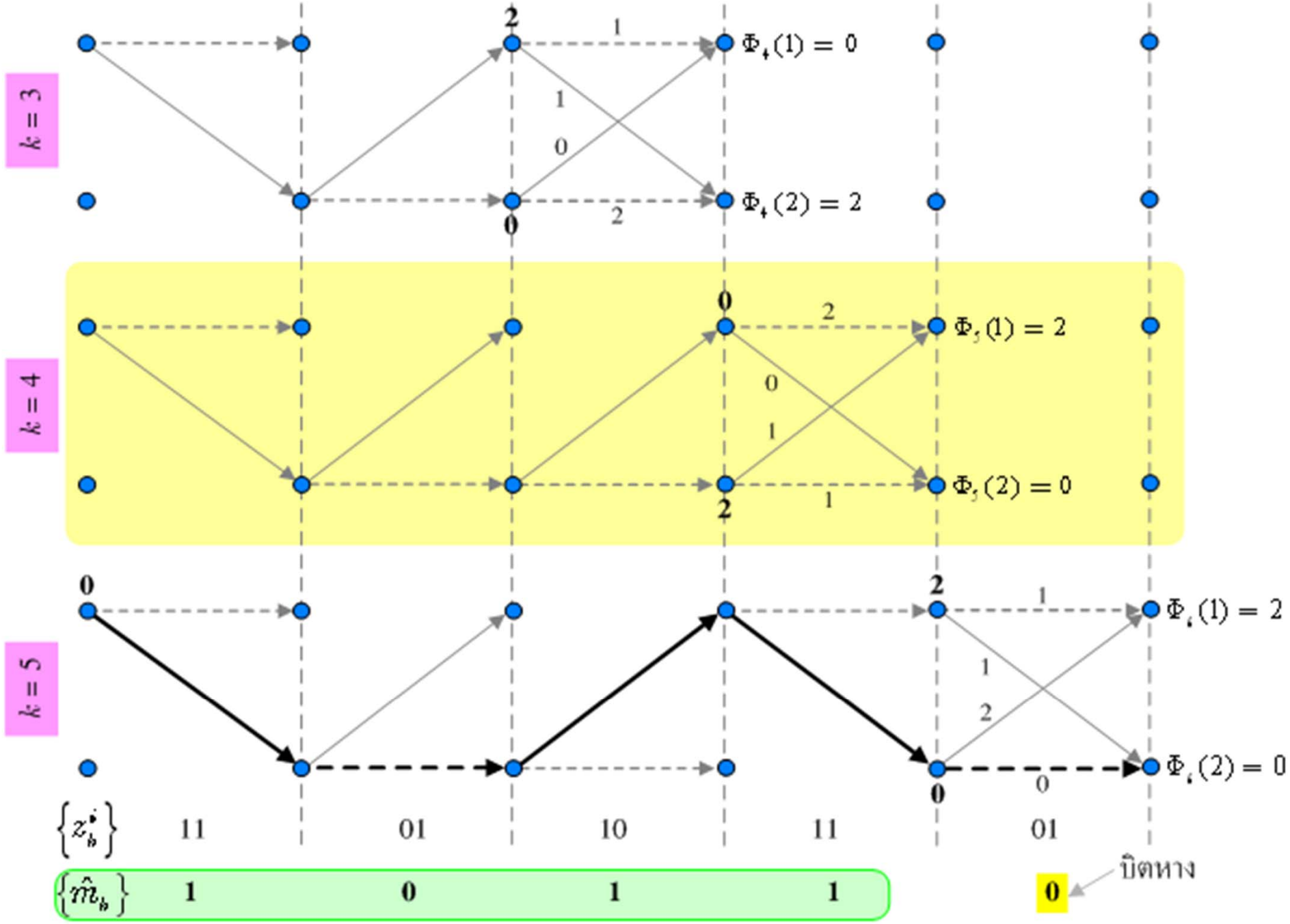


(ข) แผนภาพเทรลลิส

จงถอดรหัสลำดับข้อมูล $\{z_k^i\} = \{11\ 01\ 10\ 11\ 01\}$ ด้วยอัลกอริทึมวีเทอร์บิ โดยสมมุติว่าวงจรภาคส่งไม่ได้ส่งบิตหางมาให้

วิธีทำ รูปต่อไปนี้แสดงขั้นตอนการถอดรหัสข้อมูลด้วยอัลกอริทึมวีเทอร์บิในแต่ละช่วงเวลา k ซึ่งจะแสดงเฉพาะเส้นทางที่ยังมีชีวิตอยู่ที่มาถึงแต่ละสถานะ โดยค่าที่อยู่ติดกับแต่ละเส้นสาขาคือค่าเมตริกสาขา $\lambda_k(u,q)$ ที่สอดคล้องกับการเปลี่ยนสถานะ (u,q) นั้นๆ และตัวเลขที่อยู่ตรงโหนดของแต่ละสถานะคือค่าเมตริกเส้นทาง $\Phi_k(q)$ จากรูปจะได้ว่าวงจรถอดรหัสจะให้ค่าประมาณของบิตข้อมูลอินพุตคือ $\{\hat{m}_k\} = \{1,0,1,1\}$




k = 3
k = 4
k = 5
$\Phi_4(1) = 0$
$\Phi_4(2) = 2$
$\Phi_5(1) = 2$
$\Phi_5(2) = 0$
$\Phi_6(1) = 2$
$\Phi_6(2) = 0$
$\{z_b^i\}$
11
01
10
11
01
$\{\hat{m}_b\}$
1
0
1
1
0
บิตหาง

รหัสช่องสัญญาณที่น่าสนใจ

- ในปัจจุบันรหัส ECC มีให้เลือกใช้งานจำนวนมากตามลักษณะการใช้งานของแต่ละงานประยุกต์

รหัสเทอร์โบ

- กรรมวิธีการเข้าและถอดรหัสช่องสัญญาณที่ได้ถูกพัฒนาขึ้นมาในปี ค.ศ. 1993 โดย Berrou, Glavieux, และ Thitimajshima
  - ข้อดีคือ ทำงานได้ดีแม้ว่าช่องสัญญาณมีค่า SNR ต่ำ, แก้ไขข้อผิดพลาดได้ดี, และมีสมรรถนะเข้าใกล้ขีดจำกัดของแชนนอน โดยอาศัยกระบวนการเข้าและถอดรหัสที่ไม่ซับซ้อน
  - ใช้ในหลายๆ งานประยุกต์ เช่น ระบบโทรศัพท์เคลื่อนที่ยุคที่สาม (3G: third generation) ได้นำรหัสเทอร์โบมาใช้เป็นมาตรฐานสำหรับการติดต่อสื่อสารระหว่างสถานีฐานและเครื่องโทรศัพท์เคลื่อนที่ เป็นต้น

รหัส LDPC

- รหัสตรวจสอบภาวะคู่หรือคี่แบบความหนาแน่นต่ำ (LDPC: low-density parity-check) ⇒ เป็นรหัส ECC ที่ดีสุดในปัจจุบัน เพราะมีสมรรถนะเข้าใกล้ขีดจำกัดของแชนนอนมากกว่ารหัส ECC ชนิดอื่น
- รหัสแอลดีพีซีเป็นรหัสบล็อกเชิงเส้นประเภทหนึ่งที่ถูกกำหนดด้วยเมทริกซ์พาริตีเช็กที่มีจำนวนเลข 1 น้อยมาก เมื่อเทียบกับขนาดของเมทริกซ์พาริตีเช็ก เพื่อให้มีระยะทางน้อยสุด d_{min} ของรหัสสูง
- คิดค้นโดย Gallager ในปี ค.ศ. 1960 ณ MIT ประเทศสหรัฐอเมริกา
- ในช่วงแรกรหัสแอลดีพีซีไม่ได้รับความสนใจเท่าที่ควร เนื่องจากมีข้อจำกัดทางด้านการคำนวณ
- ในปี ค.ศ. 1990 Mackey และ Neal พบว่ารหัสแอลดีพีซีมีสมรรถนะการทำงานที่เข้าใกล้ขีดจำกัดของแชนนอนมากกว่ารหัสเทอร์โบ ⇒ ทำให้รหัสแอลดีพีซีเริ่มกลับมาเป็นที่สนใจอย่างแพร่หลายอีกครั้งหนึ่ง

*** END