บทความวิจัย/บทความวิชาการ

รหัสโพลาร์ : รหัสแก้ไขความผิดพลาดในระบบสื่อสารยุค 5G

POLAR CODES : THE ERROR CORRECTING CODES IN 5G COMMUNICATION SYSTEMS

ณัฐ ตันติบุตร[1] กันต์ ศรีรุ่งโรจน์[2] ลัญฉกร วุฒิสิทธิกุลกิจ[3]
เวธิต ภาคย์พิสุทธิ[4]

Nut Tuntiputra[1] Gun SriRungroj[2] Lunchakorn Wuttisittikulkij[3]
Watid Phakphisut[4]

สำนักกำกับดูแลกิจการโทรคมนาคม
สำนักงานคณะกรรมการกิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคมแห่งชาติ
กรุงเทพฯ 10400 [1]
ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย กรุงเทพฯ 10900 [2 และ 3]
ภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง กรุงเทพ 10520 [4]

Telecommunication Enforcement Bureau, Office of the National Office of the National Broadcasting and Telecommunications Commission, Bangkok 10400 Thailand [1]
Department of Electrical Engineering Faculty of Engineering, Chulalongkorn University, Bangkok 10330 Thailand [2 and 3]
Telecommunications Engineering Department Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520 Thailand [4]

Corresponding E-mail : Nut.t@nbtc.go.th[1]

Received Date September 28, 2018
Revised Date June 20, 2019
Accepted Date June 21, 2019

บทคัดย่อ

บทความฉบับนี้มีวัตถุประสงค์เพื่อนำเสนอองค์ความรู้เกี่ยวกับรหัสโพลาร์ ซึ่งเป็นเป็นการวิจัยเชิงทดลอง (Experimental research) เพื่อนำเสนอรหัสแก้ไขความผิดพลาดชนิดใหม่ที่ได้รับการกำหนดให้ใช้กับมาตรฐานระบบสื่อสารไร้สายในยุคที่ 5 หรือ 5G โครงสร้างพื้นฐานของรหัสโพลาร์ ซึ่งนิยามด้วยพารามิเตอร์ 4 ค่าคือ (N, K, F, v_F) โดยที่ N คือความยาวของคำรหัส K คือความยาวของบิตข้อมูล F คือ ตำแหน่งของบิตแช่แข็งจำนวน $N-K$ ค่า และคือ v_F เวกเตอร์ขนาด $N-K$ ที่บรรจุค่าไบนารี 0 หรือ 1 การเข้ารหัสโพลาร์สามารถทำได้โดยสร้างเมทริกตัวกำเนิด G ซึ่งคือการหาค่าผลคูณครอนเนกเกอร์ของ F จำนวน n ครั้งแล้วกำหนดค่าบิตแช่แข็ง จากนั้นทำการประเมินสมรรถนะของรหัสโพลาร์บนช่องสัญญาณ additive white Gaussian noise (AWGN) การถอดรหัสโพลาร์ทำได้โดยการแปลงค่าสัญญาณที่รับได้ในแต่ละบิตเป็นค่าอัตราส่วนความควรจะเป็น (Likelihood Ratio : LR) จากนั้นทำการถอดรหัสโพลาร์ ผลการทดสอบพบว่าอัตรารหัสที่ 2/3 1/2 และ 1/3 นั้นมีสมรรถนะค่าความผิดพลาดที่ดีขึ้นตามลำดับ แต่จะใช้เวลาในการส่งข้อมูลมากขึ้นตามไปด้วย

คำสำคัญ : รหัสโพลาร์ ตัวถอดรหัสหักล้างอย่างต่อเนื่อง ช่องสัญญาณ AWGN อัตรารหัส อัตราส่วนความควรจะเป็น

Abstract

This study presents the details of Polar codes which is an experimental research for new type of error correcting codes adopted for the new standard in the 5th Generation of wireless communication systems or 5G. The structure of polar codes defines by 4 parameters (N, K, F, v_F) ; N is length of codewords; K is length of data bits; F is a position of frozen bits equal to $N-K$ and v_F is vector $N-K$ that contains binary bit 0 or 1. The encoding of the polar codes was created by the generator matrix $\mathbf{G}$ which is the multiply Kronecker product of F for n times then set a position of forzen bits. Next, the perfomance of polar codes was estimated on additive white Gaussian noise (AWGN) channel. Decoding the polar codes was done by changing the signal form AWGN channel to Likelihood Ratio (LR). The bit error rate was also evaluated performances over additive white Gaussian noise (AWGN) channel. The stimulation in which of decoding the structure of polar codes at 2/3 has bit error performance better that at 1/2 and 1/3 respectively. On the other hand, the sending time for decoding of polar codes at 2/3 has greater than 1/2 and 2/3 respectively. According to this finding, the performance of bit error rate at 2/3 1.2 and 2/3 is chronological better. However, the processes for sending the data require more time period.

Keywords : Polar codes, Successive cancellation decoder, AWGN channel, Code rate, Log-Likelihood Ratio.

1. บทนำ

รหัสโพลาร์ (Polar codes) เป็นรหัสแก้ไขความผิดพลาดที่น่าสนใจยิ่งเพราะเป็นรหัสชนิดแรกที่สามารถพิสูจน์ได้อย่างเป็นระบบว่ารหัสดังกล่าวสามารถให้ความจุช่องสัญญาณได้ตามทฤษฎีข่าวสารของแชนเนล รหัสชนิดนี้ได้รับการพัฒนาขึ้นครั้งแรกโดย E. Arikan ใน ค.ศ. 2008 และได้รับการเผยแพร่ในวงกว้างในบทความวิจัย เรื่อง “Channel polarization: A method for constructing capacity achieving codes for symmetric binary-input memoryless channels” ที่ได้รับการตีพิมพ์ในวารสาร IEEE Transaction on Information Theory ใน ค.ศ. 2009 รหัสชนิดใหม่นี้ได้รับความสนใจอย่างมากจากทั้งนักวิชาการและภาคอุตสาหกรรม เพราะนอกจากจะให้สมรรถนะที่ดีเยี่ยมเทียบได้กับรหัสแอลดีพีซีแล้วการเข้ารหัสที่ภาคส่งมีโครงสร้างเรียบง่าย เป็นระบบ และไม่ซับซ้อน ในขณะที่การถอดรหัสที่ภาครับสามารถทำได้โดยใช้เทคนิคการหักล้างอย่างต่อเนื่อง (Successive cancellation) (Arikan, 2009) ที่มีความซับซ้อนน้อย ทำงานได้รวดเร็วและมีประสิทธิภาพ จึงเป็นทางเลือกใหม่ที่สำคัญสำหรับการประยุกต์ใช้งานในทางปฏิบัติ โดยในปัจจุบันรหัสโพลาร์ถูกนำไปใช้เป็นส่วนหนึ่งของมาตรฐานการสื่อสารไร้สายยุคที่ 5 หรือ 5G (Bioglio, Condo, & Land, 2018)

เนื้อหาที่จะกล่าวถึงในบทความฉบับนี้เริ่มต้นจากการศึกษาการกำหนดพารามิเตอร์ของรหัสโพลาร์ การสร้างรหัสโพลาร์ วิธีการเข้ารหัสและวิธีการถอดรหัสของรหัสโพลาร์ หลังจากได้ศึกษาเรื่องต่าง ๆ เกี่ยวกับรหัสโพลาร์จึงนำความรู้ที่ได้มาพัฒนาโปรแกรมจำลองการเข้ารหัสและถอดรหัสของรหัสโพลาร์ เพื่อที่จะประเมินสมรรถนะของรหัสโพลาร์โดยใช้การจำลองด้วยคอมพิวเตอร์ สำหรับช่องสัญญาณ Additive White Gaussian noise (AWGN) โดยประเมินสมรรถนะจากค่าอัตราความผิดพลาดบิต (Bit error rate) และค่าอัตราความผิดพลาดเฟรม (Frame error rate) ของรหัสโพลาร์ ภายใต้อัตราส่วนของสัญญาณที่ต้องการต่อสัญญาณรบกวนที่ต่างกัน เพื่อเปรียบเทียบค่าอัตราความผิดพลาดบิตและค่าอัตราความผิดพลาดเฟรมของรหัสโพลาร์ที่มีความยาวของคำรหัสค่าต่าง ๆ

2. โครงสร้างพื้นฐานและการเข้ารหัสโพลาร์

2.1 การนิยามรหัสโพลาร์

การนิยามรหัสโพลาร์สามารถทำได้โดยใช้พารามิเตอร์ 4 ค่า คือ (N, K, F, v_F) โดยที่

N คือ ความยาวของคำรหัส

K คือ ความยาวของบิตข้อมูล

F คือ ตำแหน่งของบิตแช่แข็ง (Frozen bit locations) จำนวน $N-K$ ค่า และ

v_F คือ เวกเตอร์ขนาด $N-K$ ที่บรรจุค่าไบนารี 0 หรือ 1

ยกตัวอย่างเช่น ในการเข้ารหัสโพลาร์สำหรับบิตข้อมูลจำนวน $K=4$ บิต เพื่อสร้างให้เป็นเวกเตอร์ที่มีขนาด $N=8$ บิต โดยกำหนด $F=\{0, 1, 2, 4\}$ และ v_F = [0 0 0 0] มีรายละเอียดดังนี้

1) กำหนดให้บิตข้อมูลเขียนในรูปของเวกเตอร์ $\mathrm{m} = [m_1 \quad m_2 \quad m_3 \quad m_4]^T$

2) นำบิตข้อมูลในเวกเตอร์ m เหล่านี้ไปประกอบเข้ากับบิตแช่แข็ง v_F เพื่อให้ได้เป็นเวกเตอร์ d ที่มีความยาว $N=8$ บิต ทั้งนี้ตำแหน่งของบิตแช่แข็งกำหนดให้วาง ณ ตำแหน่งที่ระบุใน F ฉะนั้น ผลที่ได้คือ $\mathrm{d} = [0 \quad 0 \quad 0 \quad m_1 \quad 0 \quad m_2 \quad m_3 \quad m_4]^T$

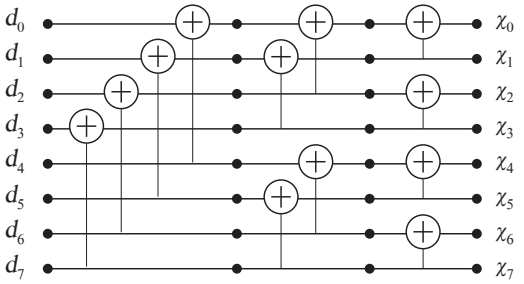
3) นำเวกเตอร์ d ที่ได้นี้ไปผ่านการแปลงด้วยเมเทริกซ์ตัวกำเนิด (Generator matrix) หรือเมทริกซ์ G โดยใช้สมการดังต่อไปนี้

$$\mathrm{x} = \mathrm{G}\,\mathrm{d}$$

หรือ

$$X = \begin{bmatrix} 1&1&1&1&1&1&1&1 \\ 0&1&0&1&0&1&0&1 \\ 0&0&1&1&0&0&1&1 \\ 0&0&0&1&0&0&0&1 \\ 0&0&0&0&1&1&1&1 \\ 0&0&0&0&0&1&0&1 \\ 0&0&0&0&0&0&1&1 \\ 0&0&0&0&0&0&0&1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ m_1 \\ 0 \\ m_2 \\ m_3 \\ m_4 \end{bmatrix} = \begin{bmatrix} m_1 + m_2 + m_3 + m_4 \\ m_1 + m_2 + m_4 \\ m_1 + m_3 + m_4 \\ m_1 + m_4 \\ m_2 + m_3 + m_4 \\ m_2 + m_4 \\ m_3 + m_4 \\ m_4 \end{bmatrix}$$

โดย X คือคำรหัสที่ได้จากการเข้ารหัสข้อมูลตามหลักการของรหัสโพลาร์ และพร้อมส่งไปยังภาครับ ทั้งนี้ กระบวนการแปลงด้วยเมทริกซ์ตัวกำเนิด G เขียนเป็นแผนภาพวงจรได้ดังในภาพที่ 1 โดย ⊕ เป็นการบวกแบบโมดูโล 2 (mod-2)



ภาพที่ 1 แผนภาพการเข้ารหัสของรหัสโพลาร์ที่มีความยาวของคำรหัสเท่ากับ 8

ที่มา : ณัฐ ตันติบุตร และคณะ (2562)

2.2 การสร้างเมทริกซ์ตัวกำเนิด (G)

การสร้างเมทริกซ์ตัวกำเนิดทำได้โดยกำหนดให้เมทริกซ์ F มีค่าเท่ากับ $\begin{bmatrix} 1&1 \\ 0&1 \end{bmatrix}$ จะได้ว่าเมทริกซ์ตัวกำเนิดมีค่าเท่ากับ $G = F^{\otimes n} = F \otimes ... F \otimes$ โดย $n = \log_2(N)$ ซึ่งคือการหาค่าผลคูณครอนเนกเกอร์ (Kronecker product) ของ F จำนวน n ครั้ง

สำหรับ

$$N = 4 : F^{\otimes 2} = F \otimes F = \begin{bmatrix} 1&1&1&1 \\ 0&1&0&1 \\ 0&0&1&1 \\ 0&0&0&1 \end{bmatrix}$$

สำหรับ $N = 8$:

$$F^{\otimes 3} = (F \otimes F) \otimes F = \begin{bmatrix} 1&1&1&1 \\ 0&1&0&1 \\ 0&0&1&1 \\ 0&0&0&1 \end{bmatrix} \otimes \begin{bmatrix} 1&1 \\ 0&1 \end{bmatrix} = \begin{bmatrix} 1&1&1&1&1&1&1&1 \\ 0&1&0&1&0&1&0&1 \\ 0&0&1&1&0&0&1&1 \\ 0&0&0&1&0&0&0&1 \\ 0&0&0&0&1&1&1&1 \\ 0&0&0&0&0&1&0&1 \\ 0&0&0&0&0&0&1&1 \\ 0&0&0&0&0&0&0&1 \end{bmatrix}$$

หมายเหตุ นิยามของผลคูณครอนเนกเกอร์ (Kronecker product) เป็นดังนี้

กำหนดให้ $A=\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ และ $B=\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$ เมื่อนำเมทริกซ์ A ครอนเนกเกอร์กับ B จะได้

$$A \otimes B=\begin{bmatrix} a_{11}B & a_{12}B \\ a_{21}B & a_{22}B \end{bmatrix}=\begin{bmatrix} a_{11}b_{11} & a_{11}b_{12} & a_{12}b_{11} & a_{12}b_{12} \\ a_{11}b_{21} & a_{11}b_{22} & a_{12}b_{21} & a_{12}b_{22} \\ a_{21}b_{11} & a_{21}b_{12} & a_{22}b_{11} & a_{22}b_{12} \\ a_{21}b_{21} & a_{21}b_{22} & a_{12}b_{21} & a_{22}b_{22} \end{bmatrix}$$

2.3 วิธีการกำหนดตำแหน่งของบิตแซ่แข็ง

ในการสร้างรหัสโพลาร์ให้มีคุณสมบัติที่ดีตามต้องการขึ้นอยู่กับการเลือกค่าให้กับเซต F เป็นสำคัญ โดยที่ผ่านมามีงานวิจัยเกี่ยวกับวิธีการเลือกที่เหมาะสมและมีประสิทธิภาพอยู่พอสมควร ดูได้จากเอกสารอ้างอิง (Arikan, 2008) (Tal & Vardy, 2013) และ (Zhao, Shi, & Wang, 2011) ในที่นี้เราพิจารณาการเลือกค่าให้กับเซต F ตามวิธีแบบดั้งเดิมที่เสนอขึ้นใน (Arikan, 2008) กระบวนการทำงานของอัลกอริทึมที่พิจารณานี้ใช้กรรมวิธีแบบรีเคอร์ซีฟสำหรับ $j=0, 1,...,n\text{-}1$ รายละเอียดเป็นดังนี้

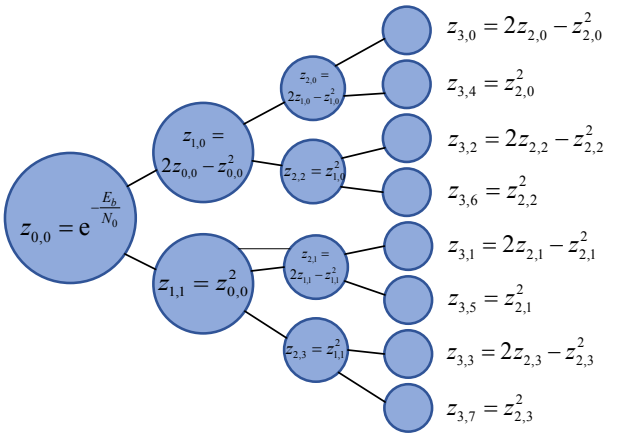
1) กำหนดค่าตั้งต้นโดยให้ $Z_{0,0}=\exp(-E_b/N_0)$ โดย E_b คือ พลังงานต่อบิต และ N_0 คือความหนาแน่นสเปกตรัมกำลังของสัญญาณรบกวน ทั้งนี้ E_b/N_0 มีหน่วยเป็นเดซิเบล (dB)

2) คำนวณหาค่า $Z_{j+1,i}$ โดยใช้สมการที่ (1)

$$Z_{j+1,i}=\begin{cases} 2z_{j,i}-z^2_{j,i}, & \text{for } 0 \le i < 2^j \\ z^2_{j,i-2^j}, & \text{for } 2^j \le i < 2^{j+1} \end{cases} \quad (1)$$

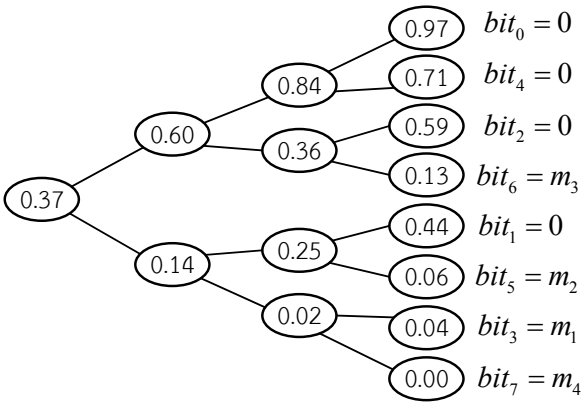
3) จากนั้นจะกำหนดให้ตำแหน่งที่ z มีค่าสูงสุด $N-K$ ค่า เป็นตำแหน่งของบิตแซ่แข็ง

หมายเหตุ เมื่อการคำนวณเสร็จสิ้นลง i เป็นดรรชนีระบุค่าตำแหน่งแต่ละบิตของคำรหัส



ภาพที่ 2 การคำนวณหาตำแหน่งของบิตแซ่แข็งตามสมการที่ (1)

ยกตัวอย่างเช่น กำหนดให้ $E_b / N_0 = 1$, $N = 8$ และ $K = 4$



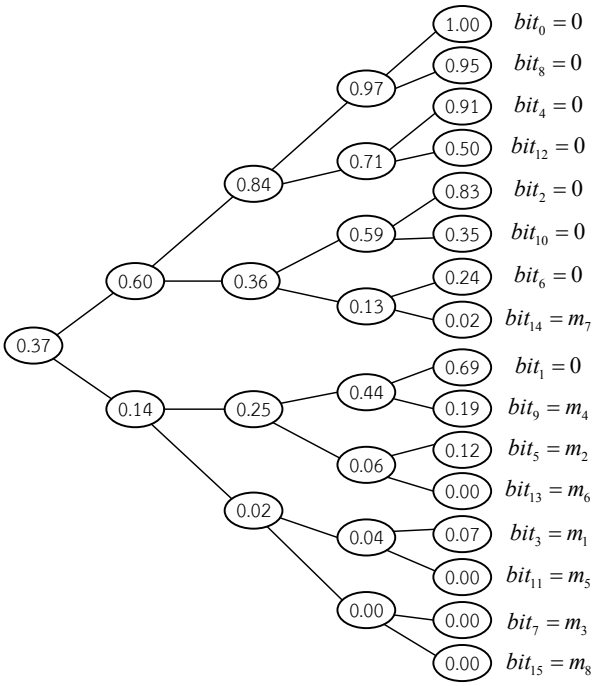
ภาพที่ 3 การคำนวณหาตำแหน่งของบิตแซ่แข็ง เมื่อ $N = 8$ และ $K = 4$

ตารางที่ 1 ตำแหน่งของบิตข้อมูลและค่า SNR ต่าง ๆ

	SNR_{dB}									
	-150	-100	0	1	2	3	4	5	10	15
d_0	m_1	m_1	0	0	0	0	0	0	0	0
d_1	m_2	m_2	0	0	0	0	0	0	0	0
d_2	m_3	0	0	0	0	0	0	0	0	0
d_3	0	m_3	m_1	m_1	m_1	m_1	m_1	m_1	m_1	m_1
d_4	0	0	0	0	0	0	0	0	0	0
d_5	0	0	m_2	m_2	m_2	m_2	m_2	m_2	m_2	m_2
d_6	0	0	m_3	m_3	m_3	m_3	m_3	m_3	m_3	m_3
d_7	m_4	m_4	m_4	m_4	m_4	m_4	m_4	m_4	m_4	m_4

ตัวอย่างที่ 1 กำหนดให้

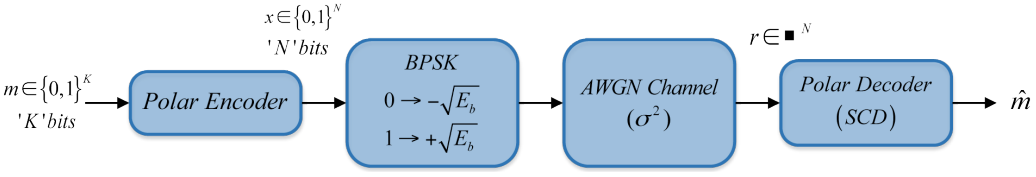
$\mathrm{m} = [m_1 \ m_2 \ m_3 \ m_4 \ m_5 \ m_6 \ m_7 \ m_8]^T = [\,1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1\,]^T$ โดยที่ $E_b / N_0 = 1$, $N = 16$ และ $K = 8$



ภาพที่ 4 การเข้ารหัสของรหัสโพลาร์ เมื่อ $N = 16$ และ $K = 8$

จะได้ว่า $d = [\,0\ \ 0\ \ 0\ \ \underline{1}\ \ 0\ \ \underline{0}\ \ 0\ \ \underline{0}\ \ 0\ \ \underline{1}\ \ 0\ \ \underline{1}\ \ 0\ \ \underline{1}\ \ \underline{1}\ \ \underline{1}\,]^T$ และคำรหัสที่ได้จากการเข้ารหัสจะเป็น $x = G \cdot d = [\,0\ \ 0\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 0\ \ 0\ \ 1\ \ 1\ \ 1\ \ 1\,]^T$

2.4 การถอดรหัสโพลาร์ โดยตัวถอดรหัสหักล้างอย่างต่อเนื่อง (Successive cancellation decoder)



ภาพที่ 5 แบบจำลองภาคการเข้ารหัส ช่องสัญญาณ AWGN และภาคการถอดรหัสของรหัสโพลาร์

หลังจากที่ข้อมูลเดินทางผ่านช่องสัญญาณมายังฝั่งรับ ขั้นตอนแรกในการถอดรหัสคือการแปลงค่าสัญญาณที่รับได้ r_i แต่ละบิตไปเป็นค่าอัตราส่วนความควรจะเป็นดังสมการต่อไปนี้

$$L = \frac{P(x_i = 0 \mid r_i)}{P(x_i = 1 \mid r_i)} = \frac{P(x_i = 0, r_i) / P(r_i)}{P(x_i = 1, r_i) / P(r_i)} = \frac{P(r_i \mid x_i = 0)\, P(x_i = 0)}{P(r_i \mid x_i = 1)\, P(x_i = 1)}$$
$$= \frac{\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(r_i + \sqrt{E_b})^2}{2\sigma^2}}}{\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(r_i - \sqrt{E_b})^2}{2\sigma^2}}} \qquad (2)$$
$$= e^{-\frac{2\sqrt{E_b} r_i}{\sigma^2}}$$

โดย r_i คือค่าของสัญญาณที่รับได้แต่ละบิตโดย $r_i = (2x_i - 1)\sqrt{E_b} = n_i$

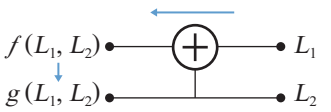
x_i คือค่าบิตแต่ละบิตที่ภาคส่งได้ส่งออกมามีค่าได้คือ 0 หรือ 1

n_i คือสัญญาณรบกวนที่มีการแจกแจงแบบเกาส์เซียน มีค่าเฉลี่ยเป็น 0 และมีความแปรปรวนเป็น σ^2

E_b คือพลังงานต่อบิต

เนื่องจากโครงสร้างของแผนภาพการเข้ารหัสโพลาร์มีคุณลักษณะเฉพาะและมีรูปแบบตายตัว ดังที่แสดงในภาพที่ 1 โครงสร้างนี้มีคุณสมบัติพิเศษที่เป็นประโยชน์ยิ่งคือ การเข้ารหัสและการถอดรหัสสามารถใช้โครงสร้างแผนภาพเดียวกันได้ โดยที่ในขั้นตอนการเข้ารหัสจะพิจารณาจากด้านซ้ายมือไปหาทางด้านขวามือ ในขณะที่ขั้นตอนการถอดรหัสให้พิจารณาในทิศตรงข้าม นั่นคือพิจารณาจากด้านขวามือไปยังทางด้านซ้ายมือ

จากค่าความควรจะเป็นที่คำนวณได้จากสัญญาณที่รับได้แต่ละบิตซึ่งจะเขียนกำกับไว้ที่ฝั่งขวามือสุดของแผนภาพตามตำแหน่งของแต่ละบิต ในขั้นตอนถัดมาให้คำนวณค่าอัตราส่วนความควรจะเป็นย้อนกลับจากด้านขวามือไปยังด้านซ้ายมือ ในการคำนวณย้อนกลับให้พิจารณาวงจรหน่วย (Unit circuit) ที่เป็นส่วนประกอบย่อยของแผนภาพดังแสดงในภาพที่ 6 อนึ่ง การทำงานของรหัสโพลาร์ประกอบขึ้นจากวงจรหน่วยที่เชื่อมต่อกันอย่างเป็นระบบตามที่ได้อธิบายไว้ข้างต้น ดังนั้น หัวใจของการถอดรหัสจึงอยู่ที่วิธีการคำนวณย้อนกลับจากขวามือไปทางซ้ายมือของวงจรหน่วยแต่ละตัวซึ่งจะได้อธิบายอย่างละเอียดในส่วนถัดไป



ภาพที่ 6 การคำนวณค่าอัตราส่วนความควรจะเป็นของวงจรหน่วย

พิจารณาวงจรหน่วย โดยในการพิจารณาจะแบ่งเป็น 2 เรื่อง

1) การคำนวณค่าอัตราส่วนความควรจะเป็น

พิจารณาจากวงจรหน่วย ค่าของ $f(L_1, L_2)$ จะถูกคำนวณก่อนค่า $g(L_1, L_2)$ เสมอโดยสามารถคำนวณได้จากค่าอัตราส่วนความควรจะเป็น L_1 และ L_2 ตามความสัมพันธ์ดังนี้

$$f(L_1, L_2) = \frac{L_1 L_2 + 1}{L_1 + L_2} \tag{3}$$

เมื่อทราบค่าของ $f(L_1, L_2)$ แล้ว การคำนวณค่าของ $g(L_1, L_2)$ จะกำหนดให้มีค่าเท่ากับ $L_2 L_1$ หากผลการตัดสินใจบิตจาก $f(L_1, L_2)$ เป็นบิต 0 และจะกำหนดให้ $g(L_1, L_2)$ มีค่าเท่ากับ L_2 / L_1 หากผลการตัดสินใจบิตจาก $f(L_1, L_2)$ เป็นบิต 1 เพื่อความสะดวกจะเขียนแสดงวิธีการคำนวณ $f(L_1, L_2)$ และ $g(L_1, L_2)$ ในรูปแบบที่กระชับได้ดังนี้

$$\begin{pmatrix} f(L_1, L_2) \\ g(L_1, L_2) \end{pmatrix} = \begin{pmatrix} \dfrac{L_1 L_2 + 1}{L_1 + L_2} \\ L_2 L_1 \ or \ L_2 / L_1 \end{pmatrix} \tag{4}$$

การคำนวณค่าอัตราส่วนความควรจะเป็นมักจะพิจารณาในรูปของล็อก หรือที่เรียกว่าค่าอัตราส่วนความควรจะเป็นแบบล็อก (Log-Likelihood Ratio : LLR) มากกว่า เพราะนอกจากจะช่วยให้การวิเคราะห์ทางคณิตศาสตร์มีรูปแบบที่เรียบง่ายขึ้นแล้ว ยังช่วยลดปัญหาความแม่นยำของการคำนวณเชิงตัวเลขของคอมพิวเตอร์ในกรณีที่ต้องมีการคูณค่าความน่าจะเป็นจำนวนหลายตัวเข้าด้วยกันซึ่งจะได้เป็นค่าตัวเลขที่มีขนาดเล็กมาก ๆ ค่าอัตราส่วนความควรจะเป็นแบบล็อกมีนิยามดังนี้

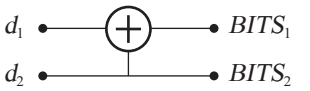
$$I = \ln L = \ln \frac{P = (x_i = 0 \mid r_i)}{P = (x_i = 1 \mid r_i)} = \ln \frac{\dfrac{1}{1+e^{\frac{2\sqrt{E_b} r_i}{\sigma^2}}}}{\dfrac{e^{\frac{2\sqrt{E_b} r_i}{\sigma^2}}}{1+e^{\frac{2\sqrt{E_b} r_i}{\sigma^2}}}} = \ln \frac{\dfrac{1}{1+e^{\frac{2\sqrt{E_b} r_i}{\sigma^2}}}}{\dfrac{e^{\frac{2\sqrt{E_b} r_i}{\sigma^2}}}{e^{\frac{2\sqrt{E_b} r_i}{\sigma^2}}+1}} = \ln e^{-\frac{2\sqrt{E_b} r_i}{\sigma^2}} = -\frac{2\sqrt{E_b} r_i}{\sigma^2} \tag{5}$$

ในกรณีที่ใช้ค่าอัตราส่วนความควรจะเป็นแบบล็อก สมการที่ (4) จะเขียนได้เป็น

$$\begin{pmatrix} \ln f(e^{I_1}, e^{I_2}) \\ \ln g(e^{I_1}, e^{I_2}) \end{pmatrix} = \begin{pmatrix} \ln \left(\dfrac{(e^{I_2 + I_1} + 1)}{(e^{I_1} + e^{I_2})} \right) \\ I_2 + I_1 \ or \ I_2 - I_1 \end{pmatrix} \tag{6}$$

2) การตัดสินใจบิต

ในการถอดรหัสการตัดสินใจบิตจะทำในทิศทางจากด้านขวามือไปหาทางด้านซ้ายมือ เมื่อทราบค่า d_1 และ d_2 ว่ามีค่าบิตเป็นเท่าไหร่ จะสามารถหาค่าบิตของ $BITS_1$, $BITS$ ได้ตามตารางที่ 2



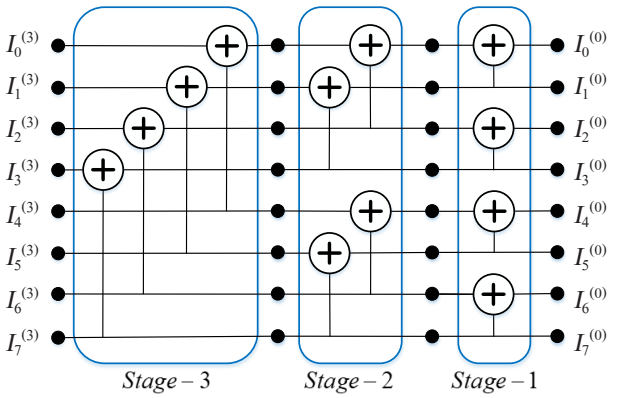
ภาพที่ 7 การตัดสินใจบิตของวงจรหน่วย

ตารางที่ 2 ค่า $BITS_1$, $BITS_2$ เมื่อ d_1 , d_2 มีค่าต่าง ๆ

	$BITS_1$	$BITS_2$
$d_1 = 0, d_2 = 0$	0	0
$d_1 = 0, d_2 = 1$	1	1
$d_1 = 1, d_2 = 0$	1	0
$d_1 = 1, d_2 = 1$	0	1

ลำดับของบิตในการถอดรหัส

ลำดับของบิตที่จะทำการถอดรหัสจะมีความพิเศษคือจะเรียงลำดับตาม Bit-reversal order ซึ่งเป็นเอกลักษณ์เฉพาะตัวสำหรับการถอดรหัสวิธีการนี้ ยกตัวอย่างเช่นกรณี $N = 8$ ลำดับก่อนที่จะผ่านกระบวนการ bit-reversal เป็น 0, 1, 2, 3, 4, 5, 6, 7 จะสามารถเขียนเป็นเลขฐานสองได้ดังนี้ 000, 001, 010, 011, 100, 101, 110, 111 เมื่อผ่านกระบวนการ Bit-reversal จะกลายเป็น 000, 100, 010, 110, 001, 101, 011, 111 นั่นคือจะได้ลำดับเป็น 0, 4, 2, 6, 1, 5, 3, 7



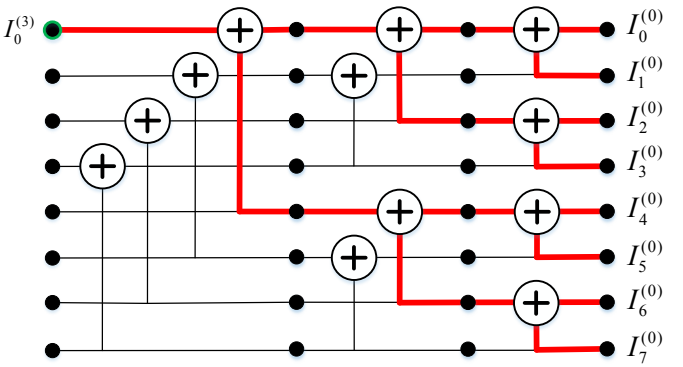
ภาพที่ 8 แผนภาพการถอดรหัสของรหัสโพลาร์ที่มีความยาวของคำรหัสเท่ากับ 8

$I_N^{(n)}$ หมายถึงค่าความควรจะเป็นแบบล็อกของบิตที่ N ใน stage-n

$BITS_N^{(n)}$ หมายถึงค่าของการตัดสินในบิตของบิตที่ N ใน stage-n

ขั้นตอนการถอดรหัสสำหรับรหัสโพลาร์ $(N, K, F, \mathrm{v}_F) = (8, 4, \{0,1,2,4\}, [0\ \ 0\ \ 0\ \ 0])$

<u>บิตที่ 0</u> เริ่มพิจารณาจากบิตที่ 0 ตามลำดับของ Bit-reversal order จะสังเกตได้ว่าในแต่ละวงจรหน่วยของเส้นทางสีแดงจะเป็นการคำนวณค่าอัตราส่วนความควรจะเป็นแบบล็อกของกิ่งด้านบนวงจรของหน่วยทั้งหมด



ภาพที่ 9 แผนภาพเส้นทางการคำนวณค่าอัตราส่วนความควรจะเป็นแบบล็อกของบิตที่ 0

เริ่มคำนวณจาก stage-1 โดยจะคำนวณค่าอัตราส่วนความควรจะเป็นแบบล็อกได้ว่า

$$I_0^{(1)} = \ln\left(\frac{e^{I_0^{(0)}+I_1^{(0)}}+1}{e^{I_0^{(0)}}+e^{I_1^{(0)}}}\right),\ I_2^{(1)} = \ln\left(\frac{e^{I_2^{(0)}+I_3^{(0)}}+1}{e^{I_2^{(0)}}+e^{I_3^{(0)}}}\right),\ I_4^{(1)} = \ln\left(\frac{e^{I_4^{(0)}+I_5^{(0)}}+1}{e^{I_4^{(0)}}+e^{I_5^{(0)}}}\right),\ I_6^{(1)} = \ln\left(\frac{e^{I_6^{(0)}+I_7^{(0)}}+1}{e^{I_6^{(0)}}+e^{I_7^{(0)}}}\right)$$

หลังจากนั้นนำค่าอัตราส่วนความควรจะเป็นแบบล็อกที่ได้จาก stage-1 ไปคำนวณใน stage-2 จะได้

$$I_0^{(2)} = \ln\left(\frac{e^{I_0^{(1)}+I_2^{(1)}}+1}{e^{I_0^{(1)}}+e^{I_2^{(1)}}}\right),\ I_4^{(2)} = \ln\left(\frac{e^{I_4^{(1)}+I_6^{(1)}}+1}{e^{I_4^{(1)}}+e^{I_6^{(1)}}}\right)$$

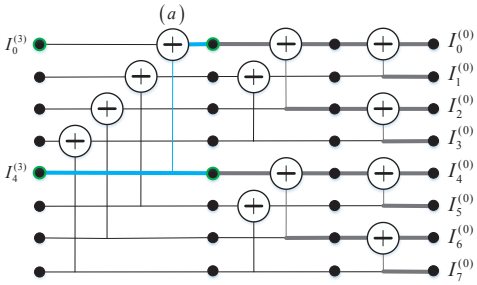
หลังจากนั้นนำค่าอัตราส่วนความควรจะเป็นแบบล็อกที่ได้จาก stage-2 ไปคำนวณใน stage-3 จะได้

$$I_0^{(3)} = \ln\left(\frac{e^{I_0^{(2)}+I_4^{(2)}}+1}{e^{I_0^{(2)}}+e^{I_4^{(2)}}}\right)$$

และเมื่อคำนวณได้ค่าอัตราส่วนความควรจะเป็นแบบล็อกใน stage-3 จะสามารถตัดสินใจได้ว่าบิตที่ 0 มีค่าเป็นเท่าไหร่ แต่สำหรับบิตที่ทราบอยู่แล้วว่าเป็นบิตแช่แข็งซึ่งมีค่าบิตเป็น 0 จะตัดสินใจทันทีว่ามีค่าเป็น 0 ($BITS_0^{(3)} = 0$) โดยไม่ต้องพิจารณาจากค่าอัตราส่วนความควรจะเป็นแบบล็อก

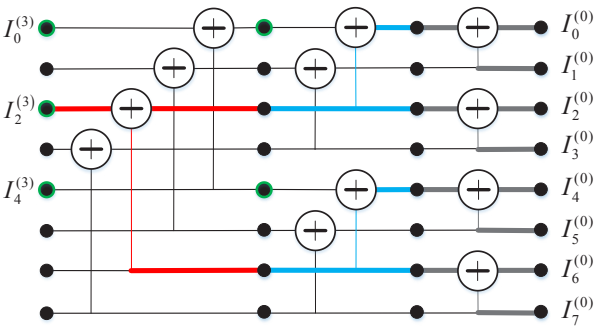
หมายเหตุ วงกลมสีเขียวแสดงถึงตำแหน่งที่ตัดสินใจบิตเรียบร้อยแล้ว

<u>บิตที่ 4</u> ลำดับต่อไปจะเป็นบิตที่ 4 จากการคำนวณค่าอัตราส่วนความควรจะเป็นแบบล็อกของกิ่งด้านของบนวงจรหน่วย (a) แล้ว จึงสามารถหาค่าอัตราส่วนความควรจะเป็นแบบล็อกของกิ่งด้านล่างของวงจรหน่วย (a) ได้ จากที่ตัดสินใจว่าบิตที่ 0 เป็น 0 ($BITS_0^{(3)} = 0$) จะได้ว่า $I_4^{(3)} = I_0^{(2)} + I_4^{(2)}$ เมื่อคำนวณได้ค่าอัตราส่วนความควรจะเป็นแบบล็อกใน stage-3 จะสามารถตัดสินใจได้ว่าบิตที่ 4 มีค่าเป็นเท่าไหร่ แต่เนื่องจากทราบอยู่แล้วว่าบิตที่ 4 เป็นบิตแช่แข็งจึงตัดสินใจว่ามีค่าเป็น 0 ($BITS_4^{(3)} = 0$) และหลังจากตัดสินใจบิตที่ 0 และ 4 แล้วจะสามารถตัดสินใจค่า $BITS_0^{(2)} = 0$ และ $BITS_4^{(2)} = 0$ ได้



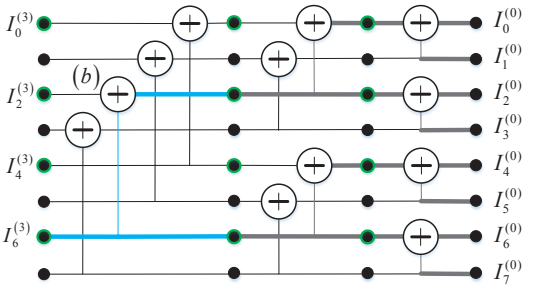
ภาพที่ 10 แผนภาพเส้นทางการคำนวณค่าอัตราส่วนความควรจะเป็นแบบล็อกของบิตที่ 4

<u>บิตที่ 2</u> เริ่มคำนวณจาก stage-2 โดยจะสามารถหาค่า $I_2^{(2)}$ และ $I_6^{(2)}$ ได้ โดย $I_2^{(2)}$ จะเท่ากับ $I_2^{(1)} + I_0^{(1)}$ หรือ $I_2^{(1)} - I_0^{(1)}$ ขึ้นอยู่กับว่า $BITS_0^{(2)}$ มีค่าเป็นเท่าไหร่และ $I_6^{(2)}$ จะเท่ากับ $I_6^{(1)} + I_4^{(1)}$ หรือ $I_6^{(1)} - I_4^{(1)}$ ขึ้นอยู่กับว่า $BITS_4^{(2)}$ มีค่าเป็นเท่าไหร่ ดูคำอธิบายได้จากสมการที่ (6) หลังจากนั้นใน stage-3 นำค่าอัตราส่วนความควรจะเป็นแบบล็อกที่ได้จาก stage-2 มาคำนวณจะได้ $I_2^{(3)} = \ln\left(\frac{e^{I_2^{(2)}+I_6^{(2)}}+1}{e^{I_2^{(2)}}+e^{I_6^{(2)}}}\right)$ และเนื่องจากทราบอยู่แล้วว่าบิตที่ 2 เป็นบิตแซ่แข็งจึงตัดสินใจว่ามีค่าเป็น 0 ($BITS_2^{(3)} = 0$)



ภาพที่ 11 แผนภาพเส้นทางการคำนวณค่าอัตราส่วนความควรจะเป็นแบบล็อกของบิตที่ 2

<u>บิตที่ 6</u> จากการคำนวณค่าอัตราส่วนความควรจะเป็นแบบล็อกของกิ่งด้านบนของวงจรหน่วย (b) แล้วจะสามารถหาค่าอัตราส่วนความควรจะเป็นแบบล็อกของกิ่งด้านล่างของวงจรหน่วย (b) ได้จากที่ตัดสินใจว่าบิตที่ 2 เป็น 0 ($BITS_2^{(3)} = 0$) จะได้ว่า $I_6^{(3)} = I_6^{(2)} + I_2^{(2)}$ เมื่อคำนวณได้ค่าอัตราส่วนความควรจะเป็นแบบล็อกใน stage-3 จะสามารถตัดสินใจได้ว่าบิตที่ 6 มีค่าเป็นเท่าไหร่ โดยถ้า $I_6^{(3)} > 0$ จะตัดสินใจว่า $BITS_6^{(3)} = 0$ และถ้า $I_6^{(3)} < 0$ จะตัดสินใจว่า $BITS_6^{(3)} = 1$ และหลังจากตัดสินใจบิตที่ 2 และ 6 แล้วจะสามารถตัดสินใจค่า $BITS_0^{(2)}$ และ $BITS_4^{(2)}$ ได้รวมถึงจะสามารถหาค่า $BITS_0^{(1)}$, $BITS_2^{(1)}$, $BITS_4^{(1)}$, $BITS_6^{(1)}$ ได้



ภาพที่ 12 แผนภาพเส้นทางการคำนวณค่าอัตราส่วนความควรจะเป็นแบบล็อกของบิตที่ 6

<u>บิตที่ 1</u> ลำดับต่อไปจะเป็นบิตที่ 1 เริ่มคำนวณจาก stage-1 โดยจะสามารถหาค่า

$I_1^{(1)}$ เท่ากับ $I_1^{(0)}+I_0^{(0)}$ หรือ $I_1^{(0)}-I_0^{(0)}$ ขึ้นอยู่กับค่า $BITS_0^{(1)}$ ว่ามีค่าเป็นเท่าไร

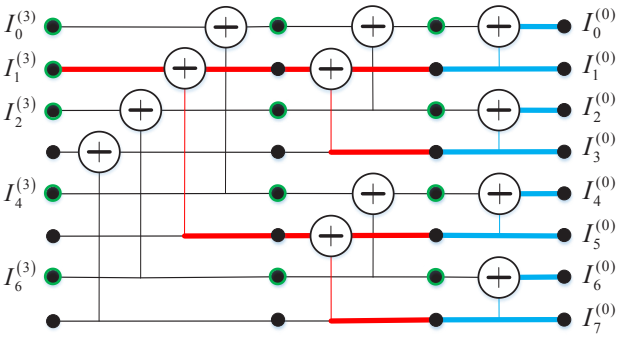
$I_3^{(1)}$ เท่ากับ $I_3^{(0)}+I_2^{(0)}$ หรือ $I_3^{(0)}-I_2^{(0)}$ ขึ้นอยู่กับค่า $BITS_2^{(1)}$ ว่ามีค่าเป็นเท่าไร

$I_5^{(1)}$ เท่ากับ $I_5^{(0)}+I_4^{(0)}$ หรือ $I_5^{(0)}-I_4^{(0)}$ ขึ้นอยู่กับค่า $BITS_4^{(1)}$ ว่ามีค่าเป็นเท่าไร

$I_7^{(1)}$ เท่ากับ $I_7^{(0)}+I_6^{(0)}$ หรือ $I_7^{(0)}-I_6^{(0)}$ ขึ้นอยู่กับค่า $BITS_6^{(1)}$ ว่ามีค่าเป็นเท่าไร

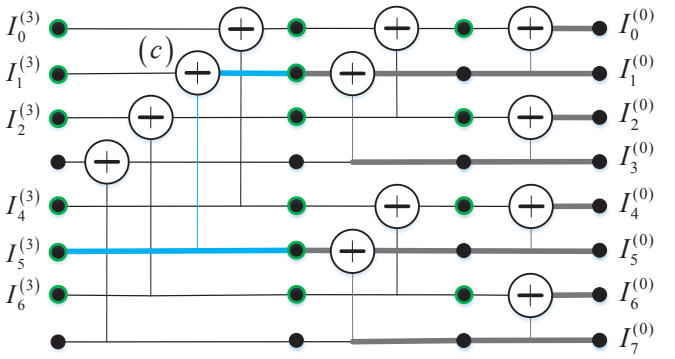
ใน stage-2 จะสามารถหาค่า $I_1^{(2)}=\ln\left(\frac{e^{I_1^{(1)}+I_3^{(1)}}+1}{e^{I_1^{(1)}}+e^{I_3^{(1)}}}\right)$ และ $I_5^{(2)}=\ln\left(\frac{e^{I_5^{(1)}+I_7^{(1)}}+1}{e^{I_5^{(1)}}+e^{I_7^{(1)}}}\right)$ ได้

ใน stage-3 จะสามารถหาค่า $I_1^{(3)}=\ln\left(\frac{e^{I_1^{(2)}+I_5^{(2)}}+1}{e^{I_1^{(2)}}+e^{I_5^{(2)}}}\right)$ ได้ และเนื่องจากทราบอยู่แล้วว่าบิตที่ 1 เป็นบิตแซ่แข็งจึงตัดสินใจว่ามีค่าเป็น 0 $(BITS_1^{(3)}=0)$



ภาพที่ 13 แผนภาพเส้นทางการคำนวณค่าอัตราส่วนความควรจะเป็นแบบล็อกของบิตที่ 1

<u>บิตที่ 5</u> จากการที่คำนวณค่าอัตราส่วนความควรจะเป็นแบบล็อกของกิ่งด้านบนของวงจรหน่วย (c) แล้วจึงสามารถหาค่าอัตราส่วนความควรจะเป็นแบบล็อกของกิ่งด้านล่างของวงจรหน่วย (c) ได้จากที่ตัดสินใจว่าบิตที่ 1 เป็น 0 $(BITS_1^{(3)}=0)$ จะได้ว่า $I_5^{(3)} = I_5^{(2)} + I_1^{(1)}$ และเมื่อคำนวณได้ค่าอัตราส่วนความควรจะเป็นแบบล็อกใน stage-3 จะสามารถตัดสินใจได้ว่าบิตที่ 5 มีค่าเป็นเท่าไหร่ โดยถ้า $I_5^{(3)}>0$ จะตัดสินใจว่า $BITS_5^{(3)}=0$ และถ้า $I_5^{(3)}<0$ จะตัดสินใจว่า $BITS_5^{(3)}=1$ และหลังจากตัดสินใจบิตที่ 1 และ 5 แล้วจะสามารถตัดสินใจค่า $BITS_1^{(2)}$ และ $BITS_5^{(2)}$ ได้



ภาพที่ 14 แผนภาพเส้นทางการคำนวณค่าอัตราส่วนความควรจะเป็นแบบล็อกของบิตที่ 5

<u>บิตที่ 3</u> เริ่มคำนวณจาก stage-2 โดยจะสามารถหาค่า $I_3^{(2)}$ และ $I_7^{(2)}$ ได้

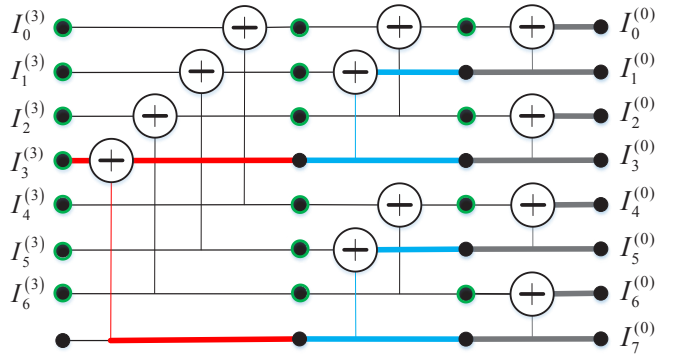
โดย $I_3^{(2)}$ จะเท่ากับ $I_3^{(1)} + I_1^{(1)}$ หรือ $I_3^{(1)} - I_1^{(1)}$ ขึ้นอยู่กับว่า $BITS_1^{(2)}$ มีค่าเป็นเท่าไหร่

และ $I_7^{(2)}$ จะเท่ากับ $I_7^{(1)} + I_5^{(1)}$ หรือ $I_7^{(1)} - I_5^{(1)}$ ขึ้นอยู่กับว่า $BITS_5^{(2)}$ มีค่าเป็นเท่าไหร่

ใน stage-3 นำค่าอัตราส่วนความควรจะเป็นแบบล็อกที่ได้จาก stage-2 มาคำนวณ จะได้

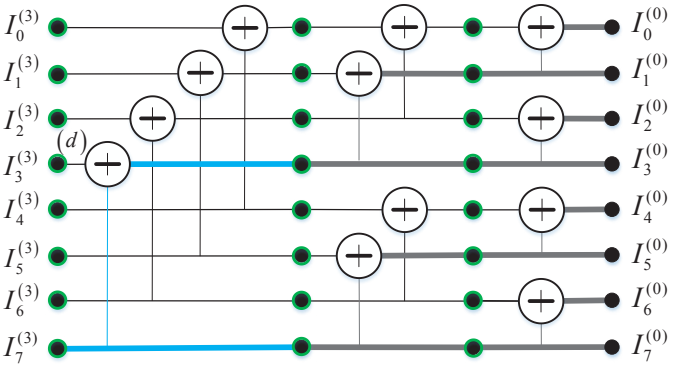
$I_3^{(3)} = \mathrm{In}\left(\frac{e^{I_3^{(2)}+I_7^{(2)}}+1}{e^{I_3^{(2)}}+e^{I_7^{(2)}}}\right)$ และจะสามารถตัดสินใจได้ว่าบิตที่ 3 มีค่าเป็นเท่าไหร่

โดยถ้า $I_3^{(3)} > 0$ จะตัดสินใจว่า $BITS_3^{(3)} = 0$ และถ้า $I_3^{(3)} < 0$ จะตัดสินใจว่า $BITS_3^{(3)} = 1$



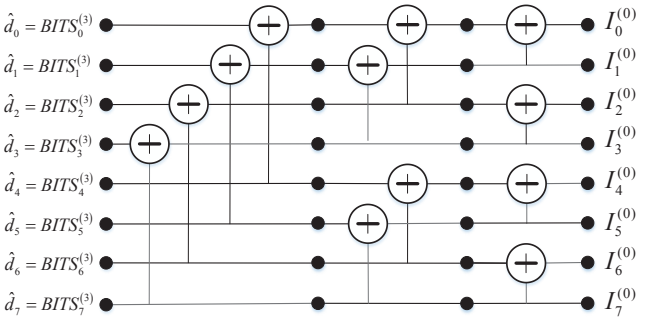
ภาพที่ 15 แผนภาพเส้นทางการคำนวณค่าอัตราส่วนความควรจะเป็นแบบล็อกของบิตที่ 3

<u>บิตที่ 7</u> จากการที่ได้คำนวณค่าอัตราส่วนความควรจะเป็นแบบล็อกของกิ่งด้านบนของวงจรหน่วย (d) แล้ว จึงสามารถหาค่าอัตราส่วนความควรจะเป็นแบบล็อกของกิ่งด้านล่างของวงจรหน่วย (d) ได้โดย $I_7^{(3)}$ เท่ากับ $I_7^{(2)} + I_3^{(2)}$ หรือ $I_7^{(2)} - I_3^{(2)}$ ขึ้นอยู่กับว่า $BITS_3^{(3)}$ มีค่าเป็นเท่าไหร่ และเมื่อคำนวณได้ค่าอัตราส่วนความควรจะเป็นแบบล็อกใน stage-3 จะสามารถตัดสินใจได้ว่าบิตที่ 7 มีค่าเป็นเท่าไหร่ โดยถ้า $I_7^{(3)} > 0$ จะตัดสินใจว่า $BITS_7^{(3)} = 0$ และถ้า $I_7^{(3)} < 0$ จะตัดสินใจว่า $BITS_7^{(3)} = 1$ หลังจากตัดสินใจบิตที่ 3 และ 7 แล้วจะสามารถตัดสินใจค่า $BITS_3^{(2)}$ และ $BITS_7^{(2)}$ ได้ รวมถึงจะสามารถหาค่า $BITS_1^{(1)}$, $BITS_3^{(1)}$, $BITS_5^{(1)}$, $BITS_7^{(1)}$ จะได้



ภาพที่ 16 แผนภาพเส้นทางการคำนวณค่าอัตราส่วนความควรจะเป็นแบบล็อกของบิตที่ 7

หลังจากทำการตัดสินใจบิตใน stage-3 เสร็จเรียบร้อย จะเรียกค่าบิตที่ตัดสินใจได้ทางด้านซ้ายสุดของแผนภาพว่า $\hat{d}$ โดย $\hat{d} = BITS_3^{(1)}$ เมื่อ $i = 0, 1, 2,..., 7$ และจะสามารถหาค่าของข้อมูลที่ถูกถอดรหัส $\hat{m}$ ได้จาก $\hat{m}_j = \hat{d}_i$ โดย $i = 0, 1, 2,3$ และ $j \in F^e$



ภาพที่ 17 แผนภาพหลังจากหาค่าอัตราส่วนความควรจะเป็นแบบล็อกของทั้ง 8 บิต

3. การทดสอบสมรรถนะของรหัสโพลาร์

3.1 โปรแกรมจำลองการเข้ารหัสและถอดรหัสของรหัสโพลาร์

โปรแกรมที่สร้างขึ้นสามารถกำหนดความยาวของคำรหัส = N บิต, ความยาวของบิตข้อมูล = K บิต และค่า SNR ได้ โดยจะมีอัตรารหัส (R) = K / N ในโปรแกรมจะจำลองการส่งข้อมูลบนช่องสัญญาณ AWGN และทำการมอดูเลตโดยใช้วิธีการเลื่อนเฟสแบบไบนารี (Binary phase-shift keying) เมื่อโปรแกรมทำงานจะแสดงค่าดังต่อไปนี้ 1) เวกเตอร์ของบิตข้อมูล (m) 2) ตำแหน่งของบิตแช่แข็ง โดยตำแหน่งที่เป็น 0 หมายถึงตำแหน่งของบิตแช่แข็ง และตำแหน่งที่เป็น -1 หมายถึงตำแหน่งของบิตข้อมูล 3) เวกเตอร์ของข้อมูลที่ถูกเข้ารหัสแล้ว (x) 4) ค่าของสัญญาณที่ภาครับรับได้ (r) 5) เวกเตอร์ของบิตข้อมูลที่ถูกถอดรหัส ($\hat{m}$)

```
m = [1, 1, 1, 0]
Frozen bits location = [0, 0, 0, -1, 0, -1, -1, -1]
d = [0, 0, 0, 1, 0, 1, 1, 0]
x = [1, 0, 0, 1, 0, 1, 1, 0]
r =  [1.8377896000480614, -0.7446171791500966, -0.8782319612026277, 0.028483937861286224, -0.6994396131761276, 0.8392653538273632, 0.9151788671069758, -0.3073939656543457]
d_hat = [0, 0, 0, 1, 0, 1, 1, 0]
m_decoded = [1, 1, 1, 0]
```

ภาพที่ 18 ตัวอย่างของผลที่ได้จากการทำงานของโปรแกรมการเข้ารหัสและถอดรหัสของรหัสโพลาร์โดยมี $N = 8, K = 4$

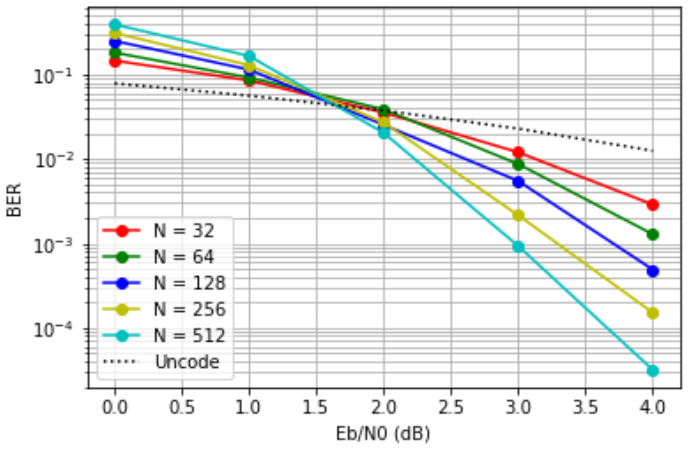
```
m = [0, 0, 1, 0, 1, 1, 0, 1]
Frozen bits location = [0, 0, 0, -1, 0, -1, 0, -1, 0, -1, 0, -1, 0, -1, -1, -1]
d = [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1]
x = [0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1]
r =  [-0.35545098715566026, 0.10466945116719983, 1.5916091390218237, 0.19728095326777795, 0.3045168698581803, 1.2338474519220073, -1.4812561530660795, -0.10357062089493185, 1.0061186175729464, 0.745597573501635, 1.2378546476719876, -1.9028004896475483, -0.11208951918812515, -0.6279688937375663, 0.29318234603745935, 1.7553596570860082]
d_hat = [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1]
m_decoded = [0, 0, 1, 0, 1, 1, 0, 1]
```

ภาพที่ 19 ตัวอย่างของผลที่ได้จากการทำงานของโปรแกรมการเข้ารหัสและถอดรหัสของรหัสโพลาร์โดยมี $N = 16, K = 8$

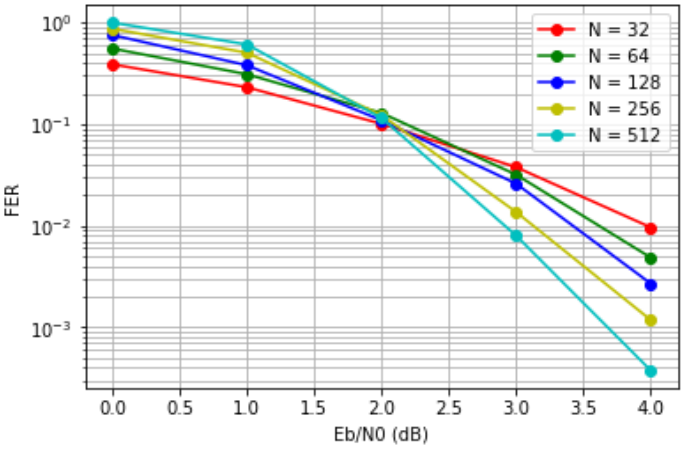
3.2 การตรวจเช็คความถูกต้องของโปรแกรมที่ใช้แสดงกราฟิกเพื่อประเมินสมรรถนะของรหัสโพลาร์

ในการตรวจเช็คความถูกของโปรแกรมที่ใช้แสดงกราฟิกเพื่อประเมินสมรรถนะของรหัสโพลาร์ทำได้โดยการจำลองการเข้ารหัสและถอดรหัสบนช่องสัญญาณ AWGN ที่ความยาวของคำรหัสค่าต่าง ๆ อัตรารหัสเท่ากับ 1/2 และถอดรหัสโดยวิธีหักล้างอย่างต่อเนื่อง ซึ่งการจำลองบนช่องสัญญาณ AWGN อัตรารหัสเท่ากับ 1/2 และถอดรหัสโดยวิธีหักล้างอย่างต่อเนื่องมีการนำเสนอผลและนำมาเขียนบทความทางวิชาการอย่างแพร่หลาย หลังจากนั้นจึงนำผลค่าอัตราความผิดพลาดบิต (BER) และค่าอัตราความผิดพลาด

เฟรม (FER) ที่ได้จากโปรแกรมที่เขียนขึ้นไปเปรียบเทียบกับบทความทางวิชาการที่มีการตีพิมพ์ (Vangala, Viterbo, & Hong, 2014) (Xiong, Lin, & Yan, 2016) และ (Chen, Niu, & Lin, 2013) จึงได้ข้อสรุปว่าผลที่ได้มีความใกล้เคียงกับบทความทางวิชาการในระดับที่เชื่อได้ว่าโปรแกรมที่เขียนขึ้นถูกต้อง ในภาพที่ 21 และ 22 นำเสนอผลการทดสอบค่าอัตราความผิดพลาดบิต และค่าอัตราความผิดพลาดเฟรม (FER) ของรหัสโพลาร์ที่มีความยาวบล็อกเท่ากับ 32, 64, 128, 256 และ 512 บิต ผลการทดสอบแสดงให้เห็นว่าเมื่อระบบใช้บล็อกที่มีความยาวมากขึ้นอัตราความผิดพลาดจะลดต่ำลงเร็วกว่าเมื่อมีการเพิ่มค่า E_b / N_0ขึ้น หรือจะกล่าวในอีกนัยหนึ่งคือ หากกำหนดเป้าหมายค่าอัตราความผิดพลาดไว้ที่ค่าเฉพาะค่าหนึ่ง รหัสโพลาร์ที่มีขนาดความยาวบล็อกสูงจะสามารถให้ค่าความผิดพลาดได้ตามที่กำหนดด้วย E_b / N_0 ที่ต่ำกว่า



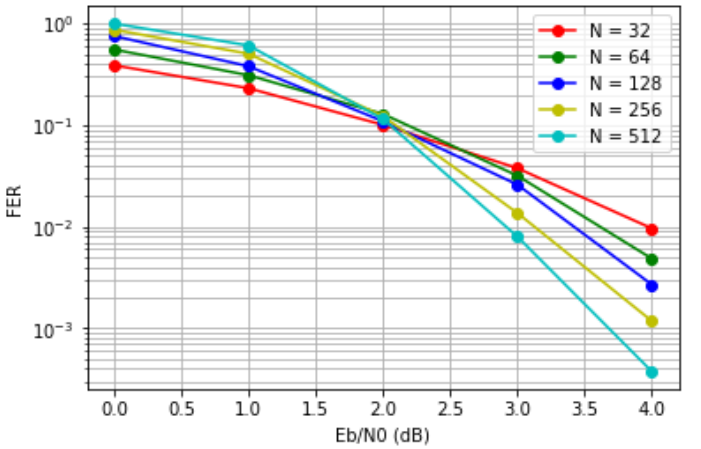
ภาพที่ 20 กราฟเปรียบเทียบค่าอัตราความผิดพลาดบิตของรหัสโพลาร์ ที่ความยาวของคำรหัสค่าต่าง ๆ



ภาพที่ 21 กราฟเปรียบเทียบค่าอัตราความผิดพลาดเฟรมของรหัสโพลาร์ ที่ความยาวของคำรหัสค่าต่าง ๆ

3.3 กราฟเปรียบเทียบสมรรถนะของรหัสโพลาร์ที่มีอัตรารหัสค่าต่าง ๆ

นำโปรแกรมที่ใช้แสดงกราฟิกเพื่อประเมินสมรรถนะของรหัสโพลาร์ในข้อ 2) มาใช้เพื่อประเมินสมรรถนะของรหัสโพลาร์ที่มีอัตรารหัสค่าต่าง ๆ โดยในการจำลองกำหนดให้รหัสโพลาร์ที่มีความยาวของคำรหัสเท่ากับ 128 บิต อัตรารหัสค่าต่าง ๆ แตกต่างกัน 3 ค่า ได้แก่ 1/3 1/2 และ 2/3 และทำการจำลองการเข้ารหัสและถอดรหัสบนช่องสัญญาณ AWGN



ภาพที่ 22 กราฟเปรียบเทียบค่าอัตราความผิดพลาดบิตของรหัสโพลาร์ ที่อัตรารหัสค่าต่าง ๆ

จากผลการจำลองในรูป 22 จะเห็นว่าเมื่ออัตรารหัสมีค่าน้อยลง จะทำให้ได้สมรรถนะที่ดีมากขึ้น โดยยกตัวอย่างที่ E_b / N_0 = dB รหัสโพลาร์ที่มีอัตรารหัสเท่ากับ 2/3 1/2 และ 1/3 จะมีค่าอัตราความผิดพลาดบิตเท่ากับ 2×10^{-3}, 5.5×10^{-4} และ 3×10^{-4} ตามลำดับ ซึ่งการที่อัตรารหัสมีค่าน้อยลง มีความหมายว่าส่งจำนวนของเช็คบิต (บิตแซ่แข็ง) ต่อหนึ่งเฟรมมากขึ้น เมื่อส่งเช็คบิตมากขึ้นจะส่งผลให้ได้สมรรถนะค่าความผิดพลาดที่ดีขึ้น แต่ก็จะใช้เวลาในการส่งข้อมูลมากขึ้นตามไปด้วย

4. บทสรุป/ข้อเสนอแนะ

บทความนี้ได้ศึกษาและพัฒนาโปรแกรมจำลองการเข้ารหัสและถอดรหัสของรหัสโพลาร์ โดยใช้โปรแกรมภาษาไพทอน เพื่อนำมาประเมินสมรรถนะของรหัสโพลาร์ภายใต้ช่องสัญญาณ AWGN โดยในส่วนของการถอดรหัสจะใช้ตัวถอดรหัสหักล้างอย่างต่อเนื่อง (Successive cancellation decoder) ที่เป็นตัวถอดรหัสที่มีประสิทธิภาพเพิ่มขึ้น ในบทความได้นำเสนอผลการประเมินสมรรถนะจากค่าอัตราความผิดพลาดบิตและค่าอัตราความผิดพลาดเฟรม โดยผลที่ได้สามารถนำไปประเมินสมรรถนะของรหัสโพลาร์ได้ภายใต้ช่องสัญญาณแบบ AWGN รวมทั้งโปรแกรมที่เขียนขึ้นสามารถกำหนดพารามิเตอร์ต่าง ๆ ได้เช่น ความยาวของคำรหัส ความยาวของบิตข้อมูล ค่า SNR เป็นต้น รวมถึงในอนาคตผู้วิจัยเห็นว่า สามารถนำโปรแกรมต้นแบบที่เขียนขึ้นไปพัฒนาต่อเพื่อประเมินสมรรถนะของรหัสโพลาร์ภายใต้เงื่อนไขต่าง ๆ และปรับปรุงให้มีสมรรถนะที่ดีขึ้นต่อไป

บรรณานุกรม

Arikan, E. (2008). A performance comparison of polar codes and Reed-Muller codes. *IEEE Communications Letters, 12*(6), (pp.447-449). doi:10.1109/LCOMM.2008.080017

Arikan, E. (2009). Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels. *IEEE Transactions on Information Theory, 55*(7), (pp.3051-3073). doi:10.1109/TIT.2009.2021379

Bioglio, V., Condo, C., & Land, I. (2018). *Design of Polar Codes in 5G New Radio.* Corenll University. Retrieved from https://arxiv.org/abs/1804.04389.

Chen, K., Niu, K., & Lin, J. (2013). Improved Successive Cancellation Decoding of Polar Codes. *IEEE Transactions on Communications, 61*(8), (pp.3100-3107). doi:10.1109/TCOMM.2013.070213.120789

Tal, I., & Vardy, A. (2013). How to Construct Polar Codes. *IEEE Transactions on Information Theory, 59*(10), (pp.6562-6582). doi:10.1109/TIT.2013.2272694

Vangala, H., Viterbo, E., & Hong, Y. (2014). *Permuted successive cancellation decoder for polar codes.* ieeeplore Digital Library. Retrieved from https://ieeexplore.ieee.org/document/8007003.

Xiong, C., Lin, J., & Yan, Z. (2016). Symbol-Decision Successive Cancellation List Decoder for Polar Codes. *IEEE Transactions on Signal Processing, 64*(3), (pp.675-687). doi:10.1109/TSP.2015.2486750

Zhao, S., Shi, P., & Wang, B. (2011). *Designs of Bhattacharyya Parameter in the Construction of Polar Codes.* ieeexplore Digital Library. Retrieved from https://ieeexplore.ieee.org/abstract/document/6040179.


5G