รายงานโครงการวิจัยโดยใช้เงินรายได้คณะวิศวกรรมศาสตร์

ประจำปี 2550

การออกแบบอุปกรณ์เข้ารหัสเพื่อปกปิดข้อมูลโดยอาศัยหลักการเข้ารหัสช่องสัญญาณสำหรับประยุกต์ใช้กับโทรศัพท์เคลื่อนที่

Design scrambles code by using channel code for mobile application

โดย

รศ.ดร.กอบชัย เดชหาญ

บทคัดย่อ

โครงการวิจัยนี้นำเสนอการออกแบบอุปกรณ์เข้ารหัสเพื่อปกปิดข้อมูลโดยอาศัยหลักการเข้ารหัสช่องสัญญาณแบบบล็อคโค้ดเชิงเส้นและสัญญาณรบกวนแบบสุ่มเทียม ซึ่งจะได้พาริตี้ออกมาจำนวนหนึ่ง หลังจากนั้นทำการรวมสัญญาณรบกวนแบบสุ่มเข้ากับรหัส พร้อมทั้งใช้พาริตี้เป็นตัวเปิดตารางการสลับตำแหน่งบิตในรหัสคำ ทำให้ได้ข่าวสารที่ผิดเพี้ยนไปจากเดิม แล้วถูกส่งออกไปในช่องสัญญาณสื่อสารครั้งละ 8 บิต

Abstract

This project proposed the design the data scramble by using block code and pseudo random sequence noise. The message code will be passed to the generator matrix of linear block code for producing a code vector. The obtained code vector will be consisted with the parity digits. Then the pseudo random sequence noise will be summed by exclusive-or to code vector in order to generate the error syndrome. The sequence of scramble code vector will be divided into 8 bits for each word before transmitting them into the transmission channel.

สารบัญ

หน้า

สารบัญตาราง

หน้า

สารบัญรูป

หน้า

บทที่ 1

บทนำ

โครงการวิจัยนี้นำหลักการออกแบบและสร้างระบบปกปิดข้อมูลด้วยการเข้ารหัสบล็อกโค้ดและสัญญาณรบกวนแบบลำดับสุ่มเทียม โดยการเข้ารหัสแบบบล็อกโค้ดนั้น ข้อมูล m ขนาด k บิตเข้ามาทางอินพุต จะถูกแปลงให้เป็นรหัสคำ u โดยการคูณกับเมตริกตัวกำเนิด G ตามหลักการเข้ารหัสแบบบล็อกโค้ดเชิงเส้น จะได้รหัสที่มีขนาด n บิต โดยในรหัสคำ u จะยังคงมีข้อมูล m ในรูปแบบเดิมอย่างครบถ้วน ซึ่งเป็นการเข้ารหัสแบบซิสเต็มเมตริกส์ โดยตัวเข้ารหัสที่ใช้ สามารถในการแก้รหัสที่มีรูปแบบที่ผิดจำนวน 1 และ 2 บิต คือมีรูปแบบที่ผิด 1 บิตจำนวน 8 รูปแบบ และรูปแบบของรหัสที่ผิด 2 บิตจำนวน 28 รูปแบบ จากนั้นต้องมีการทำให้รหัสผิดไปจำนวน 2 บิต โดยการสร้างวงจรรวมรหัสที่ผิดเข้ากับข้อมูล m ในรหัสคำ u ซึ่งรูปแบบรหัสที่ผิดนี้จะสอดคล้องกับค่าซีนโดรมที่ภาครับ

ส่วนในการแก้รหัสคำที่ได้รับที่มีบิตผิดรวมอยู่ จึงต้องตรวจสอบหาตำแหน่งบิตที่ผิดนั้นด้วยเมตริกซ์ตรวจสอบพาริตี้และเปิดตารางซีนโดรมหาตำแหน่งบิตที่ผิดแล้วแก้ไขให้ได้ข่าวสารที่ถูกต้องกลับคืนมา โดยอุปกรณ์ที่ออกแบบมีขนาดเล็ก ปลอดภัยจากการดักฟัง

บทที่ 2
ทฤษฎี

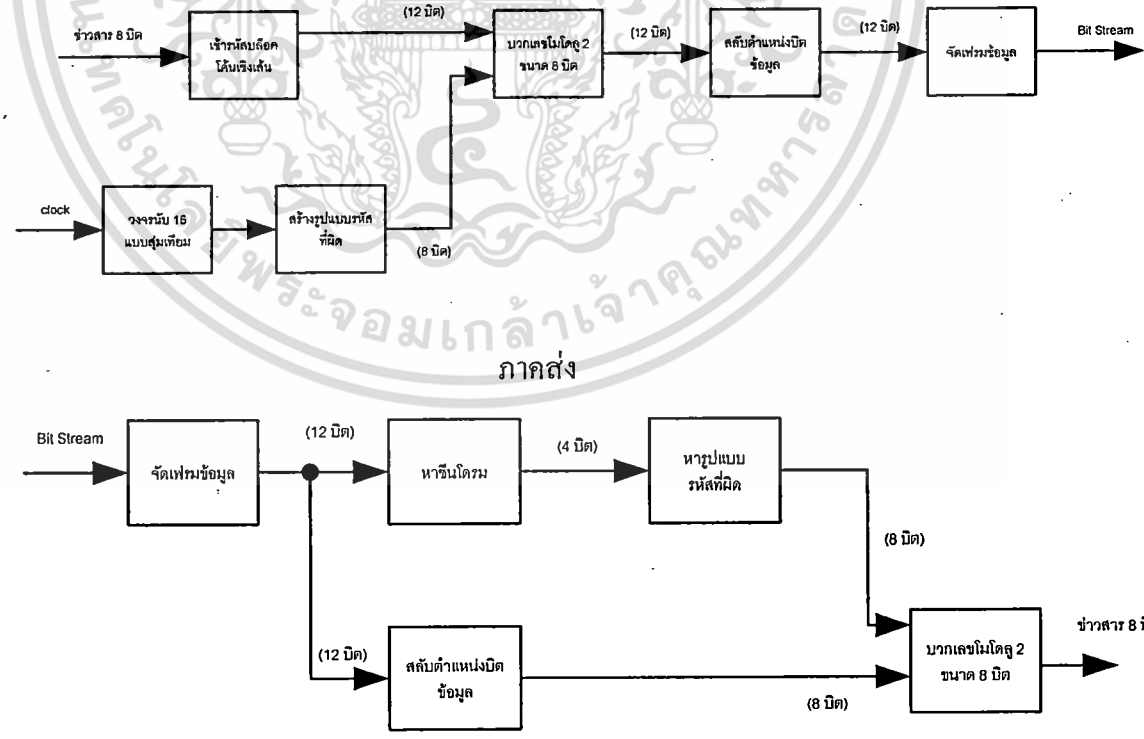
2.1 การเข้ารหัสแบบบล็อกโค้ดเชิงเส้น

ในการเข้ารหัสแบบบล็อกโค้ดเชิงเส้น ข่าวสาร m ขนาด k บิตที่เข้ามาทางอินพุต จะถูกแปลงให้เป็นรหัสคำ u โดยการคูณกับเมตริกซ์ตัวกำเนิด G ตามหลักการเข้ารหัสแบบบล็อกโค้ดเชิงเส้น จะได้รหัสคำที่มีขนาด n บิต คือมีจำนวนบิตเพิ่มขึ้นจำนวน n-k ในรหัสคำ u จะยังคงมีข่าวสาร m ในรูปแบบเดิมอย่างครบถ้วน ซึ่งเป็นการเข้ารหัสแบบซิสเต็มเมติกส์ สามารถเขียนสมาการ

$$u = mG \tag{1}$$

เมตริกซ์ G จะประกอบด้วยเมตริกซ์เอกลักษณ์ I_k ขนาด k*k และเมตริกซ์พาริตี้ P ขนาด k*(n-k) คือ

$$G = [I_k : P] \tag{2}$$



รูปที่ 2.1 แสดงผังวงจรของระบบปกปิดข้อมูล

2.2 ความสามารถในการแก้รหัสที่ผิด

ตัวเข้ารหัสที่ใช้ในงานนี้มีความสามารในการแก้รหัสที่มีรูปแบบที่ผิดจำนวน 1 และ 2 บิต คือจะมีรูปแบบที่ผิด 1 บิต จำนวน 8 รูปแบบ และรูปแบบของรหัสที่ผิด 2 บิต จำนวน 28 รูปแบบ ส่วนรูปแบบที่มีจำนวนบิตผิดมากกว่านั้นก็ไม่สามารถแก้ไขได้

2.3 การกำเนิดรูปแบบรหัสที่ผิดแบบสุ่มเทียม

เนื่องจากเป็นการเข้ารหัสแบบซิสเต็มเมติกส์ ข่าวสารสามารถถูกดักจับได้จากรหัสคำ จึงต้องมีการทำให้รหัสผิดไปจำนวน 2 บิต โดยการสร้างวงจรรวมรหัสที่ผิดเข้ากับข่าวสาร m ในรหัสคำ u ซึ่งรูปแบบรหัสที่ผิดนี้ จะสอดคล้องกับค่าซีนโดรมที่ภาครับ วงจรในส่วนนี้จะรับอินพุทขนาด 4 บิตจากวงจรนับแบบสุ่มเทียม

เพื่อให้รูปแบบของรหัสที่ผิด มีลำดับที่ยากแก่การคาดเดา จึงได้สร้างวงจรนับลำดับสุ่มเทียมด้วยโพลิโนเมียลตั้งต้นอันดับ 8 แล้วเลือกเอาท์พุทจำนวน 4 บิต ส่งให้วงจร สร้างรูปแบบรหัสที่ผิดต่อไป

เมื่อรวมรหัสคำเข้ากับรูปแบบรหัสที่ผิดแล้ว จะได้รหัสคำใหม่ที่ภาครับคือ

$$r = u \oplus e \tag{3}$$

โดยที่ $\oplus$ แทนการบวกแบบเอ็กซ์คลูซีฟออร์

2.4 การแก้รหัสที่ผิด

รหัสคำที่ได้จะมีบิตที่ผิดรวมอยู่ จึงต้องตรวจสอบหาตำแหน่งบิตที่ผิดนั้น ด้วยเมตริกซ์ตรวจสอบพาริตี้ และเปิดตารางซีนโดรมหาตำแหน่งบิตที่ผิด แล้วแก้ไขให้ได้ข่าวสารที่ถูกต้องกลับคืนมา เมตริกซ์ตรวจสอบพาริตี้ H จะถูกนำมาใช้เพื่อคำนวณหาค่าซีนโดรม โดยการนำรหัสคำมาคูณกับเมตริกซ์ตรวจสอบพาริตี้

$$S = rH^T \tag{4}$$

เมื่อ S คือ ซีนโดรม

r คือ รหัสคำที่รับได้

H คือ เมตริกซ์ตรวจสอบพาริตี้ $H = [P^T I_{n-k}]$

บทที่ 3

ผลการทดลอง

3.1 การออกแบบวงจรเข้ารหัส

เลือกเมตริกซ์ G จากรหัสในสมการที่ 4

$$G=\left[\begin{array}{cccccccc:cccc} 1&0&0&0&0&0&0&0&0&0&1&1\\ 0&1&0&0&0&0&0&0&1&0&0&1\\ 0&0&1&0&0&0&0&0&1&1&0&0\\ 0&0&0&1&0&0&0&0&0&1&1&0\\ 0&0&0&0&1&0&0&0&1&1&1&0\\ 0&0&0&0&0&1&0&0&0&1&1&1\\ 0&0&0&0&0&0&1&0&1&0&1&1\\ 0&0&0&0&0&0&0&1&1&1&0&1 \end{array}\right] \quad (6)$$

จะได้สมการสำหรับสร้างวงจรคือ

$$u_0=m_0,\ u_1=m_1,\ u_2=m_2,\ u_3=m_3,\ u_4=m_4,\ u_5=m_5,\ u_6=m_6,\ u_7=m_7 \quad (7)$$

$$u_8=m_1\oplus m_2\oplus m_4\oplus m_6\oplus m_7$$

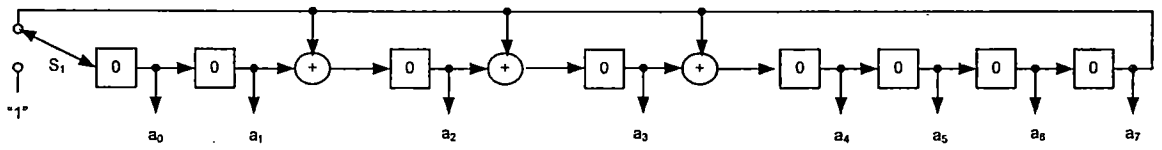
$$u_9=m_2\oplus m_3\oplus m_4\oplus m_5\oplus m_7$$

$$u_{10}=m_0\oplus m_3\oplus m_4\oplus m_5\oplus m_6$$

$$u_{11}=m_0\oplus m_1\oplus m_5\oplus m_6\oplus m_7$$

3.2 ตัวกำเนิดการนับแบบสุ่มเทียม

ตัวกำเนิดลำดับการนับแบบสุ่มเทียม อาศัยโพลิโนเมียลตั้นต้น $p(x)=x^8+x^4+x^3+x^2+1$ มาสร้างวงจรชิฟท์รีจิสเตอร์ป้องกลับ



รูปที่ 3.1แสดงวงจรชิฟท์รีจิสเตอร์ป้อนกลับของลำดับ $p(x)=x^8+x^4+x^3+x^2+1$

เมื่อ Z คือ ชิฟท์รีจิสเตอร์ขนาด 1 บิต

a_0-a_7 คือเอาพุตของการนับแบบสุ่ม

S_1 จะรับลอจิก 1 เข้ามาในช่วงรีเซ็มระบบ

3.3 วงจรสร้างรูปแบบรหัสที่ผิด

ลำดับการนับแบบสุ่ม p [7] จะถูกใช้สร้างรูปแบบรหัสที่ผิด e น้ำหนักเท่ากับ 2 เพื่อรวมกับข่าวสารในรหัสคำ (u_0 ถึง u_7) ความสัมพันธ์ระหว่าง p กับ e แสดงในตารางที่ 1

ตารางที่ 3.1 การกำเนิดรหัสที่ผิดจากตัวนับแบบสุ่ม

Random (p)	Error pattern (e)							
	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
1	0	0	1	0	0	0	0	1
2	0	1	0	0	0	0	1	0
3	0	0	0	0	1	0	0	1
4	1	0	0	0	0	1	0	0
5	0	0	0	0	1	0	1	0
6	0	0	0	0	0	0	1	1
7	0	1	0	0	1	0	0	0
8	0	0	0	1	1	0	0	0
9	0	0	0	0	0	1	0	0
10	0	0	0	0	0	1	0	1
11	0	0	1	0	0	1	0	0
12	0	0	0	0	0	1	1	0
13	1	0	0	0	1	0	0	0
14	1	0	0	0	0	0	0	1
15	0	1	0	1	0	0	0	0

3.4 การสลับตำแหน่งบิตในภาคส่ง

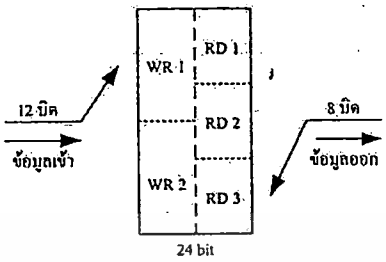
หลังจากรวมรหัสคำ u เข้ากับ e แล้ว จะส่งรหัสคำผ่านวงจรสลับตำแหน่งบิต (u_0 ถึง u_7) ซึ่งขึ้นอยู่กับค่าของพาริตี้ (u_8 ถึง u_{11})

ตารางที่ 3.2 การสลับตำแหน่งบิตในภาคส่ง

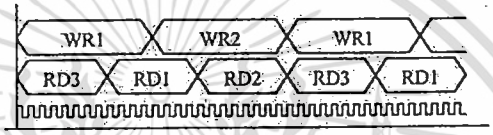
อินพุต				ตำแหน่งการสลับบิต							
u_8	u_9	u_{10}	u_{11}	u_0	u_1	u_2	u_3	u_4	u_5	u_6	u_7
0	0	0	0	O_0	O_2	O_3	O_4	O_7	O_6	O_1	O_5
0	0	0	1	O_2	O_4	O_6	O_1	O_0	O_3	O_7	O_5
0	0	1	0	O_5	O_7	O_0	O_4	O_2	O_6	O_1	O_3
0	0	1	1	O_6	O_4	O_0	O_3	O_7	O_5	O_1	O_2
0	1	0	0	O_4	O_0	O_6	O_3	O_7	O_1	O_2	O_5
0	1	0	1	O_5	O_1	O_6	O_2	O_4	O_3	O_7	O_0
0	1	1	0	O_7	O_0	O_1	O_2	O_6	O_4	O_5	O_3
0	1	1	1	O_6	O_2	O_1	O_0	O_7	O_3	O_4	O_5
1	0	0	0	O_3	O_1	O_4	O_0	O_6	O_2	O_5	O_7
1	0	0	1	O_2	O_5	O_3	O_1	O_6	O_3	O_0	O_7
1	0	1	0	O_1	O_7	O_6	O_4	O_5	O_2	O_0	O_6
1	0	1	1	O_5	O_1	O_0	O_2	O_7	O_0	O_3	O_4
1	1	0	0	O_6	O_4	O_0	O_3	O_1	O_2	O_7	O_5
1	1	0	1	O_7	O_0	O_1	O_2	O_3	O_4	O_5	O_6
1	1	1	0	O_4	O_0	O_7	O_3	O_2	O_1	O_5	O_6
1	1	1	1	O_5	O_0	O_4	O_1	O_2	O_3	O_7	O_6

3.5 การจัดเฟรมข้อมูลที่ภาคส่ง

ช่องสัญญาณจะรับข่าวสารที่มีความกว้างข้อมูล 8 บิต จึงต้องทำการจัดเฟรมข้อมูลขนาด 12 บิต ให้ส่งออกมาครั้งละ 8 บิต โดยต่อเนื่อง



(ก) ผังวงจร

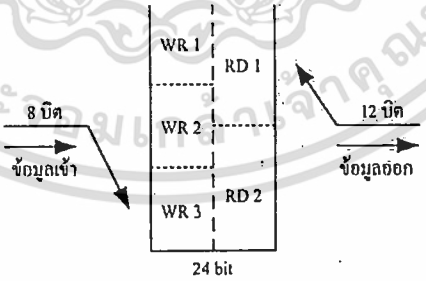


(ข) ผังเวลาการเขียน- อ่าน

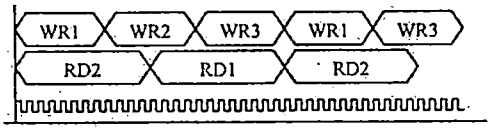
รูปที่ 3.2 วงจรจัดเฟรมข้อมูลภาคส่ง

3.6 การจัดเฟรมข้อมูลที่ภาครับ

ข่าวสารที่รับได้ จะมีความกว้าง 8 บิต จึงต้องใช้วงจรนี้ดึงรหัสคำเดิมคืนมา โดยใช้หลักการเดียวกันกับวงจรจัดเฟรมในภาคส่ง



(ก) ผังวงจร



(ข) ผังเวลาการเขียน- อ่าน

รูปที่ 3.3 วงจรจัดเฟรมข้อมูลภาครับ

3.7 การสลับตำแหน่งบิตในภาครับ

วงจรในภาคนี้จะทำการสลับตำแหน่งบิตกลับโดยใช้พาริตี้ที่รับได้ ตำแหน่งของการสลับบิตแสดงในตารางที่ 3.3

ตารางที่ 3.3 การสลับตำแหน่งบิตในภาครับ

อินพุต				ตำแหน่งการสลับบิต							
A	B	C	D	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
0	0	0	0	O_0	O_2	O_3	O_4	O_7	O_6	O_1	O_5
0	0	0	1	O_4	O_4	O_6	O_1	O_0	O_3	O_7	O_5
0	0	1	0	O_2	O_7	O_0	O_4	O_2	O_6	O_1	O_3
0	0	1	1	O_2	O_4	O_0	O_3	O_7	O_5	O_1	O_2
0	1	0	0	O_1	O_0	O_6	O_3	O_7	O_1	O_2	O_5
0	1	0	1	O_7	O_1	O_6	O_2	O_4	O_3	O_7	O_0
0	1	1	0	O_1	O_0	O_1	O_2	O_6	O_4	O_5	O_3
0	1	1	1	O_3	O_2	O_1	O_0	O_7	O_3	O_4	O_5
1	0	0	0	O_3	O_1	O_4	O_0	O_6	O_2	O_5	O_7
1	0	0	1	O_6	O_5	O_3	O_1	O_6	O_3	O_0	O_7
1	0	1	0	O_6	O_7	O_6	O_4	O_5	O_2	O_0	O_6
1	0	1	1	O_5	O_1	O_0	O_2	O_7	O_0	O_3	O_4
1	1	0	0	O_2	O_4	O_0	O_3	O_1	O_2	O_7	O_5
1	1	0	1	O_1	O_0	O_1	O_2	O_3	O_4	O_5	O_6
1	1	1	0	O_1	O_0	O_7	O_3	O_2	O_1	O_5	O_6
1	1	1	1	O_1	O_0	O_4	O_1	O_2	O_3	O_7	O_6

3.8 วงจรถอดรหัสบล็อกโค้ดเชิงเส้น

หลังจากที่ได้รหัสคำขนาด 12 บิตแล้ว ก็จะทำการหาค่าซินโดรม **S** โดยการนำรหัสคำ r คูณกับเมตริกซ์ตรวจสอบพาริตี้ **H** จากเมตริกซ์ **G** ใน (6) เราจะได้เมตริกซ์ H ที่สอดคล้องคือ

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

และจาก $\mathbf{S} = rH^T$ จะได้สมการซีนโดรมเป็น

$$
\begin{aligned}
s_0 &= m_1 \oplus m_2 \oplus m_4 \oplus m_6 \oplus m_7 \oplus m_8 \\
s_1 &= m_2 \oplus m_3 \oplus m_4 \oplus m_5 \oplus m_7 \oplus m_9 \\
s_3 &= m_0 \oplus m_3 \oplus m_4 \oplus m_5 \oplus m_6 \oplus m_{10} \\
s_4 &= m_0 \oplus m_1 \oplus m_5 \oplus m_6 \oplus m_7 \oplus m_{11}
\end{aligned}
\tag{8}
$$

3.9 การแก้รหัสที่ผิดโดยใช้ซีนโดรม

หลังจากที่ได้ซีนโดรมจากขขึ้นตอนในหัวข้อ 3.8 แล้ว ต่อไปคือการนำซีนโดรมไปเปิดตาราง เพื่อหารูปแบบของตำแหน่งของบิตที่ผิด ระบบจะมีรูปแบบที่ผิดทั้งสิ้น 15 รูปแบบ เมื่อได้รูปแบบที่ผิดแล้ว ก็จะนำไปรวมกับรหัสคำ r แล้วดึงข่าวสารที่ถูกต้องออกมา

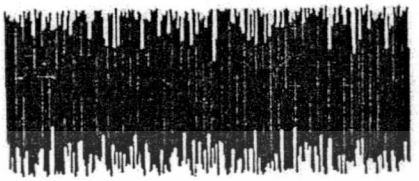
การออกแบบวงจรในส่วนนี้ ใช้วงจรลอจิกเพื่อให้มีขนาดเล็กและประหยัดทรัพยากรภายใน FPGA ดังสมการ

$$
\begin{aligned}
Y_0 &= \bar{s}_0 s_1 \bar{s}_2 \bar{s}_2 + s_0 s_1 \bar{s}_2 s_3 + s_0 s_1 s_2 \bar{s}_3 \\
Y_1 &= \bar{s}_0 \bar{s}_1 s_2 \bar{s}_3 + s_1 s_2 s_3 \\
Y_2 &= \bar{s}_0 \bar{s}_1 \bar{s}_2 s_3 + s_0 \bar{s}_1 s_2 s_3 \\
Y_3 &= s_0 \bar{s}_1 \bar{s}_2 \bar{s}_3 + s_0 s_1 s_2 s_3 \\
Y_4 &= \bar{s}_0 s_2 s_3 + s_0 \bar{s}_1 \bar{s}_2 + s_1 \bar{s}_2 s_3 \\
Y_5 &= s_0 \bar{s}_1 s_2 + s_0 \bar{s}_1 s_3 + s_1 \bar{s}_2 \bar{s}_3 \\
Y_6 &= \bar{s}_0 s_1 \bar{s}_2 s_3 + \bar{s}_0 s_2 \bar{s}_3 + s_0 s_1 \bar{s}_2 \bar{s}_3 \\
Y_7 &= \bar{s}_0 \bar{s}_1 s_3 + s_0 s_2 s_3 + s_1 s_2 \bar{s}_3
\end{aligned}
\tag{9}
$$

เมื่อ Y คือ รูปแบบรหัสที่ผิด

s คือ ซีนโดรม

ก)รูปสัญญาณต้นฉบับ



ข) รูปสัญญาณที่เข้ารหัสลับ

รูปที่ 3.4 แสดงการเข้ารหัสเสียงสัญญาณ

บทที่ 4
สรุปและข้อเสนอแนะ

จากการออกแบบ การสร้างและการทดสอบ พบว่าระบบปกปิดข้อมูลนี้ได้รับการพัฒนาขึ้นจากอุปกรณ์ FPGA มีตัวนับลำดับแบบสุ่มที่ยาวขึ้น เพิ่มเติมวงจรจัดเฟรมข้อมูล โดยให้มีการเข้ารหัสลับสัญญาณเสียงและข้อมูลที่มีประสิทธิภาพ

บรรณานุกรม

1. Shu Lin, "An Introduction to Error Correcting Code," Prentice-Hall, Inc., New jersey, 1970.
2. Xilinx, "The Programable Logic Data Book," 3rd Xilinx, Inc. 1994.
3. ฟูศักดิ์ ชีวสุวิทย์ม "การแก้รหัสที่ผิด," คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง 2528.