

НЕАЛГЕБРАЇЧНИЙ ДЕКОДЕР КОРИГУВАЛЬНИХ КОДІВ

Вступ

Висока вірогідність передачі інформації в цифрових каналах зв'язку забезпечується шляхом підвищення їх завадостійкості, насамперед, за рахунок використання завадостійкого (канального, коригувального, надлишкового) кодування. Розробка і практична реалізація так званих турбокодів дозволили впритул наблизитись до потенційної завадостійкості навіть при роботі в сильно зашумлених каналах передачі інформації [1]. Цьому сприяли значні досягнення в мікроелектроніці, які дали можливість створення досить потужних алгоритмів корекції помилок.

В той же час не втратили своєї значущості і більш короткі коди (Хеммінга, циклічні, БЧХ, Голя та ін.), які використовуються як автономно, так і в складі каскадних конструкцій. Серед схем декодерів особливе місце займають декодери Меггита з корекцією помилок, які виділяються простотою, універсальністю й ефективністю роботи [2].

Однак питання простоти реалізації кодеків і, насамперед, декодерів, забезпечення високих швидкостей обробки в них цифрової інформації і, тим самим, можливості використання їх у високошвидкісних цифрових каналах передачі і деякі інші питання, поки ще залишаються не повною мірою вирішеними і вимагають подальших досліджень.

Мета роботи – при достатньо простій схемі декодера, розширити границю його використання та зменшити час обробки кодових слів, що дозволить суттєво збільшити швидкість передачі інформації по каналах зв'язку.

Основна частина

Відповідно до теореми Шеннона для дискретного каналу з завадами, ймовірність помилки при передачі даних по каналу зв'язку може бути якою завгодно малою величиною при виборі відповідного методу кодування сигналу, тобто завада не накладає суттєвих обмежень на точність передачі даних. При цьому достовірність переданої інформації по каналу зв'язку (рис. 1) може бути забезпечена застосуванням кодера та декодера коригувальних кодів.

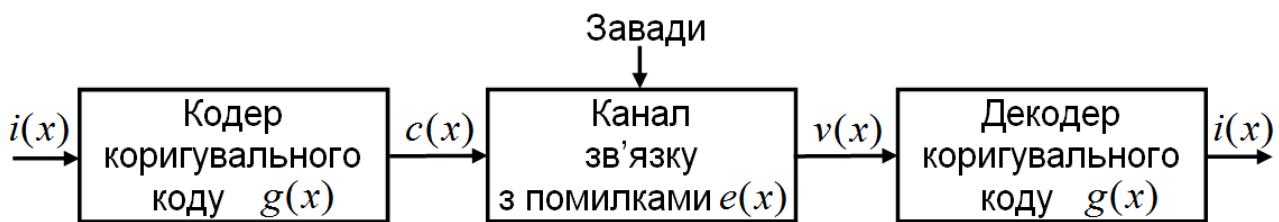


Рис. 1

На вхід кодера подається інформаційна послідовність $i(x)$, довжини k , яку кодер перетворює в послідовність коригувального коду $c(x)$, довжини n , згідно до правил кодування, при чому $n > k$. З каналу зв'язку на декодер надходить кодове слово $v(x)$, уражене помилками ($e(x)$ – вектор помилок).

Неалгебраїчний декодер коригувальних кодів (рис. 2) містить пристрій розв'язки вхідних сигналів, синдромний регістр зсуву зі зворотними зв'язками, логічний блок перевірки синдромів за заданими критеріями, буферний n -розрядний регістр зсуву, модифікатор синдрому, коректор помилок та три ключі K_2 , K_2 і K_3 .

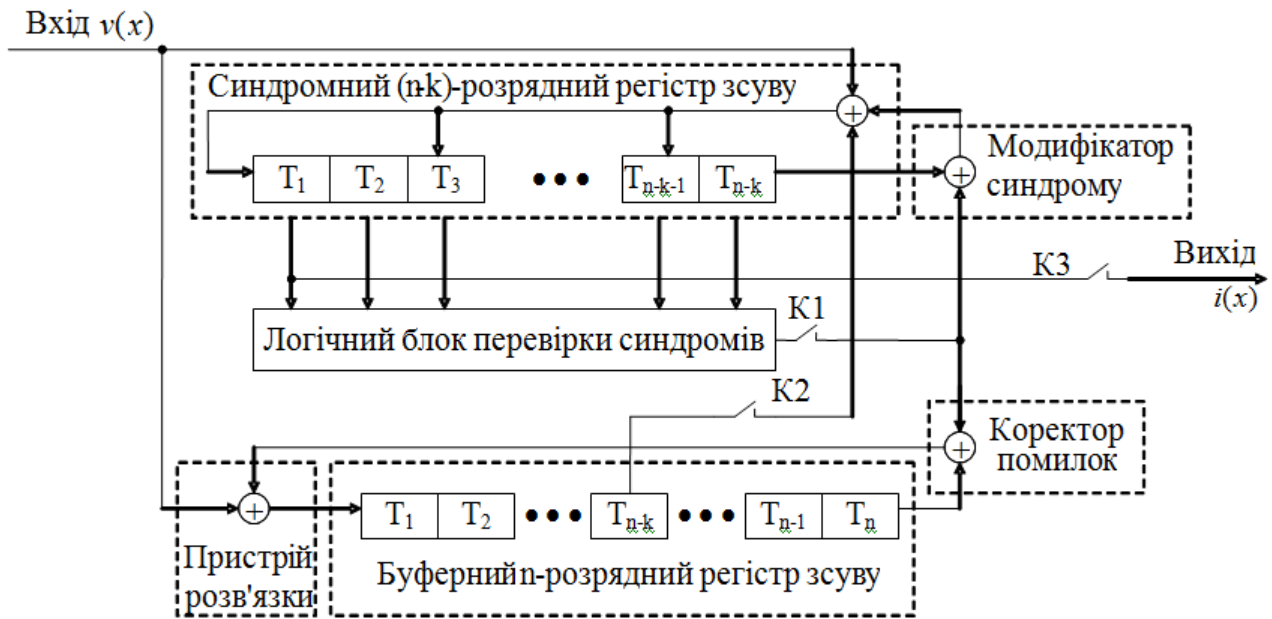


Рис. 2

Робота пропонованого пристрою відбувається в такий спосіб. Попередньо всі блоки декодера обнулені. Вхідне кодове слово $v(x)$, можливо, уражене завадами в каналі зв'язку (тобто з помилками) послідовно подається на вхід синдромного $(n-k)$ – розрядного регістра зсуву зі зворотними зв'язками, де n – довжина кодового слова, а k – довжина інформаційної частини кодового слова та, через пристрій розв'язки вхідних сигналів, на вхід буферного n -розрядного регістра зсуву. Неалгебраїчний декодер обробляє вхідні кодові слова за три цикли роботи з n часових тактів кожний.

Протягом 1-го циклу роботи в синдромному регістрі зсуву формується синдром 1, а буферний n -розрядний регістр зсуву послідовно заповнюється символами кодового слова $v(x)$, що надходить на вхід декодера. Оскільки синдром по визначенню є залишком від ділення $v(x)$ на утворюючий поліном $g(x)$, то синдромний регістр зсуву, структура якого задається структурою $g(x)$, є таким дільником (цифровим фільтром). На цьому циклі ключ К1 розімкнутий, а тому стан виходу логічного блоку перевірки синдромів за заданими критеріями не аналізується. Якщо декодоване кодове слово $c(x)$ не має помилок, то $v(x) = c(x)$, $e(x) = 0$, і до кінця 1-го циклу роботи синдромний регістр зсуву обнуляється. У випадку наявності помилок у кодовому слові, що надходить на декодер, $e(x) \neq 0$, $v(x) \neq c(x)$, і до кінця 1-го циклу роботи на виході синдромного регістра зсуву формується деяка кодова комбінація. Протягом 1-го циклу роботи ключі К2 і К3 також розімкнуті.

С початком 2-го циклу роботи ключ К1 замикається і логічний блок перевірки синдромів за заданими критеріями на кожному такті аналізує кодові комбінації на виходах осередків синдромного регістра зсуву. У випадках виявлення синдромів способом, різним для кожного типу декодера, логічний блок видає з виходу сигнал логічної одиниці U^1 , що надходить на коректор помилок, виправляючи помилку, і на модифікатор синдромів, спрощуючи структуру синдромів. За час 2-го циклу роботи ключі К2 і К3 розімкнуті. У результаті вхідна послідовність $v(x)$ примусово повторно проходить осередки буферного n -розрядного регістра зсуву, що забезпечує скорочення відстані між широко розташованими помилками (більше ніж $n-k$ біт) і дозволяє обробляти й виправляти їх.

Протягом 3-го циклу роботи синдромний регістр зсуву продовжує працювати разом з логічним блоком перевірки синдромів за заданими критеріями, виправляючи помилки, що залишилися, і модернізує синдроми. З виправленням останньої помилки синдромний регістр зсуву обнуляється.

На цьому, останньому циклі роботи, функціонування декодера залежить від способу формування кодових слів.

А. У випадку формування кодового слова систематичним методом ключ К1 замкнутий, ключ К2 розімкнений, вихідний ключ К3 замкнений протягом перших k тактів, пропускаючи на "Вихід" інформаційну групу $i(x)$, після чого розмикається.

Б. У випадку формування кодового слова несистематичним методом, починаючи з $(n-k)$ -го такту цього циклу розмикається ключ К1, а ключі К2 та К3 замикаються. При цьому, з виходу T_{n-k} буферного регістра зсуву подається прийнята кодова послідовність $c(x)$ з виправленими помилками на вхід синдромного регістра, а з виходу першого тригера цього регістра знімається виділена інформаційна послідовність $i(x)$, яка і передається на "Вихід".

Наприкінці 3-го циклу роботи регістри обнуляються, і декодер готовий до обробки чергового кодового слова, можливо, перекрученого перешкодами в каналі зв'язку.

Розглянемо роботу синдромного регістра зсуву і всього декодера на конкретному прикладі при обробці кодового слова блокового коду Голея.

Коди Голея належать до алгоритмів, що використовують кінцеві випадкові групи для кодування даних. Існує два різних типи кодів Голея, бінарний і троїчний. Бінарна версія відноситься до класу $(23,12,7)$, де $n=23$, $k=12$, з мінімальною кодовою відстанню $d_{\min}=7$. Ця версія може знайти і виправити всі трьохбітні помилки. З додаванням біта парності, вийде бінарний код Голея $(24,12,8)$. Цей розширений код Голея може знайти всі чотирьохбітні помилки і виправити всі трьохбітні помилки. Троїчний код Голея відноситься до класу $(11,3,5)$ і може знайти і виправити всі 2-бітні помилки.

Прийmemo, що на передавальній стороні кодові слова бінарного коду Голея $(23,12,7)$ формуються кодером несистематичним способом, тобто

$$c(x) = i(x) \cdot g(x). \quad (1)$$

Кодуванню підлягало інформаційне повідомлення

$$i(x) = x^{11} + x^3, \quad (2)$$

або у векторному виді $I=100000001000$.

Утворюючий поліном $g(x)$ для цього коду має вид [2, с. 89]

$$g(x) = x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1. \quad (3)$$

Кодове слово, яке надійде на вхід декодера для взятого $i(x)$, без ураження перешкодами, прийме значення

$$\begin{aligned} c(x) &= i(x) \cdot g(x) = (x^{11} + x^3)(x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1) = \\ &= x^{22} + x^{21} + x^{17} + x^{16} + x^{15} + x^{14} + x^{11} + x^9 + x^8 + x^7 + x^5 + x^3, \end{aligned} \quad (4)$$

або у векторному виді $C=11000111100101110101000$.

Нехай поліном помилок дорівнює

$$e(x) = x^{19} + x^{16} + x, \quad (5)$$

що відповідає вектору $E = 0001001000000000000010$. Тоді кодове слово (4), уражене завадами (4), буде мати вид

$$v(x) = c(x) \oplus e(x) = x^{22} + x^{21} + x^{19} + x^{17} + x^{15} + x^{14} + x^{11} + x^9 + x^8 + x^7 + x^5 + x^3 + x, \quad (6)$$

або у векторному виді

$$V = C \oplus E = 110\mathbf{1}01\mathbf{0}11001011101010\mathbf{1}0. \quad (7)$$

Тут виділені жирним шрифтом біти помилок.

Цикли	Такти	V(x)	(n-k) осередків синдромного регістру (* - місця підключення зворотних зв'язків)											Вихід ЛБПС	Коментарі			
			T ₁ *	T ₂	T ₃ *	T ₄	T ₅ *	T ₆ *	T ₇ *	T ₈	T ₉	T ₁₀	T ₁₁ *					
	20	1		1			1											
	21	0			1			1										
	22	1	1		1	1	1	1						1				
	23	0	1	1	1	1								1				
2-й цикл роботи декодера	24		1	1		1		1	1				1				1. На 44-му такті виконана умова спрацювання логічного блоку перевірки синдромів, що означає необхідність виправлення 1-ї помилки в буферному регістрі. 2. Умовою спрацювання логічного блоку є наявність логічної одиниці в старшому розряді регістра, але при цьому сума одиниць в усіх розрядах не повинна перевищувати трьох.	
	25		1	1				1		1			1					
	26		1	1			1	1			1		1					
	27		1	1			1					1	1					
	28		1	1			1		1									
	29			1	1				1		1							
	30				1	1				1		1						
	31					1	1				1		1					
	32						1	1				1		1				
	33		1		1		1						1	1				
	34		1	1	1	1	1		1									
	35			1	1	1	1	1		1								
	36				1	1	1	1	1		1							
	37					1	1	1	1	1	1		1					
	38						1	1	1	1	1	1		1				
	39		1		1		1			1	1	1	1	1				
	40		1	1	1	1	1		1		1	1	1					
41			1	1	1	1	1		1		1	1	1					
42		1			1						1							
43			1			1						1						
44				1			1						1	1				
45					1			1										
46						1			1		1							
3-й цикл роботи декодера	47							1			1						1. На 49-му та 52-му тактах виконана умова спрацювання логічного блоку перевірки синдромів, що означає необхідність виправлення відповідно 2-ї та 3-ї помилки в буферному регістрі. 2. Починаючи з 58-го такту ключ К3 замикається і на "Вихід" з першого тригера синдромного регістра передається виділена інформаційна послідовність $i(x)$	
	48								1			1						
	49									1			1	1				
	50										1							
	51											1						
	52												1	1				
	53																	
	54																	
	55																	
	56																	
	57																	
	58		1		1		1	1	1					1				
	59			1		1		1	1	1	1							
	60				1		1		1	1	1							
61					1		1		1	1	1							
62						1		1		1	1	1						
63							1		1		1	1						
64								1		1		1						
65									1		1	1						
66		1		1		1	1	1		1		1						
67			1		1		1	1	1	1		1						
68				1		1		1	1	1	1		1					
69					1		1		1	1	1	1						

$$\rightarrow i(x) = x^{11} + x^3 = 100000001000$$

В табл. 2 показано послідовне надходження в буферний регістр вхідного кодового слова $v(x)$, що має 3 помилки і процес їх виправлення. Процес початкового заповнення регістра в першому циклі роботи декодера у таблиці не показано.

Таблиця 2

Цикл	Такт	ЛБ	Буферний регістр																						
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	23	0	0	1	0	1	0	1	0	1	1	1	0	1	0	0	1	1	0	1	0	1	0	1	1
	24	0	1		1		1		1		1	1	1		1			1	1	0	1		1		1
2	25	0	1	1		1		1		1		1	1	1		1			1	1	0	1		1	
	26	0		1	1		1		1		1	1	1		1				1	1	0	1		1	
	27	0	1		1	1		1		1		1	1	1		1				1	1	0	1		1
	28	0		1		1	1		1		1		1	1	1		1				1	1	0	1	
	29	0	1		1		1	1		1		1		1	1	1		1			1	1	0	1	
	30	0	0	1		1		1	1		1		1		1	1	1		1		1		1	1	1
	31	0	1	0	1		1		1	1		1		1		1	1	1		1	1	1		1	
	32	0	1	1	0	1		1		1	1		1		1		1		1	1	1	1		1	
	33	0		1	1	0	1		1		1	1		1		1		1		1	1	1	1		1
	34	0			1	1	0	1		1		1	1		1		1		1		1	1	1	1	
	35	0	1			1	1	0	1		1		1	1		1		1		1		1	1	1	
	36	0		1			1	1	0	1		1		1	1		1		1		1		1	1	1
	37	0	1		1			1	1	0	1		1		1	1		1		1		1		1	1
	38	0	1	1		1			1	1	0	1		1		1	1		1		1		1		1
	39	0	1	1	1		1			1	1	0	1		1		1	1		1		1		1	1
	40	0		1	1	1		1			1	1	0	1		1		1	1		1		1	1	1
	41	0	1		1	1	1		1			1	1	0	1		1		1	1	1		1		1
	42	0		1		1	1	1		1			1	1	0	1		1		1	1	1		1	1
	43	0	1		1		1	1	1		1			1	1	0	1		1		1	1	1		1
	44	1		1		1		1	1	1		1			1	1	0	1		1		1	1	1	1
	45	0	0		1		1		1	1	1		1			1	1	0	1		1		1	1	1
	46	0		0		1		1		1	1	1		1			1	1	0	1		1		1	1
	3	47	0	1		0		1		1		1	1	1		1		1	1	0	1		1		1
		48	0	1	1		0		1		1		1	1	1		1		1	1	0	1		1	1
49		1	0	1	1		0		1		1		1	1	1		1		1	1	0	1		1	
50		0	0		1	1		0		1		1		1	1	1		1		1	1	0	1	1	
51		0	0	0		1	1		0		1		1		1	1	1		1		1	1	0	1	
52		1	1		0		1	1		0		1		1		1	1	1		1		1	1	0	
53		0	1	1		0		1	1		0		1		1		1	1	1		1		1	1	
54		0	1	1	1		0		1	1		0		1		1		1	1	1		1		1	
55		0	1	1	1	1		0		1	1		0		1		1		1	1	1		1		
56		0	0	1	1	1	1		0		1	1		0		1		1		1	1	1		1	
57		0	0		1	1	1	1		0		1	1		0		1		1		1	1	1		
58		0	1			1	1	1	1		0		1	1		0		1		1		1	1	1	
59		0	0	1			1	1	1	1		0		1	1		0		1		1		1	1	
60		0	1		1			1	1	1	1		0		1	1		0		1		1		1	
61		0	1	1		1			1	1	1	1		0		1	1		0		1		1	1	
62		0	1	1	1		1			1	1	1	1		0		1	1		0		1		1	
63		0	0	1	1	1		1			1	1	1	1		0		1	1		0		1	1	
64		0	1		1	1	1		1			1	1	1	1		0		1	1		0		1	
65		0	0	1		1	1	1		1			1	1	1	1		0		1	1		0		
66		0	1		1		1	1	1		1			1	1	1	1		0		1	1		0	
67		0	0	1		1		1	1	1		1			1	1	1	1		0		1	1	0	
68		0	0	0	1		1		1	1	1	1		1		1	1	1	1		0		1	1	
69		0	0	0	0	1	0	1	0	1	1	1	0	1	0	1	1	1	1	1	0	0	0	1	1

$$c(x) = x^{22} + x^{21} + x^{17} + x^{16} + x^{15} + x^{14} + x^{11} + x^9 + x^8 + x^7 + x^5 + x^3 =$$

$$= 11000111100101110101000$$

Як видно з табл. 2, на 69-му такті в буферному регістрі сформоване прийняте кодове слово $c(x)$ з виправленими помилками в відмічених розрядах. Крім того, при необхідності використання кодового слова при подальшій обробці, його можна зняти в послідовному коді з першого тригера цього регістра починаючи з 47-го такту.

Висновки

1. Запропонований декодер значно спрощений за рахунок відсутності пристрою виділення $i(x)$ з $c(x)$. Він універсальний, тому що дозволяє обробляти кодові слова, сформовані як систематичним, так і несистематичним методами.

2. Запропонований декодер має широку область застосування, оскільки може бути реалізований замість будь-якого неалгебраїчного декодера, що має у своєму складі синдромний регістр зсуву зі зворотними зв'язками й призначеного, насамперед, для обробки кодових слів, сформованих несистематичним методом.

3. Запропонований декодер не сповільнює процес обробки кодових слів і швидкість передачі їх по каналу зв'язку, тому що виділення інформаційної групи $i(x)$ відбувається в межах 3-го циклу роботи, який і так необхідний для виконання основної функції – декодування.

Список літератури

1. Блейхут Р. Теория и практика кодов, контролирующих ошибки / Пер. с англ. – М.: Мир, 1986. – С.164. – Рис. 6.17. – 576 с. 2. Кларк Дж. Кодирование с исправлением ошибок в системах цифровой связи / Пер. с англ. – М.: Радио и связь, 1987. – 392 с. 3. Макаров Л. Б., Битченко А. Н., Коняхин Г. Ф., Коваленко Н. А. Синтез инверсных пороговых схем для реализации в неалгебраических декодерах корректирующих кодов. Системы обробки інформації, вип. 8 (98), 2011. – С.87-92.