



НОВІ ЗАСОБИ КІБЕРНЕТИКИ, ІНФОРМАТИКИ, ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ ТА СИСТЕМНОГО АНАЛІЗУ

В.Н. ОПАНАСЕНКО, С.Л. КРЫВЫЙ

УДК 51.681.3

СИНТЕЗ НЕЙРОПОДОБНЫХ СЕТЕЙ НА ОСНОВЕ ПРЕОБРАЗОВАНИЙ ЦИКЛИЧЕСКИХ КОДОВ ХЕММИНГА

Аннотация. Рассматривается синтез нейроподобной сети Хемминга для реализации задачи классификации входного множества двоичных векторов. Формирование отсортированной последовательности по мере близости (расстояние Хемминга) основано на преобразованиях циклических кодов Хемминга. Доказана корректность синтеза такой реализации для произвольного значения расстояния Хемминга и произвольной разрядности входного вектора.

Ключевые слова: булева функция, нейроподобная сеть, расстояние Хемминга, циклический код.

ВВЕДЕНИЕ

Сеть Хемминга [1–4], предназначенная для решения задач распознавания образов (речевых сигналов, изображений и т.д.) посредством отнесения входного двоичного вектора к одному (или нескольким в случае одинаковой меры близости) из заданных множеств шаблонов, определяется максимальной мерой близости, в качестве которой используется расстояние Хемминга.

Достоверность классификации или процесса принятия решения зависит от разброса значений шаблонов, однако шаблон с максимальной мерой близости не всегда является наиболее правдоподобным для входной модели. Этот недостаток устраняется при реализации сети, на втором уровне обработки которой формируется упорядоченная группа, содержащая шаблоны, а меры их близости не превышают допустимой (с точки зрения правдоподобности классификации) величины. Реализация такой сети на унифицированной элементной базе весьма проблематична ввиду большой мощности множества шаблонов для многих задач распознавания.

В настоящей статье предлагается синтез структур нейроподобной сети Хемминга, ориентированных на кристаллы FPGA [5, 6] и выполняющих формирование упорядоченных групп векторов–шаблонов, мера близости которых не превышает допустимых значений [3]. За основу формирования таких групп векторов–шаблонов берутся циклические коды Хемминга [7, 8], множество которых можно использовать как близкие (по расстоянию Хеммингу) к эталонному коду.

СИНТЕЗ ПРЕОБРАЗОВАТЕЛЯ ЦИКЛИЧЕСКОГО КОДА ХЕММИНГА ДЛЯ ПРОИЗВОЛЬНОЙ РАЗРЯДНОСТИ И ПРОИЗВОЛЬНОГО РАССТОЯНИЯ ХЕММИНГА

Пусть $X = \{0, 1\}$ — двоичный алфавит, $F_n(X)$ — множество всех слов длины n в алфавите X , которое называют полным множеством слов в этом алфавите.

© В.Н. Опанасенко, С.Л. Крывый, 2017

Очевидно, что $|F_n(X)| = 2^n$. Пусть $v = (x_1 x_2 \dots x_n) \in F_n(X)$, а R — оператор циклического сдвига с шагом 1, определяемый следующим образом:

$$Rv = (x_n x_1 x_2 \dots x_{n-1}) \quad \forall v = (x_1 x_2 \dots x_{n-1} x_n) \in F_n(X).$$

Множество V слов из $F_n(X)$, замкнутое относительно оператора сдвига R , будем называть циклическим кодом Хемминга. Замкнутость множества V означает $\forall v \in V (Rv \in V)$. Циклический код Хемминга можно рассматривать как подмножество слов, близких (по расстоянию Хемминга) к заданному эталонному слову в алфавите X .

Если в качестве эталонного слова выбирается единичное слово (слово, состоящее из единиц), то в случае расстояния Хемминга, равного единице, все слова из множества V , кроме эталонного, порождаются произвольным его элементом в силу замкнутости этого множества относительно оператора циклического сдвига. Такое слово в дальнейшем будем называть порождающим словом.

Введем следующие обозначения:

$H_m(1)$ — циклический код длины n , порождающее слово которого включает m -компонентную ($m = 1, n-1$) группу циклически смежных единиц и d -компонентную ($d = (n-m)$) группу нулей;

$H(1)$ — циклический код длины n , порождающее слово которого включает n -компонентную группу, состоящую из единиц;

$H_m(0)$ — циклический код длины n , порождающее слово которого включает m -компонентную ($m = 1, n-1$) группу циклически смежных нулей и d -компонентную ($d = (n-m)$) группу циклически смежных единиц;

$H(0)$ — циклический код длины n , порождающее слово которого включает n -компонентную группу, состоящую из нулей.

ПОСТАНОВКА ЗАДАЧИ

Пусть $F_n(X)$ — полное множество слов длины n в алфавите $X = \{0, 1\}$ и $H_m(1)$ (соответственно $H_m(0)$) — циклический код длины n . Необходимо построить логическую структуру, которая реализует отображение \mathfrak{Z} , определяемое следующим образом:

$$\mathfrak{Z}(H_m(1)) = 1, \quad \mathfrak{Z}(F_n(X) \setminus H_m(1)) = 0$$

$$(\text{соответственно } \mathfrak{Z}(H_m(0)) = 1, \quad \mathfrak{Z}(F_n(X) \setminus H_m(0)) = 0).$$

Задача может быть решена последовательным соединением [9–11] циклических структур типа AND и NOR, которые имеют n входов и n выходов, а также структуры OR, имеющей n входов и один выход. В общем случае задачи синтеза многоуровневых логических структур для классификации входных двоичных векторов с одним и многими выходами рассмотрены в работах [12–15]. Структуры для поставленной задачи имеют r -уровневую организацию, k -й уровень ($k = 1, r$) которых содержит n логических элементов AND, реализующих преобразование

$$\omega(a_i, a_{(i+s) \bmod(n)}) \quad \forall i = \overline{0, n-1}, \quad 1 \leq s \leq (n-1),$$

где ω — логическая функция двух переменных; $a_i, a_{(i+s)}$ — однобитовые компоненты входного двоичного вектора; $s \in [1, n-1]$ — шаг циклического сдвига. Все элементы одного уровня настраиваются на выполнение одной и той же логической функции AND или NOR.

Мощность группы циклически смежных единиц, исходя из таблицы истинности логической функции AND и структуры связей ($\forall k, s = 1$), при увеличении номера уровня k уменьшается на единицу. Таким образом, для преобразования кодового слова с заданным значением m необходимо иметь $m-1$ уровень.

В качестве многоуровневых логических структур (рис. 1) выбираются структуры S_1 и S_2 ($n = 4$, шаг $i = 1$ в операторе R). Исходя из таблицы истинности логических операций AND и NOR, а также структуры связей ($\forall k, s = 1$), количество циклически смежных единиц при увеличении номера уровня k уменьшается на единицу для операции AND, а для операции NOR, наоборот, количество нулей увеличится на единицу, если на вход подается слово, содержащее d циклически смежных единиц. Структуры S_1 и S_2 синтезированы ранее в работах [7, 8, 11]. Если на вход k -го уровня структуры S_1 ($1 \leq k < r$) с такими связями подается слово, содержащее группу из d циклически смежных нулей, то на его выходе будет слово, содержащее группу из $d + 1$ циклически смежных нулей. В работе [7] показано, что достаточно исследовать структуру типа S_1 , поскольку структура типа S_2 после первого уровня превращается в структуру типа S_1 с входным словом $H_{n-d-1}(1)$.

Реализация таких структур (для $n = 4, s = 1$) приведена на рис. 1, а результаты преобразования даны в табл. 1.

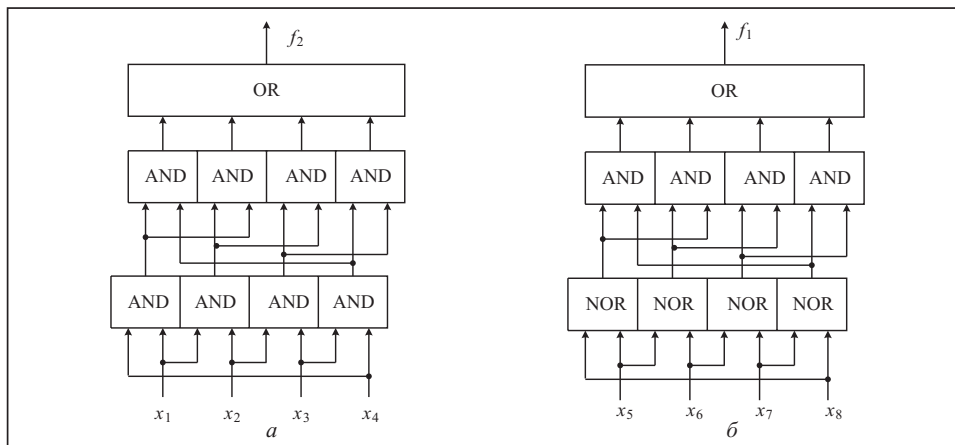


Рис. 1. Структура преобразователя циклических кодов на базе операций AND, NOR и OR: структура типа S_1 (а); структура типа S_2 (б)

Таблица 1. Преобразование циклических кодов

Результаты преобразования структуры S_1					Результаты преобразования структуры S_2																																																														
Входной код				Выход f_2	Входной код				Выход f_1																																																										
x_4	x_3	x_2	x_1		x_8	x_7	x_6	x_5																																																											
1	1	1	1	1	0	0	0	0	1																																																										
0	1	1	1		1	0	1	1		1	1	0	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	0	1	0	1	0	1	1	0	1	0	0	0	1	0	0	1	1	0	1	0	1	1	0	0
1	0	1	1		1	1	0	1		1	1	1	0	0	1	0	0		0	0	0	0		0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	0	1	0	1	0	1	1	0	1	0	0	0	1	0	0	1	1	0	1	0	1	1	0	0		
1	1	0	1		1	1	1	0		0	1	0	0		0	0	0		0		0	0		0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	0	1	0	1	0	1	1	0	1	0	0	0	1	0	0	1	1	0	1	0	1	1	0	0				
1	1	1	0	0	1	0	0	0	0																																																										
0	0	0	0		0	0	0	1			0	0	1		0		0		0		1	1		0	1	0	0	0	1	0	1	0	1	1	0	1	0	0	0	1	0	0	1	1	0	1	0	1	1	0	0																
0	0	0	1		0	0	1	0			0	0	1		1		0		1		0	0		0	1	0	1	0	1	1	0	1	0	0	0	1	0	0	1	1	0	1	0	1	1	0	0																				
0	0	1	0		0	0	1	1			0	1	0		0		0		1		0	1		0	1	1	0	1	0	0	0	1	0	0	1	1	0	1	0	1	1	0	0																								
0	0	1	1		0	1	0	0			0	1	0		1		0		1		1	0		1	0	0	0	1	0	0	1	1	0	1	0	1	1	0	0																												
0	1	0	0		0	1	0	1			0	1	1		0		1		0		0	0		1	0	0	1	1	0	1	0	1	1	0	0																																
0	1	0	1		0	1	1	0			1	0	0		0		1		0		0	1		1	0	1	0	1	1	0	0																																				
0	1	1	0		1	0	0	0			1	0	0		1		1		0		1	0		1	1	0	0																																								
1	0	0	0		1	0	0	1			1	0	1		0		1		1		0	0																																													
1	0	0	1		1	0	1	0			1	1	0		0																																																				
1	0	1	0		1	1	0	0																																																											
1	1	0	0																																																																

Обоснование свойств приведенных структур для циклических кодов Хемминга проведено в работе [7], где была доказана следующая теорема.

Теорема 1. Если на вход первого уровня вычислительной структуры S_2 с операцией NOR подается слово с d циклически смежными единицами, то на выходе имеем слово с $d+1$ циклически смежными нулями. Если на вход k -го уровня структуры S_1 ($1 \leq k < r$) подается слово с d циклически смежными нулями, то на выходе этого уровня имеем слово с $d+1$ циклически смежными нулями.

Структура S_1 идентифицирует полубайт (4-разрядное слово, состоящее из нулей и единиц), который содержит три или четыре единицы, а структура S_2 идентифицирует полубайт, содержащий одну единицу или только нули. Полубайты, состоящие только из нулей $H(0)$ или только из единиц $H(1)$, будем называть особыми точками, идентификация которых изложена ниже.

Рассмотренные структуры относятся к структурам последовательного типа, структуры параллельно-последовательного типа могут быть реализованы аналогичным образом на $r = \log_2 n$ уровнях. Каждый уровень параллельно-последовательной структуры имеет регулярную структуру связей, т.е. на вход любого i -го логического элемента уровня k поступают компоненты с индексами i и $(i+s)$ (значение $i+s$ берется по модулю n).

Далее рассмотрим общую задачу классификации входных полубайтов, которые могут иметь нециклическую структуру. С этой целью рассмотрим базовую структуру, состоящую из двух однотипных 4-разрядных подструктур, построенных с помощью подструктур S_2 и S_1 , выходы которых обозначены символами f_1, f_2 и g_1, g_2 соответственно (рис. 2).

Особенность этой структуры обосновывает следующее утверждение.

Лемма 1. Если особые точки не принимают участия в классификации, то с помощью приведенных подструктур идентифицируется l единиц во входном байте, где $1 \leq l \leq 6$.

Доказательство. Из теоремы 1 и табл. 1 для подструктур следует, что выходы обеих подструктур взаимоисключающие, т. е. если на выходе f_1 (f_2) имеем единицу, то на выходах f_2 и $f_1 f_2$ (f_1 и $f_1 f_2$) имеем значение 0. Это справедливо и для g_1 (g_2).

На выходах подструктур f_2, g_2 имеем единицу, если на входе подструктур S_1 и S_2 полубайт имеет только одну единицу или все нули. Выходы подструктур

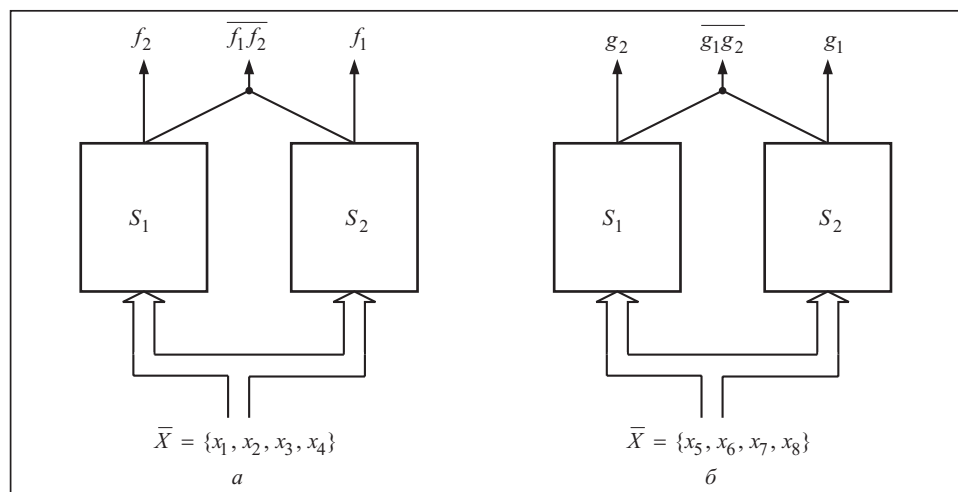


Рис. 2. Базовая структура преобразователей циклических кодов для первого полубайта (а) и для второго полубайта (б)

f_1, g_1 дают единицу, если на входе подструктур S_1 и S_2 полубайт имеет точно три единицы или имеет все единицы. На выходе подструктур $f_1 f_2$ и $(g_1 g_2)$ имеем единицу, если входной полубайт подструктур S_1 и S_2 содержит точно две единицы.

Из этого следует, что настройка обеих подструктур на инициализацию в байте трех единиц принимает вид $(f_2 \& g_1 g_2) + (g_2 \& f_1 f_2) + (f_1 \oplus g_1)$, где символ \oplus обозначает операцию XOR, а символ $+$ обозначает операцию дизъюнкции. Настройка на четыре единицы в байте (без учета особой точки $H(1)$) принимает вид $(f_2 \& g_1) + (g_2 \& f_1) + (g_1 g_2 \& f_1 f_2)$. Настройка на пять единиц в байте (без учета особой точки $H(1)$) принимает вид $(f_1 \& g_1 g_2) + (g_1 \& f_1 f_2)$. Настройка на шесть единиц в байте (без учета особой точки $H(1)$) имеет вид $f_1 \& g_1$. Утверждение доказано.

На рис. 3 дана структура сети для идентификации трех единиц в байте.

Замечание 1. В общем случае если на вход такой структуры подается выборка, состоящая не из 2^k разрядов, а, например, из 11 разрядов, то она выделяет два полубайта, а оставшиеся три бита дополняются одним битом до полубайта, и тем самым задача сводится к предыдущему случаю.

ИДЕНТИФИКАЦИЯ ОСОБЫХ ТОЧЕК

Из леммы 1 следует, что настройка на семь и восемь единиц в байте требует учета особых точек, в частности точек $H(1)$ во входных полубайтах. Поскольку структура S_2 (соответственно S_1) идентифицирует вместе с тремя единицами (тремя нулями) и четыре единицы (четыре нуля), то ее необходимо модифицировать так, чтобы можно было выделять особые точки. Это можно сделать простой модификацией базовой структуры S_2 (соответственно S_1), заменив в ней выходную функцию OR на AND. Структура преобразования кодов приведена на рис. 4: для особой точки $H(1)$ — структура T_1 , а для особой точки $H(0)$ — структура T_2 , а результаты преобразования даны в табл. 2.

Обозначим выходы этих подструктур $f_3^0 (g_3^0)$ и $f_3^1 (g_3^1)$ для идентификации четырех нулей и четырех единиц соответственно. Исходя из вышеприведенных таблиц на выходе f_3^1 модифицированной структуры имеем 1, если значением входного полубайта является особая точка $H(1)$. Аналогично для второй модифицированной структуры — на выходе f_3^0 имеем 1, если значением входного полубайта является особая точка $H(0)$.

Таким образом, соединяя выходы f_3^1 и f_1 с помощью операции AND, получаем на выходе единицу только тогда, когда на входе в полубайте будет четыре единицы (особая точка $H(1)$). Аналогично строится полуструктура для идентификации особой точки $H(0)$ на базе полуструктур f_3^0 и f_2 . Вышеописанная конструкция показана на рис. 5.

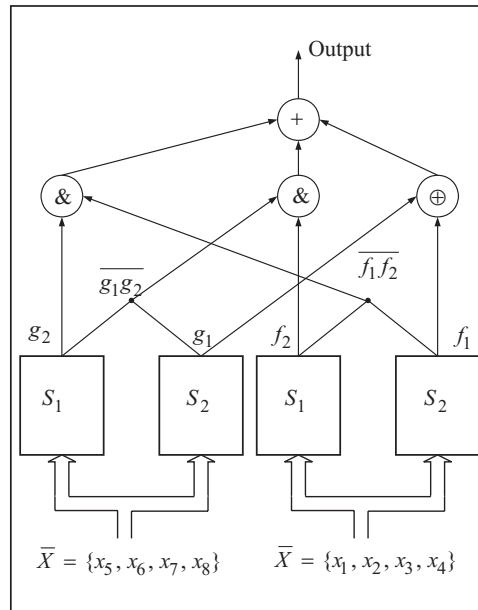


Рис. 3. Структура сети для идентификации трех единиц в байте

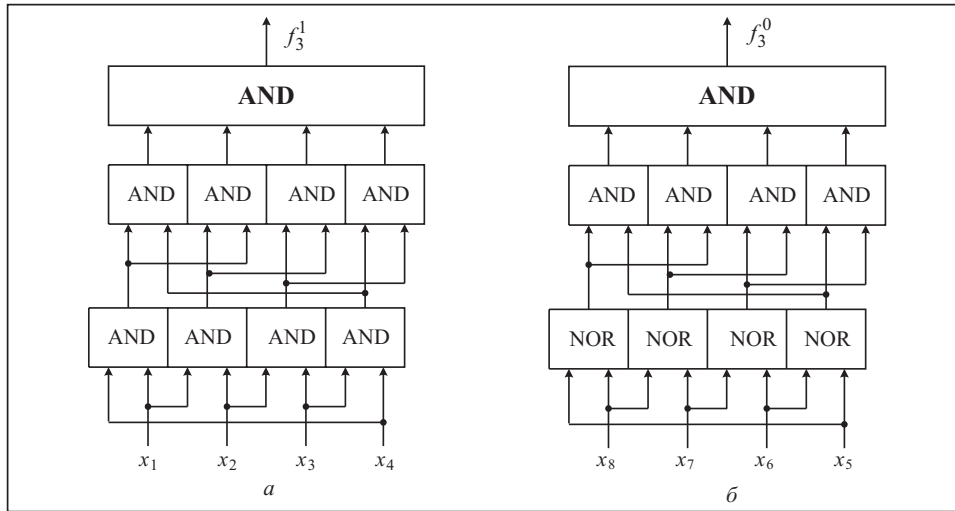


Рис. 4. Структура преобразователя циклических кодов на базе операций AND и NOR для определения особых точек: структура типа T_1 (а); структура типа T_2 (б)

Таблица 2. Преобразование циклических кодов для особых точек

Результаты преобразования структуры T_1				Результаты преобразования структуры T_2					
Входной код				Выход f_3^1	Входной код				Выход f_3^0
x_4	x_3	x_2	x_1		x_8	x_7	x_6	x_5	
1	1	1	1	1	0	0	0	0	1
0	1	1	1	0	0	0	0	1	0
1	0	1	1		0	0	1	0	
1	1	0	1		0	1	0	0	
1	1	1	0		1	0	0	0	
0	0	0	0		0	0	1	1	
0	0	0	1		0	1	0	1	
0	0	1	0		0	1	1	0	
0	0	1	1		0	1	1	1	
0	1	0	0		1	0	0	1	
0	1	0	1		1	0	1	0	
0	1	1	0		1	0	1	1	
1	0	0	0		1	1	0	0	
1	0	0	1		1	1	0	1	
1	0	1	0		1	1	1	0	
1	1	0	0		1	1	1	1	

Обоснование свойств приведенной структуры дает следующая теорема.

Теорема 2. Общая структура идентифицирует произвольное содержимое входного байта путем настройки ее выходов на число единиц (нулей) в байте.

Доказательство. Из леммы 1 и приведенных выше модифицированных подструктур вытекает, что для идентификации четырех единиц в байте функция идентификации имеет вид

$$\begin{aligned}
 & (\overline{g_3^1} \& f_2 \& g_1 \& \overline{f_3^0}) + (\overline{f_3^1} \& g_2 \& f_1 \& \overline{g_3^0}) + (\overline{g_1 g_2} \& \overline{f_1 f_2}) + \\
 & + [(f_1 \& f_3^1) \oplus (g_1 \& g_3^1)].
 \end{aligned}$$

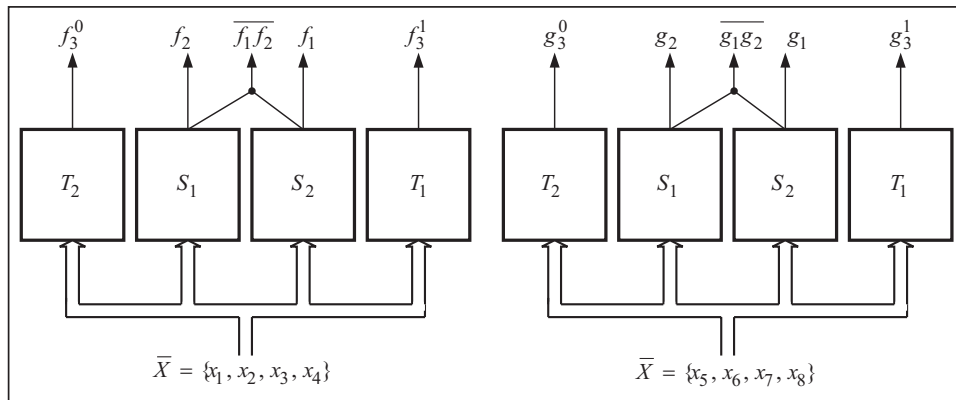


Рис. 5. Общая структура сети с учетом особых точек

Действительно, единицы во входном байте могут распределиться так, что в первом (во втором) полубайте первой подструктуры будет одна единица, а во втором (первом) полубайте второй подструктуры будет три единицы.

Из первых двух членов в вышеприведенном выражении вытекает следующее:

- если $g_3^1 = 0$, то входной полубайт второй подструктуры содержит не более трех единиц, и если $g_1 = 1$, то он содержит в точности три единицы;
- если $f_3^1 = 0$, то входной полубайт первой подструктуры содержит не более трех нулей, и если $f_2 = 1$, то он содержит только одну единицу.

Тогда в байте будет в точности четыре единицы. Справедливость сказанного для второго выражения следует из симметрии расположения, а для остальных двух выражений это очевидно. Идентификация пяти единиц в байте выполняется с помощью настройки на функцию

$$(f_2 \& f_3^0 \& g_1 \& g_3^1) + (g_2 \& g_3^0 \& f_1 \& f_3^1) + (\overline{f_1 f_2} \& g_1 \& \overline{g_3^1}) + (\overline{g_1 g_2} \& f_1 \& \overline{f_3^1}).$$

Аналогичным способом можно получить остальные комбинации единиц в байте.

Идентификация восьми единиц в байте выполняется с помощью настройки на функцию $(f_1 \& f_3^1) + (g_1 \& g_3^1)$.

Теорема доказана.

С учетом замечания 1 отметим, что теорема 2 справедлива для произвольного расстояния Хемминга и произвольной разрядности входного вектора.

СТРУКТУРНАЯ ОРГАНИЗАЦИЯ СЕТЕЙ ХЕММИНГА

Постановка задачи [5]. Задано множество шаблонов-векторов $B \subset U$ (U — полное множество n -мерных двоичных векторов $\{u_p\}$, $p = \overline{1, 2^n}$; $B = \{b_h\}$, $h = \overline{1, q}$, $q < 2^n$, $\bigcap_h b_h = \emptyset$).

Для заданного множества шаблонов-векторов B и произвольного множества $U_1 \subseteq U$ ($U_1 = \{u_j\}$, $j = \overline{1, \alpha}$) определить множества $C_j = \{c_{jh}\}$ и сформировать множества $Y_j = \{y_j^i: i = \overline{1, z_j}\}$, причем

$$y_j^i = \mu_{jh} \mapsto b_h, \quad (4)$$

где символом \mapsto обозначена конкатенация; c_{jh} — двоичный вектор (непозиционный двоичный код), определяющий расстояние Хемминга между n -мерными двоичными векторами u_j и b_h ,

$$c_{jh} = \sum_{\tau=1}^n (u_{j\tau} \oplus b_{h\tau}); \quad (5)$$

символ \oplus обозначает операцию XOR; $u_{j\tau}, b_{h\tau}$ — τ -е двоичные компоненты, $u_{j\tau}, b_{h\tau} \in \{0, 1\}$, векторов u_j и b_h соответственно; μ_{jh} — двоичный код значения расстояния Хемминга (определяется количеством единичных компонент непозиционного двоичного кода c_{jh}).

В соответствии с (4), (5) сеть Хемминга включает следующие основные функциональные блоки (рис. 6): ОЗУ шаблонов; преобразователь (5) на базе логического элемента XOR; преобразователь $\Lambda(c_{jh} \Rightarrow \mu_{jh})$ и ОЗУ результатов.

На вход преобразователя (5) поступает множество векторов $C_j = \{c_{jh}\}$. На выходе формируется множество $Y_j = \{y_{jh}^i : i = \overline{1, z_j}\}$.

Рассмотрим пример функционирования нейроподобной сети Хемминга.

Заданы мера близости $\mu_{jh} \leq 2$ и множество шаблонов:

$$B = \{b_1 = 10101011; b_2 = 11111001; b_3 = 00011101; b_4 = 00010111; b_5 = 10100100\}.$$

На вход нейроподобной сети подается входной двоичный вектор $u = 10100111$. Определить шаблоны-векторы b_n , в которых мера близости входного двоичного вектора c_{jh} удовлетворяет ограничению $\mu_{jh} \leq 2$. С учетом теорем 1, 2 и синтезированных структур (см. рис. 3, 5) получим результаты преобразования сети Хемминга (табл. 3).

Заданному ограничению удовлетворяют шаблоны $b_1 = 10101011$ и $b_5 = 10100100$, которые и будут записаны в ОЗУ результатов в соответствующую область памяти.

Расстояние Хемминга μ_{jh} имеет максимальную размерность $(\log_2 n + 1)$; таким образом, вектор y_j^i согласно (4) будет иметь размерность (длину), определяемую величиной $(n + \log_2 n + 1)$. В ОЗУ результата последовательно записываются векторы множества y_j^i (4). Объем памяти в ОЗУ результатов определяется суммарной мощностью множеств Y_j следующим выражением:

$$\text{Card}\{Y\} = \sum_j \text{Card}\{Y_j\},$$

где

$$\text{Card}\{Y_j\} = 1 \quad (\mu_{jh} = 0);$$

$$\text{Card}\{Y_j\} = n \quad (\mu_{jh} = 1);$$

⋮

$$\text{Card}\{Y_j\} = \frac{n!}{(n - \mu_{jh})! \mu_{jh}!} \quad (\mu_{jh} = n).$$

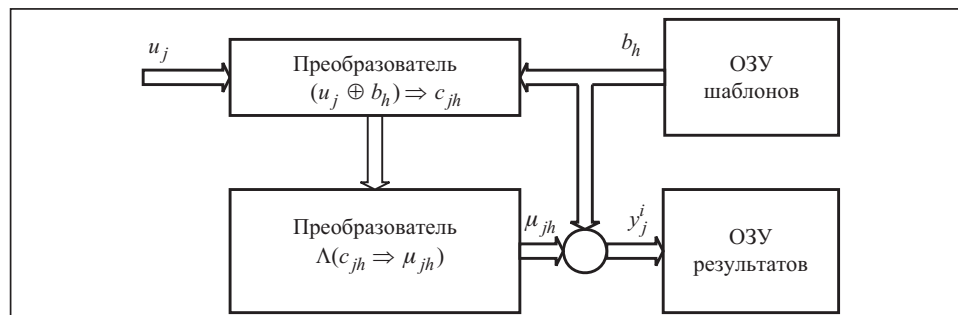


Рис. 6. Функциональная схема сети Хемминга

Таблица 3. Результаты преобразования в нейроподобной сети Хемминга

Входной вектор u_j	Результаты преобразования		
	Шаблоны-векторы b_h	c_{1h}	μ_{1h}
$u_1 = 10100111$	$b_1 = 10101011$	$c_{11} = 00001100$	$\mu_{11} = 2$
	$b_2 = 11111001$	$c_{12} = 01011110$	$\mu_{12} = 5$
	$b_3 = 00011101$	$c_{13} = 10110010$	$\mu_{13} = 4$
	$b_4 = 00010100$	$c_{14} = 10111010$	$\mu_{14} = 5$
	$b_5 = 10100100$	$c_{15} = 00000011$	$\mu_{15} = 2$

Для подмножества входных векторов, имеющих одинаковое значение расстояния Хемминга μ_{jh} , в соответствии с вышеприведенными выражениями в ОЗУ результатов отводится фиксированная область памяти для соответствующего значения μ_{jh} , размер которой четко соответствует значению $\text{Card} \{Y_j\}$.

ЗАКЛЮЧЕНИЕ

В настоящей статье обосновывается корректность синтеза многоуровневой структуры комбинационного типа с помощью базовых операций AND и NOR (преобразователи циклических кодов Хемминга) для распознавания подмножества двоичных векторов произвольной разрядности, мера близости (расстояние Хемминга) которых принадлежит некоторому диапазону.

На основе рассмотренных структур реализована нейроподобная сеть Хемминга, которая распознает входной двоичный вектор путем определения его меры близости с заданными векторами-шаблонами и записывает (вместе со значением расстояния Хемминга) в соответствующую область памяти результатов.

Рассмотренные структуры также могут быть использованы для решения широкого класса задач распознавания образов, где в качестве меры близости используется расстояние Хемминга.

СПИСОК ЛИТЕРАТУРЫ

1. Винцюк Т.К. Анализ, распознавание и интерпретация речевых сигналов. Киев: Наук. думка, 1987. 262 с.
2. Bruck J., Blaum M. Neural networks, error-correcting codes, and polynomials over the binary n -cube. *IEEE Transactions on Information Theory*. 1989. Vol. 35, N 5. P. 976–987.
3. Robinson M.E., Yoneda H., Sanchez-Sinencio E. A modular CMOS design of a Hamming network. *IEEE Transactions on Neural Network*. 1992. Vol. 3, N 3. P. 444–456.
4. Дмитриенко В.Д., Заковоротный А.Ю. Нейронная сеть, использующая расстояние Хемминга, для распознавания изображений на границах нескольких классов. *Вестник НТУ «ХПИ»*. 2013. № 39 (1012). С. 57–67.
5. Palagin A.V., Opanasenko V.N., Chigirik L.G. Synthesis of a Hamming network on a basis of programmable logic integrated circuits. *Engineering Simulation*. 1996. Vol. 13. P. 651–666.
6. Kondratenko Y.P., Gordienko E. Implementation of the neural networks for adaptive control system on FPGA. *Annals of DAAAM for 2012 & Proceeding of 23rd DAAAM International Symposium on Intelligent Manufacturing and Automation*. 23 (1). B. Katalinic (Ed.), Published by DAAAM International. Vienna. Austria, 2012. P. 0389–0392.
7. Палагин А.В., Опанасенко В.Н., Крытый С.Л. Реконфигурируемые структуры на базе FPGA: Синтез проблемно-ориентированных структур. Verlag: Lambert Academic Publishing, 2014. 54 p.
8. Palagin A., Opanasenko V., Kryvyi S. The structure of FPGA-based cyclic-code converters. *Optical Memory & Neural Networks (Information Optics)*. 2013. Vol. 22, N 4. P. 207–216.

9. Palagin A.V., Opanasenko V.N. Reconfigurable computing technology. *Cybernetics and Systems Analysis*. 2007, Vol. 43, N. 5. P. 675–686.
10. Palagin A.V., Opanasenko V.N. Design and application of the PLD-based reconfigurable devices. In: Design of Digital Systems and Devices. *Lecture Notes in Electrical Engineering*. M. Adamski, A. Barkalov, M. Wegryzn (Eds.). Berlin; Heidelberg: Springer-Verlag, 2011. Vol. 79. P. 59–91.
11. Палагин А.В., Опанасенко В.Н., Кривый С.Л. Метод синтеза структур для преобразований циклического кода на базе FPGA. *Электронное моделирование*. 2014. Т. 36, № 2. С. 27–48.
12. Брейтон Р.К., Хэтчел Г.Д., Санджованни-Винченцелли А.Л. Синтез многоуровневых комбинационных логических схем. *ТИИЭР*. 1990. Т. 78, № 2. С. 38–83.
13. Opanasenko V.N., Kryvyi S.L. Partitioning the full range of boolean functions based on the threshold and threshold relation. *Cybernetics and Systems Analysis*. 2012. Vol. 48, N 3. P. 459–468.
14. Opanasenko V.N., Kryvyi S.L. Synthesis of adaptive logical networks on the basis of Zhegalkin polynomials. *Cybernetics and Systems Analysis*. 2015. Vol. 51, N 6. P. 969–977.
15. Opanasenko V., Kryvyi S. Synthesis of multilevel structures with multiple outputs. *CEUR Workshop Proceeding of 10th International Conference of Programming, UkrPROG 2016, Kyiv, Ukraine*. 2016. Vol. 1631, Code 122904. P. 32–37.

Надійшла до редакції 17.02.2017

В.М. Опанасенко, С.Л. Кривий
СИНТЕЗ НЕЙРОПОДІБНИХ МЕРЕЖ НА ОСНОВІ ПЕРЕТВОРЕНЬ ЦИКЛІЧНИХ КОДІВ ХЕМІНГА

Анотація. Розглянуто синтез нейроподібної мережі Хемінга для реалізації задачі класифікації вхідної множини двійкових векторів. Формування відсортованої послідовності за мірою близькості (відстань Хемінга) базується на перетвореннях циклічних кодів Хемінга. Доведено коректність синтезу такої реалізації для довільного значення відстані Хемінга та довільної розрядності вхідного вектора.

Ключові слова: булева функція, нейроподібна мережа, відстань Хемінга, циклічний код.

V.N. Opanasenko, S.L. Kryvyi
SYNTHESIS OF NEURAL-LIKE NETWORKS BASED ON THE CONVERSION OF CYCLIC HAMMING CODES

Abstract. The authors consider the problem of synthesis of neural-like Hamming networks for the implementation of the problem of classifying an input set of binary vectors. Formation of the sorted sequence by the criterion of proximity measures (Hamming distance) is based on the conversion of cyclic Hamming codes. The correctness of the synthesis of such implementation for an arbitrary value of Hamming distance and arbitrary bit capacity of the input vector is proved.

Keywords: Boolean function, neural-like network, Hamming distance, cyclic code.

Опанасенко Владимир Николаевич,
 доктор техн. наук, профессор, ведущий научный сотрудник Института кибернетики им. В.М. Глушкова НАН Украины, Киев, e-mail: opanasenko@incyb.kiev.ua.

Кривый Сергей Лукьянович,
 доктор физ.-мат. наук, профессор кафедры Киевского национального университета имени Тараса Шевченко, Киев, e-mail: krivoi@i.com.ua.