Лабораторно упражнение № 6

НАДЕЖДНОСТ НА КОМПЮТЪРНАТА ОБРАБОТКА И ПРИЛОЖЕНИЕ НА ОТКАЗОУСТОЙЧИВИ КОДОВЕ

6.1. Теоретична постановка

6.1.1. Същност и предназначение на кодирането при компютърната обработка

Кодирането на информацията в общ смисъл е важен проблем в областта на компютърната обработка, информационните технологии и програмирането. То се свързва пряко със следните насоки:

- кодиране на различни по характер данни (числа, текст, графика) и представянето им в паметта на компютъра;
- защита на информацията от несанкциониран достъп;
- осигуряване на шумоустойчивост при предаване на данни по канали за връзка;
- компресиране на данни при съхраняване и предаване;
- самото създаване на текста на програмата се нарича кодиране.

Код е система от символи и правила за еднозначно представяне на данните в КС, а кодирането е функционално преобразуване на данните при това представяне. Ако са зададени две азбуки $A = \{a_1, \ldots, a_n\}$ и $B = \{b_1, \ldots, b_m\}$, то кодиране се нарича функцията $FS \rightarrow B^*$, където A^* и B^* са множествата на всички думи над съответните азбуки, а $S \subset A^*$ е подмножество на A^*, включващо съобщенията за кодиране. Елементите $\beta = F(s)$, където $s \in S$ и $\beta \in B^*$, се наричат кодови думи или накратко кодове на съответните съобщения. Обратната функция F^{-1} (ако съществува) се нарича декодиране. Ако мощността на множеството B е $|B| = m$, функцията F се нарича $\underline{m}$-ично кодиране. В КС се използва двоично кодиране, където $|B| = 2$ и $B = \{0,1\}$.

Една типична задача пред теорията на кодирането е намиране на оптимално кодиране (дефиниране на подходяща функция F) при зададени азбуки A, B и множество на съобщенията S. Критерият за оптималност обикновено е минимална дължина на кодовите думи, а свойствата на кода могат да се свържат с изискванията:

- да съществува обратна функция F^{-1} (декодиране) – твърде естествено свойство при представяне и предаване на информацията, но понякога

то не се изисква; например, транслацията на програма от език на високо ниво в машинен език е кодиране, за което не се изисква еднозначно декодиране;
- да е устойчив на грешки или да има възможност за корекция на грешки;
- да притежава определена простота или достатъчна сложност за функциите F и F^{-1}. Например, в криптографията се разглеждат такива методи за кодиране, при които F е просто изчислима, но определянето на F^{-1} изисква много сложни изчисления.

Голямото значение за задачите на кодирането има природата на S (типа на самите съобщения). При еднакви азбуки A, B и изисквания към F, оптималните решения могат съществено да се различават при различни множества S.

За описване на множеството на съобщенията S се прилагат различни методи:

- Теоретико-множествено описание, напр. $S = \{s \,/\, (s \in A^*) \,\&\, (|s| = k)\}$.
- Вероятностно описание, напр. $S = A^*$ и вероятности p_i за появя на съответна буква в съобщението, като $\Sigma p_i = 1$.
- Логико-комбинаторно описание, напр. S се задава чрез пораждаща формална граматика.

6.1.2. Побуквено кодиране

Най-често прилаганото кодиране е *побуквеното*. Съобщенията $s \in S$ се състоят от части (букви) $a_i \in A$, като всяко съобщение $s = a_1a_2\ldots a_k \in A^*$ има фиксирана дължина от k символа. За всяка буква a_i се дефинира еднозначен уникален код $\beta_i \in B^*$, като кодовата функция се задава чрез *кодова таблица (схема)*: $\sigma := \langle a_1 \to \beta_1, a_2 \to \beta_2, \ldots a_k \to \beta_k \rangle$; $a_i \in A$; $\beta_i \in B^*$, където β_i, се наричат елементарни кодове, а $V = \{\beta_1, \beta_2, \ldots \beta_n\}$ – множество на елементарните кодове.

Изисква се схемата да осигурява еднозначно кодиране. Буквеното кодиране може да се приложи за всяко множество от съобщения: $F{:}A^* \to B^*$; $s \in A^*$; $F(s) := \beta_a\beta_b,\ldots\beta_q$.

Пример:

Зададени са азбуките $A := \{0,1,2,3,4,5,6,7,8,9\}$ и $B := \{0,1\}$, както и кодовата схема:

$\sigma' := \langle 0 \to 0,1 \to 1,2 \to 10,3 \to 11,4 \to 100,5 \to 101,6 \to 110,7 \to 111,8 \to 1000,9 \to 1001 \rangle$

Схемата е еднозначна, но не осигурява еднозначно кодиране, защото допуска ситуацията $F_{\sigma'}(333) = 111111 = F_{\sigma'}(77)$. За преодоляване на този недостатък се прилага схемата за двоично-десетично кодиране, представяна чрез:

$\sigma'' := \langle 0 \to 0000, 1 \to 0001, 2 \to 0010, 3 \to 0011, \ldots, 8 \to 1000, 9 \to 1001 \rangle$

Схемата σ на побуквеното кодиране е *разделима*, ако всяка дума, съставена от елементарни кодове, може да се разложи на елементарните кодове. Побуквено кодиране с разделима схема допуска декодиране. Ако кодовата таблица съдържа еднакви елементарни кодове, т.е. ако $\exists i,j$ $(i \neq j)$, за които $\beta_i = \beta_j$, $\beta_i, \beta_j \in V$, то кодовата схема σ не е разделима и не води до еднозначно кодиране.

6.1.3. Кодиране с минимален излишък

На практика е важно кодираните съобщения да бъдат с минимална дължина. Това е свързано с икономия на ресурс (памет за съхраняване, канали за предаване, време за обработка и пр.). Ако е зададена разделима схема на азбучно кодиране $\sigma = <a_i \rightarrow \beta_i / i = 1 \div n>$, то всяка схема $\sigma' = <a_i \rightarrow \beta_i / i = 1 \div n>$, където $<\beta_1, \beta_2, \ldots, \beta_n>$ е някакво подреждане на $<\beta_1, \beta_2, \ldots, \beta_n>$, също ще бъде разделима. Ако елементарните кодове имат еднаква дължина, то пренареждането им няма да се отрази върху дължината на кода на съобщението. При различна дължина, обаче, дължината на кода на съобщението ще зависи от съставящите го букви.

Ако са зададени конкретни съобщения и конкретна кодираща схема, то не е трудно да се избере такова подреждане на елементарните кодове, че дължината на кода на съобщението да е минимална.

Дължината на кода е свързана с т.нар. *цена (дължина) на кодирането*. Нека за азбука $A = \{a_1, \ldots, a_n\}$ и вероятности $P = <p_1, \ldots, p_n>$ за попадане на всяка буква от A в дадено съобщение е дефинирана кодова схема $\sigma = <a_i \rightarrow \beta_i / i = 1 \div n>$ на азбучно кодиране ($\Sigma p = 1$).

Средна цена на схема σ се нарича математическото очакване за нарастване на дължината на кода на съобщението (l_σ) при кодирането и се определя:

$$l_\sigma(P) := \sum_{i=1}^{n} p_i l_i, \text{ където } l_i := |\beta_i|.$$

Пример:

$A = \{a,b\}$; $B = \{0,1\}$; $\sigma: = \{a \rightarrow 0;\ b \rightarrow 01\}$ (разделима схема)

$P = <0{,}5;\ 0{,}5> \Rightarrow l_\sigma(P) = 0{,}5.1 + 0{,}5.2 = 1{,}5$

$P = <0{,}9;\ 0{,}1> \Rightarrow l_\sigma(P) = 0{,}9.1 + 0{,}1.2 = 1{,}1$

Винаги може да се намери разделима схема $\sigma: = <a_i \rightarrow \beta_i / i = 1 \div n>$, за която $\forall i,\ |\beta_i| = L = \text{const}$. Такава схема се нарича *равномерно кодиране*. Схема σ с минимална цена за дадено разпределение на вероятностите P се нарича *кодиране с минимален излишък* или *оптимално кодиране*.

6.1.4. Отказоустойчиво кодиране и класификация на грешките

При работа на електронните устройства е възможно да възникнат грешки (систематични или случайни). Причина за това могат да бъдат различни външни въздействия, изкривявания на сигнала по линиите за връзка, дефекти в магнитната повърхност на диска и пр. Такива грешки водят до промяна или загуба на предаваните или съхраняваните данни. За намаляване на външните въздействия, както и за откриване на грешки в информацията, се прилагат *отказоустойчиви кодове*.

Нека С е канал за пренасяне на съобщения, съдържащ източник на грешки. Това може да се запише чрез $s \xrightarrow{C} s'$, където $s \in A^*$ е предаденото съобщение, а $s' \in B^*$ е приетото съобщение. Ако с S и S' се отбележат множествата на предадените и приетите съобщения, то кодиране F, което притежава свойството:

$$S \xrightarrow{F} K \xrightarrow{C} K \xrightarrow{F^{-1}} S;\ \forall s \in S;\ F^{-1}(C(F(s))) = S,,$$

се нарича *отказоустойчиво кодиране* или *кодиране с откриване на грешки.*

В КС се използва двоичната азбука, т.е. $A = B = \{0,1\}$ и множествата A^* и B^* съдържат думи над $\{0,1\}$. Ако се приеме, че при самото кодиране не се въвежда грешка, то при предаване са възможни следните *видове грешки* (с Λ е означена празна дума):

(1) $0 \to 1;\ 1 \to 0$ (промяна на разряд);
(2) $0 \to \Lambda;\ 1 \to \Lambda$ (пропадане на разряд);
(3) $\Lambda \to 0;\ \Lambda \to 1$ (добавяне на разряд).

Канал C се характеризира с горна граница δ_i за броя на грешките от вид (i), които са възможни при предаване на съобщения с определена дължина. Общата оценка се означава с $\delta = <\delta_1, \delta_2, \delta_3>$.

Пример (отказоустойчиво кодиране):

Нека канал C има характеристика $\delta = <1,0,0>$, т.е. възможна е една грешка от първи вид (промяна). Предаваните съобщения са $s = a \in \{0,1\}$.

Кодиране: $F(a) = aaa$

Декодиране: $F^{-1}(abc) = a + b + c > 1$ (мажоритарно възстановяване)

Възможни ситуации:

$$a=1 \Rightarrow a \xrightarrow{F} K = aaa = 111 \xrightarrow{C} K' = 110 \xrightarrow{F^{-1}} a = [1+1+0>1] \Rightarrow a = 1$$

$$a=0 \Rightarrow a \xrightarrow{F} K = aaa = 000 \xrightarrow{C} K' = 010 \xrightarrow{F^{-1}} a = [0+1+0 \le 1] \Rightarrow a = 0$$

6.1.5. Кодово разстояние

Разстояние (метрика) на множество M е неотрицателна функция $d(x,y)$: $M \times M \rightarrow \mathbf{R}_+$, за която са изпълнени условията, наричани аксиоми на метриката:

1) $d(x,y) = 0 \Leftrightarrow x = y$
2) $d(x,y) = d(x,y)$
3) $d(x,y) \leq d(x,z) + d(y,z)$

Кодово разстояние (разстояние по Хеминг) е метриката на всяко множество A^*, представляващо *кодовото пространство* за даден код F над множеството $A = \{0,1\}$. При двоичното кодиране в КС, използващо ограничен брой разряди на машинната дума, дължината на кодовата дума определя броя на възможните комбинации, които дефинират кодово пространство. За n-разрядна кодова дума мощността на кодовото пространство A^* над множеството $\{0,1\}$ е $N = 2^n$. Две различни кодови думи $\alpha = \{\alpha_1\alpha_2\ldots\alpha_n\}$ и $\beta = \{\beta_1\beta_2\ldots\beta_n\}$ от едно кодово пространство се намират на кодово разстояние $d(\alpha,\beta)$ една от друга, определяно от броя на различаващите се по стойност разряди за две думи, т.е.

$$d(\alpha,\beta) = \alpha_1 \oplus \beta_1 + \alpha_2 \oplus \beta_2 + \alpha_3 \oplus \beta_3 + \ldots + \alpha_n \oplus \beta_n .$$

Пример:

$$A = \{0,1\};\ n = 4 \Rightarrow |A^*| = 2^4 = 16;\ \alpha,\beta \in A^*$$
$$\alpha = <1001>;\ \beta = <0010> \Rightarrow d(\alpha,\beta) = 3$$

Най-малкото кодово разстояние $d_{\min} = \text{MIN}\{d_j/j = 1 \div N\}$, осигурено от даден код, се нарича *минимално кодово разстояние* и определя степента на защитеност на кода. За да бъде един код защитен, в неговото кодово пространство трябва да е осигурено $d_{\min} \geq 2$. В такъв код е въведен излишък от кодови комбинации, които не се използват като правилни кодови думи. Така множеството A^* на кодовите комбинации се разделя на две непресичащи се подмножества: A^*_1 – на правилните кодови думи; A^*_2 – на неизползваните кодови комбинации.

Ако α е изходното кодирано съобщение, а β е полученото след пренасяне по канал съобщение, то възможни са следните ситуации:

а) $\beta = \alpha$; $\alpha,\beta \in A^*_1$ – пренасяне без грешка;
б) $\beta \neq \alpha$; $\alpha \in A^*_1$ и $\beta \in A^*_2$ – възникване на откриваема грешка;
в) $\beta \neq \alpha$; $\alpha,\beta \in A^*_1$ – възникване на неоткриваема грешка.

Въвеждането на излишък става чрез допълване на информационната дума (от n разряда) с контролно поле (k на брой контролни разряди), с което се

увеличава d_{min}. Така се формира кодова дума с дължина $n+k$ разряда. Отказоустойчивите възможности на такъв код се определят от условията:

- $d_{min} \geq r+1$ – за откриване на r-кратна грешка;
- $d_{min} \geq 2r+1$ – за откриване и коригиране на r-кратна грешка.

6.1.6. Разделими кодови схеми за отказоустойчиво кодиране

Разделим код е код (кодова схема), при който могат да бъдат отделени контролните разряди от информационните. При отказоустойчивото кодиране на информацията в КС се прилагат основно два типа кодове – кодове, които могат да откриват наличие на грешка и кодове, които освен откриване, могат и да коригират възникнала грешка. По-долу са представени базови кодове, прилагани в КС.

- ***Двоичен код по четност/нечетност.*** Използва един контролен разряд, добавян към информационното поле X от 8 бита. Това е разделим код, т.е. допуска разпознаване и отделяне на контролното поле. Кодовата дума може да се запише $<x_7x_6x_5x_4x_3x_2x_1x_0k>$, където:

$$k = \begin{cases} x_0 \oplus x_1 \oplus x_2 \oplus x_3 .... \oplus x_7; \text{ код по четност;} \\ x_0 \oplus x_1 \oplus x_2 \oplus x_3 .... \oplus x_7 \oplus 1; \text{ код по нечетност.} \end{cases}$$

Поддържа $d_{min} = 2$, което определя откриваемост на единична или нечетно-кратна грешка.

Пример:

Информационна дума:	$X = <10011001>$; $n = 8 \Rightarrow \lvert A^* \rvert = 2^8$
Контролен бит по четност:	$k = 1 \oplus 0 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 0 \oplus 1 = 0$
Изпратена кодова дума:	$\alpha = X$; $k = <10011001;0>$
Получена кодова дума:	$\beta = <10011\underline{0}1;0>$ (с еднократна грешка в маркиран разряд)
Проверка:	$k' = 1 \oplus \underline{1} \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 0 \oplus 1 = 1 \neq k = 0$

- ***Код с остатък.*** Двоичен код по *mod m* (m>2), който е разделим код с дължина на контролното поле $\log_2 m$. Ако информационната част е $<x_0x_1 \ldots x_{n-1}>$, то контролното поле се определя като $K = \text{REST}[(\Sigma x_j.2^j) \text{mod } m]$. За двоичен код ($p = 2$) е в сила $m = p + 1 = 3$, като се осигурява $d_{min} = 2$.

Пример:

Информационна дума: $X = <00010011>$

Изчислява се $(\Sigma x_j.2^j) = 19$ и се определя контролно поле

$K = \text{REST}[(19) \text{mod } 3] = 01$

Изпратена кодова дума: $\alpha = X$; $K = <00010011;01>$
Получена кодова дума: $\beta = <00011011;01>$ (с еднократна грешка в посочения разряд)
Проверка (при декодиране): $(\Sigma x_j.2^j) = 2^0 + 2^1 + 2^3 + 2^4 = 27 \Rightarrow K' =$
$= \text{REST}[(27) \bmod 3] = 00 \neq K = 01 \Rightarrow$ грешка

• ***Самокоригиращ се код на Хеминг.*** Самокоригиращите кодове позволяват откриване и коригиране на възникнали при пренасяне грешки, за което се поддържа по-голямо минимално кодово разстояние. Постига се чрез формиране на m-разрядна кодова дума, като към информационната част (n разряда) се добавят k контролни разряда ($m = n + k$). Типичен представител е *код на Хеминг*, който е разделим код и може да открива и коригира еднократна грешка от тип промяна ($d_{min} = 3$). За откриване на еднократна грешка броят на контролните разряди k се избира чрез $2^k \geq m + 1$, т.е.

$$2^k \geq m+1 \Rightarrow \frac{2^m}{m+1} \geq 2^{m-1} \Rightarrow \frac{2^m}{m+1} \geq 2^n$$

Например: ако $n = 32 \Rightarrow 2^k \geq m + 1 \Rightarrow 2^k \geq n + k + 1 \Rightarrow 2^k \geq 32 + k + 1$. От последното възможностите са:

$k = 4 \Rightarrow 2^k = 16$;
$k = 5 \Rightarrow 2^k = 32$;
$k = 6 \Rightarrow 2^k = 64$ и т.н.

Както се вижда, условието $2^k \geq 32 + k + 1$ се изпълнява при $k = 6 \Rightarrow 2^6 =$
$= 64 \geq 32 + 6 + 1 = 39$.

При *формиране на кодовата дума* контролните разряди k_i се разполагат на позиции с индекс $i = 2^j$ ($j = 0,1,2,\ldots$), т.е. на позиции $i = 1, 2, 4, 8$ и т.н.:

$$\alpha = <\alpha_m \ldots \alpha_3\alpha_2\alpha_1> = <\alpha_m \ldots \alpha_9 k_8 \alpha_7\alpha_6\alpha_5 k_4 \alpha_3 k_2 k_1>,$$

където броят k на контролните разряди се определя чрез $2^k \geq m + 1$ (при $m = n + k$), а номерирането е от 1 до m.

Кодирането на информацията се основава на формиране на групи V_i от индекси, които имат стойност '1' за i-тия разряд на двоичните им кодове:

$V_1 = 1, 3, 5, 7, 9, 11, \ldots$ (с '1' в най-младшия – първия разряд на двоичния им код)
$V_2 = 2, 3, 6, 7, 10, 11, \ldots$ (с '1' във втория разряд на двоичния им код)
$V_3 = 4, 5, 6, 7, 12, \ldots$ (с '1' в третия разряд на двоичния им код)
$V_4 = 8, 9, 10, 11, 12, \ldots$ (с '1' в четвъртия разряд на двоичния им код)

Разрядите от кодовата дума α, които попадат в една и съща група, се сумират по модул 2 за формиране на условията $P_j = 0$ ($j = 0, 1, 2, \ldots$), чрез които се определят стойностите за контролните разряди:

$$P_0 = k_1 \oplus \alpha_3 \oplus \alpha_5 \oplus \alpha_7 \oplus \alpha_9 \oplus \alpha_{11} \ldots = 0 \Rightarrow k_1 = \alpha_3 \oplus \alpha_5 \oplus \alpha_7 \oplus \alpha_9 \oplus \alpha_{11} \ldots$$
$$P_1 = k_2 \oplus \alpha_3 \oplus \alpha_6 \oplus \alpha_7 \oplus \alpha_{10} \oplus \alpha_{11} \ldots = 0 \Rightarrow k_2 = \alpha_3 \oplus \alpha_6 \oplus \alpha_7 \oplus \alpha_{10} \oplus \alpha_{11} \ldots$$
$$P_2 = k_4 \oplus \alpha_5 \oplus \alpha_6 \oplus \alpha_7 \oplus \alpha_{12} \ldots = 0 \quad \Rightarrow k_4 = \alpha_5 \oplus \alpha_6 \oplus \alpha_7 \oplus \alpha_{12} \ldots$$
$$P_3 = k_8 \oplus \alpha_9 \oplus \alpha_{10} \oplus \alpha_{11} \oplus \alpha_{12} \ldots = 0 \quad \Rightarrow k_8 = \alpha_9 \oplus \alpha_{10} \oplus \alpha_{11} \oplus \alpha_{12} \ldots$$

При *декодиране* на получената кодова дума $\beta = (\beta_1\beta_2\beta_3 \ldots \beta_m)$ се проверява изпълнението на условията $P_j = 0$ ($j = 0,1,2,\ldots$). При неизпълнение, числото

$$N = \ldots P_3.2^3 + P_2.2^2 + P_1.2^1 + P_0.2^0$$

определя номера на сгрешения разряд в β, който се коригира чрез инвертиране.

Пример:

Изходна информационна дума:
$X = x_8x_7x_6x_5x_4x_3x_2x_1 = 10101001$ ($n = 8$);
от $2^k \geq n + k + 1 \Rightarrow 2^k \geq 9 + k \Rightarrow k = 4 \Rightarrow m = n + k = 12$.

Кодова дума: $\alpha = \alpha_{12}\alpha_{11}\alpha_{10}\alpha_9\alpha_8\alpha_7\alpha_6\alpha_5\alpha_4\alpha_3\alpha_2\alpha_1$
$\alpha = x_8x_7x_6x_5k_8x_4x_3x_2k_4x_1k_2k_1 = 1010k_8100k_41k_2k_1$

Кодиране: $k_1 = 1 \oplus 0 \oplus 1 \oplus 0 \oplus 0 = 0$ $\quad k_4 = 0 \oplus 0 \oplus 1 \oplus 1 = 0$
$k_2 = 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 = 1$ $\quad k_8 = 0 \oplus 1 \oplus 0 \oplus 1 = 0$

Предадена дума: $\alpha = \alpha_{12}\alpha_{11}\alpha_{10}\alpha_9k_8\alpha_7\alpha_6\alpha_5k_4\alpha_3k_2k_1 = 101001000110$

Приета дума (с грешка в трети разряд): $\beta = 101001000\underline{0}10$

Декодиране: $P_0 = \beta_1 \oplus \beta_3 \oplus \beta_5 \oplus \beta_7 \oplus \beta_9 \oplus \beta_{11} = 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 0 = 1$
$P_1 = \beta_2 \oplus \beta_3 \oplus \beta_6 \oplus \beta_7 \oplus \beta_{10} \oplus \beta_{11} = 1 \oplus 0 \oplus 0 \oplus 1 \oplus 1 \oplus 0 = 1$
$P_2 = \beta_4 \oplus \beta_5 \oplus \beta_6 \oplus \beta_7 \oplus \beta_{12} = 0 \oplus 0 \oplus 1 \oplus 0 \oplus 1 = 0$
$P_3 = \beta_8 \oplus \beta_9 \oplus \beta_{10} \oplus \beta_{11} \oplus \beta_{12} = 0 \oplus 0 \oplus 1 \oplus 0 \oplus 1 = 0$

Корекция: $N = P_3.2^3 + P_2.2^2 + P_1.2^1 + P_0.2^0 = 0.2^3 + 0.2^2 + 1.2^1 + 1.2^0 = 3$

6.1.7. Методи за повишаване на надеждността

В КС се използват различни методи за повишаване достоверността на информацията, като изборът зависи от степента на отговорност на изчисленията. Тези методи използват излишък (допълнителни средства), което повишава стойността на компютърната обработка. Основни методи са следните:

- *Информационен излишък* – използване на контролиращи и коригиращи кодове с определена отказоустойчивост, която се дефинира чрез въведе-

ното контролно поле K, допълващо информацията X до кодова дума $W = XK$. Прилагат се при предаване на данни, като на базата на приетата дума $W' = X'K'$ се изчислява отново стойността $K'' = f(X')$ на контролното поле и се сравнява с приетото K'. При съвпадение $K'' = K'$ се приема, че грешка не е настъпила.

- *Апаратен излишък* – прилагат се при обработващи устройства за изчисляване на аритметични функции $C = F(A,B)$ над защитени данни ($A = XK_X$; $B = YK_Y$; $C = ZK_Z$). Използват се допълнителни апаратни средства за изчисляване на стойността на контролното поле на резултата $K'_Z = f(K_X, K_Y)$ паралелно с извършване на основната операция, след което изчислената контролна стойност K'_Z се сравнява с тази от изчисления резултат $C = ZK_Z$.
- *Резервиране* – изпълнение на една и съща операция в две (дублиране) или повече (мажориране) устройства едновременно и следващо сравняване на получените резултати. Ако се открие несъвпадение при дублирането се съобщава за грешка. При мажорирането верният отговор се определя на принципа на гласуването. Тази група методи изисква значителен апаратен излишък, което повишава стойността на компютърната обработка и затова се прилагат при изчисления, които изискват висока надеждност и сигурност.

6.2. Задание за лабораторна работа

1. Да се разучат особеностите и предназначението на кодирането в компютърните системи и възможностите за повишаване на надеждността на компютърната обработка чрез приложение на отказоустойчиво кодиране на информацията.

2. Да се извърши подходящо побуквено кодиране с минимален излишък при зададени азбуки $A = \{a, b, c, d, e, f, h, g, i, j, k, l\}$ и $B = \{a, b\}$, като се осигури разделимост и еднозначност.

3. Да се анализират възможните видове грешки при предаване на двоична информация по канали в КС. Да се предложи схема за отказоустойчиво кодиране при предаване на съобщения $s = a \in \{0,1\}$ по канал с характеристика $\delta = <2,0,0>$.

4. Да се определи кодовото разстояние за кодовите думи 110011 и 110101. Какво е минималното кодово разстояние за кодово пространство над множеството $\{0,1\}$, съдържащо 2^n на брой n-разрядни думи?

5. За информационната дума <11001100> определете стойностите на необходимия брой контролни разряди съгласно следните кодове: код по четност; код по нечетност; код с остатък; код на Хеминг. Анализирайте параметрите на отказоустойчивост при всеки от тези кодове и въведения излишък.

(Приложението не е самостоятелна единица, а онагледява преподаван по предмета урок)

КОД НА ХЕМИНГ И EXCEL ПРИЛОЖЕНИЕ ЗА ГЕНЕРИРАНЕ И КОДИРАНЕ

Коригиращи кодове.

Важна характеристика на всеки код е т.н**. кодово разстояние, D,** което се изразява като различие между две кодови думи.

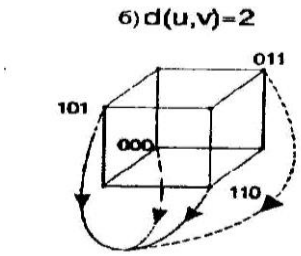
Най-разпространената мярка за разстояние между кодови думи е ***кодово разстояние по Хеминг***, което се дефинира като **разлика в съдържанието на разрядите на две кодови думи** – D(u, v).

Нека u=1**1**0**1**0, а v=1**0**0**0**1 – вижда се, че разлика има в **три разряда** т. е. D(u, v)=3.

Най-малката разлика в разрядите на две кодови думи (два кодови вектора) за цялото кодово пространство се дефинира като минимално кодово разстояние по Хеминг – Dmin. Този параметър показва ефикасността на кода за откриване и коригиране на грешки.

Примери: Нагледен начин за представяне на кодовите думи е обемният, при който на всяка дума се съпоставя точка от "n" мерното пространство. Кодовите думи са разположени във върховете на куб.

А) D_{min} = 1. Този код няма никаква възможност за откриване на грешка, защото при всяка грешка ще се попадне в друга разрешена комбинация.



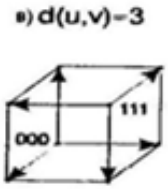
Фиг. 1.

Таблица 1.

X	0	0	0	0	1	1	1	1
Y	0	0	1	1	0	0	1	1
Z	0	1	0	1	0	1	0	1

Б) D_{min} = 2. От възможните 8 кодови думи са използвани 4, подбрани

така, че кодовото разстояние между две кои да е от тях да бъде 2. Този код има възможност за откриване на еднократна грешка, но не и за корекция.

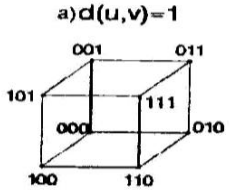


Фиг.2.

Таблица 2.

X	0	0	1	1
Y	0	1	0	1
	0	1	1	0

В) D_{min} = 3. От всички кодови комбинации са избрани 2. Съществува възможност за откриване на двукратна грешка и корекция на еднократна.



•

Фиг.3

Условието да бъде коригирана t-кратна грешка се дава чрез израза:

$$D_{min} \geq 2t+1, \quad (1)$$

- където t е кратността на грешката при корекция.
- Условието да бъде гарантирано откриването на t-кратна грешка се дава чрез израза:

$$D_{min} \geq t+1, \quad (2)$$

- където t е кратността на грешката при контрол.

3.1.Въвеждащ пример за коригиране на грешка.

Първоначално са необходими статистическите и аналитични изчисления, при каква максимална дължина на съобщението може да се получи не повече от една грешка.

За примера се приема, че при изпращането на съобщение с дължина 20 или по-малко нули и единици е възможно само на едно единствено място да се получи грешка. Нулата да бъде заменена с единица или обратно. Изключва се възможността за две и повече грешки.

Ако съобщението се състои от 4bit-а, въпросът е: „***Може ли да се допълни четири-символното съобщение с допълнителна информация, с която да се установи има ли грешка и ако има – да се поправи?***”

Един от възможните начини е три пъти да се изпрати съобщението с дължина 4bit.

Таблица 3. *Информативна част на съобщението*

1	2	3	4
0	1	1	1

Таблица 4. *Изпратено кодирано съобщение*

1	2	3	4	5	6	7	8	9	10	11	12
0	1	1	1	0	1	1	1	0	1	1	1

Полученото съобщение е с дължина 12 и в него е възможна една грешка, например в девета позиция.

Таблица 5.

1	2	3	4	5	6	7	8	9	10	11	12
0	1	1	1	0	1	1	1	1	1	1	1

За да се коригира грешката, съобщението се разделя на три части и се подреждат една под друга.

Таблица 6.

1	2	3	4
0	1	1	1
0	1	1	1
1	1	1	1
0	1	1	1

Грешка при предаването се открива, когато два от символите в колоната съвпадат, а третия се различава, както е в първа позиция. Правилният символ се среща два пъти – в случая нула.

3.2. Код на Хеминг.

Сн. 1.„Хеминг с котката”

Хеминг е открил код с минимална дължина на съобщението, което трябва да бъде изпратено, за да се открият и коригират грешките.

Ричард Уесли Хеминг (1915 –1998) започва изследователската си кариера в лабораторията на Бел, където открива коригиращия грешките код, носещ неговото име.

Кодът на Хеминг е един от най-известните систематични кодове за коригиране на единични грешки и има минимално кодово разстояние d_{min} =3.

Кодът на Хеминг се строи по такъв начин, че към началните информационни разряди се добавят известен брой контролни разряди, които се формират преди предаване на информацията на базата на определяне на четността на сумата на единиците в съответни групи информационни разряди.

След приемане на информацията контролната апаратура образува от получените информационни и контролни разряди т.нар. коригиращо число на базата на аналогично преброяване на единиците. При липса на грешка коригиращото число е равно на нула. При грешка стойността му посочва мястото на грешката - двоичния номер на грешния разряд в думата. Грешният разряд автоматично се коригира чрез инвертиране на стойността му.

3.2.1. Кодиране с код на Хеминг.

Кодът на Хеминг е с $D_{min}=3$ и следователно може да открива двукратни грешки и да коригира еднократни.

Таблица 7.

		k1	k2	k3	k4	n	2^n
0	00000000					0	1
1	00000001	X				1	2
2	00000010		X			2	4
3	00000011	X	X			3	8
4	00000100			X		4	16
5	00000101	X		X		5	32
6	00000110		X	X		6	64
7	00000111	X	X	X		7	128
8	00001000				X	8	256
9	00001001	X			X		
10	00001010		X		X		
11	00001011	X	X		X		
12	00001100			X	X		
13	00001101	X		X	X		
14	00001110		X	X	X		
15	00001111	X	X	X	X		
16	00010000						

При кодирането трябва да се получат контролните разряди. Те се намират в позициите на думата W*, които са точни степени на 2 / 2^0=1поз., 2^1=2поз, 2^2=4поз и т.н./. Контролните разряди се получават като се приравнят горните изрази на “0” и се изразят членовете, точни степени на 2 / W_1, W_2, W_4, W_8…./.

$к_1 \rightarrow W_1 = W_3 \oplus W_5 \oplus W_7 \oplus W_9 \oplus \ldots$ всички разряди през един;

$к_2 \rightarrow W_2 = W_3 \oplus W_6 \oplus W_7 \oplus W_{10} \oplus \ldots$ всички разряди през 2 по 2;

$к_3 \rightarrow W_4 = W_5 \oplus W_6 \oplus W_7 \oplus W_{12} \oplus W_{13} \oplus \ldots$ всички разряди през 4 по 4

$к_4 \rightarrow \ldots$ всички разряди през 8 по 8

Броят на контролните разряди “к” зависи от броя на информационните разряди “n” и се изчислява по формулата:

$$2^k \geq n + к + 1 \qquad (3)$$

Позициите на контролните разряди са строго фиксирани.

3.2.2. Пример - кодиране:

Информативната част на съобщението е с дължина n=4. От формулата се изчислява, че

$2^3 \geq 4 + 3 + 1$, изпратеното съобщение ще е с дължина m=7. Същото се вижда и от табл. 7.

Позициите с номера, които не са степени на две (3, 5, 6, 7) са за самото съобщение, а позициите с номера степените на двойката (1, 2, 4) са служебни.

Попълват се позициите за съобщението, а служебните остават свободни:

Таблица 8.

1	2	3	4	5	6	7
		0		1	1	1

Построява се матрица представляваща числата от 1 до 7 в двоична бройна система (извадка от табл. 7). Това са трибуквени думи от азбуката {0,1} , подредени лексикографски (както в речник). Номерата на редовете са отляво.

Таблица 9.

1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Таблица 10.

Редът с номер 1 е равен на сумата по модул две от трите реда с номера 3 , 5 и 7;

$[1]=[3] \oplus [5] \oplus [7]$, защото

$(001)= (011) \oplus (101) \oplus (111)$.

С аналогични изчисления проверяваме, че:

$$\begin{array}{rrrr} & 0 & 1 & 1 \\ \oplus & 1 & 0 & 1 \\ \hline & 1 & 1 & 0 \\ \oplus & 1 & 1 & 1 \\ \hline & 0 & 0 & 1 \end{array}$$

$[2]=[3] \oplus [6] \oplus [7] \Leftrightarrow (010)= (011) \oplus (110) \oplus (111)$

$[4]=[5] \oplus [6] \oplus [7] \Leftrightarrow (100)= (101) \oplus (110) \oplus (111)$

Сега служебната позиция с номер 1, по подобие на равенството $[1]=[3] \oplus [5] \oplus [7]$ се попълва със сума по модул две от числата, стоящи на позиции 3, 5 и 7:

$0 \oplus 1 \oplus 1=0$

Таблица 11.

1	2	3	4	5	6	7
0		0		1	1	1

Позицията с номер 2, по подобие на равенството $[2]=[3] \oplus [6] \oplus [7]$ се попълва със сумата от числата, стоящи на позиции 3, 6 и 7:

$0 \oplus 1 \oplus 1=0$

Таблица 12.

1	2	3	4	5	6	7
0	0	0		1	1	1

$[4]=[5] \oplus [6] \oplus [7]$

$1 \oplus 1 \oplus 1=1$

Позицията с номер 4, по подобие на равенството се попълва със сумата от числата, стоящи на позиции 5, 6 и 7:

Таблица 13.

1	2	3	4	5	6	7
0	0	0	1	1	1	1

Това е кодираното съобщение.

3.2.3.Декодиране на код на Хеминг.

При декодирането се образува двоично число “А”, наречено контролно. Ако “А” е “0”-няма грешка, ако е различно от нула-показва позицията, в която има грешка.

Разрядите на контролното число се получават с помощта на операцията сума по модул 2 от разрядите на получената по канала за връзка дума W*, като се използва определен алгоритъм.

Алгоритъм за получаване на контролното число А=a_1 a_2 a_3 ……..$a_к$ от кодовата дума W*=W_1 W_2…….Wl_{-1} W_l:

1	2	3	4	5	6	7
0	0	0	1	1	0	1
0	1		0			

+	0	0	1
	0	1	0
	0	1	1

Получават се “к” равенства, така се получават всички разряди на числото “А”.

От получените информационни битове се изчисляват повторно контролните битове.

Сравняват се получените и изчислените контролни битове.

Ако има поне два различни бита на съответните си позиции – това е индикатор за грешка при предаването на данните.

Обръщат се получените цифри (най-младшият разряд става най-старши) 011 -> 110 **и това е номерът на сгрешената позиция в двоична бройна система**.

$110_{(2)} = 6_{(10)}$ - Грешната позиция е с номер 6 и информативното съобщение не е 0101 а 0111.

3.2.5.Интегрални схеми, реализиращи код на Хеминг

Съществуващите интегрални схеми, реализиращи код на Хеминг, могат да се разделят според разрядността си:

- 8 битови - 74636 и 74637;
- 16 битови - 74616, 74617, 74630 и 74631;
- 32 битови - 74632, 74633, 74634 и 74635.

3.2.6.Приложение на кода на Хеминг

Кодът на Хеминг се използва за контрол и корекция на грешки в запомнящия масив на ОП на компютрите**. При запис** в масива информацията се кодира, т.е. към нея се добавя **k**-разрядна контролна дума, която се записва заедно с основната. **При четене** на n-разрядната дума от паметта се извършва декодиране, при което се коригират единичните грешки, а двойните се откриват.