Improving error correction in data information protected by linear correction codes

I. Kolev, B. Aleksandrov

In this paper an algorithm is presented on data linear coding, for multiple error correction. The memory or data information is divided into fragments, where each fragment includes “a” words. The proposed technique is based on combination error detection and correction with a data linear code for each word and control words for every fragment. The presented algorithm can be used for error correction in areas which operate in high level error sources.

1. Въведение.

Навлизането на персоналната изчислителна техника в нашето ежедневие, налага решаването на редица проблеми една част от които са свързани с обезпечаването на качествен обмен и съхранение на информацията. Процесите на съхранение или обмен на информация в една или друга компютърна система са съпроводени със смущаващи въздействия които околната среда внася спрямо свързващата или заемящата среда. В резултат на тези смущаващи въздействия обменните информационни масиви се модифицират и в приемните страни се получават данни различни от предадените.

Отстраняването на тези смущаващи въздействия най-често се извършва по някой от подходите, един от който се състои в използването на по-шумоустойчиви свързващи и заемящи среди. Друг възможен подход се явява окомплектовване на масивите от данни с определен излишък от информация (контролни разряди), на базата на които да може да се установи дали данните са правилно приети (прочетени) или в тях са настъпили някакви модификации.

В зависимост от начина по който информацията се съхранява, обменя, и какви са смущаващите въздействия, се ползват различни кодове за откриване и отстраняване на грешки. Едни от кодовете намерили широко приложение в компютърната техника се явяват линейните коригиращи кодове. Вследствие техните особенности,
паралелност на процесите кодиране и синдромно декодиране, тези кодове са намерили приложение за откриване и отстраняване на грешки в оперативни памети, където информацията се съхранява паралелно най-често като 8,16,32… битови думи [2; 3].

В настоящия материал е описан подход при които чрез разделяне паметта на фрагменти и формиране на контролни думи за всеки отделен фрагмент, е възможно да се откриват и отстраняват грешки с кратност по-голяма от кратността на грешките, които могат да се откриват и отстраняват посредством ползване линеен код.

2. Защитно кодиране на информационни масиви с линейни коригиращи кодове и допълнителни контролни думи.

При линейните коригиращи кодове всяка една от думите подлежащи на съхранение се окомплектова с определен излишък от информация - г броя kontrolni разреди, с помощта на които при четене се установява наличието (отсъствието) на грешки в съответната дума. Увеличаване дължината на линейните кодови вектори води до повишаване вероятността от поява на двойни, тройни и по-вече разредни грешки, откриването и отстраняването, на които е свързано със значително увеличаване броя на контролни разреди във всеки кодов вектор. Тези контролнi разредi се съхраняват паралелно към отделните думи, с което се увеличавa общия обем памет, необходимa за съхранение на информационния масив. Този начин на защитa на информация най-общо е показан на фиг. 1а.

Фиг 1.
Същността на дефинирания подход при кодиране се състои в разделяне паметта на фрагменти и формиране по една kontrolна дума \( X_k \) за всеки отделен фрагмент (фиг. 16), където: a- брой думи във фрагмента

\[
X_k = \oplus \sum_{j=1}^{d} X_j
\]

Всяка една от думите във фрагмента представлява кодов вектор с които могат да се откриват и отстраняват грешки с предварително избрана кратност, ползвана при определяне броя на kontrolните разряди [3]. При възникване и откриване в една дума \( X_e \) от фрагмента на грешки с кратност по-голяма от избранията, тази грешка се отстранява с помощта на kontrolната дума за съответния фрагмент. Битовете на \( X_k \) се получават след сумиране на битовете с еднакви номера за всички думи във фрагмента. В пример 1А е представен фрагмент от пет кодови вектора. Всеки един от тях е кодов вектор на \((13,8,4)\) линеен код с посочените по-долу проверочна и пораждаша матрица.

\[
G_{13,8} = \begin{bmatrix}
10000000 & 000111 & 00000000 & 000011 & 00000001 & 00010000 & 00000100 & 00000001 & 11100000 & 10101000 & 01001101 & 01001100 & 01001011 & 01001001 & 01001000
\end{bmatrix}
\]

\[
H_{5,13} = \begin{bmatrix}
100011110 & 10000111 & 00010110 & 01001011 & 00101101 & 01001100 & 11100000 & 00000001
\end{bmatrix}
\]

Процеса декодиране се извършва в следната последователност от действия. Синдромно се декодира kontrolната за фрагмента дума. В нея както и във всички останали думи е

Пример 1

A)

\[
\begin{array}{c}
x_1 \quad 0111 \quad 1110 \quad 01101
\end{array}
\]

\[
\begin{array}{c}
x_2 \quad 1100 \quad 0011 \quad 11110
\end{array}
\]

\[
\begin{array}{c}
x_3 \quad 0000 \quad 0011 \quad 11000
\end{array}
\]

\[
\begin{array}{c}
x_4 \quad 1111 \quad 0101 \quad 01010
\end{array}
\]

\[
\begin{array}{c}
x_5 \quad 1011 \quad 0111 \quad 11001
\end{array}
\]

\[
\begin{array}{c}
x_6 \quad 1111 \quad 1100 \quad 11000
\end{array}
\]

Б)

\[
\begin{array}{c}
x_k \quad 1111 \quad 1100 \quad 11000 \quad 0 \quad 0 \quad 0 \quad 0
\end{array}
\]

\[
\begin{array}{c}
x_{k0} \quad 0000 \quad 0000 \quad 0000
\end{array}
\]

\[
\begin{array}{c}
x_{k1} \quad 0111 \quad 1110 \quad 01101 \quad 0 \quad 0 \quad 0 \quad 0
\end{array}
\]

\[
\begin{array}{c}
x_{k2} \quad 1100 \quad 0011 \quad 11110 \quad 0 \quad 0 \quad 0 \quad 0
\end{array}
\]

\[
\begin{array}{c}
x_{k3} \quad 1011 \quad 1101 \quad 10011
\end{array}
\]

\[
\begin{array}{c}
x_{k4} \quad 0001 \quad 0011 \quad 11000 \quad 1 \quad 0 \quad 0 \quad 1
\end{array}
\]

\[
\begin{array}{c}
x_{k5} \quad 0000 \quad 0111 \quad 11000 \quad 0 \quad 0 \quad 0 \quad 0
\end{array}
\]

\[
\begin{array}{c}
x_{k6} \quad 0101 \quad 01010 \quad 11110
\end{array}
\]

\[
\begin{array}{c}
x_{k7} \quad 1111 \quad 1100 \quad 11000
\end{array}
\]

\[
\begin{array}{c}
x_{k8} \quad 0100 \quad 0010 \quad 10011
\end{array}
\]

\[
\begin{array}{c}
x_{k9} \quad 1011 \quad 0111 \quad 11001 \quad 0 \quad 0 \quad 0 \quad 0
\end{array}
\]

\[
\begin{array}{c}
x_{k10} \quad 1111 \quad 0101 \quad 01010 \quad 0 \quad 0 \quad 0 \quad 0
\end{array}
\]

95
възможно възникване на грешки с кратност по-голяма от предварително избраната. За изясняване на подхода нека се допусне че в $X_k$ не са настъпили такива грешки.

Декодира се първата дума от фрагмента за грешки които могат да се открият и отстранят с помоща на ползвания конкретен линеен код (пример 1б). Ако декодирането е успешно се формира първа междуна контролна дума $X'_{k1}$ (2)

$$X'_{k1} = X'_{k0} + X_1 ; \quad X_{k0} = 0$$

(2)

Декодира се следващата дума от фрагмента за грешки които могат да се открият и отстранят с конкретния линеен код. Ако при декодиране се установи, че липсват грешки, или ако има такива и те са отстранени се формира следващата междуна контролна дума.

$$X'_{k2} = X'_{k1} + X_2 , \quad X'_{k3} = X'_{k2} + X_3$$

(3)

Ако на определен етап при декодиране на думите от фрагмента се открие дума $X_e$ с възникнали в нея грешки които не могат да се отстранят с линейния код, то тази дума не може да участва във формирането на междуните контролни думи. По нататъшното формиране на междуните контролни думи се извършва като вместо $X_e$ участва $X_k$. Процеса продължава до декодиране на последната дума($X_a$) от фрагмента. Последно формираната междуна kontrolна дума ще съвпада с оригинала на думата в която е възникнала многоразредната грешка.

$$X_e = X_k - \sum_{j=1}^{n} X_j$$

(4)

В пример 1Б е посочено декодиране на масив съгласно описания подход. Единичната грешка в $X_3$ е отстранена след синдромно декодиране. Многоразрядната грешка в $X_4$ е отстранена посредством контролната за фрагмента дума. По този начин могат да се отстраняват грешки, които с ползвания линеен код не могат да бъдат отстранени.

3. Заключение.

Описаният подход предполага отстраняване на грешки с кратност за която линейните коригиращи кодове изискват значително по-голям брой kontrolни разряди. Особеност на подхода се явява това че в рамките на един фрагмент може да се въстановява след многократна грешка най-много една дума. За целта е необходимо да се подбере подходящ брой думи във фрагмента. Познавайки вероятностите на появяване на двойни,
тройни и по-вече кратни грешки може да се вземе решение колко думи ще участват в отделните фрагменти. Подобен подход на кодиране най-общо може да се опише със следните параметри \((a,n,k,d)\) където: \(a\) - брой думи във фрагмента; параметрите \(n,k,d\) са същите както при линейните кодове.

В табл.1 са посочени количествата памет необходима за защита на оперативната памет, разглеждана като съвкупност от \(X.10^5\) думи. Всяка една дума съдържа 16 информационни бита и определен брой контролни разряди за откриване и отстраняване на единични, двойни и тройни грешки. Посочени са необходимите количества памет \((M_{r1})\) за съхранение на контролните разряди при ползване на \((22,16,4), (26,16,5)\) и \((29,16,7)\) линейни кодове.

\[
M_{r1} = \frac{X.10^5}{8}(n-k) [MB]
\]  
(5)

В същата таблица а посочени и необходимите количества памет \((M_{r2})\) за съхранение на контролните разряди при ползване на \((a,22,16,4)\) кодове.

\[
M_{r2} = \frac{X.10^5}{8}(r+n) [MB]
\]  
(6)

Вместо използване на кодове с разстояние 5,6,7,8,... позволяващи откриване и отстраняване на двойни, тройни и т.н. грешки, съгласно описания подход е целесъобразно ползване на \((a,n,k,4)\) кодове. Имайки в предвид особеностите на \((a,n,k,d)\) кодовете да отстраняват многократни грешки само в една дума от фрагмента и познавайки вероятностите на появяване на подобни грешки, може да се подбере подходяща стойност за „а”.

На фиг.2 са посочени в графичен вид зависимостите между броя думи \((X.10^5)\) и необходимите количества памет за съхранение на контролните разряди при \((n,k,d)\) и \((a,n,k,d)\) кодове. На фиг. 2а тези зависимости са за кодове защитаващи 16 разрядни думи срещу единични грешки \((22,16,4)\) срещу двойни грешки \((26,16,5)\) и срещу тройни грешки \((29,16,7)\). На същата фигура са посочени аналогични зависимости за \((a,n,k,d)\) кодове, базирани на \((22,16,4)\) линеен код и голяма на фрагмента 16,32,64,128 думи. От посочената таблица и графика се вижда, че \((a,22,16,4)\) кодове с един по малък информационен излишък спрямо \((26,16,5)\) кодове, позволяват откриване и отстраняване не само на двойни но и на многоразредни грешки в една дума от фрагмента.
При увеличаване на d с цел отстраняване на тройни и по-вече кратни грешки, се увеличават и разликите между $M_{r1}$ и $M_{r2}$ като с

![Diagram](image1)

**Фиг. 2**

включването на по-голям брой думи в паметта, тези разлики се увеличават значително. Така например една една 16MB памет за да бъде защитена срещу тройни грешки, са необходими $M_{r1}$=13MB допълнителна памет, а при защита с използване на предложения подход са необходими $M_{r2}$=7,357MB, с които може да се отстранява по една многоразрядна грешка във всеки фрагмент. Познавайки вероятностите на появяване на двойни и многоразрядни грешки, може да се намери "а" за която $M_{r2}$>$M_{r1}$. От посочените графически зависимости се вижда че намаляване големината на фрагмента не влияе същественно върху разликите между $M_{r2}$, докато различията между $M_{r1}$ стават значително по-големи с използване на кодове позволяващи откриване на двойни, тройни и по-вече разрядни грешки.
Предложения подход на кодиране може да намери приложение за защита на оперативни памети работещи в условията на смущаващи въздействия, които водят до едновременно модифициране на няколко бита в рамките на един кодов вектор. Познавайки модела на смущаващите въздействия, може да се определи и големината на фрагментите, с която да се осигури и определена защитеност на информацията съхранявана в съответната памет.

Предложения подход може да бъде мултиплициран към циклическите кодове, и да намери приложение при обмен на информация между ресурси в компютърни мрежи. Подхода с успех може да се ползва и при информационен обмен в промишленни компютърни среди, работещи в условията на различни смущаващи въздействия, с висока интензивност на появяване.

Използвана литература:
2. Мак-Вильмс Ф. Дж., Слоен А.Х.- Теория и практика кодов изправляющиъх ошибки. Москва 1979г.