

**KODIRANJE
IN DEKODIRANJE
KOREKTURNEGA KODA
Z MIKRO RAČUNALNIKOM**

**M. ŠUBELJ
J. KORENINI
F. NOVAK
R. TROBEC**

UDK: 681.327.63 : 621.3

INSTITUT JOŽEF STEFAN, LJUBLJANA

V članku je obravnavano kodiranje in dekodiranje korekturnega koda realizirano z mikro računalnikom. Najprej je definirana Hammingova razdalja in princip dekodiranja "idealni opazovalec". Tipi korekturnih kodov so razdeljeni na kode, ki e napak popravijo in kode, ki (e-1) napak popravijo in e-to napako detektirajo.

Primer prikazuje princip oddaje kodiranih znakov in sprejem in dekodiranje znakov. V zbirnem jeziku za procesor F-8 sta dodana tudi programska modula za kodiranje in dekodiranje.

CODING AND DECODING OF A CORRECTION CODE BY A MICROCOMPUTER - Coding and decoding of a correction code implemented in microcomputer is discussed in the paper. The definition of "Hamming distance" is given and the "ideal observer" decoding technique is described.

Different correction codes are classified as e-error-correcting codes and (e-1)-error-correcting, e-error-detecting codes.

An example of transmission and reception of correction codes in practice is shown. Program modules for coding and decoding written in F8 assembly language are also given.

1. UVOD

Pri prenosu informacije preko telefonskega kabla se pojavi problem zanesljivega prenosa. Motnje, ki jih lahko povzročijo preklopi na liniji, distorzija signala, atmosfarske motnje, presluh med linijama, lahko povzročijo, da bit informacije spremeni svojo električno vrednost.

Za zanesljiv prenos poznamo več metod kontrole napake pri prenosu. Mi se bomo omejili na kontrolo napake, ki omogoča popravljanje napake, v nekaterih primerih pa samo odkrivanje. Izvedba je programska in nam omogoča veliko prilagodljivost in relativno enostavno modifikacijo na drugačen kod. Znaki se lahko prenašajo paralelno (vsi biti znaka istočasno) ali serijsko. Sinhronizacija in paralelno serijska in serijsko paralelna pretvorba je lahko izvedena aparaturno (ACIA) ali programsko.

2. KOREKTURNI KOD IN RAZDALJE

Predpostavlja se, da se informacija prenaša preko simetričnega binarnega kanala. Vhod v kanal tvori zaporedje binarnih kodnih besed $W = \{w_1, w_2, \dots, w_s\}$ dolžine n. Vse besede imajo isto verjetnost, da pridejo pri oddaji na vrsto. Na izhodu iz kanala imamo sprejemnik, ki se odloči po pravilu "idealnega opazovalca".

Dekoder "idealni opazovalec" se odloča za tisto varianto, kjer je verjetnost napake najmanjša ali drugače povedano, za tisto besedo koda $W = \{w_1, \dots, w_s\}$, do katere je Hammingova razdalja najmanjša. Hammingova razdalja $d(w_1, w_2)$ med dvema binarnima besedama w_1 in w_2 dolžine n je število binarnih znakov, v katerih se razlikujeta besedi w_1 in w_2 .

Predpostavimo, da je zaradi motenj možna napaka na vsakem binarnem znaku. Ker je vhodna beseda dolžine n, lahko na izhodu sprejemno 2^n različnih besed dolžine n.

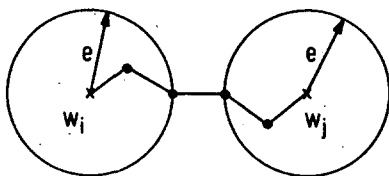
Predpostavimo, naj vir pošilja zaporedje besed

$W = \{w_1, w_2, \dots, w_s\}$ in sprejemnik sprejema zaporedje besed $V = \{v_1, v_2, \dots, v_{2^n}\}$.

Zanima nas, kakšna je povezava med odposlani mi in sprejetimi besedami. Oglejmo si Hammingovo razdaljo med besedami koda $W = \{w_1, w_2, \dots, w_s\}$:

$$a) d(w_i, w_j) = 2e + 1 \quad \begin{array}{l} i = 1, 2, \dots, s \\ j = 1, 2, \dots, s \\ i \neq j \end{array}$$

Predočimo si še grafično razdaljo med besedama w_i in w_j .

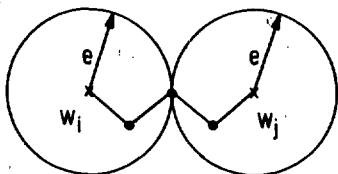


$$d(w_i, w_j) = 5 \Rightarrow e = 2$$

Vidimo, da vsaka sprejeta beseda $v_j \in V$ $j=1, 2, \dots, 2^n$ pade v okolje samo ene izmed besed $W = \{w_1, w_2, \dots, w_s\}$ s Hammingovo razdaljo $d(w_i, v_j) \leq e$ $i = 1, 2, \dots, s$ in $j = 1, 2, \dots, 2^n$. Naloga dekoderja "idealni opazovalec" je, da najde to besedo $w_i \in W$, $i = 1, 2, \dots, s$. Beseda dekodirana po tem principu je bila pravilno dekodirana v odposlano besedo, če je bila napaka e ali manj kratna.

$$b) d(w_i, w_j) = 2e \quad \begin{array}{l} i = 1, 2, \dots, s \\ j = 1, 2, \dots, s \\ i \neq j \end{array}$$

Predočimo si grafično razdaljo med besedama.



$$d(w_i, w_j) = 4 \Rightarrow e = 2$$

Oglejmo si Hammingovo razdaljo med sprejeto besedo $v_j \in V$ $j = 1, 2, \dots, 2^n$ in besedami koda $W = \{w_1, w_2, \dots, w_s\}$.

Razdalja je:

$$\begin{array}{l} - d(w_i, v_j) = e \quad \text{ali} \quad \begin{array}{l} i = 1, 2, \dots, s \\ j = 1, 2, \dots, 2^n \end{array} \\ - d(w_i, v_j) = d(w_k, v_j) = e \quad \begin{array}{l} i = 1, 2, \dots, s \\ j = 1, 2, \dots, 2^n \\ k = 1, 2, \dots, s \end{array} \end{array}$$

V prvem primeru se lahko odločimo. Sprejeto besedo v_j , dekodiramo kot w_i . Beseda je pravilno dekodirana, če je bila napaka $e=1$ ali manj kratna.

V drugem primeru besede ni možno dekodirati, saj je razdalja besede v_j enaka do dveh besed w_i in w_k . Napako se lahko samo detektira.

3. PRINCIP ODDAJE IN SPREJEMA KOREKTURNEGA KODA

Zgled obravnava primer prenosa informacije s kodom 4 korekturnih in 4 informacijskih bitov. Hammingova razdalja med besedami koda je 4, zato tak kod popravlja enojne in detektira dvojne napake na 8 bitih. Za 8 bitno besedo smo se odločili zaradi najpogostejše aparaturne organizacije mikroročunalniških komponent (CPU, ACIA). Seveda je možno izbrati tudi kod, ki bo imel drugačno razmerje med informacijskimi in korekturnimi biti pri enaki dolžini besede.

V primeru koda 4 informacijskih in 4 korekturnih bitov, oddajnik odda zlog (8 bitov) informacije kot dve 8 bitni kodi. Najprej kodira zgornje 4 bite informacije s Hammingovim kodom (8 bitov) in ga odda. Za tem kodira še spodnje 4 bite informacije in kod odda.

Sprejemnik dela v obratnem vrstnem redu. Sprejme kod (8 bitov), ga dekodira (4 biti) in shrani na zgornja 4 mesta vmesnika informacije. Za tem sprejme še drugi kod, ga dekodira in prida na spodnja štiri mesta vmesnika informacije. Modula za kodiranje in dekodiranje se sklicujeta na isto tabelo Hammingovih kodov. Osnovni moduli za sprejem in oddajo so napisani kot subrutine. Ogljemo si jih podrobneje opisane v psevdokodu:

a) ODDAJA:

TRBY: subrutina odda zlog (8 bitov) informacije kodirane s kodom Hammingove razdalje 4.

Vhodni parameter : adresa lokacije, iz katere bo zlog oddan.

SUBROUTINE TRBY

VLOŽITEV ZLOGA;
 POMIK ZA 4 MESTA V. DESNO;
CALL KOHE
CALL TRC
 VLOŽITEV ZLOGA;
 OHRANITEV SP. 4 BITOV;
CALL KOHE
CALL TRC

ENDSUBROUTINE

KOHE : subrutina 4 bitom priredi kod Hammingove razdalje 4 (8 bitov).
 Vhodni parameter: 4 informacijski biti v reg. DIO.
 Izhodni parameter: kod Hammingove razdalje (8 bitov) v reg. DIO.

SUBROUTINE KOHE

.ADRESIRANJE ZAČETNE ADRESE TABELE
 HAMMINGOVIH KODOV;
 RELATIVNI PREMİK ADRESE ZA 4 INF. BITE
 (IZRAČUN ADRESE KODA);
 VLOŽITEV IN SHRANITEV KODA;

ENDSUBROUTINE

TRC : subrutina odda znak na izhod za oddajo.
 Vhodni register: znak (Hammingov kod) v reg. DIO.

SUBROUTINE TRC

DOUNTIL (MOŽNOST ODDAJE ZNAKA)
ENDDO
 ODDAJA ZNAKA NA IZHOD;

ENDSUBROUTINE

b) SPREJEM :

REBY : sprejme zlog (8 bitov) informacije kodirane s kodom Hammingove razdalje 4.
 Vhodni parameter je adresa lokacije, kamor bo sprejeti zlog shranjen.

SUBROUTINE REBY

CALL REC
CALL DEHE
 SHRANITEV ZG. 4 BITOV ZLOGA;
CALL REC
CALL DEHE
 SHRANITEV SP. 4 BITOV ZLOGA;

ENDSUBROUTINE

DEHE : subrutina kod Hammingove razdalje 4 (8 bitov) dekodira v 4 bite. V primeru, ko se ne more odločiti ($d(w_i, v_j) = d(w_k, v_j)$) postavi zastavico "programska detekcija napake".
 Vhodni parameter: kod Hammingove razdalje v reg. DIO.
 Izhodni parameter: kod dekodiran v 4 informacijske bite v reg. DIO ali DIO nespremenjen in zastavica "programska detekcija napake" v reg. DRS.

SUBROUTINE DEHE

ŠTEVEC KODOV=0;
 ADRESIRANJE ZAČETKA TABELE HAMMINGOVIH KODOV;
DOUNTIL (TEST VSEH KODOV ALI DEKODIRANJE)
ŠTEVEC BITOV=8
 EXCLUSIVNI OR MED SPREJETIM ZNAKOM IN
 ADRESNIM ZNAKOM TABELE KODOV
 (ADRESIRANJE LOKACIJE NASLEDNJEGA KODA)
IF (STA ENAKA, REZULTAT=0)
THEN
 DEKODIRANI SPREJETI ZNAK=ŠTEVEC KODOV,
 IZTOP;
ELSE
DOUNTIL (POJAVITEV 1. ENICE V REZULTATU)
 PRIPRAVA NASLEDNJEGA MESTA;
ŠTEVEC BITOV=ŠTEVEC BITOV-1;
ENDDO
DOUNTIL (TEST OSTALIH MEST REZULTATA)
ENDDO
IF (NI VEČ ENIC)
THEN
 SPREJETI ZNAK=ŠTEVEC KODOV,
 IZTOP;
ELSE
ŠTEVEC KODOV=ŠTEVEC KODOV+1;
ENDIF
ENDIF
ENDDO
 POSTAVITEV ZASTAVICE PROG.DET.NAPAKE;
ENDSUBROUTINE

REC : subrutina sprejme znak preko vhoda za sprejem
 Izhodni parameter: sprejeti znak (Hammingov kod) v reg. DIO.

SUBROUTINE RECDOUNTIL (MOŽNOST SPREJEMA ZNAKA)ENDDO

SPREJEM ZNAKA PREKO VHODA;

ENDSUBROUTINE

5. LITERATURA

- /1/ L. Gyergyek: Statistične metode v teoriji sistemov, teorija o informacijah, Fakulteta za elektrotehniko 1971.
- /2/ Electronics Book Series: Basics of data communications, Mc Graw-Hill 1976.
- /3/ Mostek: Programming Guide

4. SKLEP

Cilj članka je predstaviti možnost prenosa podatkov kodiranih s konkretnim kodom pri komunikaciji med mikro računalniki. Ta dograditev, sicer zmanjša množino koristne informacije prenešene v časovnem intervalu, vendar poveča zanesljivost prenosa in izboljša kvaliteto kanala.

```

*          1.2.1979 M.SUBELJ
*
*          TITLE SUBROUTINA KODIRANJE HAMMINGA
*
*
*SUB.KOHE DOLOCI 4 INFORMACIJSKIM BITOM KOD
*HAMMINGOVE RAZDALJE 4.KOD SESTAVLJAJO 4
*INFORMACIJSKI BITI + 4 KOREKTURNI BITI.
*
*
*VHOD: DIO = 4 INFORMACIJSKI BITI (SP.4 MESTA)
*
*IZHOD: DIO = KOD HAMMINGOVE RAZDALJE
*
*UPOR.REG.:ACC,W,K,DIO
*
DIO    EQU    H'07'          PODATKI V I/O SUB .
*
*
*
0000 08    KOHE    LR    K,P
*
0001 2A 00 34    DCI    TAHEM          VLOZITEV ADRESE TABELA KODOV
0004 47          LR    A,DIO
0005 8E          ADC          IN ADRESIRANJE KODA
*
0006 16          LM          VLOZITEV IN
0007 57          LR    DIO,A          SHRANITEV KODA
*
0008 0C          PK
*
*          TITLE SUBROUTINA DEKODIRANJE HAMMINGA
*
*
*SUB. DEHE DEKODIRA KOD HAMMINGOVE RAZDALJE 3
* ( 7 BITNA BESEDA) ALI RAZDALJE 4 ( 8 BITNA BESEDA).
*PRI KODU RAZDALJE 3 POPRAVI ENOJNO NAPAKO.
*PRI KODU RAZDALJE 4 POPRAVI ENOJNO NAPAKO IN
*DETEKTIRA DVOJNO NAPAKO.
*
*
*
*VHOD: DIO= KOD HAMMINGOVE RAZDALJE
*
*
*NAPAKA 0 REDA ALI 1. REDA
*
*IZHOD: DIO= DEKODIRANA BESEDA (SPODNJI 4.BITI)
*
*NAPAKA VEC KOT 1.REDA
*
*IZHOD: DIO= NESPREMENJEN
*          ERFL= B'XXX1XXXX' FLAG NAPAK
*
*UPOR.REG.:ACC,W,K,XU,XL,DRS,DIO

```

```

*
*
*
XU      EQU      H'00'      DELOVNI REGISTER
XL      EQU      H'01'      DELOVNI REGISTER
DRS     EQU      H'05'      PODATKI V SUB. LRB IN SRS
DIO     EQU      H'07'      PODATKI V I/O SUB .
*
*
0009 08      DEHE  LR      K,P
*
000A 70      LIS      H'00'
000B 50      LR       XU,A      VLOZITEV STEVCA KODOV XU=0
*
000C 2A 00 34  DCI      TAKEH      ADRESIRANJE ZACETKA TABELA KODOV
*
000F 78      DEHE0  LIS      H'08'
0010 51      LR       XL,A      DOUNTIL (TESTIRANJE VSEH KODOV)
                                VLOZITEV STEVCA BITOV XL=8
*
0011 47      LR       A,DIO     VLOZITEV SPREJETEGA ZNAKA IN
0012 8C      XH       PRIMERJANJE Z ZNAKOM TABELA
0013 84 11    BZ       DEHE3     IF ( STA ENAKA) THEN3 DEHE3,ELSE...
*
0015 21 FF    DEHE1  NI      H'FF'
0017 91 09    BH      DEHE6     DOUNTIL (POJAVITEV 1.ENICA)
                                IF (ENICA) THEN DEHE6,ELSE...
*
0019 13      SL       1
001A 31      DS      XL
001B 90 F9    BR      DEHE1     PRIPRAVA NASLEDNJEGA BITA
                                STEVEC BITOV XL=XL-1
                                ENDDO
*
001D 21 FF    DEHE2  NI      H'FF'
001F 91 09    BH      DEHE4     TESTIRANJE ALI JE 2.ENICA
                                IF (ENICA) THEN DEHE4,ELSE...
*
0021 13      DEHE6  SL       1
0022 31      DS      XL
0023 94 F9    BNZ     DEHE2     (DEHE6 PRIPRAVA NOVEGA NESTA)
                                IF (TEST VSI BITI) THEN...,ELSE DEHE2
*
0025 40      DEHE3  LR      A,XU
0026 57      LR      DIO,A     VLOZITEV DEKODIRANE BESEDE
0027 90 0B    BR      DEHE5     IZHOD IZ SUB.
*
0029 40      DEHE4  LR      A,XU
002A 1F      INC
002B 50      LR      XU,A
002C 23 10    XI      H'10'
002E 94 E0    BNZ     DEHE0     POVECANJE STEVCA KODOV IN TEST
                                ALI SO TESTIRANI VSI KODI
                                ENDDO
*
0030 20 10    LI      H'10'
0032 55      LR      DRS,A     POSTAVITEV FLAG-A PROGRAMSKE
                                DETEKCIJE NAPAKE PRI SPREJEMU
*
0033 0C      DEHE5  PK
*
*
*
*
TABELA KODOV (8 BITNA KODA)
*
0034 00      TAKEH  DC      H'00'      W0
0035 02      DC      H'D2'      W1
0036 55      DC      H'55'      W2
0037 87      DC      H'87'      W3
0038 99      DC      H'99'      W4
0039 4B      DC      H'4B'      W5
003A CC      DC      H'CC'      W6
003B 1E      DC      H'1E'      W7
003C E1      DC      H'E1'      W8
003D 33      DC      H'33'      W9
003E 84      DC      H'B4'      W10
003F 66      DC      H'66'      W11
0040 78      DC      H'78'      W12
0041 AA      DC      H'AA'      W13
0042 2D      DC      H'2D'      W14
0043 FF      DC      H'FF'      W15
*
*
*
END
//NUMBER OF ERRORS= 0

```