

Efektivní kódy a algoritmy jejich zabezpečení

Martin Vysloužil

Bakalářská práce
2005



Univerzita Tomáše Bati ve Zlíně
Fakulta technologická

ABSTRAKT

Abstrakt česky

Tato bakalářská práce poskytuje stručný přehled problematiky týkající se kódování, zabezpečování kódů a opravou chyb vzniklých při přenosu. Obsahuje historii vzniku kódů, obecné informace o teorii kódování, stručné dělení kódů na rovnoměrné, nerovnoměrné, efektivní a bezpečnostní. Obsahuje problémy, které vznikají při zabezpečení kódu, způsoby zabezpečení a využití kódů, geometrický model kódu, vzdálenost kódových složek, detekci i korekci chyb, matici kódu, kontrolní matici kódu, ochranu zprávy před chybami, opravu jedné chyby Hammingových kódů i opravu více chyb u Reedových-Mullerových kódů.

Abstrakt ve světovém jazyce

This Bachelor thesis offers a brief survey of problems concerning with coding, suppling code and repairing errors on broadcast. It contains history origin code, general information about theory coding, brifing division code on the even, not even, effective and safety. It contains problems about security code, mode security and use code, geometric model code, distance components code, detection and correction errors, matrix code, control matrix code, protection information before errors, repairing one error Hamming code and repairing more errors by R-M code.

Chtěl bych tímto poděkovat vedoucímu mé bakalářské práce, panu RNDr.Ing.Miloši Krčmářovi, za odborné vedení, rady, připomínky a poznatky, které mi poskytoval v průběhu řešení mé bakalářské práce a za jeho vstřícný přístup. Také bych chtěl poděkovat svým rodičům za jejich podporu při mém studiu a při tvorbě této práce.

Prohlašuji, že jsem na celé bakalářské práci pracoval samostatně a použitou literaturu jsem citoval.

Ve Zlíně, 07. 06. 2005

.....

podpis

ÚVOD.....	7
1 HISTORIE TEORIE KOMUNIKACE.....	9
2 OBECNÉ INFORMACE O TEORII KÓDOVÁNÍ.....	11
3 DĚLENÍ KÓDŮ.....	13
3.1 ROVNOMĚRNÉ KÓDY	13
3.1.1 Telegrafní kód	13
3.1.2 ASCII kód	13
3.2 NEROVNOMĚRNÉ KÓDY	13
3.2.1 Morseova abeceda.....	14
3.3 EFEKTIVNÍ KÓDY	15
3.3.1 Shannonova -Fanova metoda	15
3.3.2 Huffmanova metoda.....	16
3.4 BEZPEČNOSTNÍ KÓDY	16
3.4.1 Zabezpečující schopnosti kódu	17
3.4.2 Lineární kódy	18
3.4.3 Hammingovy kódy.....	19
3.4.4 Cyklické kódy	20
3.4.5 Nelineární kódy - inverzní kódy	21
4 PROBLÉMY ZABEZPEČENÍ KÓDŮ	23
4.1 ZPŮSOBY ZABEZPEČENÍ A VYUŽITÍ KÓDŮ	23
4.2 GEOMETRICKÝ MODEL KÓDU	24
4.3 VZDÁLENOST KÓDOVÝCH SLOŽEK.....	26
4.4 HAMMINGOVA VÁHA KÓDOVÉ SLOŽKY	29
4.5 SČÍTÁNÍ (MOD 2)	29
4.6 DISTRIBUCE CHYBOVÝCH MÍST V KÓDOVÉ SLOŽCE	30
5 CHYBY.....	32
5.1 OCHRANA ZPRÁVY PŘED VLIVEM CHYB.....	32
5.2 DETEKCE CHYB	32
5.2.1 Lineární binární blokové kódy	32
5.2.2 Generující matice kódu	33
5.2.3 Kontrolní matice.....	35
5.2.4 Kontrolní matice celkové parity.....	36
5.3 OPRAVA JEDNÉ CHYBY	37
5.3.1 Hammingovy kódy.....	37
5.3.2 Algoritmické vytváření kódových slov	39
5.3.3 Opravování kódových slov	40
5.4 OPRAVA VÍCE CHYB.....	41
5.4.1 Reedovy-Mullerovy kódy	41
5.4.2 Algoritmus dekódování Reedových-Mullerových kódů.....	44
5.4.3 Oprava chyb Reedových-Mullerových kódů	45

ZÁVĚR.....	48
SEZNAM POUŽITÉ LITERATURY	49
SEZNAM OBRÁZKŮ	50
SEZNAM TABULEK.....	51
SEZNAM PŘÍLOH.....	52

ÚVOD

Mezi základní vlastnosti nejrozličnějších typů informačních systémů patří schopnost efektivně a kvalitně komunikovat. Pod pojmem komunikace se rozumí předávání zpráv mezi informačními systémy, nacházejícími se na různých místech. Za určitý způsob komunikace, u kterého se jedná o předávání zpráv ne v prostoru, ale v čase, je možné považovat ukládání zpráv do paměti a jejich pozdější čtení. Srovnání komunikace v prostoru a v čase není založeno jenom na formální podobnosti. U obou způsobů je podstatné i to, že zprávy procházejí fyzikálním prostředím, pro které se používá název *medium*. Medium je zdrojem rušení. Rušení způsobuje vznik chyb a ty snižují kvalitu předávaných zpráv.

Abecedou informace jsou obecně „nuly“ a „jedničky“ a sdělovacím či přenosovým kanálem paměť počítače nebo komunikační počítačová síť. Naším hlavním zájmem je, aby se zprávy přenášely co nejúčinněji a aby měly co nejméně chyb. Obě veličiny: efektivita přenosu zpráv i množství chyb ve zprávách jsou dány kapacitou přenosového kanálu. Tato skutečnost, kterou matematicky formuloval C.E.Shannon v roce 1948, říká, že i ve špatném kanálu je možné zakódovat zprávy tak, aby chyby byly při přenosu potlačeny na minimum. Není to návod, jak takový kód zkonstruovat, ale je to důkaz, že takový kód existuje. Teoretické poznatky o „dobrých“ kódech, které byly po desítky let připravovány v pracích matematiků se nyní stávají předmětem zájmu techniků. Rozvoj nových technologií výroby a způsob návrhu číslicových systémů umožňují tyto poznatky využívat.

Nezavádí se pojem signál, tedy fyzikální veličina jako například průběh napětí v čase. Fyzikální popis média používá výkonu šumu a přenos informace signálem je charakterizován poměrem výkonu signálu k výkonu šumu SNR (Signal to Noise Ratio). Hodnotu SNR, tedy i kvalitu komunikace, lze zlepšit změnou fyzikální veličiny: zvětšením výkonu signálu tak, aby byl dostatečně velký ve srovnání s výkonem šumu charakterizujícím medium, to však není vždy možné.

Každá zpráva obsahuje nějaké nadbytečné (redundantní) informace. Má-li komunikace probíhat rychle, musí být nadbytečných informací ve zprávě co nejméně. Pro tento účel se používá kódování, které je někdy označováno jako *kódování zdroje zpráv*. Entropii zdroje zpráv popsal Claude E. Shannon jako průměrné množství informace připadající na jeden symbol zprávy. Zpráva nese maximální hodnotu entropie (má

minimální redundanci), je-li vyjádřena vzájemně nezávislými symboly s pokud možno stejnou pravděpodobností výskytu.

Budou-li ve zprávě některé symboly obsaženy častěji než jiné, je naděje, že se podaří nalézt nějaký nerovnoměrný kód, který umožní zprávu zestručnit. Je-li zestručněná zakódovaná zpráva plně obnovitelná do původní podoby je komprese bezztrátová, pokud není možné zprávu obnovit v plné míře, je komprese ztrátová. Při zpracování zpráv pro účinnou a bezpečnou komunikaci jsou důležité dva kroky: prvním krokem je odstranit ze zprávy redundanci, zvýšit tak efektivitu komunikace a druhým krokem opět do zprávy redundanci zavést. Nově zavedená redundance má za úkol minimalizovat ztráty informace, které mohou při komunikaci nastat.

V reálném kódovém systému probíhá komunikace vždy za přítomnosti rušení. Přijaté zprávy tedy obsahují chyby. Aby se snížila pravděpodobnost chyb ve zprávě, používají se tzv. bezpečnostní kódy. Kódování bezpečnostními kódy se nazývá také *kanálové kódování*. Ve své druhé větě o kódování za přítomnosti šumu Claude Shannon dokazuje, že jestliže diskrétní zdroj zpráv vytváří menší množství informace za dobu trvání symbolu zprávy, než je hodnota veličiny nazývané *kapacita kanálu*, je možné dosáhnout i při přenosu za přítomnosti rušení libovolně nízkého počtu chyb v přenášené zprávě. Předpokladem pro toto zlepšení odolnosti proti chybám je použití vhodného zakódování zprávy před komunikací a její dekodování po skončení komunikace.

Přestože historie kódování je krátká pouhých padesát let, jsou výsledky obdivuhodné. Problematika teorie kódování vycházela vždy z praktických problémů. Cesta k nalezení nejlepšího kódu pro daný problém vede přes zvládnutí zákonitostí moderní algebry, zejména znalost maticového počtu, Booleovských funkcí, řešení mnohočlenů a dalších vlastností. Teoretická odvození vlastností kódů jsou mnohdy obtížná, praktická řešení jsou však často překvapivě obvodově jednoduchá.

1 HISTORIE TEORIE KOMUNIKACE

Všeobecně rozšířený názor o prvenství autorské publikace říká, že autorem je ten, kdo jako první publikoval důležitý poznatek. Ačkoliv je znění tohoto pravidla jednoduché, jeho použití může být komplikované. Právě tato situace nastala při publikování Hammingových kódů. První publikaci, která na Hammingovy kódy upozorňuje, napsal Shannon, který uvádí ve své práci z roku 1948 Hammingův (7,4) – kód jako příklad s odkazem na Hammingovo autorství. Potom co si Shannonovu práci přečetl Golay, začal sám pracovat na kódech pro opravu chyb a dosáhl pozoruhodných výsledků. Kód, který objevil byl nazván Golayův (23,12) – kód. Golay přitom provedl zobecnění Hammingova (7,4)-kódu na všechny Hammingovy kódy. Obě tyto práce byly vydány před první Hammingovou prací o kódech, která vyšla až roku 1950. To byly pozoruhodné začátky, které provázely Shannonovy podnětné publikace z let 1948 a 1949. Tyto Shannonovy práce, ve kterých vyslovil matematickou definici informace a formuloval věty o kompresi zpráv a o kapacitě kanálu. Tyto věty se staly základem oboru nazývaného *teorie komunikace*.

Na začátku padesátých let po vydání Shannonových prací zavládlo velké očekávání při praktickém uplatnění kódování. Tyto praktické aplikace však předběhl pokrok technologický. Hlavní aplikace kódování se očekávaly v telekomunikační technice, ale právě na začátku padesátých let se podařilo dosáhnout významných zlepšení fyzikálních vlastností v té době používaných komunikačních kanálů. Díky zlepšení technologie a díky použití nových materiálů došlo v té době ke zlevnění dálkových komunikačních linek a současně ke zkvalitnění jejich parametrů.

Technické aplikace kódování se však překvapivě uplatnily v nové oblasti: použití hromadných pamětí počítačů. Zde je kódové zabezpečení používáno k překlenutí nedokonalosti materiálu, který tvoří paměťové médium. Vzhledem ke stále se zvyšující hustotě magnetického i optického záznamu bude tato oblast ještě dlouhou dobu odběratelem aplikací kódování zdroje i kanálového kódování. Pro další zlevnění komunikací je využívána komprese zpráv pomocí kompresních kódů. Avšak zpráva, která obsahuje málo redundance je vlivem rušení náchylná ke ztrátě informace. Proto je nezbytné i kódové zabezpečení.

Optická media jsou všeobecně náchylnější ke vzniku shlukových chyb. Detekce nebo korekce shlukových chyb vhodným kanálovým kódováním je náročnější než

zabezpečení zpráv před ojedinělými chybami. Při konstrukci bezpečnostních kódů je v takovém případě nutné vycházet z abecedy zdroje který používá symbolů vyjadřujících více než dva stavy.

Nových vlastností bezpečnostních kódů je možné dosáhnout jejich zřetězením. Zlepší se tím vlastnosti bezpečnostního kódu. Základní rozdělení zřetězených kódů je reprezentováno dvěma odlišnými způsoby uspořádání: sériovým s vnitřním a vnějším kódem a paralelním s prokládáním, které se dále dělí podle způsobu prokládání na sériové a paralelní. Zřetězené kódy jsou v současné době předmětem intenzivního výzkumu, protože výsledné řešení je jednodušší, než by muselo být, kdyby se jednalo o použití pouze jednoho kódu s odpovídajícími vlastnostmi.

2 OBECNÉ INFORMACE O TEORII KÓDOVÁNÍ

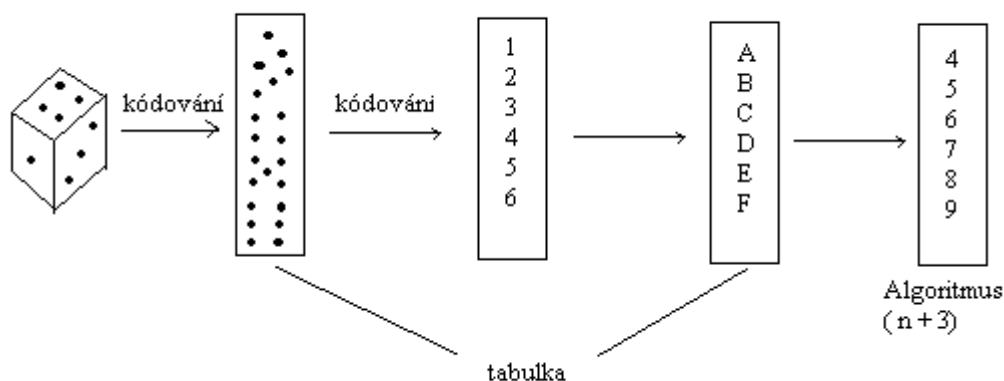
Teorie kódování se vyvíjela paralelně s teorií informace. Zpočátku se v ní využíval poměrně jednoduchý matematický aparát. Později, když začala využívat různá odvětví matematiky, se stala přitažlivou disciplínou pro matematické teoretiky. Dvojková soustava, s níž počítače pracují, je zřejmě nevýhodná pro lidské uvažování – neumíme v ní běžně počítat. Proto k převodu mezi lidskými informacemi a počítači existují převodní pravidla - kódy. Obecně můžeme říct, že kód je množina symbolů, kterými vyjadřujeme různé stavy systému. Kód lze libovolně převádět jeden na druhý, aniž by se změnila velikost a množství informace. Kódování pak můžeme charakterizovat jako převod jednoho kódu na druhý a to buď pomocí tabulky anebo algoritmu.

Pro ukázkou si zvolme jako zkušební systém hrací kostku. Po hodu se může kostka nacházet v jednom ze šesti stavů. Stav kostky se nejčastěji vyjadřuje pomocí množiny tečkových symbolů.



Obr. 1. Množina tečkových symbolů

Když si chceme poznamenat jak nám kostka při hře padla neuděláme to zápisem tečkových symbolů ale arabskými číslicemi (1,2,...,6) odpovídajícími počtu teček. To znamená, že převedeme kódování z jedné množiny symbolů do druhé. Vztah mezi symbolem a kódem a kódováním nám zobrazuje Obr.2



Obr.2. Vztah mezi symbolem a kódem

Vyjádřeno matematicky : Při přenosu informací používáme dvě základní abecedy.

Zdrojová abeceda (abeceda zdroje) $A = \{ a_1, \dots, a_N \}$ je konečná množina znaků (symbolů), kterou chceme přenést přes nějaké přenosové zařízení (přenosový kanál).

Přenosová abeceda (abeceda přenosového kanálu) $Y = \{ y_1; \dots, y_R \}$ je konečná množina znaků y_j , kterou můžeme přenášet přes nějaké konkrétní přenosové zařízení. Např. zařízení se dvěma úrovněmi řízení (0,1) bude mít abecedu $Y = \{ y_1, y_2 \} = \{ 0, 1 \}$. Pod *kódováním* pak rozumíme zobrazení, které každému znaku $a_i \in A$ přiřadí nějakou skupinu znaků $y_j \in Y$ - *kódované slovo*. Množina všech kódovaných slov se pak nazývá *kód*. Kódování však nemá pouze mechanicky nahrazovat jednu konkrétní abecedu jinou. Mění-li se při kódování množina použitých symbolů i rozložení pravděpodobnosti, mění se také entropie na symbol zprávy. Hlavním cílem je zajistit nejmenší počet symbolů na dané množství informace.

Důležité je, aby bylo možné ze zakódované zprávy získat zprávu zdrojovou; tj. kódování musí být *jednoznačně dekódovatelné*. Kódování je jednoznačně dekódovatelné tehdy, jestliže zobrazení, které každému znaku $a_i \in A$ přiřadí nějakou skupinu znaků $y_j \in Y$ je prostým zobrazením.

3 DĚLENÍ KÓDŮ

Kvůli rozsáhlosti dané kapitoly jsem provedl pouze stručný přehled nejdůležitějších kódů a principu jejich tvoření. Zdaleka nejde o celkovou problematiku, která je mnohem rozsáhlejší.

3.1 Rovnoměrné kódy

Rovnoměrné kódy jsou takové kódy, u kterých se každému znaku zdrojové abecedy přiřadí stejný počet kódových znaků. Z toho důvodu mají tyto kódy velkou výhodu při dekódování. Nerozšířenější příklady rovnoměrných kódů jsou: *telegrafní kód* a *ASCII kód*.

3.1.1 Telegrafní kód

Zdrojová abeceda obsahuje všechny písmena anglické abecedy, číslice 0, 1, ..., 9 a některá interpunkční znaménka. Každému znaku zdrojové abecedy $2^5=32$ znaků je přiřazena konkrétní pětice kódových znaků 0 a 1.

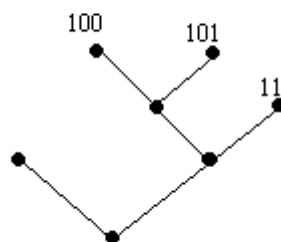
3.1.2 ASCII kód

ASCII (American Standard Code for Information Interchange). Zdrojová abeceda obsahuje kromě písmen anglické abecedy a číslic i znaky používané ve výpočetní technice (malá písmena, algebraické znaky +, -, *, = ...). Každému z $2^7=128$ znaků zdrojové abecedy se přiřadí sedmice kódových znaků 0 a 1.

3.2 Nerovnoměrné kódy

Nerovnoměrné kódy jsou takové kódy, kdy se jednotlivým znakům zdrojové abecedy přiřazují kódová slova, která mají různý počet znaků. Nejznámějším a nejrozšířenějším příkladem tohoto kódování je *Morseova abeceda*.

Ke konstrukci nerovnoměrných kódů se využívá graf (viz Obr.3). V teorii grafů se takový graf nazývá strom. Strom obsahuje uzly a z každého uzlu stromu vycházejí dvě hrany. Posloupnost za sebou následujících hran se nazývá cesta.



Obr.3. Konstrukce nerovnoměrného kódu

Po dohodě, že při pohybu po stromě od jednoho uzlu k druhému přiřadíme:

a) směrem doprava 1

b) směrem doleva 0

3.2.1 Morseova abeceda

Je obecně známo, že Morseova abeceda obsahuje znaky ”.” a ”-”. Znakům s vyšší frekvencí výskytu je přiřazeno kratší kódové slovo a naopak znakům s nižší frekvencí výskytu delší kódové slovo.

Tab.1. Tabulka Morseové abecedy

Písmena	Kód	Písmena	Kód	Písmena	Kód	Písmena	Kód
A	. -	K	- . -	U	.. -	5	
B	-	L	. - ...	V	... -	6	-
C	- . . .	M	- -	W	. - -	7	- - . . .
D	- ..	N	- .	X	- . . -	8	- - - . .
E	.	O	- - -	Y	- . - -	9	- - - - .
F	.. - .	P	. - - .	Z	- - . .	0	- - - - -
G	- - .	Q	- - . -	1	. - - - -	pozor	- . - . - .
H		R	. - .	2	.. - - -	nerozum	
I	..	S	...	3	... - -	konec	... - . -
J	. - - -	T	-	4	 -	rozumím	- - - - . -

3.3 Efektivní kódy

U běžného kódování známe pravděpodobnosti výskytu znaku a_i zdrojové abecedy $P(A=a_i) = p_i$ ale neznáme délky kódových slov n_i . Musíme proto hledat délky kódových slov tak, aby průměrný počet kódových znaků připadajících na jeden zdrojový znak byl minimální. Takové kódy nazýváme *efektivní* nebo *minimální*. Myšlenka efektivního kódu je založena na následujícím tvrzení:

Pro nezávislé vysílání znaků zdrojové abecedy $A = \{ a_1, a_2, \dots, a_N \}$ s pravděpodobnostmi jednotlivých znaků $P(A=a_i) = p_i, i=1, \dots, N$, zvolme délku kódového slova n_i tak, že platí:

$$2^{-n_i} \leq p_i < 2^{-n_i+1} \quad (1)$$

Takové číslo existuje jediné a platí pro něj:

$$\sum_{i=1}^N 2^{-n_i} \leq \sum_{i=1}^N p_i = 1 \quad (2)$$

3.3.1 Shannonova -Fanova metoda

Tato metoda je nejpoužívanější metodou pro sestrojení minimálního kódu.

Algoritmus:

1. Zdrojové znaky se uspořádají postupně podle pravděpodobnosti p_i v klesajícím pořadí.
2. Zdrojové znaky se rozdělí na dvě podskupiny tak, aby součet pravděpodobností v obou skupinách byl přibližně stejný.
3. První skupině se přiřadí kódový znak **1** a druhé skupině **0**.
4. Každou podskupinu opět rozdělíme na dvě podskupiny s přibližně stejným součtem pravděpodobností.
5. Prvním podskupinám se přiřadí znak **1** a druhým **0**. Tak získáme druhý kódový znak.
6. V dělení na skupiny pokračujeme dokud v každé podskupině nezůstane pouze jeden zdrojový znak.

3.3.2 Huffmanova metoda

Tato matematická metoda konstrukce minimálního kódu je složitější než předchozí metoda, vede ale vždy ke kódu s maximální efektivností.

(Vypracoval ji O. Huffman v r. 1952).

Algoritmus:

1. Zdrojové znaky se uspořádají postupně podle pravděpodobnosti výskytu p_i .
2. Sečteme poslední dvě pravděpodobnosti a výsledek zařadíme podle velikosti mezi ostatní pravděpodobnosti - *redukce*.
3. Znovu sečteme dvě poslední pravděpodobnosti a výsledek opět zařadíme podle velikosti.
4. Sečítání pravděpodobností provádíme tak dlouho, až dojdeme k součtu 1.
5. Posledním dvěma znakům přiřadíme kódové znaky **0** a **1**.
6. Zpětným postupem přiřazujeme jednotlivým sčítancům vždy kódové znaky **0** a **1**, dokud nepřiřadíme kódové znaky všem zdrojovým znakům.

3.4 Bezpečnostní kódy

V reálném životě často dochází při přenosu zakódovaných informací k chybám způsobeným šumem prostředí (přenosového kanálu). V každé oblasti přenosu a zpracování diskretních informací se proto postupně stabilizuje používání určitých kódů, které se ukazují jako nejvýhodnější z hlediska kompromisu mezi:

- a) stupněm zabezpečení proti chybám,
- b) jednoduchostí (efektivností),
- c) cenou realizace příslušných zařízení.

Snažíme se tedy najít kód, který má co nejkratší délku a přitom opravuje co největší počet chyb.

Snížení výskytu chyb při přenosu dat je možné zabezpečit několika způsoby:

Bez zabezpečovacích zařízení tj. úpravou zprávy. Je to velice účinný způsob ochrany, nevyžaduje žádná přídavná zařízení. Nevýhodou je pracnost a zdlouhavost.

Nejjednodušším způsobem snížení chybnosti je zvýšení redundance (nadbytečnosti) zprávy. Jednou z možností je opakování vysílané zprávy, tedy z následného porovnání je možné odhalit případné chyby (kdybychom přenesli zprávu třikrát, bylo by možné zjištěné chyby většinou i odstranit).

U číselných položek se používá *úprava kontrolním součtem*, např. chceme-li přenést číslo **54321** sečteme $5+4+3+2+1=15$, poslední číslici přidáme k původnímu číslu a pak přenášíme číslo **543215**.

S pomocí zabezpečovacích zařízení tj. bezpečnostní kódy.

Kontrolou kvality signálu tj. při vybočení sledovaného parametru signálu z tolerance se žádá o opakované zaslání posledního bloku dat.

3.4.1 Zabezpečující schopnosti kódu

Detekční schopnost Pod detekčními schopnostmi kódu rozumíme schopnost kódu objevovat - *detekovat* chyby vzniklé při přenosu informací. Z důvodu, aby bylo možné objevovat t -násobné chyby, byla definována tzv. *Hammingova vzdálenost* – **d** . (Pro dvě kódová slova popsaná vektory $a = \{a_1, a_2, a_3 \dots a_n\}$ a $b = \{b_1, b_2, b_3 \dots b_n\}$ je Hammingova vzdálenost definovaná jako počet míst, na kterých se hodnoty symbolů vektoru a a b liší). Pro objevování 1 -násobných chyb se používají **kódy kontroly parity**. V tom případě k binárnímu slovu přidáme znak **0** nebo **1** tak, aby výsledné slovo mělo sudý počet jedniček - *sudá parita* (lichý počet jedniček - *lichá parita*). Kód sudé parity je obvykle nazýván **kódem celkové kontroly parity**. Kód (8, 7) - 7 znaků nese informaci, 1 znak je redundantní a slouží k zabezpečení kódu. Výsledný kód skutečně objevuje všechny jednoduché (1 -násobné) chyby.

Kód objevuje t -násobné chyby, když platí:

$$t < d \quad (3)$$

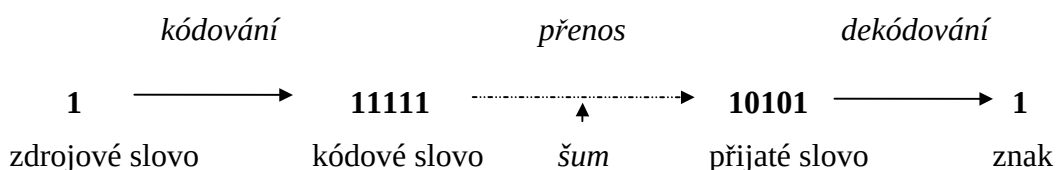
U kódů celkové parity mají všechna kódová slova Hammingovou vzdálenost minimálně $d = 2$.

Korekční schopnost Pod pojmem korekční schopnosti kódu rozumíme schopnost kódu chyby objevené při přenosu i opravit - *korigovat*. Abychom mohli opravit t -násobní chybu, musí platit:

$$t < \frac{d}{2} \quad (4)$$

Kód s minimální Hammingovou vzdáleností $d = 3$ je schopen opravit 1-násobné chyby a 2-násobné chyby pouze objeví (opravil by je nesprávně).

Např. pro nekvalitní přenosové kanály se často používá **opakovací kód**. Přenášený znak se vyšle vícekrát (lichý počet krát - např. 5). Pro dekódování pak platí většinou zastoupení - větší počet stejných znaků je uznán.



Obr. 4. Korekční schopnost kódu

3.4.2 Lineární kódy

Lineární kódy se vyznačují tím, že libovolná lineární kombinace kódových slov je opět kódovým slovem. Vyplývá to z toho, že lineární kód tvoří lineární podprostor n -rozměrného prostoru binárních proměnných. Tento podprostor je možné popsat jeho *bází*. Každý prvek báze je sám kódovým slovem. Báze by měla mít zařazená kódová slova s co nejmenším počtem jedniček, vzájemně nezávislá.

Z důvodu zpřehlednění manipulace s kódovými slovy se kódová slova vyjadřují jako řádkové vektory, které se zapisují do maticových tvarů. Kódová slova báze zapsána pod sebe tvoří **generující matici** G :

$$G = \begin{bmatrix} g_{11} & \cdots & g_{1n} \\ \vdots & & \vdots \\ g_{k1} & \cdots & g_{kn} \end{bmatrix} \quad (5)$$

Generující matice má u kódu (n, k) n -sloupců a k -řádků, přičemž n je celkový počet znaků, k je počet znaků nesoucích informaci a $n-k=r$ je počet zabezpečovacích znaků.

Pro matici G musí platit:

- a) každý její řádek je kódovým slovem,
- b) její řádky jsou lineárně nezávislé (hodnota matice $G=k$).

Zamění-li se sloupce generující matice G , vzniká jiný kód, ale zabezpečující schopnost kódu se nemění. Tak lze soustředit prvky nesoucí informace na prvních k místech tj. *jednotková matice* E_k a zbylých r místech tj. *matice zabezpečujících prvků* R :

Kódy, u kterých lze generující matici napsat jako součin jednotkové matice a matice zabezpečujících prvků se nazývají *systematické kódy*.

1. Vysílané zabezpečené kódové slovo z ($1 \times n$) se vytváří z nezabezpečeného kódového slova i ($1 \times k$) pomocí generující matice G :

$$z = i \cdot G \quad (6)$$

2. Přijímané zabezpečené kódové slovo p ($1 \times n$) se bude rovnat vysílanému kódovému slovu z tehdy, jestliže:

$$H \cdot p^T = z \cdot H^T = 0 \quad (7)$$

Matice H je **kontrolní matice** ($r \times r$). Definovaná je pomocí generující matice systematických kódů $G=E_k R$:

$$H = R^T E_r \quad (8)$$

Kontrolní matice se využívá při dekódování zpráv ke kontrole, zda při přenosu kódované informace nedošlo k chybě.

Pro objevování případně i opravu chyb se používá vektor s ($1 \times r$) zvaný **syndrom slova** (soubor příznaků) a je definován jako:

$$s = p \cdot H^T \Rightarrow \begin{array}{ll} \text{jestli } s = 0 & \text{bezchybný přenos} \\ s \neq 0 & \text{chyba v přenosu} \end{array} \quad (9)$$

3.4.3 Hammingovy kódy

Hammingovy kódy jsou lineární kódy, které jsou určeny pro zabezpečování údajů ukládaných do pamětí počítačů, resp. pro přenosové kanály s nezávislými chybami. Tyto bezpečnostní kódy mají nejmenší možnou redundanci. Kód byl sestaven tak, aby syndrom s , pokud není nulový, udával svými prvky polohu chybného místa v kódovém slově.

Prvky vektoru s jsou chápány jako zápis čísla ve dvojkové soustavě:

$$s = \begin{bmatrix} 2^{r-1} & 2^{r-2} & \dots & 2^0 \\ s_r & s_{r-1} & \dots & s_1 \end{bmatrix} \quad (10)$$

Je proto výhodné uspořádat sloupce kontrolní matice H tak, aby v i -tém sloupci bylo dvojkové vyjádření čísla i . Protože platí:

$$s = p \cdot H^T = (z \oplus c) \cdot H^T = z \cdot H^T + c \cdot H^T = 0 + c \cdot H^T = c \cdot H^T \quad (11)$$

má při jednonásobné chybě vektor $chyb - c$ pouze jednu jedničku. Z toho vyplývá, že syndrom s bude roven řádku matice H^T , který odpovídá jedničce v chybovém vektoru c a bude udávat pořadové číslo místa chybného znaku.

3.4.4 Cyklické kódy

Cyklické kódy patří k lineárním bezpečnostním kódům. Jejich specifická vlastnost, podle které se také nazývají, spočívá v tom, že *cyklickou záměnou prvků v použitém kódovém slově vzniká opět platné kódové slovo*. Cyklický kód můžeme definovat pomocí *generujícího polynomu* $G(x)$. Generující polynom cyklického (n,k) kódu je polynom stupně $n-k=r$, který musí dělit polynom (x^n-1) .

Je-li kódové slovo Z_1 platným kódovým slovem

$$Z_1 = [z_0 \ z_1 \ z_2 \ z_3 \ \dots \ z_{n-1}] \quad (12)$$

pak pro cyklický kód jsou platným kódovým slovem i kódová slova

$$Z_2 = [z_1 \ z_2 \ z_3 \ \dots \ z_{n-1} \ z_0] \quad (13)$$

Je výhodné aby byla matematicky kódová slova cyklických kódů reprezentována pomocí polynomů. Kódové slovo délky n je pak vyjádřené polynomem $(n-1)$ -ho stupně:

$$Z_1(x) = z_{n-1}x^{n-1} + z_{n-2}x^{n-2} + \dots + z_3x^3 + z_2x^2 + z_1x + z_0 \quad (14)$$

Cyklický posun prvků kódové složky pak odpovídá násobení polynomu proměnnou x :

$$Z_2(x) = x \cdot Z_1(x) = z_{n-1}x^n + z_{n-2}x^{n-1} + \dots + z_3x^4 + z_2x^3 + z_1x^2 + z_0x \quad (15)$$

Pro modulo x^n platí:

$$x^{n+k} = x^k \quad (16)$$

z toho vyplývá: $x^{n+k} - x^k = (x^n - 1)x^k = 0 \Rightarrow x^n - 1 = 0 \Rightarrow x^n = 1$.

Tímto způsobem dostaneme nejvyšší mocninu na začátek kódového slova. Kódová slova pak budou ve tvaru:

$$Z_2(x) = z_{n-2}x^{n-1} + \dots + z_3x^4 + z_2x^3 + z_1x^2 + z_0x + z_{n-1}$$

$$Z_3(x) = x \cdot Z_2(x)$$

$$Z_4(x) = x \cdot Z_3(x) \quad (17)$$

:

$$Z_m(x) = x \cdot Z_{m-1}(x)$$

3.4.5 Nelineární kódy - inverzní kódy

Inverzní kód patří do blokových, systematických, nelineárních kódů. Tento kód má Hammingovou kódovou vzdálenost $d = 4$. Může objevovat až trojnásobné chyby a může opravovat jednoduché chyby.

Prvním krokem postupu při použití inverzního kódu je, že vysílané nezabezpečené kódové slovo se zopakuje beze změny, je-li v něm sudý počet nul. V případě, že je v něm počet nul lichý, vyšle se v druhé části inverze nezabezpečeného kódového slova. Tak se zabezpečené kódové slovo bude skládat z dvou částí: *informační* a *zabezpečovací*. Princip je zřejmý na přenosu znaků G, L, X zakódovaných telegrafním kódem, které chceme zabezpečit inverzním kódem. Způsob kódování je podrobně zobrazen v tabulce (viz .Tab.2).

Tab.2. Tabulka inverzního kódu

	G	L	X
Vyslané kódové slovo	01011 01011	01001 10110	10111 01000
Přijaté kódové slovo	00011 01011	01011 10110	10111 00000
Zaznamenané	00011 10100	01011 10110	10111 11111
<i>Rozbor</i>	00011	01011	10111
	$\oplus$ 10100	$\oplus$ 10110	$\oplus$ 11111
<i>místo korekce</i>	10111	11101	01000
	chyba	chyba	chyba

Dojde-li k chybě v informační části, pak při součtu *mod 2* se shoduje jediný pár prvků (bitů) v místě chyby (u znaků G a L). Nachází-li se chybný prvek v zabezpečovací části, pak při součtu *mod 2* se pár prvků v místě chyby neshoduje (u znaku X).

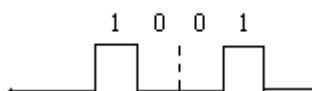
4 PROBLÉMY ZABEZPEČENÍ KÓDŮ

4.1 Způsoby zabezpečení a využití kódů

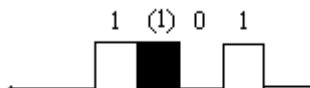
Množina kódových složek je rozložena v signálním prostoru. Každá složka, kterou si představíme jako bod, má přesně definované umístění. V důsledku rušivých vlivů se složka vysune do jiné polohy (směr a velikost posuvu záleží na velikosti a fázi rušivého signálu). Při náhodném charakteru rušivého vlivu se poloha kódové složky neustále mění a představuje celou soustavu bodů kolem místa určeného souřadnicemi signálu bez poruch. Je-li rozložení náhodných poruch dáno Gaussovou distribucí, bude velkých odchylek daleko méně než malých. V případě výskytu velkých odchylek je možné, že se budou prostory jednotlivých kódových složek prolínat. Je zřejmé, že správné vyhodnocení takového signálu je nejisté a je předmětem rozhodování na přijímací straně.

Již zjednodušená představa rozložení množiny signálů v prostoru napovídá, že pravděpodobnost chybného přenesení zprávy se zmenší vzdálením jednotlivých symbolů množiny. Toho můžeme dosáhnout např. zvýšením výkonu přenášeného signálu. Tato cesta není však prakticky vždy použitelná z různých důvodů (přeslechy, ekonomie zařízení, zdroj signálu mimo náš vliv). Výhodnější je, dosáhneme-li zvětšení vzdálenosti zvětšením kombinační kapacity signálových symbolů – tj. vyjádříme-li signálové symboly nějakým kódem a pak je vzdálíme vložím dalších míst do každé kódové složky. Opatření, která provádíme ke zvýšení odolnosti kódových složek proti vlivům poruch, nazýváme souborně bezpečnostním kódováním (nebo též kódovým zabezpečením). Zabezpečení může spočívat jednak v dodatečné úpravě kódové složky nesoucí informaci, jednak v konstrukci kódu.

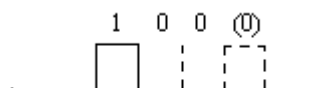
Víme že s nadbytečností vyjádřenou zkrácením délky kódu ($L_z < L$) roste bezpečnost přenosu. Mějme přímý binární kód o délce $L = 2^m$. Přiřadíme-li ke každé kódové složce významový prvek, pak jsme využili celou délku kódu. Pro $m = 4$ je $L = 2^4 = 16$. Jedna z kódových složek odpovídá **Obr.5**. Při přenosu může dojít ke změně tvaru složky působením rušivých vlivů (impulsní hluky, přeslech, atmosférické poruchy atd.) např. podle **Obr. 6,7**. V prvním případě (**Obr.6**) se objevila místo nuly jednička, ve druhém případě tomu bylo naopak (**Obr.7**). Protože je všem kombinacím přiřazen nějaký význam (informace) má každá z chybně přenesených kódových složek některý z významů přiřazených zvoleným šifrovacím klíčem (i nesprávně přenesená) kódová složka se proto vyhodnotí jako správná, bez možnosti kontroly správnosti.



Obr.5 Kódová složka nezkreslená



Obr.6 Kódová složka pod vlivem rušení



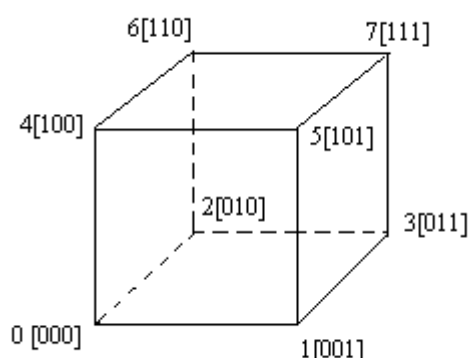
Obr.7 Kódová složka pod vlivem rušení

Nejjednodušší možností zabezpečení kódu je opakování každé kódové složky. Možnost použití platí pro kód vyjádřený libovolnou číselnou soustavou.

4.2 Geometrický model kódu

Vzdálení kódových složek zmenšuje pravděpodobnost chybného vyhodnocení. Pojem vzdálenost vede ke geometrickému modelování kódu [5] [9]. Takový model je velmi prospěšný pro odvození zabezpečujících vlastností kódu. Umožňuje názorné odvození vzdálenosti kódových složek potřebné pro detekci α chyb a korekci β chyb.

Hamming stanovil, že pro dvojkový m -místný kód je třeba m -rozměrné krychle, pro zobrazení jeho 2^m kódových složek. Nejvýhodnější je představa v trojrozměrném prostoru, tedy při $m = 3$. Předpokládáme, že délka hrany krychle je rovna jedné. Taková krychle zobrazí trojmístný dvojkový kód o osmi různých kombinacích (viz Obr.8)



Obr.8 Zobrazení kódu trojrozměrnou krychlí

Když zvolíme jeden (libovolný) z vrcholů krychle jako výchozí, a opíšeme kouli se středem ve zvoleném vrcholu a bude-li poloměr koule respektovat vzdálenost d mezi jednotlivými kódovými složkami (umístěnými na vrcholech krychle), pak kombinace $\binom{m}{d}$

určuje počet vrcholů ležících na opsané kouli. Vzdálenost mezi jednotlivými kódovými složkami označíme d ; m je počet míst kódové složky, tedy $m = 3$. Vrcholy vymezené koulí jsou ekvidistantní (mají stejnou vzdálenost) vzhledem ke středovému vrcholu. Pro $m = 3$ přicházejí v úvahu poloměry koule $r = 1$, $r = \sqrt{2}$, $r = \sqrt{3}$. Pro

$$r = 1 \quad \text{je } d = 1$$

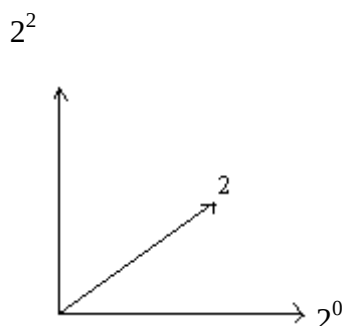
$$r = \sqrt{2} \quad \text{je } d = 2$$

$$r = \sqrt{3} \quad \text{je } d = 3$$

(poloměry zde představují hranu krychle, stěnovou úhlopříčku a tělesovou úhlopříčku).

Na krychli jsou zobrazeny kódové složky ve tvaru: $C_1 - 000$, $C_2 - 001$, $C_3 - 010$, $C_4 - 011$, $C_5 - 100$, $C_6 - 101$, $C_7 - 110$, $C_8 - 111$. Rozložení dvojkových koeficientů 0; 1 v trojrozměrném prostoru bylo provedeno podle diagramu Obr.9, kde osa x odpovídá dvojkovému řádu 2^0 , osa y dvojkovému řádu 2^1 a osa z dvojkovému řádu 2^2 . Rozložení dvojkových řádů je zaměnitelné.

Trojrozměrná krychle poskytovala kód o délce $L = 2^3 = 8$. Geometrické prostředky zobrazení jsou výhodné, pro přehlednost, kterou poskytují zejména při aplikaci různých kódovacích klíčů, při analýze vlastností navrženého kódu apod.



Obr.9 Diagram rozložení dvojkových
řádů na geometrickém modelu kódu

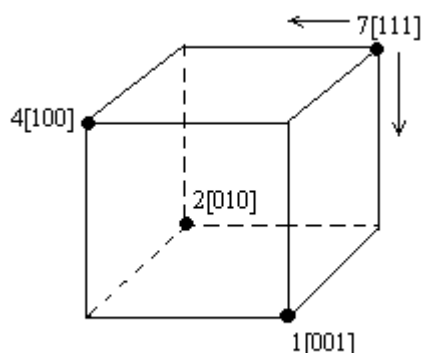
4.3 Vzdálenost kódových složek

Při bližším pozorování kódové krychle a rozložení složek kódu na jejich vrcholech zjistíme že, vzdálenost d mezi dvěma sousedními vrcholy je rovna jedné. Přiřadíme významy kódovým složkám tak, že se všechny stanou nositeli informace - dostaneme kód o maximální délce $L = 2^3 = 8$ (nezkrácený). Jestliže např. složka 010 se vlivem rušení (posune o $d = 1$) změní na 011, dostaneme opět složku kódu a nesoucí informaci. Je nemožné vyhodnotit přijatou zprávu jako chybnou či správnou. Kód neumožňuje registraci chyby. Vzdálenost složek C_1, C_2 symbolicky značíme: $d(C_1; C_2) = 1$

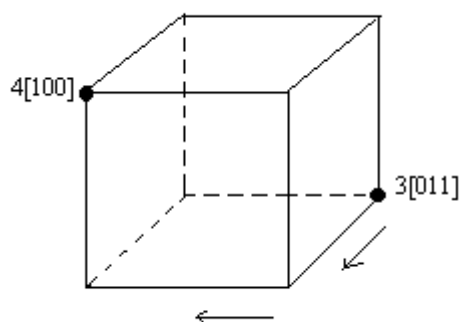
Sledujeme, jaký vliv na zabezpečení vlastnosti kódu bude mít zvětšení vzdálenosti na hodnoty $d > 1$ (d jsou vždy celá kladná čísla). Necht' $d = 2$. Potom zahrne kód pouze ty složky, které vymezuje koule o poloměru $r = \sqrt{2}$. Je-li středem např. vrchol s kódovou složkou $C_5 = 100$ pak dalšími složkami kódu jsou: $C_2 = 001$; $C_3 = 010$; $C_8 = 111$. Zbývajícím čtyřem trojmístným kombinacím není přiřazen žádný význam. Délka kódu je $L_z < L$; $L_z = 4$. Z geometrického modelu kódu je zřejmé, že použité dvojkové kombinace jsou vzájemně vzdáleny o délku dvou hran, tedy vzdálenost kódových složek je $d(C_1; C_2) = 2$

Dojde-li k chybě na jednom řádovém místě, kódová složka C_1 se změní na jednu ze tří sousedních dvojkových posloupností, kterým není přiřazen žádný význam a které nejsou součástí uvedeného kódu (Obr.10a). To umožňuje zjistit chybu v přijaté kódové složce. Protože však lze k téže nevyužitě dvojkové posloupnosti dospět z různých vrcholů krychle (z různých použitých kombinací), můžeme pouze zjistit, zda chyba nastala, či ne.

Takto chybu pouze detekujeme. Vzdálenost $d = 2$ umožňuje tedy pouze zjištění chyby, nedovoluje však určit správný tvar vyslané kódové složky.



Obr.10a



Obr.10b

Obr.10 Posun kódové složky

Zvětšením vzdálenosti kódových složek na $d(C_1; C_2) = 3$ dospějeme k takovému zkrácení kódu, že $L_z = 2$ a informaci nesou ty vrcholové kombinace, z nichž jedna je středem opsané koule a druhá jako jediná na jejím povrchu při poloměru (daném tělesovou úhlopříčkou) $r = \sqrt{3}$. Je-li středem např. $C_5 = 100$, pak další složkou kódu je $C_4 = 011$.

Vznik chyby v kódové složce si opět představíme jako posun po jednotkové hraně krychle. Vychýlí-li se kódová složka o vzdálenost $d = 1$ (posun na jeden ze tří sousedních vrcholů; změna na jediném řádovém místě) dostaneme novou posloupnost jedniček a nul, která není součástí kódu a je vzdálena o $d = 2$ od druhé složky kódu. Tato rozdílná vzdálenost od složky výchozí a od další složky (složek) kódu dává možnost určení správného tvaru chybně přenesené kódové složky. Příklad je graficky znázorněn na Obr.10b

Z uvedených úvah vyplývá, že kód se vzdáleností $d = 1$ nemá zabezpečující vlastnosti. Vzdálenost $d = 2$ umožňuje zjištění (detekci) chyby. Vzdálenost $d = 3$ umožňuje opravu (korekci) chyby. Označíme-li počet chyb, které chceme zjistit (detekovat) α , a počet chyb, které chceme opravit (korigovat) β , platí:

- a) $d \geq \alpha + 1$ zjistíme α chyb v jedné kódové složce
- b) $d \geq 2\beta + 1$ opravíme β chyb v jedné kódové složce
- c) $d \geq \alpha + \beta + 1$ zjistíme α chyb a opravíme β chyb (musí platit $\beta < \alpha$)

Mezi schopnosti kódu detekovat či korigovat chyby a mezi nejmenší dovolenou vzdáleností sousedních kódových složek platí pevné vztahy. Z toho také plyne větší zabezpečovací schopnost kódu se složkami o větším počtu míst.(Tab.3)

Tab.3 Vlastnosti kódu s různými vzdálenostmi d

d	vlastnost kódu
2	detekuje 1 chybu
3	koriguje 1 chybu nebo detekuje 2 chyby
4	koriguje 1 chybu a detekuje 2 chyby nebo detekuje 3 chyby
5	koriguje 2 chyby nebo koriguje 1 chybu a detekuje 2 chyby nebo detekuje 4 chyby
6	koriguje 2 chyby a detekuje 1 chybu nebo koriguje 1 chybu a detekuje 4 chyby nebo detekuje 5 chyb

Vzdálenost kódových složek je velmi důležitým kritériem zabezpečujících vlastností kódu. Tato vzdálenost se někdy také označuje jako Hammingova vzdálenost. Je definována jako počet míst na nichž se dvě kódové složky vzájemně liší.

Obecně je vzdálenost dvou kódových složek definována jako suma součtů (mod 2) koeficientů na odpovídajících řákových místech dvou kódových složek.Tedy

$$d(C_i; C_j) = \sum_{i=0}^{m-1} (a_i + a_j)(\text{mod } 2) \quad (18)$$

Všechny kódové složky přirozeně nemusí mít vzájemnou vzdálenost konstantní. Vlastnosti kódu podle předchozích předpokladů charakterizuje nejmenší vzdálenost $d(C_i; C_j)_{\min}$.

4.4 Hammingova váha kódové složky

Je definována jako součet nenulových míst dvojkové posloupnosti tvořící kódovou složku.

$$\sum_{i=0}^{m-1} a_i = w \quad (19)$$

Protože vzdálenost dvou sousedních kódových složek je rovna počtu míst v nichž se obě liší a na těchto místech dostaneme při sčítání (mod 2) vždy jedničku, odpovídá hodnota vzdálenosti daná součtem jedniček Hammingově váze w .

4.5 Sčítání (mod 2)

S touto operací se setkáváme u velké části přeměn dvojkových kódů.

Součty (mod 2) můžeme psát

$$(0 + 0) \bmod 2 = 0 \oplus 0 = 0$$

$$(0 + 1) \bmod 2 = 0 \oplus 1 = 1$$

$$(1 + 0) \bmod 2 = 1 \oplus 0 = 1$$

$$(1 + 1) \bmod 2 = 1 \oplus 1 = 0$$

Součet (mod 2) značíme symbolicky znaménkem $\oplus$ a jeho pravidla sestavujeme do Tab.4

Tab.4.Tabulka operace součet (mod 2)

$\oplus$	0	1
0	0	1
1	1	0

Počet operace součet (mod 2) odvozené z operace platných pro komutativní grupu o dvou elementech odpovídá logické operaci negace ekvidistance nebo též antivalence. Na pořadí sčítanců (mod 2) nezáleží.

Platí pro ni jak komutativní zákon

$$0 \oplus 1 = 1 \oplus 0 = 1 \quad (20)$$

tak asociativní zákon

$$0 \oplus (1 \oplus 0) = (0 \oplus 1) + 0 = 1 \quad (21)$$

4.6 Distribuce chybových míst v kódové složce

Je vyslána kódová složka ve tvaru $C_{iv} = 0110$. Na složku působí během přenosu rušivé vlivy, které můžeme vyznačit také jako dvojkovou posloupnost, např. $C_{ir} = 0011$, kterou označíme jako chybovou složku. Jedničky v chybové složce označují místa na nichž došlo k chybě, nuly správně přenesená místa. Na přijímací straně obdržíme výslednou kódovou složku C_{ip} jako součet (mod 2) složek $C_{iv} = C_{ir}$. Tedy

$$C_{ip} = C_{iv} \oplus C_{ir} = 0110 \oplus 0011 = 0101 \quad (22)$$

Také platí

$$d(C_{iv}; C_{ip}) = w(C_{iv} \oplus C_{ir}) \quad (23)$$

Ve složce C_{iv} byla rušením zasažena řádová místa a_0, a_1 . Tedy dvě místa. Lze však rušit kterákoli dvě místa v kódové složce, také však kterékoli jedno místo a kterákoli tři místa atd. Přitom pravděpodobnost změny 0 na 1 může být jiná, než pravděpodobnost změny 1 na 0.

Obecně, je-li kódová složka m -místná a z toho je s míst zasaženo rušením (tedy chybných), dostaneme $\binom{m}{s}$ různých s míst chybných. Pro 1, 2, 3 atd. chybných s míst musí platit

$$\sum_{s=0}^m \binom{m}{s} = 2^m \quad (24)$$

možnosti nulové až m -násobné chyby.

Přitom pravděpodobnost chyby na určitých s místech je vyjádřena jako součin pravděpodobnosti vzniku chyby a pravděpodobnosti, že chyba na ostatních místech nevznikne. Pro výskyt chyby platí pravděpodobnost p a pravděpodobnost $(1-p)$ pro ostatní místa složky. Pak pravděpodobnost, že na s místech kódové složky se vyskytne chyba, a na $(m-s)$ místech nikoli, bude

$$P_{ch} = p^s (1 - p)^{m-s} \quad (25)$$

Protože však libovolných s z m míst a libovolných $(m - s)$ z m míst může být vyjádřeno, platí pro pravděpodobnost P_{ch} výskytu s násobné chyby v libovolné posloupnosti

$$P_{ch} = \binom{m}{s} p^s (1 - p)^{m-s} \quad (26)$$

což odpovídá binomickému rozdělení (Bernoulli).

Z pravděpodobnosti P_{ch}

$$P_{ch} = P^s (1 - p)^{m-s} \quad (27)$$

můžeme odvodit důležitý poznatek pro výskyt chybových složek. Označíme-li pravděpodobnost vzniku chyby na jednom místě kódové složky $P_{ch\ 1}$, pravděpodobnost vzniku chyby na dvou místech $P_{ch\ 2}$ a analogicky pravděpodobnost vzniku chyby na n -tém místě $P_{ch\ n}$, pak platí podle předchozích pravděpodobností nerovnost.

$$P_{ch\ 1} > P_{ch\ 2} > \dots > P_{ch\ n} \quad (28)$$

5 CHYBY

5.1 Ochrana zprávy před vlivem chyb

Kódové zabezpečení (nazývané podle účelu použití také kanálové kódování) je vyjádření zprávy, které obsahuje redundanci za účelem snížení pravděpodobnosti chyby při přenosu zprávy kanálem. Zpráva, která obsahuje chybu může být chráněna dvěma způsoby.

- Chyba ve zprávě je rozpoznána (detekována) a zpráva musí být přenesena znovu bez chyby.
- Chyba ve zprávě je rozpoznána, klasifikována a opravena (korigována); zpráva nemusí tedy být znovu přenášena.

Podle tohoto rozdělení jsou úpravy zprávy, které mají chránit informační obsah, nazývány *detekční kódování a korekční kódování*. Je zřejmé, že nároky se budou u obou způsobů kódování zpráv lišit. Bezpečnostní kód, má-li být účinný, musí mít dobrý informační poměr a kódová slova musí mít dostatečně velkou kódovou vzdálenost. Při hledání kódu máme tedy stanoveny vlastnosti kódování, ale nemáme předepsán způsob, jak tento „ideální kód“ máme nalézt. Dobrý informační poměr kódu bude snáze dosažitelný, budou-li slova kódu co nejdelší. Z tohoto důvodu bude jistě výhodou, jestliže kódování bude popsáno jako algoritmus výpočtu a ne jako vyhledávání kódových slov v tabulce.

Výhodou vlastností kódování, které je popsáno jako algoritmus, jistě bude linearita matematického vztahu. Při splnění požadavků linearity, bude možné popsat řadu vlastností, které jsou platné pro obecná řešení lineárních algebraických rovnic. Pro soustavy lineárních rovnic budou platit pravidla pro operace s maticemi nebo pro operace s mnohočleny. Toho je možné využít při hledání efektivního řešení kódových soustav.

5.2 Detekce chyb

5.2.1 Lineární binární blokové kódy

Binární blokový kód je lineární, jestliže výsledkem součtu dvou kódových slov je opět kódové slovo. Pro vytváření účinných způsobů kódování se nejčastěji používá úprava zprávy vyjádřené v binární podobě. Předpis pro úpravu zpráv je vhodné navrhnout jako lineární závislost. Bude se vždy jednat o vztah mezi informačními (zdrojovými) slovy a

kódovými slovy. Tento vztah je výhodné vytvořit jako jednoznačný předpis pro výpočet kódového slova ze slova informačního a jednoznačný výpočet původního informačního obsahu z kódového slova.

Praktické zjednodušení synchronizace slov zprávy poskytuje tzv. **blokové kódování**.

Při použití systematického binárního blokového kódu je možné rozlišit v kódovém slově část, která je původním informačním slovem (informační bity) a část, která byla kódováním přidána (zabezpečovací nebo také kontrolní bity). Způsob vytváření slov je možné zobrazit následovně: informační slovo o délce k bitů $u_1u_2\dots u_k$ je kódováním zpracováváno na kódové slovo o délce n bitů $u_1u_2\dots u_ku_{k+1}u_{k+2}\dots u_n$.

Výhodou binárních lineárních kódů je to, že předpis pro kódování může být vždy popsán jako soustava rovnic, ve kterých jsou použity operace sčítání násobení binárních čísel. Pravidla platná pro binární číslice vyjádřené znaky 0 a 1 jsou vyjádřena v tabulkách:

Tab.5.Pro sčítání

$\oplus$	0	1
0	0	1
1	1	0

Tab.6.Pro násobení

$\oplus$	0	1
0	0	0
1	0	1

Tabulka pro násobení binárních čísel odpovídá obvyklým zvyklostem, ale tabulka pro sčítání definuje výsledek $1 + 1 = 0$.

5.2.2 Generující matice kódu

Maticový popis kódovacího předpisu binárního lineárního kódu nelineární soustava rovnic s binárními koeficienty a aritmetikou. Často používaným lineárním binárním kódem

je tzv. kontrola parity, který je správněji nazýván kód celkové parity. Spočívá v tom, že k binárnímu slovu $u_1u_2\dots u_k$ připojíme pouze znak u_{k+1} , který je vyjádřen rovnicí pro kódování

$$U_{k+1} = U_1 \oplus U_2 \oplus \dots \oplus U_k \quad (29)$$

Protože se jedná o lineární binární kódování, můžeme použít vyjádření pomocí maticového popisu. Jeho výhody nebudou zřejmé ihned. Na tomto jednoduchém příkladu kódování kódem celkové parity se může použití maticového popisu jevit jako zbytečná komplikace.

Označíme nejdříve změny, které je nutné rozlišovat, je-li posláno kódové slovo $u_1u_2\dots u_ku_{k+1}$ komunikačním kanálem. Přijaté slovo označíme $v_1v_2\dots v_kv_{k+1}$. Přijaté slovo které bylo poškozené chybou, bude $w_1w_2\dots w_kw_{k+1}$. Nyní můžeme definovat chybové slovo pomocí relace

$$w_1w_2\dots w_kw_{k+1} = v_1v_2\dots v_kv_{k+1} + e_1e_2\dots e_ke_{k+1}, \quad (30)$$

přitom pro obecné vyjádření libovolného znaku chybového slova bude platit $e_i = 1$ ($v_i \neq w_i$), $e_i = 0$ ($v_i = w_i$). Budeme se zabývat případem, kdy vybraná kódová slova obsahují v informačních bitech pouze jednu jedničku a ostatní bity jsou nulové. (Z kódových slov je možné vypočítat další slova kódu).

Definujeme *generující matici* $[G]$ kódu. Tato matice je tvořena kódovými slovy tzv. *báze kódu* (jsou lineárně nezávislá). Chceme-li nalézt kódová slova lineárního binárního kódu, stačí, když nalezneme bázi kódu. Báze kódu je tvořena slovy, která jsou lineárně nezávislá. Ostatní slova lineárního kódu mohou být získána sčítáním libovolných dvou slov báze kódu. Matice G typu (k, n) je generující maticí lineárního kódu, jestliže:

- a) každý její řádek je kódovým slovem,
- b) každé kódové slovo je lineární kombinací řádků,
- c) řádky jsou lineárně nezávislé.

Pro paritní systematický kód má generující matice tvar:

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \quad (31)$$

Není to jediná báze kódu. Báze kódu je tvořena všemi kombinacemi kódových slov, která jsou lineárně nezávislá. (Dokonce mohou být sečtena dvě stejná kódová slova; to ovšem znamená, že výsledkem je nulové kódové slovo. Z tohoto pravidla plyne, že nulové slovo vždy patří do lineárního kódu; ze stejného důvodu však nemůže být obsaženo v generující matici.)

Vybraná kódová slova mohou tvořit jednotkovou matici. Zabezpečovací bitová slova vytváříme podle algoritmu daného kódování. Seřadíme-li takto vytvořená kódová slova do řádků a uzavřeme je do matice, získali jsme tzv. *generující matici systematického kódu*:

$$G = [E|B]. \quad (32)$$

Generující matice má význam pro vytváření kódu v tom, že součinem informačního slova s generující maticí získáme kódové slovo. Tomuto postupu se říká zakódování, ale častěji kódování. Kódováním je tedy myšlen předpis (algoritmus) pro přeměnu informačních slov na slova kódová, ale také operace prováděná podle tohoto algoritmu.

5.2.3 Kontrolní matice

Opačný postup nazývaný *dekódování* se provádí po přijetí slova z komunikačního kanálu pomocí tzv. *kontrolní matice*, která představuje homogenní soustavu lineárních rovnic. Jestliže kódové slovo $v_1v_2\dots v_kv_{k+1}$ (respektive $v_1v_2\dots v_kv_n$, neboť kód celkové parity obsahuje pouze jeden kontrolní bit) přijmeme nepoškozené, jedná se o správné řešení lineární soustavy rovnic; její koeficienty tvoří prvky kontrolní matice.

V tom případě platí:

$$H \cdot \begin{bmatrix} V_1 \\ V_2 \\ \dots \\ V_n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \dots \\ 0 \end{bmatrix} \quad (33)$$

Vztah mezi generující maticí a kontrolní maticí je popsán pomocí matice B.

Kontrolní matice H je definována vztahem:

$$H = [-B^T \mid E'] \quad (34)$$

kde E' je jednotková matice.

5.2.4 Kontrolní matice celkové parity

Provedeme-li výpočet kontrolní matice pro kód celkové parity, získáme sloupcové matice B transponováním řádkovou matici $-B^T$. Záporné znaménko u všech prvků matice můžeme s ohledem na shodný výsledek sčítání nebo odečítání zanedbat.

$$H = [1 \ 1 \ 1 \ .. \ 1 \ 1] \quad (35)$$

Poslední jednička, která je k matici $-B^T$ připojena, je jednotkovou maticí rozměru 1. Součinem této matice H se sloupcovým vektorem, který vznikl z vektoru přijatého slova $v_1 v_2 \dots v_k v_n$, získáváme homogenní soustavu rovnic.

Tato soustava je tvořena jedinou rovnicí, která je totožná s definičním vztahem kódu celkové parity, který je tentokrát vyjádřený pro slovo přenesené komunikačním kanálem a přijaté.

$$v_{k+1} = v_1 \oplus v_2 \oplus \dots \oplus v_k \quad (36)$$

Z poslední rovnice je zřejmé, že nejmenší vzdálenost mezi kódovými slovy kódu celkové parity je vždy $d = 2$. Změnou hodnoty v libovolném informačním bitu se změní také hodnota kontrolního bitu. Odvození generující a kontrolní matice je základním postupem, který bude uplatňován i u dalších lineárních binárních kódů. Pro dekódování je pro kód celkové parity použit implicitní vztah $v_1 + v_2 + \dots + v_k + v_n = 0$. Bude-li rovnice homogenní, bude přijaté kódové slovo bez chyb. Bude-li pravá strana rovnice nenulová, signalizuje to, že během přenosu došlo k chybě.

5.3 Oprava jedné chyby

5.3.1 Hammingovy kódy

Hammingův kód je důležitým lineárním binárním kódem se schopností opravy chyb. Jeho vlastnosti jsou dány kódovou vzdáleností $d_{\min} = 3$. (Vznikají tedy výběrem jedné osminy slov z přirozeného binárního kódu). Kód s minimální vzdáleností 3 detekuje až dvojnásobnou chybu nebo opravuje jednonásobnou chybu. Hammingovy kódy se snadno dekódují a jsou perfektní, tj. mají nejmenší myslitelnou redundanci. Jestliže počet kontrolních bitů m roste po jedné, je celková délka Hammingových kódů $n = 2^m - 1$, takže dostáváme binární (3,2) – kód, (7,4) – kód, (15,11) – kód, atd.

Tab.7 Konfigurace Hammingových kódů

m	2	3	4	5	6	...
n	3	7	15	31	63	...
k	1	4	1	26	57	...

Informační poměr roste rychle k 1, např. Pro kód délky $2^6 - 1 = 63$ je informační poměr

$$\frac{63-6}{63} \succ 0,9 \quad (37)$$

Sloupce kontrolní matice jsou dány dvěma podmínkami: sloupce musí být nenulové (tím je zaručena detekce chyby) a nenulový musí být součet dvou sloupců (což zaručuje detekci dvojnásobné chyby).

Sloupce kontrolní matice musí být nenulové a různé a žádný sloupec se neopakuje. Na příkladu (7,4)- kódu je možné ukázat uspořádání kontrolní matice.

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \quad (38)$$

Abychom našli generující matici, přemístíme sloupce 1 $\Leftrightarrow$ 7, 2 $\Leftrightarrow$ 6 a 4 $\Leftrightarrow$ 5. Získáme matici H' ; která je maticí ekvivalentního Hammingova kódu

$$H' = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \quad (39)$$

Relace $H' = [-B^T|E']$ je možné vypočítat generující matici $G' = [E|B]$:

$$G' = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \quad (40)$$

a zpětným přeházením sloupců získáme generující matici původního Hammingova kódu

$$G = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \quad (41)$$

Kódová slova generovaná maticí G získáme jako vlastní řádky matice a jejich lineární kombinace. Celkem se jedná o šestnáct slov. Mezi kódová slova binárního lineárního kódu a tedy i kódu Hammingova patří nulové kódové slovo. Další, nenulová slova, získáme jako účelové součty řádků generující matice.

Tab.8 Součty řádků generující matice

sčítané řádky	kódové slovo	sčítané řádky	kódové slovo
např.: 1, 1	0000000	1, 2	1000011
1, 2, 4	0001111	4	1001100
2, 3, 4	0010110	1, 3, 4	1010101
1, 3	0011001	2, 3	1011010
1, 4	0100101	2, 4	1100110
2	0101010	1	1101001
1, 2, 3	011011	3	1110000
3, 4	0111100	1, 2, 3, 4	1111111

5.3.2 Algoritmické vytváření kódových slov

Pro praktické zapojení je výhodné vypočítat zabezpečovací bity kódového slova na základě toho, že se jedná o blokový kód. Jednotlivé zabezpečovací bity jsou vyjádřeny relacemi, které můžeme získat z homogenní soustavy vyjádřené maticí H.

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \end{bmatrix} = 0 \quad (42)$$

Násobením matice a transponovaného vektoru kódového slova a postupnou eliminací rovnic můžeme získat vyjádření pro všechny informační bity: $v_5 = v_2 \oplus v_3 \oplus v_4$, $v_6 = v_1 \oplus v_3 \oplus v_4$ a konečně i $v_7 = v_1 \oplus v_2 \oplus v_4$.

Pomocí těchto rovnic pak je možné i ověřit, zda kódové slovo patří do Hammingova (7,4)-kódu. Tato vlastnost je popsána úpravou matice H na tzv.kanonický tvar. Postupné úpravy matice H jsou naznačeny ve schématu.

Výsledkem je matice ve tvaru:

$$\begin{aligned} & \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 & \oplus \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & \uparrow \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 & \downarrow \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & \oplus \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 & \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & \oplus \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & \uparrow \end{bmatrix} \rightarrow \\ & \rightarrow \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

Zabezpečovací bity pak získáme za zdrojových jako vektor:

$$[v_5 v_6 v_7] = [v_1 v_2 v_3 v_4] \cdot \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad (43)$$

Použitím tohoto postupu můžeme vypočítat kódová slova Hammingova(7,4) – kódu. Prakticky používaným Hammingovým kódem je rovněž (15,11) – kód. Jeho kontrolní matice je vytvořena podle stejných pravidel.

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \quad (44)$$

S kontrolní maticí Hammingova (15,11) – kódu můžeme provádět stejné manipulace jako ty, které jsou uvedeny výše.

5.3.3 Opravování kódových slov

Postup, který umožňuje opravovat chyby, je založen na tom, že pravděpodobnost jedné chyby je mnohem vyšší, než pravděpodobnost více chyb v jednom kódovém slově. Jedna chyba způsobí, že chybné slovo bude mít od kódového slova vzdálenost 1. Ztráta informace jednoho bitu v kódovém slově se dá opravit. Chybné slovo způsobí, že součin kontrolní matice a přijatého slova je nenulový. Nazývá se syndrom. Předpokládejme, že při přenosu došlo k jednonásobné chybě a přijali jsme slovo [0111010].

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \end{bmatrix} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} \quad (45)$$

Vynásobením přijatého slova s kontrolní maticí dostaneme na pravé straně rovnice vektor syndromu. Syndrom musí být roven sloupci kontrolní matice, jehož pořadí v matici je rovné pořadí nenulového bitu v přijatém slově. Porovnáním syndromu se sloupci kontrolní matice vidíme, že syndrom je shodný se třetím sloupcem.

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \quad (46)$$

K chybě tedy došlo na třetím místě. Změnou znaku na tomto místě opravíme přijaté slovo na správné kódové slovo $[0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0]$. Kdyby bylo přijaté slovo $[0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0]$, znamená to, že došlo ke dvojnásobné chybě. Syndrom pro kontrolní matici by byl $[1 \ 0 \ 0]$. Syndrom je nenulový; dekódování správně detekuje chybu. Kdybychom ovšem nyní chtěli detekovanou chybu opravit, opravíme na špatnou hodnotu kódového slova. Kód není schopen opravovat dvojnásobné a vícenásobné chyby.

5.4 Oprava více chyb

5.4.1 Reedovy-Mullerovy kódy

Reedovy-Mullerovy kódy jsou lineární kódy definované nad $GF(2)$ viz. příloha, které mohou být dekódovány jednoduchou technikou rozhodování. Z těchto důvodů jsou Reedovy-Mullerovy kódy důležité i přesto, že někdy nesplňují nejmenší kódovou vzdálenost pro každé celé m a pro každé celé r menší než m je Reedův-Mullerův kód blokovým kódem o délce 2^m a nejmenší kódové vzdálenosti $d_{\text{nim}} = 2^{m-r}$. Reedův-Mullerův kód je definován pomocí konstrukce generující matice; konstrukce generující matice tohoto nesystematického kódu je zároveň návodem pro postup dekódování. Základem definice generující matice je součin vektorů a a b . Tento součin je definován pro vektory:

$${}^1a = ({}^1a_0, {}^1a_1, \dots, {}^1a_{n-1}) \quad (47)$$

a

$${}^2a = ({}^2a_0, {}^2a_1, \dots, {}^2a_{n-1}) \quad (48)$$

jako

$${}^1a \cdot {}^2a = ({}^1a_0 \cdot {}^2a_0, {}^1a_1 \cdot {}^2a_1, \dots, {}^1a_{n-1} \cdot {}^2a_{n-1}) \quad (49)$$

Generující matice Reedova-Mullerova kódu řádu r vytváří kód délky 2^m a je definována jako matice, která vznikla z $r+1$ dílčích submatic:

$$G = \begin{bmatrix} G_0 \\ G_1 \\ \dots \\ G_r \end{bmatrix} \quad (50)$$

$$G_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \begin{matrix} {}^1a \\ {}^2a \\ {}^3a \\ {}^4a \end{matrix} \quad (56)$$

Řádky submatice G_2 obsahují součiny všech vektorů submatice G_1 právě jednou, protože G_1 má čtyři řádky, G_2 má

$$\binom{4}{2} = 6 \text{ řádků:} \quad (57)$$

$$G_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{matrix} {}^1a \cdot {}^2a \\ {}^1a \cdot {}^3a \\ {}^1a \cdot {}^4a \\ {}^2a \cdot {}^3a \\ {}^2a \cdot {}^4a \\ {}^3a \cdot {}^4a \end{matrix} \quad (58)$$

Submatice G_3 má

$$\binom{4}{3} = 4 \text{ řádky} \quad (59)$$

$$G_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{matrix} {}^1a \cdot {}^2a \cdot {}^3a \\ {}^1a \cdot {}^2a \cdot {}^4a \\ {}^1a \cdot {}^3a \cdot {}^4a \\ {}^2a \cdot {}^3a \cdot {}^4a \end{matrix} \quad (60)$$

Výsledná matice G vznikne spojením submatic do tvaru:

$$G = \begin{bmatrix} G_0 \\ G_1 \\ G_2 \\ G_3 \end{bmatrix} \quad (61)$$

Tato matice G je generující maticí (16, 15)- kódu nad $GF(2)$ je maticí kódu celkové parity s patnácti informačními bity a jedním zabezpečovacím (kontrolním) paritním bitem. O tom se můžeme přesvědčit výpočtem:

$$k = 1 + \binom{4}{1} + \binom{4}{2} + \binom{4}{3} = 1 + \frac{4}{1} + \frac{4 \cdot 3}{2 \cdot 1} + \frac{4}{1} = 1 + 4 + 6 + 4 = 15 \quad (62)$$

Výpočet nejmenší kódové vzdálenosti je dán vztahem $d_{\min} = 2^{4-3} = 2$. Přísnějším výběrem kódových slov může být zkonstruován kód s nejmenší kódovou vzdáleností $d_{\min} = 2^{4-2} = 4$. Číslo r , které představuje stupeň booleovského mnohočlenu, tedy musí být menší, než v předchozím příkladu. Generující matice G bude mít v tom případě tvar:

$$G = \begin{bmatrix} G_0 \\ G_1 \\ G_2 \end{bmatrix} \quad (63)$$

Počet informačních bitů bude:

$$k = 1 + \binom{4}{1} + \binom{4}{2} = 1 + \frac{4}{1} + \frac{4 \cdot 3}{2 \cdot 1} = 1 + 4 + 6 = 11 \quad (64)$$

Je to tedy matice (16, 11)-kódu, což je rozšiřitelný Hammingův kód. Při délce kódového slova šestnáct bitů by bylo možné zkonstruovat ještě Reedův-Mullerův kód s nejmenší kódovou vzdáleností $d_{\min} = 2^{4-1} = 8$. Kódové slovo obsahuje maximálně pět informačních bitů:

$$k = 1 + \binom{4}{1} = 1 + \frac{4}{1} = 1 + 4 = 5 \quad (65)$$

a tvar generující matice je:

$$G = \begin{bmatrix} G_0 \\ G_1 \end{bmatrix} \quad (66)$$

Jedná se o nesystematický rozšiřitelný BCH kód proto mohou být informační bity na různých místech kódového slova. Největší dosažitelná kódová vzdálenost Reedova-Mullerova kódu délky 16 bitů je $d_{\min} = 2^{4-0} = 16$, který může provést opravu kódového slova ještě při výskytu sedmi chyb. Má však pouze jeden informační bit. Je zřejmé, že se jedná o opakovací kód.

5.4.2 Algoritmus dekódování Reedových-Mullerových kódů

Reedův algoritmus je navržen speciálně pro Reedovy-Mullerovy kódy. Reedův algoritmus je neobvyklý ve srovnání s většinou algoritmů pro většinu kódů, ve kterých je informace opravována přímo z přijatého slova a není přímo vypočítávána chyba. Nejsou používány proměnné, které jsou mezivýsledkem.

Předpokládáme, že má být dekodován Reedův-Mullerův kód $(r-1)$ -tého řádu, který obsahuje

$$\left(\frac{1}{2} \cdot 2^{m-(r-1)} - 1 \right) \quad (67)$$

chyb. Zkonstruuje dekodér pro R-M kód r -tého řádu s opravou

$$\left(\frac{1}{2} \cdot 2^{m-r} - 1 \right) \quad (68)$$

chyb a zjednodušíme jej na předcházející případ. Protože již víme, že R-M kód nultého řádu můžeme dekodovat podle majoritního principu, vytvoříme dekodovací algoritmus indukci.

Je obvyklé rozdělit informační vektor na $r+1$ segmentů zapsaných jako

$i = [l_0, l_1, \dots, l_r]$, kde segment l_1 obsahuje $\binom{m}{1}$ informačních bitů. Každý segment násobí

jeden blok matice G . Zakódování může být popsáno jako součin matice a vektoru:

$$V = [l_0, l_1, \dots, l_r] \cdot \begin{bmatrix} G_0 \\ G_1 \\ \vdots \\ G_r \end{bmatrix} = i \cdot G \quad (69)$$

Informační posloupnost je rozdělena do takových částí. Každá část odpovídá jednomu z r bloků generující matice a je násobena touto částí v průběhu procesu zakódování.

5.4.3 Oprava chyb Reedových-Mullerových kódů

Jsme-li schopni nalézt rozmístění informačních bitů v r -té části, jsme to také schopni rozlišit u přijatého slova a odečíst jeho příspěvek. Tím se zmenšuje problém dekodování na menší R-M kód $(r-1)$ řádu. Proces dekodování může postupovat podle majoritního rozhodování až k rozlišení jednotlivých bitů v r -té části.

Přijaté slovo je:

$$V = [l_0, l_1, \dots, l_r] \cdot \begin{bmatrix} G_0 \\ G_1 \\ \vdots \\ G_r \end{bmatrix} + e \quad (70)$$

Dekódovací algoritmus nejdříve provede výpočet l_r ze slova v . Pak je vypočítáno slovo:

$$V' = V - l_r G_r = [l_0, l_1, \dots, l_{r-1}] \cdot \begin{bmatrix} G_0 \\ G_1 \\ \vdots \\ G_{r-1} \end{bmatrix} + e \quad (71)$$

což je kódové slovo Reedova-Mullerova kódu $(r-1)$ řádu s chybou. Nejdříve uvažujeme dekódování informačního bitu i_{k-1} , který je násoben posledním řádkem matice G_r . Je dekódován při nastavení kontrolního součtu 2^{m-r} vypočítaného z 2^m přijatých bitů; každý takový součet zahrnuje 2^r přijatého slova a každý bit je použit pouze v jednom kontrolním součtu. Nevyskytne-li se chyba, je kontrolní součet roven bitu i_{k-1} . Je-li však přítomno

$$\left(\frac{1}{2} \cdot 2^{m-r} - 1 \right) \quad (72)$$

chyb, většina kontrolních součtů bude nadále rovna i_{k-1} . První kontrolní součet je součet modulo-2 z prvních 2^r bitů přijatého slova; druhý je součet modulo-2 z druhých 2^r bitů a tak dále. Existuje-li 2^{m-r} takových kontrolních součtů a za předpokladu $\left(\frac{1}{2} \cdot 2^{m-r} - 1 \right)$ chyb bude většina kontrolních součtů rovna i_{k-1} . Pro Reedův-Mullerův kód, tedy počítáme rovnice:

$$\begin{aligned} i_{10}^{(1)} &= v_0 + v_1 + v_2 + v_3 \\ i_{10}^{(2)} &= v_4 + v_5 + v_6 + v_7 \\ i_{10}^{(3)} &= v_8 + v_9 + v_{10} + v_{11} \\ i_{10}^{(4)} &= v_{12} + v_{13} + v_{14} + v_{15} \end{aligned} \quad (73)$$

Objeví-li se pouze jedna chyba, jedna z těchto rovnic bude chybná; většina rovnic však bude dávat správný výsledek bitu i_{10} . Vyskytnou-li se dvě chyby, výsledek nebude možné rozhodnout na základě většiny a to znamená, že kód detekuje dvě chyby.

Stejným způsobem, jakým je každý informační bit násoben řádkem v G_r , může být i dekódován. Je to možné proto, že každý řádek G_r má rovnocennou úlohu; žádný řádek není

preferován co do důležitosti. Permutací sloupců v každém řádku může být dosaženo pořadí symbolů, jaké má poslední řádek. Tedy k libovolnému kódovému zabezpečení může být použito. Každý bit může být dekodován pomocí až 2^{m-r} lineárních kontrolních součtů vytvořených z 2^m přijatých bitů pomocí majoritního rozhodování.

Potom co jsou určeny bity, informace jsou odečteny od přijatého slova. Tento výsledek je kódovým slovem R-M kódu $(r-1)$ řádu. Tím je ukončena oprava chyb ve zprávě.

ZÁVĚR

Hlavním cílem této bakalářské práce bylo shrnout poznatky o kódování, zabezpečování kódů a o opravách chyb vzniklých při přenosu.

Vzhledem k rozsáhlosti tématu jsem provedl pouze stručné dělení kódů a proto doporučuji, při hlubším zájmu o vytváření různých druhů kódů, dále se zabírat danou problematikou hlouběji. Kódy jsem rozdělil na rovnoměrné, nerovnoměrné, efektivní a bezpečnostní a ke každé skupině uvedl jejich nejznámější nebo nejrozšířenější kódy. V této práci jsem se spíše zaměřil na problémy zabezpečení kódů a vzniku chyb. Byly probrány pojmy jako geometrický model kódu, vzdálenost kódových složek, ochrana zprávy před vlivem chyb, generující matice kódu, kontrolní matice kódu, detekce a korekce chyb, oprava jedné chyby či oprava více chyb atd.

Hlavním zájmem kódování je, aby se zprávy přenášely co nejúčinněji a aby měly pokud možno co nejméně chyb. Obě veličiny tj. efektivita přenosu zpráv a množství chyb ve zprávách jsou dané kapacitou kanálu, kterou matematicky formuloval C. E. Shannon a která mluví o tom že i ve špatném kanále je možné zprávu zakódovat tak, aby byl šum při přenosu potlačen na minimum. S rozvojem technologií se teoretické poznatky o kódech stávají předmětem zájmu techniků po celém světě a umožňují, hlavně pomocí číslicových systémů, mikropočítačů a integrovaných obvodů, tyto teoretické poznatky využívat.

SEZNAM POUŽITÉ LITERATURY

- [1] KLIKA Otakar, LÉBL Miroslav : *Teorie diskrétních kódů*. Nakladatelství dopravy a spojů, Praha. 191s.
- [2] VLČEK, Karel : *Komprese a kódová zabezpečení v multimediálních komunikacích*
2.vyd.Praha: BEN, 2004. 258s. ISBN 80-7300-134-9
- [3] PROKOPOVA, Z : *Základy informatiky*. 1.vyd.VUT v Brně, FT ve Zlíně, 1998.
98s. ISBN 80-214-1129-5.
- [4] IVÁČEK Ján : *Optimální kódy*. Zlín, 2004. 49s.Bakalářská práce na FT ve Zlíně na Institutu řízení procesů a aplikované informatiky.Vedoucí práce RNDr.Ing. Miloš Krčmář

SEZNAM OBRÁZKŮ

Obr. 1. Množina tečkových symbolů.....	11
Obr. 2. Vztah mezi symbolem a kódem.....	12
Obr. 3. Konstrukce nerovnoměrného kódu.....	14
Obr. 4. Korekční schopnost kódu	18
Obr. 5. Kódová složka nezkreslená	24
Obr. 6. Kódová složka pod vlivem rušení	24
Obr. 7. Kódová složka pod vlivem rušení	24
Obr. 8. Zobrazení kódu trojrozměrně krychlí.....	25
Obr. 9. Diagram rozložení dvojkových řádů	26
Obr. 10. Posun kódové složky	27

SEZNAM TABULEK

Tab. 1. Tabulka Morseové abecedy	14
Tab. 2. Tabulka Inverzního kódu.....	22
Tab. 3. Vlastnosti kódu s různými vzdálenostmi d.....	28
Tab. 4. Tabulka operace součet (mod 2).....	29
Tab. 5. Tabulka pro sčítání	33
Tab. 6. Tabulka pro násobení.....	33
Tab. 7. Konfigurace Hammingových kódů	37
Tab. 8. Součty řádků generující matice	38
Tab. 9. Tabulka pro sčítání $[GF(2)]$	příloha PI
Tab. 10. Tabulka pro násobení $[GF(2)]$	příloha PI
Tab. 11. Tabulka pro sčítání $[GF(3)]$	příloha PI
Tab. 12. Tabulka pro násobení $[GF(3)]$	příloha PI
Tab. 13. Tabulka pro sčítání $[GF(4)]$	příloha PI
Tab. 14. Tabulka pro násobení $[GF(4)]$	příloha PI

SEZNAM PŘÍLOH

P I Galoisova tělesa

PŘÍLOHA: GALOISOVA TĚLESA

Pro popis kódů hrají důležitou roli pouze konečná tělesa. U konečných těles (která jsou nazývána Galoisova tělesa), není nutné axiomy ověřovat, stačí znát jednoduchý popis všech těles. Galoisovo těleso je určeno počtem svých prvků q , je označováno $GF(q)$.

I ta nejmenší Galoisova tělesa (s konečným počtem prvků) musí mít definován nulový a jednotkový prvek. Tyto prvky mohou být definovány tabulkou. Pro $GF(2)$ jsou to tabulky:

Tab.9 Pro sčítání platí

$\oplus$	0	1
0	0	1
1	1	0

Tab.10 Pro násobení platí

$\otimes$	0	1
0	0	0
1	0	1

Dalšími příklady nejjednodušších polí jsou $GF(3) = \{0,1,2\}$ s operacemi.

Tab.11 Pro sčítání platí

$\oplus$	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

Tab.12 Pro násobení platí

$\otimes$	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

a pro $GF(4) = \{0,1,2,3\}$ s operacemi.

Tab.13 Pro sčítání platí

$\oplus$	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

Tab.14 Pro násobení platí

$\otimes$	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	3	1
3	0	3	1	2

U $GF(4)$ může být sčítání i násobení definováno jinak, než modulo – 4 sčítání a násobení. Je to způsobeno tím, že se jedná o Galoisovo těleso s algebraickým rozšířením. Galoisových těles může tedy existovat více. Algebraické rozšíření je definováno následovně:

Pro každé těleso T a každý mnohočlen $f(x)$ stupně n je definován okruh $T[f(x)]$, který se nazývá algebraické rozšíření tělesa T . Prvky tohoto okruhu jsou všechny mnohočleny nad tělesem T stupně většího nebo rovného $n - 1$. Algebraické rozšíření $T[f(x)]$ splňuje podmínky toho, aby bylo tělesem právě tehdy, je-li mnohočlen $f(x)$ nerozložitelný