

UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

Kodér/dekodér Reed-Mullerových kódů
Martin Beran

Bakalářská práce

2014

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2013/2014

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Martin Beran**
Osobní číslo: **I10490**
Studijní program: **B2612 Elektrotechnika a informatika**
Studijní obor: **Komunikační a mikroprocesorová technika**
Název tématu: **Kodér/dekodér Reed-Mullerových kódů**
Zadávací katedra: **Katedra elektrotechniky**

Z á s a d y p r o v y p r a c o v á n í :

V teoretické části proveďte analýzu Reed- Mullerových kódů z hlediska jejich vlastností a využití v praxi. Součástí teoretické části bude popis vlastního kódování a dekódování (kodéru/dekodéru).

V praktické části vytvořte SW modul umožňující pro zadanou či importovanou zprávu kódování/ dekódování této zprávy se zobrazením jednotlivých kroků kódování/dekódování. Součástí SW bude i možnost zanést chybu vzniklou při přenosu, čímž lze ověřit schopnosti detekce/opravy chyb daného kódu.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. ADÁMEK, Jiří. Kódování. Praha: SNTL - Nakladatelství technické literatury, 1989.
2. COUCH, Leon W. Digital and Analog Communication systems, 7th edition. Prentice Hall, 2007.
3. VLČEK, Karel. Kompresí a kódová zabezpečení v multimediálních komunikacích. Praha, BEN - Technická literatura, 2004. ISBN 80-86056-68-6.
4. DOBEŠ, Josef a Václav ŽALUD. Moderní radiotechnika, BEN, Praha 2006.
5. LIN, Shu a Daniel J., COSTELLO. Error Control Coding, 2/E. Prentice Hall, 2005. ISBN-10: 0130426725.

Vedoucí bakalářské práce:

Ing. Jan Pidanič, Ph.D.

Katedra elektrotechniky

Datum zadání bakalářské práce:

20. prosince 2013

Termín odevzdání bakalářské práce:

9. května 2014



A handwritten signature in blue ink, belonging to prof. Ing. Simeon Karamazov, Dr.

prof. Ing. Simeon Karamazov, Dr.
děkan

L.S.

A handwritten signature in blue ink, belonging to Ing. Zdeněk Němec, Ph.D.

Ing. Zdeněk Němec, Ph.D.
vedoucí katedry

V Pardubicích dne 31. března 2014

Prohlášení autora

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne: 9. 5. 2014

Martin Beran

Poděkování

Děkuji vedoucímu práce, panu Ing. Janu Pidaničovi, Ph.D. za informace, spolupráci a čas, který mi věnoval při tvorbě této bakalářské práce a rodině za podporu při studiu.

Název

Kodér/dekodér Reed-Mullerových kódů

Anotace

Cílem této bakalářské práce je vytvořit kodér/dekodér vybraných Reed -Mullerových kódů. Software umožňuje zobrazovat jednotlivé kroky kódování/dekódování a umožňuje zavedení chyb, simulujících chyby vzniklé v přenosovém kanálu. Podpůrný software je vytvořen pomocí programu MATLAB.

Klíčová slova

Kódování, Reed-Mullerův kód, MATLAB

Title

The Encoder/Decoder of Reed-Muller Codes

Annotation

The objective of bachelor thesis is aimed to develop an encoder/decoder of selected Reed-Muller codes. Software shows the individual steps of encoding/decoding. The software supports addition of errors, simulating faults in the transmission channel. The software is developed in MATLAB.

Keywords

Coding, Reed-Muller Code, MATLAB

Obsah

Seznam zkratk.....	9
Seznam obrázků.....	10
Seznam tabulek.....	10
Úvod.....	11
1 Kódování	12
1.1 Binární kódování.....	12
1.2 Zdrojové kódování	12
1.3 Kanálové kódování	13
1.4 Příklad binárního kódování.....	13
1.5 Parametry kódování	13
2 Booleovské funkce	15
2.1 Úvod.....	15
2.2 Booleovský polynom	16
2.3 Booleovská množina.....	17
3 Reed-Mullerovo kódování.....	18
4 Příklad Reed-Mullerova kódování/dekódování.....	24
4.1 Zadání	24
4.2 Kódování.....	25
4.3 Dekódování bez chyb.....	26
4.4 Dekódování s opravitelnými chybami	33
4.5 Dekódování bez možnosti opravení.....	38
5 Rozbor programu	43
5.1 Úvod.....	43
5.2 Hlavní okno.....	44
5.3 Okno Zadání	45

5.4	Okna matic	47
5.5	Okna výpočtu slova v	48
5.6	Okno výsledek kodéru	48
5.7	Okno vkládání chyb	49
5.8	Okna pro výpočet koeficientů polynomů.....	50
5.9	Okno výsledek dekodéru	52
Závěr		53
Literatura		54
Přílohy		55

Seznam zkratk

RM Reed-Mullerův kód

LSB (Least Significant Bit) nejméně významný bit

MSB (Most Significant Bit) nejvýznamnější bit

Seznam obrázků

Obrázek 1 - fotografie Marsu přenesená pomocí Reed-Mullerova kódování	18
Obrázek 2 - Hlavní okno programu	44
Obrázek 3 - okno "Zadání"	45
Obrázek 4 - Error Dialog okna "Zadání"	46
Obrázek 5 - okno "Matice 15x16"	47
Obrázek 6 - okno kódování	48
Obrázek 7 - okno "Výsledek kodéru"	48
Obrázek 8 - okno "Vkládání chyb"	49
Obrázek 9 - okno "Výpočet hodnot koeficientů polynomů 3. stupně"	50
Obrázek 10 - panel výpočtu hodnoty koeficientu q_7	50
Obrázek 11 - panel výpočtu slova W'	51
Obrázek 12 - okno "Výpočet hodnot koeficientů polynomů 2. stupně"	51
Obrázek 13 - okno "Výpočet hodnot koeficientů polynomů 1. stupně"	52
Obrázek 14 - okno "Výsledek dekodéru"	52

Seznam tabulek

Tabulka 1 - přehled booleovských funkcí	15
Tabulka 2 - pravdivostní tabulka	16
Tabulka 3 - Reed-Mullerovy kódy $R(r,2)$	19

Úvod

Cílem bakalářské práce je seznámit čtenáře s problematikou zabezpečovacích technik při přenosu informace, aby se zamezilo zkreslení nebo ztrátě dat. Zabezpečení informace se provádí pomocí kanálového kódování. Kódování informací se začalo používat s vývojem telekomunikací a narůstajícího počtu dat, která bylo nutno odeslat. Kódování je proces, který přidělí informaci formu, která je lépe přenosná a méně náchylná na rušení při přenosu. V této práci se věnuji Reed-Mullerovým kódům.

Reed-Mullerovy kódy patří mezi nejstarší a nejznámější druhy kódování, které umožňují opravení libovolného počtu chyb. Objevil/vynalezl je David E. Muller společně s Irvingem S. Reedem v roce 1954. Začali se používat se vzrůstající náročností na telekomunikace, kvůli své schopnosti detekce a korekce chyb. Spadají pod lineární samostatně opravitelné kódy. Jsou základem pro Hadamardovo, Walsh-Hadamardovo a Reed-Solomonovo kódování. V literatuře se značí mnoha způsoby, ale asi nejčastějším je označení $RM(r, m)$.

Hlavní výhodou Reed-Mullerových kódů je snadné kódování a dekódování. Pro kódování se používá generující matice a dekódování probíhá pomocí většinové logiky. Další předností těchto kódů je možnost skládání složitých kódů z již vytvořených kódů jednodušších.

V teoretické části bude obecně popsáno kódování, jeho rozdělení podle druhu a typu, doplněné příkladem vybraného druhu, kterému se tato bakalářská práce bude věnovat. Budou také vyzvednuty parametry binárního kódování, podle kterých se jednotlivé kódy dělí. Další část práce se bude věnovat obecnému popisu Booleovských funkcí, jejich důležitosti a významu v oblasti kódování Reed-Mullerových kódů. Součástí kapitoly bude několik obecných příkladů používaných při kódování a dekódování.

Následující kapitola bude pojednávat obecně o Reed-Mullerových kódech, jejich vzniku a použití. Součástí bude postupný proces kódování, kdy se ze vstupního informačního slova vytvoří kód, který je poté přiveden na dekodér. Dále zde bude popsáno dekódování. Tedy proces kontroly přijatých dat a případná oprava chyb.

V další kapitole bude rozebrán příklad kódování a dekódování konkrétního Reed-Mullerova kódu. Postup bude obsahovat zjištění parametrů přiloženého kódu dle požadavku na opravu chyb a na délce vstupních informací. Následovat bude zakódování vstupních informací a jeho opětovné dekódování pomocí většinové logiky.

Vlastní proces dekódování je ukázán na třech typických případech: (a) při přenosu nevznikla chyba, (b) při přenosu vznikla opravitelná chyba, (c) při přenosu vznikla neopravitelná chyba.

Poslední kapitola bude obsahovat kompletní popis vytvořeného softwaru pro kódování a dekódování Reed-Mullerových kódů.

1 Kódování

Kódováním se rozumí zápis informace do předem daného kódu pomocí kódovací tabulky.

Obecněji je kódování zápis, který určitému počtu prvků přiřadí určitý počet předem daných znaků, kterým příslušný systém rozumí. Kódování je tedy zobrazení prvků množiny A do množiny B :

$$K : A \rightarrow B^* \quad (1.1)$$

Množinou A se rozumí konečná množina prvků neboli zdrojová abeceda obsahující zdrojové znaky. Množina B se nazývá kódová abeceda a její prvky kódové znaky [1].

1.1 Binární kódování

Bezpochyby nejpoužívanějším druhem kódování je kódování binární, které obsahuje pouze dva prvky $\{0,1\}$.

Jde o posloupnost nul a jedniček, do kterých se převede informace. Zápis pomocí logických nul a jedniček nazýváme zkráceně kód.

Vlastní kód má specifické přiřazení bitů příslušnému znaku. Kombinaci těchto bitů nazýváme kódové slovo [1]. Samotné kódování provádí generátor kódu, který ve většině případů obsahuje logické a paměťové obvody. Opačný proces, tedy získání námi zvoleného slova z binárního kódového slova se nazývá dekódování.

Pro zpracování binárního kódování se využívá booleovských funkcí, které mohou také nabývat pouze dvou hodnot. Tyto funkce jsou rozebrány v Kapitole 2.

Existují dva typy kódování, a to zdrojové a kanálové kódování.

1.2 Zdrojové kódování

Zdrojové kódování se používá pro snížení redundance a irelevance v přenášeném signálu. Používá se většinou při přenášení velkého množství informací například obrazových signálů. Snížení redundance, tedy nadbytečnosti, která je definována jako: „*množství dat které je větší, než je množství dat nezbytně nutné pro přenos dané informace*“ [7], vede ke snížení přenosové rychlosti. Redundanci je možné plánovitě zvýšit přidáním kontrolních bitů do kódového slova, dosáhne se tím částečného odhalení chyb při přenosu. Irelevance je definována jako „*nepodstatná složka informace, kterou je možné ve zdrojovém kodéru zcela potlačit a dále již nepřenášet*“ [7].

1.3 Kanálové kódování

Kanálové kódování (bezpečnostní kódování), kterému se budeme věnovat v této bakalářské práci, slouží k zabezpečení signálu v komunikačním kanálu (koaxiální kabel, optický kabel, Wifi). Oproti zdrojovému kódování se v kanálovém kódování úmyslně zvyšuje redundance přidáním kontrolních bitů, které slouží k ochraně dat před chybami způsobenými šumem, rušením nebo odrazy. Rozdíl oproti redundanci v zdrojovém kódování je v tom, že námi přidaná redundance u bezpečnostního kódování má specifické vlastnosti, které slouží k ochraně informace při přenosu (detekce či korekce chyby vzniklé při přenosu).[2]

1.4 Příklad binárního kódování

Na vstupu přijímače máme číslici 108, postupným celočíselným dělením tohoto čísla číslem 2 dostáváme výsledek, ale co je důležitější, dostáváme i zbytek po dělení, což je žádaný binární kód:

$$108 : 2 = 54(\text{zbytek} = 0)$$

$$54 : 2 = 27(\text{zbytek} = 0)$$

$$27 : 2 = 13(\text{zbytek} = 1)$$

$$13 : 2 = 6(\text{zbytek} = 1)$$

$$6 : 2 = 3(\text{zbytek} = 0)$$

$$3 : 2 = 1(\text{zbytek} = 1)$$

$$(\text{zbytek} = 1)$$

Hledané kódované slovo je tedy:

$$(1101100)_2$$

1.5 Parametry kódování

Každý kód má svoji délku, kterou označíme k . Délka kódu je dána vztahem:

$$k = 2^m, \tag{1.2}$$

kde m odpovídá počtu informačních bitů, tedy počtu bitů, které nesou příslušnou informaci.

Dalším parametrem kódu je Hammingova váha w , což je počet jedniček v kódovém slovu [8], např.:

$$A = 01101110 \Rightarrow w = 5$$

Při kódování se mohou vyskytnout chyby při přenosu. Pomocí Hammingovy vzdálenosti je možné zjistit kolik chyb je daný kód schopen odhalit nebo opravit.

Hammingova vzdálenost, udává počet bitů, které se musí změnit, aby se z jednoho kódového slova stalo jiné kódové slovo, např.:

$$A = 01101110$$

$$B = 11011011$$

Hammingova vzdálenost se vyjadřuje vztahem:

$$d_{\min}(A, B) = \sum_{i=1}^l |A_i - B_i| \quad (1.3)$$

Parametry A_i a B_i jsou prvky vektorů A a B .

$$d_{\min}(A, B) = \sum_{i=1}^l |A_i - B_i| = 1 + 0 + 1 + 1 + 0 + 1 + 0 + 1 = 5 \quad (1.4)$$

Hammingova vzdálenost kódu A od kódu B je 5. Jinak řečeno, v kódu A se musí změnit celkem 5 bitů, aby odpovídal kódu B .

Hammingova vzdálenost je dále používána ve vztahu pro detekci (1.5) a korekci (1.6) chyb:

$$t \leq d_{\min} - 1 \quad (1.5)$$

$$u \leq \frac{d_{\min} - 1}{2} \quad (1.6)$$

Při zavedení Hammingovy vzdálenosti ze vztahu (1.3) platí:

$$\begin{array}{ll} t \leq d_{\min} - 1 & u \leq \frac{d_{\min} - 1}{2} \\ t \leq 5 - 1 & u \leq \frac{5 - 1}{2} \\ t \leq 4 & u \leq 2 \end{array}$$

Tento kód tedy dokáže detekovat 4 chyby a opravit chyby 2.

2 Booleovské funkce

2.1 Úvod

Booleovské funkce jsou binární funkce, které nabývají pouze hodnot 0 a 1. Při jejich tvorbě využíváme logické operace součtu OR (+), součinu AND (×) a negace NOT, které jsou popsány v následující tabulce[6]:

Tabulka 1 - přehled booleovských funkcí

A	B	OR	AND	NOT(A)
0	0	0	0	1
0	1	1	0	1
1	0	1	0	0
1	1	1	1	0

Dále platí vztahy:

$$\begin{aligned} NOT(A) &= A+1 \\ A \times A &= A \end{aligned} \tag{2.1}$$

Z předešlých vztahů lze odvodit, že negaci funkce provedeme přičtením jednotkového vektoru a vynásobením dvou stejných funkcí dostaneme tu samou funkci.

Pro lepší pochopení logických operací funkcí je zde uvedena ukázka:

Mějme funkce $f = 1011010010$ a $g = 0101010010$ na kterých jsou provedeny výše uvedené logické operace:

$$a) \quad fg = \begin{array}{r} 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \\ \times \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \\ \hline \end{array} = 0001010010$$

$$b) \quad f+g = \begin{array}{r} 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \\ + \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \\ \hline \end{array} = 1111010010$$

$$c) \quad \bar{f} = \begin{array}{r} 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \\ + \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \\ \hline \end{array} = 0100101101$$

$$d) \quad ff = \begin{array}{r} 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \\ \times \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \\ \hline \end{array} = 1011010010$$

Logických funkcí se při kódování používá při vytvoření pravdivostní tabulky. Příklad pravdivostní tabulky pro proměnné X, Y a Z :

Tabulka 2 - pravdivostní tabulka

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

2.2 Booleovský polynom

Booleovský polynom je jiné vyjádření booleovské funkce pomocí součtu a součinu jednotlivých funkcí a jejich negací.

Polynomy můžeme obecně vyjádřit vztahem

$$f(x_1 \dots x_m) = \sum_{i=0}^{2^m-1} q_i x_1^{i_1} x_2^{i_2} \dots x_m^{i_m} \quad (2.2)$$

Koeficient q nabývá hodnot 0 nebo 1 a číslo i má binární rozvoj $i_m \dots i_2 i_1$, který určuje mocnitele.[1]

Mocnitéle těchto polynomů mohou mít hodnotu maximálně 1, podle vzorce:

$$x_i^n = x_i \quad (2.3)$$

Pro $m = 4$ bude mít tedy polynom rozvoj:

$$\begin{aligned} f(x_1, x_2, x_3, x_4) = & q_0 x_1^0 x_2^0 x_3^0 x_4^0 + q_1 x_1^1 x_2^0 x_3^0 x_4^0 + q_2 x_1^0 x_2^1 x_3^0 x_4^0 + q_3 x_1^1 x_2^1 x_3^0 x_4^0 + q_4 x_1^0 x_2^0 x_3^1 x_4^0 + \\ & q_5 x_1^1 x_2^0 x_3^1 x_4^0 + q_6 x_1^0 x_2^1 x_3^1 x_4^0 + q_7 x_1^1 x_2^1 x_3^1 x_4^0 + q_8 x_1^0 x_2^0 x_3^0 x_4^1 + q_9 x_1^1 x_2^0 x_3^0 x_4^1 + \\ & q_{10} x_1^0 x_2^1 x_3^0 x_4^1 + q_{11} x_1^1 x_2^1 x_3^0 x_4^1 + q_{12} x_1^0 x_2^0 x_3^1 x_4^1 + q_{13} x_1^1 x_2^0 x_3^1 x_4^1 + q_{14} x_1^0 x_2^1 x_3^1 x_4^1 + \\ & q_{15} x_1^1 x_2^1 x_3^1 x_4^1 = \\ & q_0 + q_1 x_1 + q_2 x_2 + q_3 x_1 x_2 + q_4 x_3 + q_5 x_1 x_3 + q_6 x_2 x_3 + q_7 x_1 x_2 x_3 + q_8 x_4 + \\ & q_9 x_1 x_4 + q_{10} x_2 x_4 + q_{11} x_1 x_2 x_4 + q_{12} x_3 x_4 + q_{13} x_1 x_3 x_4 + q_{14} x_2 x_3 x_4 + q_{15} x_1 x_2 x_3 x_4 \end{aligned} \quad (2.4)$$

2.3 Booleovská množina

Pro každé číslo $i = 0, 1, \dots, 2m-1$ označuje $M(i)$ množinu těchto čísel, která ve svém binárním rozvoji má jedničku jen tam, kde ji má číslo i . Takto jsou definovány množiny, použité pro dekódování námi zakódovaného slova:

$$\begin{array}{ll}
 M(0) = \{0\} & M(8) = \{0; 8\} \\
 M(1) = \{0; 1\} & M(9) = \{0; 1; 8; 9\} \\
 M(2) = \{0; 2\} & M(10) = \{0; 2; 8; 10\} \\
 M(3) = \{0; 1; 2; 3\} & M(11) = \{0; 1; 2; 3; 8; 9; 10; 11\} \\
 M(4) = \{0; 4\} & M(12) = \{0; 4; 8; 12\} \\
 M(5) = \{0; 1; 4; 5\} & M(13) = \{0; 1; 4; 5; 8; 9; 12; 13\} \\
 M(6) = \{0; 2; 4; 6\} & M(14) = \{0; 2; 4; 6; 8; 10; 12; 14\} \\
 M(7) = \{0; 1; 2; 3; 4; 5; 6; 7\} & M(15) = \{0; 1; 2; 3; 4; 5; 6; 7; 8; 9; 10; 11; 12; 13; 14; 15\}
 \end{array}$$

Například pro množinu $M(6)$ platí:

$$\begin{array}{llll}
 0 & 0 & 0 & 0=0 \\
 0 & 0 & 1 & 0=2 \\
 0 & 1 & 0 & 0=4 \\
 0 & 1 & 1 & 0=6
 \end{array}$$

Číslo 0 se využívá u všech množin jako neutrální prvek, naopak číslo 2 má jedničku ve stejném místě jako číslo 6 a číslo 4 má také jedničku na stejném místě jako číslo 6. Proto se tedy množina $M(6)$ dá zapsat jako $\{0; 2; 4; 6\}$.

Prvky množin se při dekódování používají jako indexy pořadí bitů v zakódovaném slově. Bity na příslušných místech se poté binárně sečtou a porovnají s ostatními výsledky pro danou množinu. Pokud jsou všechny výsledky stejné, tedy rovnají se 1 nebo 0, rovná se i příslušný koeficient polynomu danému číslu. Pokud se některý výsledek liší, využívá se většinového kritéria. Pokud je počet nul a jedniček ve výsledcích stejný je o hodnotě koeficientu rozhodnuto pomocí vzorce:

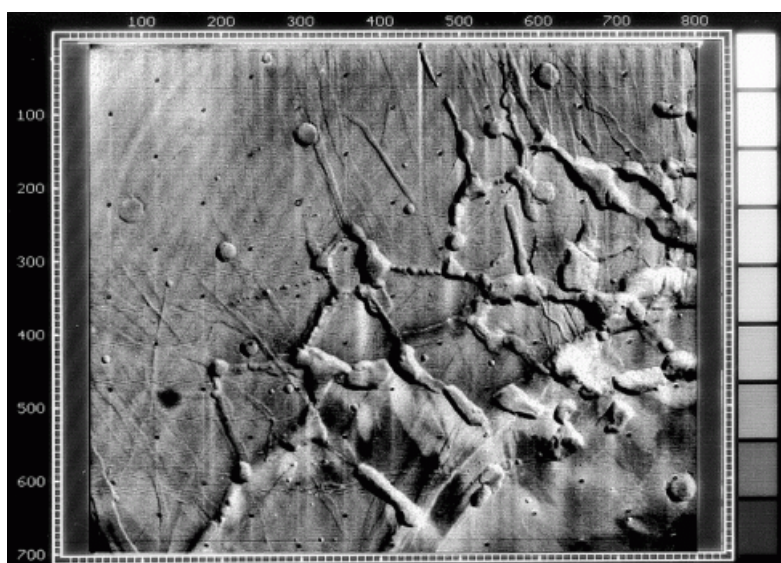
$$q_i = \sum_{j \in M(i)} w_j \quad (2.5)$$

Jinými slovy, binárně sečteme bity na příslušných j -*pozicích* matice $M(i)$, kde w_j je bit na příslušné pozici.

3 Reed-Mullerovo kódování

Reed-Mullerovy kódy patří mezi často používané kódování. Kódování objevil/vynalezl David E. Muller v roce 1954 a zdokonaleny byly téhož roku s pomocí Irvinga S. Reeda. Původně byly zaměřeny pouze na binární kódování, ale dnes je již dokážeme použít pro víceznakové kódy (např. hexadecimální).

V roce 1972 bylo Reed-Mullerovo kódování použito k přenosu fotografií Marsu ze sondy Mariner 9. Tato sonda během svého letu pořídila 7329 fotografií celého povrchu a tím dokázala, že se jedná o neobydlenou planetu bez zásob vody.



Obrázek 1 - fotografie Marsu přenesená pomocí Reed-Mullerova kódování

Reed-Mullerův kód $R(r, m)$ existuje pro každé kladné celé číslo m a každé celé číslo $r \geq m$. Jedná se o kód r -tého stupně s 2^m znaků.

Při $r = 1$ se jedná o polynom prvního stupně, takový to polynom může vypadat následovně:

$$\begin{aligned} & x_2 \\ & x_1 + x_2 \end{aligned} \tag{3.1}$$

Pro $r = 2$, tedy polynom druhého stupně, platí například rovnice tvaru:

$$x_1 x_2 \tag{3.2}$$

Následující tabulka uvádí všechny kódy $R(r,2)$, což jsou polynomy opakovacích kódů $R(-1,2)$ a $R(0,2)$, polynomy prvního stupně $R(1,2)$ a polynomy druhého stupně $R(2,2)$ [1]:

Tabulka 3 - Reed-Mullerovy kódy $R(r,2)$

0	0	0	0	0	$R(-1,2)$
1	1	1	1	1	$R(0,2)$
x_1	0	1	0	1	$R(1,2)$
x_2	1	0	0	1	
x_1+x_2	1	1	0	0	
$1+x_1$	1	0	1	0	
$1+x_2$	0	1	1	0	
$1+x_1+x_2$	0	0	1	1	
x_1x_2	0	0	0	1	
$1+x_1x_2$	1	1	1	0	
$x_1+x_1x_2$	0	1	0	0	
$x_2+x_1x_2$	1	0	0	0	
$x_1+x_2+x_1x_2$	1	1	0	1	
$1+x_1+x_1x_2$	1	0	1	1	
$1+x_2+x_1x_2$	0	1	1	1	
$1+x_1+x_2+x_1x_2$	0	0	1	0	

Mezi parametry Reed-Mullerových kódů $R(r,m)$ patří počet informačních znaků k a minimální vzdálenost d_{\min} . Pro zjištění počtu informačních znaků slouží vzorec:

$$k = \binom{m}{0} + \binom{m}{1} + \dots + \binom{m}{r} \quad (3.3)$$

Pomocí minimální kódové vzdálenosti uvedené ve vztahu (3.4) lze stanovit počet detekovatelných a opravitelných chyb

$$d_{\min} = 2^{m-r} \quad (3.4)$$

Počet detekovatelných chyb (3.5) a počet opravitelných chyb (3.6) jsou dány vztahy:

$$t < 2^{m-r}, \quad (3.5)$$

$$u < 2^{m-r-1} \quad (3.6)$$

Kódování

Princip kódování lze rozdělit do níže uvedených kroků:

- a) Zavedení vstupního informačního slova
- b) Vytvoření generující matice
- c) Zakódování vstupního slova pomocí generující matice

Na vstup kodéru je přivedeno informační slovo l , délky k , se kterým je dále pracováno.

Při kódování se používá generující matice. Jedná se o soustavu vektorů, generující všechna možná kódová slova. Matice vznikne z libovolného lineárního kódu $R(r, m)$, jako matice

o r^m sloupců a $\binom{m}{r} + \binom{m}{r-1} + \dots + \binom{m}{0}$ řádků.

$$\mathbf{G}_{R(r,m)} = \begin{bmatrix} 0 \\ x_1 \\ x_2 \\ \vdots \\ x_m \\ x_1x_2 \\ x_1x_3 \\ \vdots \\ x_{m-1}x_m \\ x_1x_2x_3 \\ \vdots \\ x_{m-r+1}x_{m-r+2} \cdots x_m \end{bmatrix} \quad (3.7)$$

Matice pro kód $R(2,3)$ vypadá následovně[10]:

$$\mathbf{G} = \begin{bmatrix} 0 \\ x_1 \\ x_2 \\ x_3 \\ x_1x_2 \\ x_1x_3 \\ x_2x_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \quad (3.8)$$

První řádek matice je tvořen počtem 2^m jedniček. Další tři řádky jsou tvořeny pomocí jednotlivých sloupců, tak že každý sloupec je tvořen binárním vyjádřením svého pořadí, počínaje nultým sloupcem. Poslední 3 řádky jsou zapsány jako součiny předešlých řádků.

Pomocí této generující matice již vytvoříme z binárního slova \mathbf{l} , délky k , zakódované slovo \mathbf{v} , délky n :

$$\mathbf{v} = \mathbf{G} \times \begin{bmatrix} l_6 \\ l_5 \\ l_4 \\ l_3 \\ l_2 \\ l_1 \\ l_0 \end{bmatrix} = \begin{bmatrix} v_6 \\ v_5 \\ v_4 \\ v_3 \\ v_2 \\ v_1 \\ v_0 \end{bmatrix} \quad (3.9)$$

Dekódování

- a) Přijmutí zakódovaného slova
- b) Zavedení chyb do slova
- c) Zjištění polynomů zakódovaného slova
- d) Dekódování pomocí většinové logiky
- e) Porovnání výsledku dekodéru se vstupem na kodér

Na vstupu dekodéru je zakódované slovo \mathbf{w} , délky n . Během přenosu se ovšem může vyskytnout chyba, tuto chybu nazýváme chybným vektorem \mathbf{e} . Chybný vektor je vyjádřen vztahem[9]:

$$\begin{aligned} \mathbf{w} - \mathbf{e} &= \mathbf{v} \\ (w_k w_{k-1} \cdots w_0) - (e_k e_{k-1} \cdots e_0) &= (v_k v_{k-1} \cdots v_0) \end{aligned} \quad (3.10)$$

Parametr k značí maximální délku vektoru.

Jak již bylo uvedeno v kapitole 2.2 a 2.3 pro dekódování se využívá booleovských polynomů a množin. Každý polynom má koeficient, který se musí pomocí množin vypočítat.

Pro každý Reed-Mullerův kód $R(r, m)$ existují polynomy r -tého až 1.stupně.

Pro Reed-Mullerův kód $R(2, 3)$ existují polynomy druhého a prvního stupně s odpovídajícími koeficienty:

$$\begin{aligned} q_1 \times x_1 \\ q_2 \times x_2 \\ q_4 \times x_3 \\ q_3 \times x_1 x_2 \\ q_5 \times x_1 x_3 \\ q_6 \times x_2 x_3 \end{aligned} \quad (3.11)$$

Například pro výpočet hodnoty koeficient q_6 se využívá množiny $M(6) = \{0; 2; 4; 6\}$ a k ní odpovídající množiny $M(1) = \{0; 1\}$ podle vzorce:

$$M((2^m - 1) - 6) \quad (3.12)$$

Parametr m vychází ze zadání $R(r, m)$ a číslo 6 je hodnota matice $M(6)$.

Následující výpočet má zápis:

$$\begin{aligned} w_0 + w_2 + w_4 + w_6 &= q_{6_1} \\ w_1 + w_3 + w_5 + w_7 &= q_{6_2} \end{aligned} \quad (3.13)$$

Do prvního řádku se zapíše jednotlivé bity zakódovaného slova \mathbf{w} , jejichž indexy odpovídají hodnotám množiny $M(6)$ a do druhého řádku se postupně zapisují bity, jejichž indexy jsou zvětšeny o hodnotu množiny $M(1)$.

Hodnota koeficientu se poté určí z výsledků jednotlivých řádků, jak bylo uvedeno v kapitole 2.3.

Po vypočítání všech hodnot koeficientů polynomů, dojde k vynásobení příslušných koeficientů s polynomy a poté se polynomy přičtou k zakódovanému slovu \mathbf{w} .

Takto se pokračuje, dokud nejsou vypočítány všechny koeficienty kromě koeficientu q_0 , který se určí pomocí většinového kritéria z výsledného slova \mathbf{w} , které vznikne sečtením zakódovaného slova \mathbf{w} a všech vynásobených polynomů.

Po zjištění všech koeficientů dojde k sestavení dekódovaného slova pomocí hodnot koeficientů. Pro kód $R(2,3)$ je pořadí následující:

$$q_6 \quad q_5 \quad q_3 \quad q_4 \quad q_2 \quad q_1 \quad q_0 \quad (3.14)$$

4 Příklad Reed-Mullerova kódování/dekódování

V této kapitole dojde k praktickému kódování a dekódování Reed-Mullerova kódu s ukázkou dekódování bez chyb, s opravitelnými chybami a s chybami neopravitelnými.

Vstupní informace:

l - vstupní kód do kodéru a výstupní kód z dekodéru

v - výstupní kód z kodéru

w - vstupní kód do dekodéru

e - chybový vektor přenosu

4.1 Zadání

Mějme informační slovo tvaru:

$$l = 00111000101$$

Toto informační slovo je délky $k = 11$, proto bude nejvhodnější zvolit RM kód $R(2,4)$, který odpovídá právě této délce, podle vztahu:

$$k = \binom{4}{2} + \binom{4}{1} + \binom{4}{0} = 6 + 4 + 1 = 11$$

Počet chyb, které dokážeme tímto RM kódem objevit:

$$\begin{aligned} t &< 2^{(m-r)} \\ t &< 2^{(4-2)} \\ t &< 4 \end{aligned} \tag{4.1}$$

Počet chyb, kolik jich dokáže opravit:

$$\begin{aligned} u &< 2^{(m-r-1)} \\ u &< 2^{(4-2-1)} \\ u &< 2 \end{aligned} \tag{4.2}$$

4.2 Kódování

Pro kód velikosti $R(2,4)$ existuje kódovací matice velikosti 16 sloupců a 11 řádků:

$$\mathbf{G} = \begin{bmatrix} 0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_1x_2 \\ x_1x_3 \\ x_1x_4 \\ x_2x_3 \\ x_2x_4 \\ x_3x_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Touto generující maticí se vynásobí zvolený kód:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \times \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

Kód o velikosti 1×11 se postupně násobí s jednotlivými sloupci matice 11×16 a vznikne kód s 16 bity.

Výpočet probíhá jako postupné násobení zvoleného kódu a jednotlivých sloupců matice.

$$\begin{aligned}
 v_0 &= (1 \times 0) + (1 \times 0) + (1 \times 1) + (1 \times 1) + (1 \times 1) + (1 \times 0) + (1 \times 0) + (1 \times 0) + (1 \times 1) + (1 \times 0) + (1 \times 1) = 1 \\
 v_1 &= (1 \times 0) + (1 \times 0) + (1 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 0) + (1 \times 0) + (1 \times 0) + (1 \times 1) + (0 \times 0) + (1 \times 1) = 1 \\
 v_2 &= (1 \times 0) + (0 \times 0) + (0 \times 1) + (1 \times 1) + (1 \times 1) + (0 \times 0) + (1 \times 0) + (1 \times 0) + (0 \times 1) + (1 \times 0) + (1 \times 1) = 1 \\
 v_3 &= (1 \times 0) + (0 \times 0) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 0) + (1 \times 0) + (1 \times 0) + (0 \times 1) + (0 \times 0) + (1 \times 1) = 1 \\
 v_4 &= (0 \times 0) + (1 \times 0) + (0 \times 1) + (1 \times 1) + (0 \times 1) + (1 \times 0) + (1 \times 0) + (0 \times 0) + (1 \times 1) + (1 \times 0) + (1 \times 1) = 1 \\
 v_5 &= (0 \times 0) + (1 \times 0) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 0) + (1 \times 0) + (0 \times 0) + (1 \times 1) + (0 \times 0) + (1 \times 1) = 0 \\
 v_6 &= (0 \times 0) + (0 \times 0) + (0 \times 1) + (1 \times 1) + (0 \times 1) + (0 \times 0) + (1 \times 0) + (0 \times 0) + (0 \times 1) + (1 \times 0) + (1 \times 1) = 0 \\
 v_7 &= (0 \times 0) + (0 \times 0) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 0) + (1 \times 0) + (0 \times 0) + (0 \times 1) + (0 \times 0) + (1 \times 1) = 1 \\
 v_8 &= (0 \times 0) + (0 \times 0) + (1 \times 1) + (0 \times 1) + (1 \times 1) + (1 \times 0) + (0 \times 0) + (1 \times 0) + (1 \times 1) + (1 \times 0) + (1 \times 1) = 0 \\
 v_9 &= (0 \times 0) + (0 \times 0) + (1 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 0) + (0 \times 0) + (1 \times 0) + (1 \times 1) + (0 \times 0) + (1 \times 1) = 1 \\
 v_{10} &= (0 \times 0) + (0 \times 0) + (0 \times 1) + (0 \times 1) + (1 \times 1) + (0 \times 0) + (0 \times 0) + (1 \times 0) + (0 \times 1) + (1 \times 0) + (1 \times 1) = 0 \\
 v_{11} &= (0 \times 0) + (0 \times 0) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 0) + (0 \times 0) + (1 \times 0) + (0 \times 1) + (0 \times 0) + (1 \times 1) = 1 \\
 v_{12} &= (0 \times 0) + (0 \times 0) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (1 \times 0) + (0 \times 0) + (0 \times 0) + (1 \times 1) + (1 \times 0) + (1 \times 1) = 0 \\
 v_{13} &= (0 \times 0) + (0 \times 0) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (1 \times 1) + (0 \times 0) + (1 \times 1) = 0 \\
 v_{14} &= (0 \times 0) + (0 \times 0) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 1) + (1 \times 0) + (1 \times 1) = 1 \\
 v_{15} &= (0 \times 0) + (0 \times 0) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 1) + (0 \times 0) + (1 \times 1) = 1
 \end{aligned}$$

Při vynásobení kódu s posledním sloupcem matice vyjde LSB (Least Significant Bit), tedy nejméně významný bit slova v . Takto pokračuje výpočet až po MSB (Most Significant Bit), nejvýznamnější bit daného slova, tedy vynásobený kód s prvním sloupcem matice.

Výsledné slovo v , tedy výsledek kódování je:

$$v = 1100101010011111$$

4.3 Dekódování bez chyb

V této části dojde k dekodování slova w bez chyb způsobených přenosem. Výsledný kód kodéru, chybný vektor přenosu a dekodované slovo, mají tedy tvar:

$$\begin{array}{rcl}
 v & & 1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \\
 e & + & 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \\
 w & = & 1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1
 \end{array}$$

Kód $R(2,4)$ obsahuje 6 polynomů 2. stupně:

$$q_3 \times x_1 x_2$$

$$q_5 \times x_1 x_3$$

$$q_6 \times x_2 x_3$$

$$q_9 \times x_1 x_4$$

$$q_{10} \times x_2 x_4$$

$$q_{12} \times x_3 x_4$$

A 4 polynomy 1. stupně:

$$q_1 \times x_1$$

$$q_2 \times x_2$$

$$q_4 \times x_3$$

$$q_8 \times x_4$$

Výpočet koeficientů probíhá jako binární sčítání jednotlivých bitů slova w . K tomuto účelu slouží množiny $M(i)$, jejich účel byl vysvětlen v kapitole 2.2.

Výpočet koeficientů polynomů 2. stupně

Pro koeficient q_3 se využívá booleovské množiny $M(3) = \{0; 1; 2; 3\}$ zároveň s množinou $M((2^m - 1) - 3)$, kde m vychází z $R(r, m)$ a číslo 3 je značení množiny $M(3)$. Druhou množinou je tedy množina $M(12) = \{0, 4, 8, 12\}$. Tvorba množin je vysvětlena v kapitole 3. Množina $M(3)$ se zapíše do prvního řádku. Množina $M(12)$ se zapíše do prvního sloupce. Prvek na pozici druhý řádek a druhý sloupec vznikne sečtením druhého čísla v množině $M(3)$ a $M(12)$. Takto se pokračuje, dokud se nezjistí všechny prvky výpočtu:

$$w_0 + w_1 + w_2 + w_3 = 1 + 1 + 1 + 1 = 0$$

$$w_4 + w_5 + w_6 + w_7 = 1 + 0 + 0 + 1 = 0$$

$$w_8 + w_9 + w_{10} + w_{11} = 0 + 1 + 0 + 1 = 0$$

$$w_{12} + w_{13} + w_{14} + w_{15} = 0 + 0 + 1 + 1 = 0$$

Všechny řádky vyšly rovny 0, tedy:

$$q_3 = 0$$

Pro výpočet hodnoty koeficientu q_5 se využívají matice $M(5) = \{0; 1; 4; 5\}$ a $M(10) = \{0; 2; 8; 10\}$.

$$\begin{aligned} w_0 + w_1 + w_4 + w_5 &= 1 + 1 + 1 + 0 = 1 \\ w_2 + w_3 + w_6 + w_7 &= 1 + 1 + 0 + 1 = 1 \\ w_8 + w_9 + w_{12} + w_{13} &= 0 + 1 + 0 + 0 = 1 \\ w_{10} + w_{11} + w_{14} + w_{15} &= 0 + 1 + 1 + 1 = 1 \end{aligned} \Rightarrow q_5 = 1$$

Pro výpočet koeficientu q_6 se využívají matice $M(6) = \{0; 2; 4; 6\}$ a $M(9) = \{0; 1; 8; 9\}$.

$$\begin{aligned} w_0 + w_2 + w_4 + w_6 &= 1 + 1 + 1 + 0 = 1 \\ w_1 + w_3 + w_5 + w_7 &= 1 + 1 + 0 + 1 = 1 \\ w_8 + w_{10} + w_{12} + w_{14} &= 0 + 0 + 0 + 1 = 1 \\ w_9 + w_{11} + w_{13} + w_{15} &= 1 + 1 + 0 + 1 = 1 \end{aligned} \Rightarrow q_6 = 1$$

Pro výpočet hodnoty koeficientu q_9 se využívají matice $M(9) = \{0; 1; 8; 9\}$ a $M(6) = \{0; 2; 4; 6\}$.

$$\begin{aligned} w_0 + w_1 + w_8 + w_9 &= 1 + 1 + 0 + 1 = 1 \\ w_2 + w_3 + w_{10} + w_{11} &= 1 + 1 + 0 + 1 = 1 \\ w_4 + w_5 + w_{12} + w_{13} &= 1 + 0 + 0 + 0 = 1 \\ w_6 + w_7 + w_{14} + w_{15} &= 0 + 1 + 1 + 1 = 1 \end{aligned} \Rightarrow q_9 = 1$$

Pro výpočet hodnoty koeficientu q_{10} se využívají matice $M(10) = \{0; 2; 8; 10\}$ a $M(5) = \{0; 1; 4; 5\}$.

$$\begin{aligned} w_0 + w_2 + w_8 + w_{10} &= 1 + 1 + 0 + 0 = 0 \\ w_1 + w_3 + w_9 + w_{11} &= 1 + 1 + 1 + 1 = 0 \\ w_4 + w_6 + w_{12} + w_{14} &= 1 + 0 + 0 + 1 = 0 \\ w_5 + w_7 + w_{13} + w_{15} &= 0 + 1 + 0 + 1 = 0 \end{aligned} \Rightarrow q_{10} = 0$$

Pro výpočet hodnoty koeficientu q_{12} se využívají matice $M(12) = \{0; 4; 8; 12\}$ a $M(3) = \{0; 1; 2; 3\}$.

$$\begin{aligned} w_0 + w_4 + w_8 + w_{12} &= 1 + 1 + 0 + 0 = 0 \\ w_1 + w_5 + w_9 + w_{13} &= 1 + 0 + 1 + 0 = 0 \\ w_2 + w_6 + w_{10} + w_{14} &= 1 + 0 + 0 + 1 = 0 \\ w_3 + w_7 + w_{11} + w_{15} &= 1 + 1 + 1 + 1 = 0 \end{aligned} \Rightarrow q_{12} = 0$$

Když jsou zjištěny všechny koeficienty polynomů 2. stupně, dojde k jejich vynásobení s polynomy, ke kterým náleží:

$$q_3 \times x_1x_2 = 0 \times (0001000100010001)$$

$$q_5 \times x_1x_3 = 1 \times (0000010100000101)$$

$$q_6 \times x_2x_3 = 1 \times (0000001100000011)$$

$$q_9 \times x_1x_4 = 1 \times (0000000001010101)$$

$$q_{10} \times x_2x_4 = 0 \times (0000001100000011)$$

$$q_{12} \times x_3x_4 = 0 \times (0000000000001111)$$

V dalším kroku se výsledek z předchozího kroku přičte ke slovu w :

$$\begin{array}{rcl}
 \mathbf{w} & = & 1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \\
 x_1x_2 & + & 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \\
 x_1x_3 & + & 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \\
 x_2x_3 & + & 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \\
 x_1x_4 & + & 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \\
 x_2x_4 & + & 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \\
 x_3x_4 & + & 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0
 \end{array}$$

$$\mathbf{w}' = 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0$$

Slovo w' se dále používá místo slova w pro výpočet koeficientů polynomů 1. stupně.

Výpočet koeficientů polynomů 1. stupně

Pro výpočet hodnoty koeficientu q_1 se využívají matice $M(1) = \{0;1\}$ a $M(14) = \{0;2;4;6;8;10;12;14\}$.

$$\begin{aligned}w_0' + w_1' &= 0 + 0 = 0 \\w_2' + w_3' &= 1 + 1 = 0 \\w_4' + w_5' &= 0 + 0 = 0 \\w_6' + w_7' &= 1 + 1 = 0 \\w_8' + w_9' &= 0 + 0 = 0 \\w_{10}' + w_{11}' &= 1 + 1 = 0 \\w_{12}' + w_{13}' &= 0 + 0 = 0 \\w_{14}' + w_{15}' &= 1 + 1 = 0\end{aligned} \Rightarrow q_1 = 0$$

Pro výpočet hodnoty koeficientu q_2 se využívají matice $M(2) = \{0;2\}$ a $M(13) = \{0;1;4;5;8;9;12;13\}$.

$$\begin{aligned}w_0' + w_2' &= 0 + 1 = 1 \\w_1' + w_3' &= 0 + 1 = 1 \\w_4' + w_6' &= 0 + 1 = 1 \\w_5' + w_7' &= 0 + 1 = 1 \\w_8' + w_{10}' &= 0 + 1 = 1 \\w_9' + w_{11}' &= 0 + 1 = 1 \\w_{12}' + w_{14}' &= 0 + 1 = 1 \\w_{13}' + w_{15}' &= 0 + 1 = 1\end{aligned} \Rightarrow q_2 = 1$$

Pro výpočet hodnoty koeficientu q_4 se využívají matice $M(4) = \{0;4\}$ a $M(11) = \{0;1;2;3;8;9;10;11\}$.

$$\begin{aligned}w_0' + w_4' &= 0 + 0 = 0 \\w_1' + w_5' &= 0 + 0 = 0 \\w_2' + w_6' &= 1 + 1 = 0 \\w_3' + w_7' &= 1 + 1 = 0 \\w_8' + w_{12}' &= 0 + 0 = 0 \\w_9' + w_{13}' &= 0 + 0 = 0 \\w_{10}' + w_{14}' &= 1 + 1 = 0 \\w_{11}' + w_{15}' &= 1 + 1 = 0\end{aligned} \Rightarrow q_4 = 0$$

Pro výpočet hodnoty koeficientu q_8 se využívají matice $M(8) = \{0; 8\}$ a $M(7) = \{0; 1; 2; 3; 4; 5; 6; 7\}$.

$$\begin{aligned}
 w_0' + w_8' &= 0 + 0 = 0 \\
 w_1' + w_9' &= 0 + 0 = 0 \\
 w_2' + w_{10}' &= 1 + 1 = 0 \\
 w_3' + w_{11}' &= 1 + 1 = 0 \\
 w_4' + w_{12}' &= 0 + 0 = 0 \\
 w_5' + w_{13}' &= 0 + 0 = 0 \\
 w_6' + w_{14}' &= 1 + 1 = 0 \\
 w_7' + w_{15}' &= 1 + 1 = 0
 \end{aligned}
 \Rightarrow q_8 = 0$$

Po zjištění všech koeficientů polynomů 1. stupně se koeficienty vynásobí s polynomy, kterým náleží:

$$\begin{aligned}
 q_1 \times x_1 &= 0 \times (0101010101010101) \\
 q_2 \times x_2 &= 1 \times (0011001100110011) \\
 q_4 \times x_3 &= 0 \times (0000111100001111) \\
 q_8 \times x_4 &= 0 \times (0000000011111111)
 \end{aligned}$$

V dalším kroku se výsledek z předchozího kroku přičte ke slovu \mathbf{w}' :

$$\begin{array}{rcccccccccccccccc}
 \mathbf{w}' & = & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
 x_1 & + & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 x_2 & + & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
 x_4 & + & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 x_8 & + & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 \\
 \mathbf{w}'' & = & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
 \end{array}$$

Nyní se ze slova \mathbf{w}'' , podle většinového kritéria zjistí hodnota koeficientu q_0 :

$$q_0 = 1$$

Výsledek dekodování

Pokud jsou známy všechny koeficienty polynomů, je možné přejít k určení přenášeného kódu.

Koeficienty se seřadí podle své váhy v pořadí, odpovídajícímu řádku matice, ve kterém se polynomy těchto koeficientů nacházejí.

Výsledný kód po dekodování je:

$$\begin{array}{cccccccccccc} q_{12} & q_{10} & q_6 & q_9 & q_5 & q_3 & q_8 & q_4 & q_2 & q_1 & q_0 \\ \mathbf{1} & = & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \end{array}$$

Výsledný kód se rovná kódu, který byl na vstupu. Kodování i dekodování proběhlo bez chyb.

4.4 Dekódování s opravitelnými chybami

V této kapitole dojde k ukázce dekódování při zavedení opravitelného počtu chyb do kódového slova. Původní zakódované slovo bylo:

$$\mathbf{v} = 1100101010011111$$

Kód $R(2,4)$ obsahuje 1 opravitelnou chybu, podle vzorce (4.1). Tudíž nové slovo \mathbf{w} se může lišit pouze v jednom bitu, zvolme např. slovo:

$$\mathbf{w} = 1100101010010111$$

Nové slovo \mathbf{w} se liší ve 4. bitu od LSB, chybný vektor má tedy tvar:

$$\mathbf{e} = 0000000000001000$$

Výpočet koeficientů polynomů 2. stupně

Dekódování probíhá stejně jako při dekódování bez chyb. Pro výpočet jednotlivých koeficientů se používá shodného postupu (se shodnými množinami).

Výpočet hodnoty koeficientu q_3 :

$$\begin{aligned}w_0 + w_1 + w_2 + w_3 &= 1 + 1 + 1 + 0 = 1 \\w_4 + w_5 + w_6 + w_7 &= 1 + 0 + 0 + 1 = 0 \\w_8 + w_9 + w_{10} + w_{11} &= 0 + 1 + 0 + 1 = 0 \\w_{12} + w_{13} + w_{14} + w_{15} &= 0 + 0 + 1 + 1 = 0\end{aligned}$$

Kvůli zavedení chyby se jeden řádek výpočtu liší, v tomto případě se využívá většinového kritéria, tedy:

$$q_3 = 0$$

Pro další koeficienty, již bude uveden postup bez popisu.

Výpočet hodnoty koeficientu q_5 :

$$\begin{aligned}w_0 + w_1 + w_4 + w_5 &= 1 + 1 + 1 + 0 = 1 \\w_2 + w_3 + w_6 + w_7 &= 1 + 0 + 0 + 1 = 0 \\w_8 + w_9 + w_{12} + w_{13} &= 0 + 1 + 0 + 0 = 1 \\w_{10} + w_{11} + w_{14} + w_{15} &= 0 + 1 + 1 + 1 = 1\end{aligned} \Rightarrow q_5 = 1$$

Výpočet hodnoty koeficientu q_6 :

$$\begin{aligned}w_0 + w_2 + w_4 + w_6 &= 1 + 1 + 1 + 0 = 1 \\w_1 + w_2 + w_5 + w_7 &= 1 + 0 + 0 + 1 = 0 \\w_8 + w_{10} + w_{12} + w_{14} &= 0 + 0 + 0 + 1 = 1 \\w_9 + w_{11} + w_{13} + w_{15} &= 1 + 1 + 0 + 1 = 1\end{aligned} \Rightarrow q_6 = 1$$

Výpočet hodnoty koeficientu q_9 :

$$\begin{aligned}w_0 + w_1 + w_8 + w_9 &= 1 + 1 + 0 + 1 = 1 \\w_2 + w_3 + w_{10} + w_{11} &= 1 + 0 + 0 + 1 = 0 \\w_4 + w_5 + w_{12} + w_{13} &= 1 + 0 + 0 + 0 = 1 \\w_6 + w_7 + w_{14} + w_{15} &= 0 + 1 + 1 + 1 = 1\end{aligned} \Rightarrow q_9 = 1$$

Výpočet hodnoty koeficientu q_{10} :

$$\begin{aligned}w_0 + w_2 + w_8 + w_{10} &= 1 + 1 + 0 + 0 = 0 \\w_1 + w_3 + w_9 + w_{11} &= 1 + 0 + 1 + 1 = 1 \\w_4 + w_6 + w_{12} + w_{14} &= 1 + 0 + 0 + 1 = 0 \\w_5 + w_7 + w_{13} + w_{15} &= 0 + 1 + 0 + 1 = 0\end{aligned} \Rightarrow q_{10} = 0$$

Výpočet hodnoty koeficientu q_{12} :

$$\begin{aligned}w_0 + w_4 + w_8 + w_{12} &= 1 + 1 + 0 + 0 = 0 \\w_1 + w_5 + w_9 + w_{13} &= 1 + 0 + 1 + 0 = 0 \\w_2 + w_6 + w_{10} + w_{14} &= 1 + 0 + 0 + 1 = 0 \\w_3 + w_7 + w_{11} + w_{15} &= 0 + 1 + 1 + 1 = 1\end{aligned} \Rightarrow q_{12} = 0$$

Po zjištění všech koeficientů polynomů 2. stupně se koeficienty vynásobí s polynomy, kterým náleží:

$$\begin{aligned}q_3 \times x_1 x_2 &= 0 \times (0001000100010001) \\q_5 \times x_1 x_3 &= 1 \times (0000010100000101) \\q_6 \times x_2 x_3 &= 1 \times (0000001100000011) \\q_9 \times x_1 x_4 &= 1 \times (0000000001010101) \\q_{10} \times x_2 x_4 &= 0 \times (0000001100000011) \\q_{12} \times x_3 x_4 &= 0 \times (0000000000001111)\end{aligned}$$

V dalším kroku se výsledek z předchozího kroku přičte ke slovu w :

$$\begin{array}{rcl}
 w & = & 1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1 \\
 x_1x_2 & + & 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \\
 x_1x_3 & + & 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \\
 x_2x_3 & + & 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \\
 x_1x_4 & + & 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \\
 x_2x_4 & + & 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \\
 x_3x_4 & + & 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \\
 \\
 w' & = & 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0
 \end{array}$$

Slovo w' se dále používá místo slova w pro výpočet koeficientů polynomů 1. stupně.

Výpočet koeficientů polynomů 1. stupně

Výpočet hodnoty koeficientu q_1 :

$$\begin{array}{l}
 w_0' + w_1' = 0 + 0 = 0 \\
 w_2' + w_3' = 1 + 0 = 1 \\
 w_4' + w_5' = 0 + 0 = 0 \\
 w_6' + w_7' = 1 + 1 = 0 \\
 w_8' + w_9' = 0 + 0 = 0 \\
 w_{10}' + w_{11}' = 1 + 1 = 0 \\
 w_{12}' + w_{13}' = 0 + 0 = 0 \\
 w_{14}' + w_{15}' = 1 + 1 = 0
 \end{array} \Rightarrow q_1 = 0$$

Výpočet hodnoty koeficientu q_2 :

$$\begin{array}{l}
 w_0' + w_2' = 0 + 1 = 1 \\
 w_1' + w_3' = 0 + 0 = 0 \\
 w_4' + w_6' = 0 + 1 = 1 \\
 w_5' + w_7' = 0 + 1 = 1 \\
 w_8' + w_{10}' = 0 + 1 = 1 \\
 w_9' + w_{11}' = 0 + 1 = 1 \\
 w_{12}' + w_{14}' = 0 + 1 = 1 \\
 w_{13}' + w_{15}' = 0 + 1 = 1
 \end{array} \Rightarrow q_2 = 1$$

Výpočet hodnoty koeficientu q_4 :

$$\begin{aligned}w_0' + w_4' &= 0 + 0 = 0 \\w_1' + w_5' &= 0 + 0 = 0 \\w_2' + w_6' &= 1 + 1 = 0 \\w_3' + w_7' &= 0 + 1 = 1 \\w_8' + w_{12}' &= 0 + 0 = 0 \\w_9' + w_{13}' &= 0 + 0 = 0 \\w_{10}' + w_{14}' &= 1 + 1 = 0 \\w_{11}' + w_{15}' &= 1 + 1 = 0\end{aligned} \Rightarrow q_4 = 0$$

Výpočet hodnoty koeficientu q_8 :

$$\begin{aligned}w_0' + w_8' &= 0 + 0 = 0 \\w_1' + w_9' &= 0 + 0 = 0 \\w_2' + w_{10}' &= 1 + 1 = 0 \\w_3' + w_{11}' &= 0 + 1 = 1 \\w_4' + w_{12}' &= 0 + 0 = 0 \\w_5' + w_{13}' &= 0 + 0 = 0 \\w_6' + w_{14}' &= 1 + 1 = 0 \\w_7' + w_{15}' &= 1 + 1 = 0\end{aligned} \Rightarrow q_8 = 0$$

Po zjištění všech koeficientů polynomů 1. stupně se koeficienty vynásobí s polynomy, kterým náleží:

$$\begin{aligned}q_1 \times x_1 &= 0 \times (0101010101010101) \\q_2 \times x_2 &= 1 \times (0011001100110011) \\q_4 \times x_3 &= 0 \times (0000111100001111) \\q_8 \times x_4 &= 0 \times (0000000011111111)\end{aligned}$$

V dalším kroku se výsledek z předchozího kroku přičte ke slovu \mathbf{w}' :

$$\begin{array}{rcccccccccccccccc}
 \mathbf{w}' & = & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
 x_1 & + & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 x_2 & + & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
 x_4 & + & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 x_8 & + & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 \\
 \mathbf{w}'' & = & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1
 \end{array}$$

Nyní se ze slova \mathbf{w}'' , podle většinového kritéria zjistí hodnota koeficientu q_0 :

$$q_0 = 1$$

Výsledek dekódování s opravitelnými chybami

$$\begin{array}{cccccccccccc}
 & q_{12} & q_{10} & q_6 & q_9 & q_5 & q_3 & q_8 & q_4 & q_2 & q_1 & q_0 \\
 \mathbf{l} & = & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1
 \end{array}$$

Výsledný kód se rovná kódu, který byl na vstupu. I když přenos kódu proběhl s chybou, dekódování tuto chybu opravilo.

4.5 Dekódování bez možnosti opravení

V této kapitole dojde k ukázce dekódování při zavedení neopravitelného počtu chyb do kódového slova. Původní zakódované slovo bylo:

$$\mathbf{v} = 1100101010011111$$

Kód $R(2,4)$ obsahuje 1 opravitelnou chybu, podle vzorce (4.1) a 3 chyby zjistitelné, podle vzorce (4.2). Tudíž nové slovo \mathbf{w} se musí lišit minimálně ve 2 bitech, zvolme např. slovo:

$$\mathbf{w} = 1110101110001110$$

Nové slovo \mathbf{w} se liší ve 4 bitech od původního zakódovaného slova, chybný vektor má tedy tvar:

$$\mathbf{e} = 0010000100010001$$

Dá se tedy očekávat, že dekodér nedokáže chyby ani rozpoznat, ani opravit.

Výpočet hodnot koeficientů polynomů 2. stupně

Výpočet hodnoty koeficientu q_3 :

$$\begin{aligned}w_0 + w_1 + w_2 + w_3 &= 0 + 1 + 1 + 1 = 1 \\w_4 + w_5 + w_6 + w_7 &= 0 + 0 + 0 + 1 = 1 \\w_8 + w_9 + w_{10} + w_{11} &= 1 + 1 + 0 + 1 = 1 \\w_{12} + w_{13} + w_{14} + w_{15} &= 0 + 1 + 1 + 1 = 1\end{aligned} \Rightarrow q_3 = 1$$

Výpočet hodnoty koeficientu q_5 :

$$\begin{aligned}w_0 + w_1 + w_4 + w_5 &= 0 + 1 + 0 + 0 = 1 \\w_2 + w_3 + w_6 + w_7 &= 1 + 1 + 0 + 1 = 1 \\w_8 + w_9 + w_{12} + w_{13} &= 1 + 1 + 0 + 1 = 1 \\w_{10} + w_{11} + w_{14} + w_{15} &= 0 + 1 + 1 + 1 = 1\end{aligned} \Rightarrow q_5 = 1$$

Výpočet hodnoty koeficientu q_6 :

$$\begin{aligned}w_0 + w_2 + w_4 + w_6 &= 0 + 1 + 0 + 1 = 0 \\w_1 + w_3 + w_5 + w_7 &= 1 + 1 + 0 + 1 = 1 \\w_8 + w_{10} + w_{12} + w_{14} &= 1 + 0 + 0 + 1 = 1 \\w_9 + w_{11} + w_{13} + w_{15} &= 1 + 1 + 1 + 0 = 1\end{aligned} \Rightarrow q_6 = 1$$

Výpočet hodnoty koeficientu q_9 :

$$\begin{aligned} w_0 + w_1 + w_8 + w_9 &= 0 + 1 + 1 + 1 = 1 \\ w_2 + w_3 + w_{10} + w_{11} &= 1 + 1 + 0 + 1 = 1 \\ w_4 + w_5 + w_{12} + w_{13} &= 0 + 0 + 0 + 1 = 1 \\ w_6 + w_7 + w_{14} + w_{15} &= 0 + 1 + 1 + 1 = 1 \end{aligned} \Rightarrow q_9 = 1$$

Výpočet hodnoty koeficientu q_{10} :

$$\begin{aligned} w_0 + w_2 + w_8 + w_{10} &= 0 + 1 + 1 + 0 = 0 \\ w_1 + w_3 + w_9 + w_{11} &= 1 + 1 + 1 + 1 = 0 \\ w_4 + w_6 + w_{12} + w_{14} &= 0 + 0 + 0 + 1 = 1 \\ w_5 + w_7 + w_{13} + w_{15} &= 0 + 1 + 1 + 1 = 1 \end{aligned}$$

V tomto případě nevycházejí všechny řádky stejně a nelze využít většinového kritéria, neboť dva řádky jsou rovny 0 a dva jsou rovny 1. Proto se využívá vzorce (2.5):

$$q_{10} = \sum_{j \in M(10)} w_j = w_0 + w_2 + w_8 + w_{10} = 0 + 1 + 1 + 0 = 0$$

Výpočet hodnoty koeficientu q_{12} :

$$\begin{aligned} w_0 + w_4 + w_8 + w_{12} &= 0 + 0 + 1 + 0 = 1 \\ w_1 + w_5 + w_9 + w_{13} &= 1 + 0 + 1 + 1 = 1 \\ w_2 + w_6 + w_{10} + w_{14} &= 1 + 0 + 0 + 1 = 0 \\ w_3 + w_7 + w_{11} + w_{15} &= 1 + 1 + 1 + 1 = 0 \end{aligned}$$

Dva výsledky vyšly 0 a dva 1, tedy:

$$q_{12} = \sum_{j \in M(12)} w_j = w_0 + w_4 + w_8 + w_{12} = 0 + 0 + 1 + 0 = 1$$

Po zjištění všech koeficientů polynomů 2. stupně se koeficienty vynásobí s polynomy, kterým náleží:

$$\begin{aligned} q_3 \times x_1 x_2 &= 1 \times (0001000100010001) \\ q_5 \times x_1 x_3 &= 1 \times (0000010100000101) \\ q_6 \times x_2 x_3 &= 1 \times (0000001100000011) \\ q_9 \times x_1 x_4 &= 1 \times (0000000001010101) \\ q_{10} \times x_2 x_4 &= 0 \times (0000001100000011) \\ q_{12} \times x_3 x_4 &= 1 \times (0000000000001111) \end{aligned}$$

V dalším kroku se výsledek z předchozího kroku přičte ke slovu \mathbf{w} :

$$\begin{array}{rcl}
 \mathbf{w} & = & 1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1 \\
 x_1x_2 & + & 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \\
 x_1x_3 & + & 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \\
 x_2x_3 & + & 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \\
 x_1x_4 & + & 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \\
 x_2x_4 & + & 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \\
 x_3x_4 & + & 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \\
 \\
 \mathbf{w}' & = & 1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0
 \end{array}$$

Slovo \mathbf{w}' se nadále používá místo slova \mathbf{w} pro výpočet hodnot koeficientů polynomů 1. stupně.

Výpočet hodnot koeficientů polynomů 1. stupně

Výpočet hodnoty koeficientu q_1 :

$$\begin{array}{l}
 w_0' + w_1' = 0 + 1 = 1 \\
 w_2' + w_3' = 0 + 1 = 1 \\
 w_4' + w_5' = 1 + 0 = 1 \\
 w_6' + w_7' = 1 + 1 = 0 \\
 w_8' + w_9' = 1 + 0 = 1 \\
 w_{10}' + w_{11}' = 1 + 1 = 0 \\
 w_{12}' + w_{13}' = 1 + 0 = 1 \\
 w_{14}' + w_{15}' = 1 + 1 = 0
 \end{array} \Rightarrow q_1 = 1$$

Výpočet hodnoty koeficientu q_2 :

$$\begin{array}{l}
 w_0' + w_2' = 0 + 0 = 0 \\
 w_1' + w_3' = 1 + 1 = 0 \\
 w_4' + w_6' = 1 + 1 = 0 \\
 w_5' + w_7' = 0 + 1 = 1 \\
 w_8' + w_{10}' = 1 + 1 = 0 \\
 w_9' + w_{11}' = 0 + 1 = 1 \\
 w_{12}' + w_{14}' = 1 + 1 = 0 \\
 w_{13}' + w_{15}' = 0 + 1 = 1
 \end{array} \Rightarrow q_2 = 0$$

Výpočet hodnoty koeficientu q_4 :

$$\begin{aligned}
 w'_0 + w'_4 &= 0 + 1 = 1 \\
 w'_1 + w'_5 &= 1 + 0 = 1 \\
 w'_2 + w'_6 &= 0 + 1 = 1 \\
 w'_3 + w'_7 &= 1 + 1 = 0 \\
 w'_8 + w'_{12} &= 1 + 1 = 0 \\
 w'_9 + w'_{13} &= 0 + 0 = 0 \\
 w'_{10} + w'_{14} &= 1 + 1 = 1 \\
 w'_{11} + w'_{15} &= 1 + 1 = 0
 \end{aligned}$$

Počet výsledků rovnajících se 0 a 1 je stejný, použijeme tedy vzorec:

$$q_4 = \sum_{j \in M(4)} w_j = w_0 + w_4 = 0 + 1 = 1$$

Výpočet hodnoty koeficientu q_8 :

$$\begin{aligned}
 w'_0 + w'_8 &= 0 + 1 = 1 \\
 w'_1 + w'_9 &= 1 + 0 = 1 \\
 w'_2 + w'_{10} &= 0 + 1 = 1 \\
 w'_3 + w'_{11} &= 1 + 1 = 0 \\
 w'_4 + w'_{12} &= 1 + 1 = 0 \\
 w'_5 + w'_{13} &= 0 + 0 = 0 \\
 w'_6 + w'_{14} &= 1 + 1 = 0 \\
 w'_7 + w'_{15} &= 1 + 1 = 0
 \end{aligned} \Rightarrow q_8 = 0$$

Po zjištění všech koeficientů polynomů 1. stupně se koeficienty vynásobí s polynomy, kterým náleží:

$$\begin{aligned}
 q_1 \times x_1 &= 1 \times (01010101010101) \\
 q_2 \times x_2 &= 0 \times (0011001100110011) \\
 q_4 \times x_3 &= 1 \times (0000111100001111) \\
 q_8 \times x_4 &= 0 \times (0000000011111111)
 \end{aligned}$$

V dalším kroku se výsledek z předchozího kroku přičte ke slovu \mathbf{w}' :

$$\begin{array}{rcccccccccccccccc}
 \mathbf{w}' & = & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
 x_1 & + & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
 x_2 & + & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 x_4 & + & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
 x_8 & + & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 \\
 \mathbf{w}'' & = & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0
 \end{array}$$

Nyní se ze slova \mathbf{w}'' , podle většinového kritéria zjistí hodnota koeficientu q_0 :

$$q_0 = 1$$

Výsledek dekódování bez možnosti opravení

$$\begin{array}{cccccccccccc}
 & q_{12} & q_{10} & q_6 & q_9 & q_5 & q_3 & q_8 & q_4 & q_2 & q_1 & q_0 \\
 \mathbf{l} & = & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1
 \end{array}$$

Výsledný kód je jiný, než vstupní. Během přenosu mezi kodérem a dekodérem došlo k celkem 4 chybám, jelikož kód $R(2,4)$ je schopný opravit pouze 1 chybu a odhalit chyby 3, liší se výsledek v 5 bitech, což je přibližně 45% poškození kódu.

5 Rozbor programu

5.1 Úvod

Programu je vytvořen jako učební pomůcka Fakulty elektrotechniky a informatiky na Univerzitě Pardubice. Celý projekt je tvořen v programovacím jazyce MATLAB [3], tudíž by měl být srozumitelný pro většinu studentů této fakulty.

Program slouží ke kódování/dekódování informací pomocí Reed-Mullerových kódů.

Program obsahuje také funkce automatického zadání, automatického výpočtu a exportu dat. S těmito funkcemi lze rychle vstupní informaci vytvořit, zakódovat a dekódovat. Výstupy z programu lze uložit v externím textovém souboru.[5]

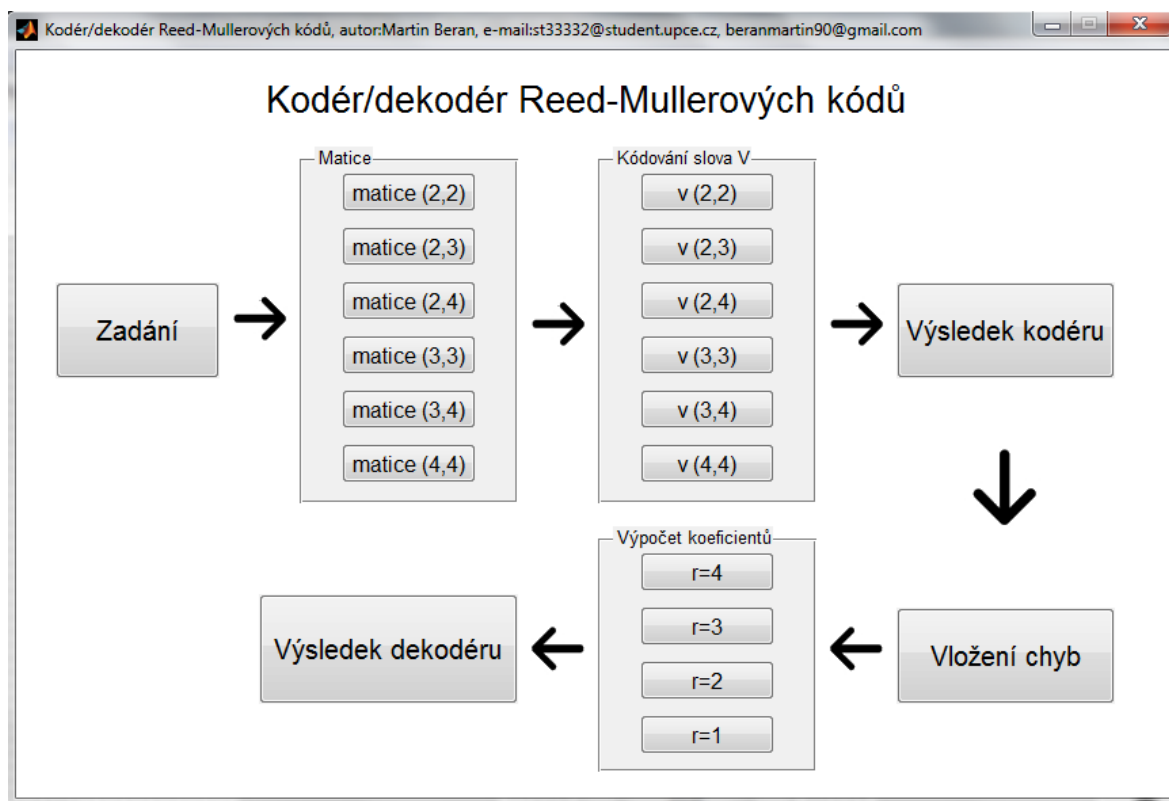
Základní funkce programu:

- volba parametrů RM kódu
- dopočet vhodného RM kódu dle zadaných parametrů
- vlastní RM kodér
- možnost vložení opravitelných/neopravitelných/náhodných chyb
- RM dekodér
- automatické výběr kódu
- automatický kodér/dekodér
- export dat

V následujících kapitolách budou popsány jednotlivé části programu.

5.2 Hlavní okno

Hlavní okno aplikace (Obr. 2) obsahuje přehledně zobrazené jednotlivé fáze kódování a dekodování. Program je ošetřen proti vstupu uživatele do částí SW, kde výpočet prozatím neproběhl.



Obrázek 2 - Hlavní okno programu

Při prvním spuštění programu je nutné vybrat pod tlačítkem „Zadání“ druh RM kódu. Možnosti této funkce budou podrobněji popsány v kapitole 5.3.

Další součástí okna je panel tlačítek s názvem „Matice“, kde se podle druhu kódu zvolí velikost matice pro zakódování námi zapsaného zdrojového slova. V panelu „Kódování slova v “ proběhne již samotné kódování.

Tlačítkem „Výsledek kodéru“ si zobrazíme kód, který je zakódován a výsledné zakódované slovo, s nímž bude pracovat dekodér. Do slova v si můžeme v nabídce tlačítka „Vložení chyb“, vložit libovolný počet chyb, omezený pouze délkou kódu.

Samotné dekodování probíhá v panelu „Výpočet koeficientů“, jehož funkce bude popsána v kapitole 5.8. Výsledek programu je uložen v okně „Výsledek dekodéru“. Zde je zobrazeno, jak se námi zvolené kódované slovo zakódovalo do slova v a poté zpět dekodovalo.

5.3 Okno Zadání

Při prvním spuštění programu je jediné přístupné okno „Zadání“ (Obr. 3).

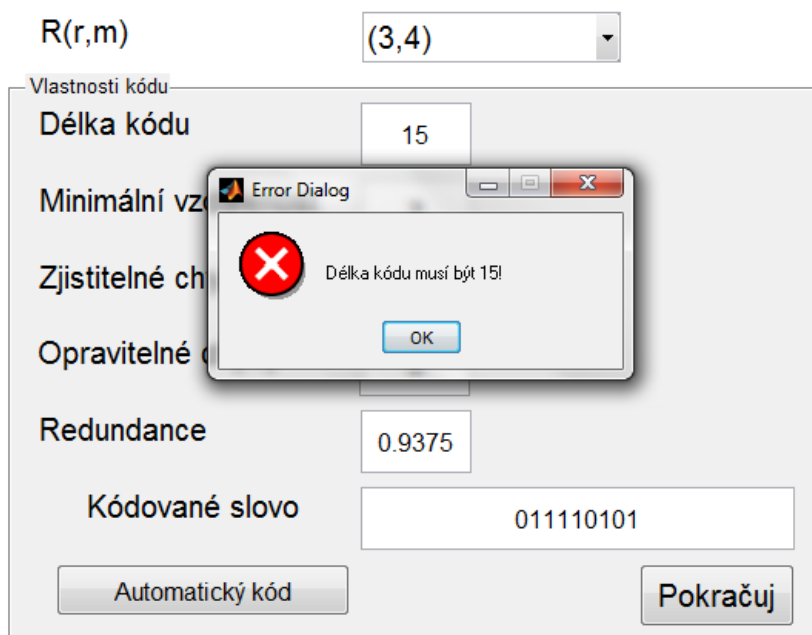
The screenshot shows a software window titled "Kodér/dekodér Reed-Mullerových kódů - zadání". The main heading is "Zadání". Below it is a dropdown menu labeled "R(r,m)". A section titled "Vlastnosti kódu" contains five input fields: "Délka kódu", "Minimální vzdálenost", "Zjistitelné chyby", "Opravitelné chyby", and "Redundance". Below these is a text input field labeled "Kódované slovo". There are two buttons: "Automatický kód" and "Pokračuj". At the bottom, there is an "Export" section with three buttons: "Automatický výpočet", "Export dat", and "Pokračuj".

Obrázek 3 - okno "Zadání"

Uživatel si nejprve zvolí druh Reed-Mullerova kódu $R(r, m)$, tj. jeho stupeň r a délku kódu odpovídající hodnotě 2^m . Celkem je zde na výběr 6 možností od nejjednoduššího kódu (2,2), což je kód druhého stupně se 4 znaky v kódovaném slově, až po kód (4,4), tedy kód čtvrtého stupně s 16 znaky v kódovaném slově. Výběr RM kódů byl zvolen s ohledem na účel aplikace, jako učební pomůcky a tudíž neobsahuje kódy vyšších řádů.

Po výběru se automaticky dopočítají vlastnosti kódu, tj. jeho délka, minimální vzdálenost, počet zjistitelných a opravitelných chyb a redundance kódu (vzorce pro tyto parametry jsou uvedeny v Kapitole 3). Zároveň se zpřístupní tlačítko „Pokračuj“, pomocí kterého se po zapsání kódovaného slova, přistoupí k dalšímu kroku a to ke kódování příslušného slova.

Toto slovo se zapíše do kolonky „Kódované slovo“ a to v délce, kterou nám udává příslušný kód dle našeho výběru. Při zapsání špatné délky, nás program upozorní na chybu a oznámí nám, jakou délku kódu máme zvolit (Obr. 4).



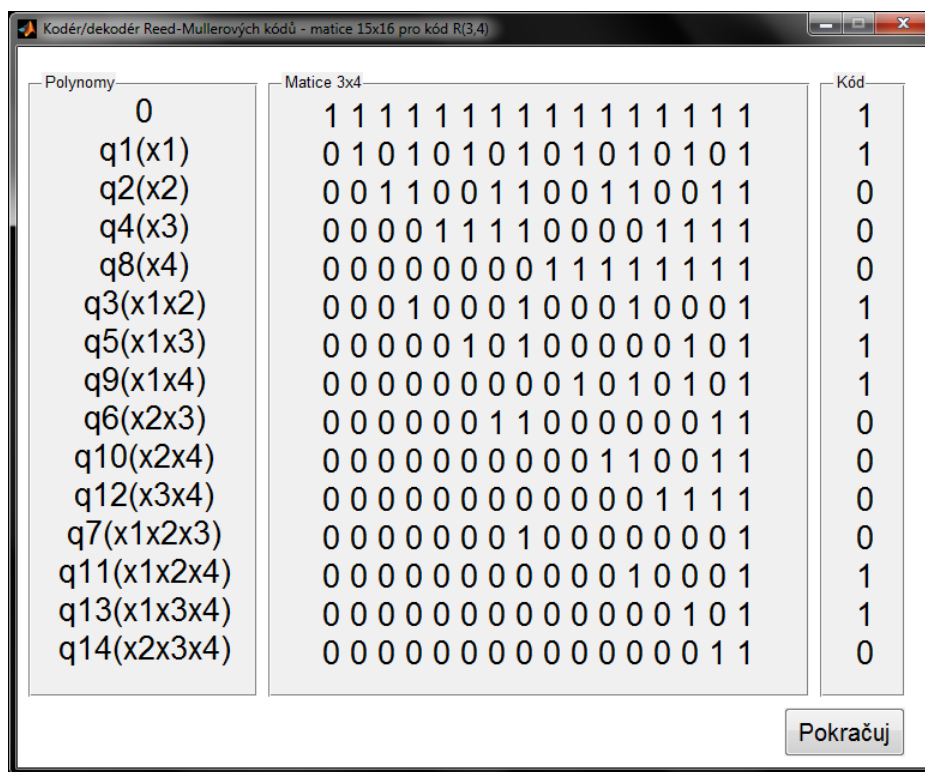
Obrázek 4 - Error Dialog okna "Zadání"

Pro jednoduchost je možné použít tlačítko „Automatický kód“, které samostatně vytvoří kód, který budeme dále v programu používat. S takto vygenerovaným kódem můžeme použít tlačítko „Automatický výpočet“. Při využití automatického výpočtu dojde k automatizovanému procesu kódování/dekódování, kdy lze zobrazit jednotlivé kroky bez nutnosti „krokování“ jednotlivých fází kódování/dekódování. Při automatickém doplnění kódu můžeme samozřejmě celé kódování a dekódování provést postupně, stejně jako u manuálního zadání kódu.

Po výběru „Automatického kódu“ a „Automatického výpočtu“ je možné veškerá data uložit do externího textového souboru formátu *.txt* pomocí tlačítka „Export dat“.

5.4 Okna matic

Každý Reed-Mullerův $R(r, m)$ kód má vlastní matici v závislosti na stupni r a délce 2^m .



Obrázek 5 - okno "Matice 15x16"

Matice pracuje s booleovskými polynomy, které jsou probrány v Kapitole 2. Tyto polynomy jsou zapsány v levém sloupci před samotnou maticí a každý z nich odpovídá jednomu řádku binární matice (Obr. 5).

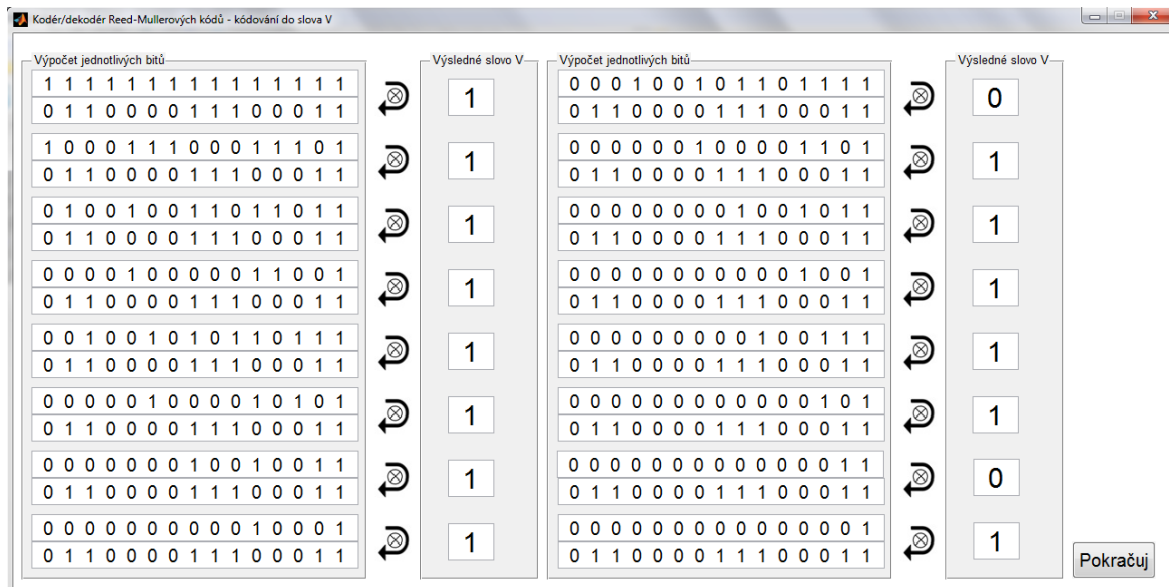
První řádek matice odpovídá počtu 2^m binárních jedniček. Poté následuje m řádků, kde zapisujeme postupně do jednotlivých sloupců binární vyjádření pořadí sloupce, kdy druhý řádek matice je brán jako LSB.

Dalších $\binom{m}{2}$ řádků vyjadřuje polynomy druhého stupně, které vzniknou vynásobením vždy dvou polynomů prvního stupně. Pokud je zvolen kód, který obsahuje i polynomy třetího stupně, vznikne nám $\binom{m}{3}$ řádků vyjádřených jako vynásobení třech polynomů prvního stupně. Taktéž u kódu, který obsahuje polynom čtvrtého stupně, vznikne $\binom{m}{4}$ řádků, kde jsou vynásobeny všechny čtyři polynomy prvního stupně.

V pravém sloupci je nakonec uveden zadaný kód, který je řazen způsobem, že LSB se nachází u prvního řádku a MSB u řádku posledního.

5.5 Okna výpočtu slova v

V těchto oknech probíhá samotné kódování zapsaného kódu do výsledného slova v , které je po možnosti zavedení chyb odesláno do dekodéru (Obr. 6).



Obrázek 6 - okno kódování

Kódování probíhá jako postupné binární násobení a sčítání po jednotlivých bitech. Vždy se násobí zadaný kód s příslušným sloupcem předchozí matice. Tedy binárně vynásobený a sečtený kód s prvním sloupcem matice nám dá MSB výsledného zakódovaného slova v . Výsledné slovo má vždy 2^m bitů, kde m je vyjádřeno z $R(r, m)$.

5.6 Okno výsledek kodéru

Okno „Výsledek kodéru“ slouží pouze k souhrnu kódování. Je zde zapsán druh kódu, informační slovo, a kódové slovo v (Obr. 7).



Obrázek 7 - okno "Výsledek kodéru"

5.7 Okno vkládání chyb

V tomto okně může uživatel vložit do kódového slova uměle zavedenou chybu/y, která simuluje vznik chyby při přenosu. Dle počtu vložených chyb mohou nastat situace: (1) přenos proběhl bez chyby, (2) kódové slovo obsahuje detekovatelné/opravitelné chyby, (3) kódové slovo obsahuje více chyb, než je daný RM kód schopen detekovat/opravit (Obr. 8).

Kodér/dekodér Reed-Mullerových kódů - vkládání chyb

Vkládání chyb

Zadání

R(r,m) (3,4)

Zjistitelné chyby 1

Opravitelné chyby 0

Kódované slovo: 0 1 1 0 0 0 0 1 1 1 0 0 0 1 1

Vložení chyb

Zakódované slovo V: 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1

Počet vložených chyb 1 přepočítat

Nové slovo W: 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1

Pokračuj

Obrázek 8 - okno "Vkládání chyb"

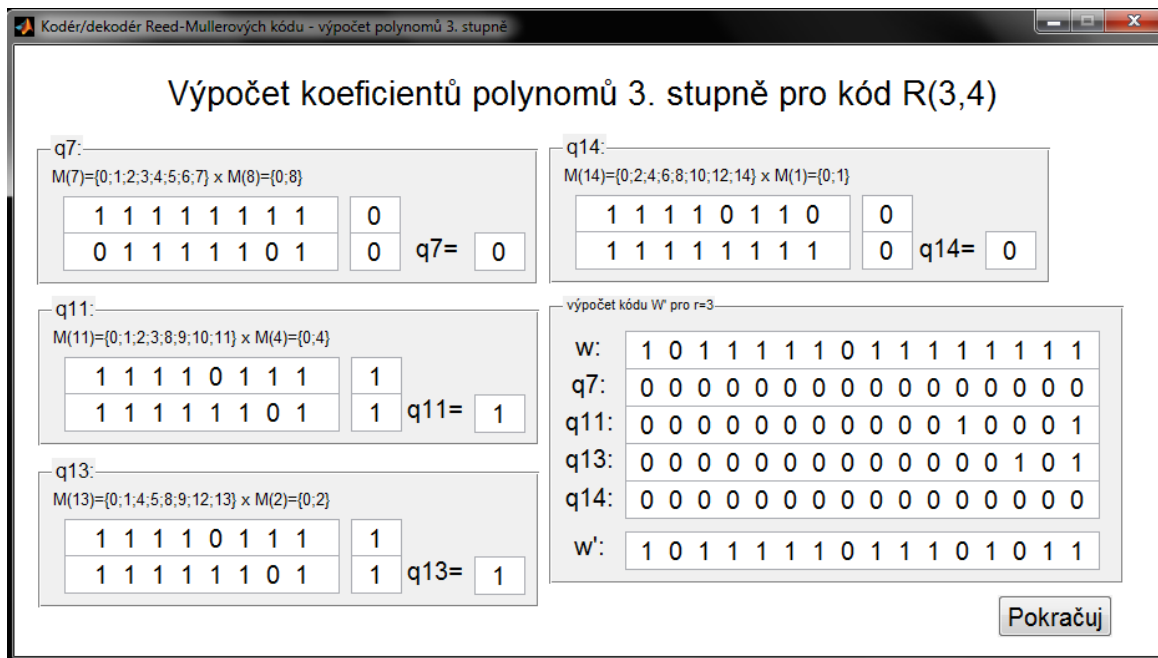
V panelu „Zadání“ je souhrn všeho, co je o kódu známo z hlediska jeho bezpečnosti. Podle složitosti $R(r,m)$ existuje určitý počet zjistitelných a opravitelných chyb. Dále je zde zapsán kód, který byl kódován.

V panelu „Vložení chyb“ je zapsáno zakódované slovo v , do kterého byl kód zakódován. Je možné zde vybrat, kolik chyb se do slova v vloží. Celkem je možno zavést až 16 chyb, musí se ovšem počítat i s délkou slova. Poté se pomocí tlačítka „Přepočítat“ zavedou chyby do slova w .

Pokud se zvolí vyšší počet chyb než je daný kód schopný opravit, pak dekodér nebude schopen „poškozené“ slovo w správně dekodovat do původního kódovaného slova I .

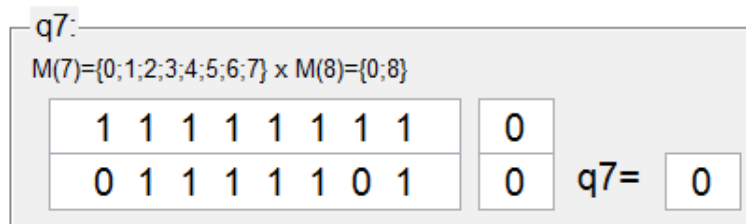
5.8 Okna pro výpočet koeficientů polynomů

V těchto oknech probíhá podle složitosti kódu $R(r, m)$, kde r je stupeň polynomu, dekódování jednotlivých bitů zapsaného kódovaného slova ze slova w , do kterého byl tento kód zakódován (Obr. 9).



Obrázek 9 - okno "Výpočet hodnot koeficientů polynomů 3. stupně"

V jednotlivých panelech koeficientů probíhá výpočet hodnot těchto koeficientů (Obr. 10):



Obrázek 10 - panel výpočtu hodnoty koeficientu q_7

V panelu jsou zapsány „ $M(i)$ “ množiny. Jedná se o výpočet hodnot koeficientů, odpovídající příslušným polynomům x_1, x_2, x_3 .

Množiny odpovídající koeficientu q_7 jsou množiny $M(7)$ a $M((2^m - 1) - 7)$, kde m vychází z $R(r, m)$.

V prvním řádku se tedy nachází bity slova w odpovídající pozicím podle množiny $M(7)$. V druhém řádku jsou bity slova w odpovídající pozicím podle množiny $M(7)$ zvětšeným o $(2^m - 1 - 7)$, což je v našem případě číslo 8. Tedy:

$$W(0) + W(1) + W(2) + W(3) + W(4) + W(5) + W(6) + W(7) \\ W(8) + W(9) + W(10) + W(11) + W(12) + W(13) + W(14) + W(15)$$

Při binárním sčítání oba dva řádky vyjdou rovny 0 a tedy i koeficient $q_7 = 0$.

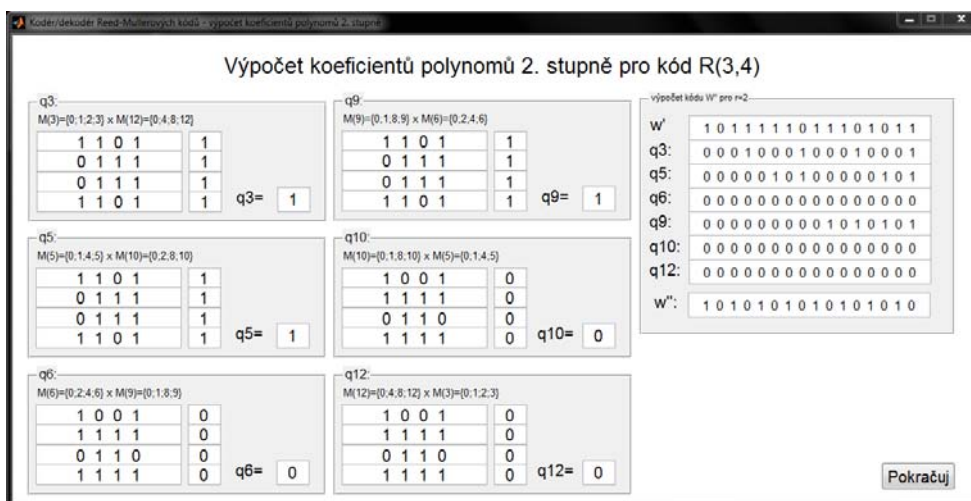
Takto probíhá výpočet pro všechny další koeficienty daného stupně polynomu. Při získání všech koeficientů pokračuje dekódování sečtením slova w s polynomy vynásobenými jejich koeficienty (Obr. 11).

výpočet kódu W' pro $r=3$

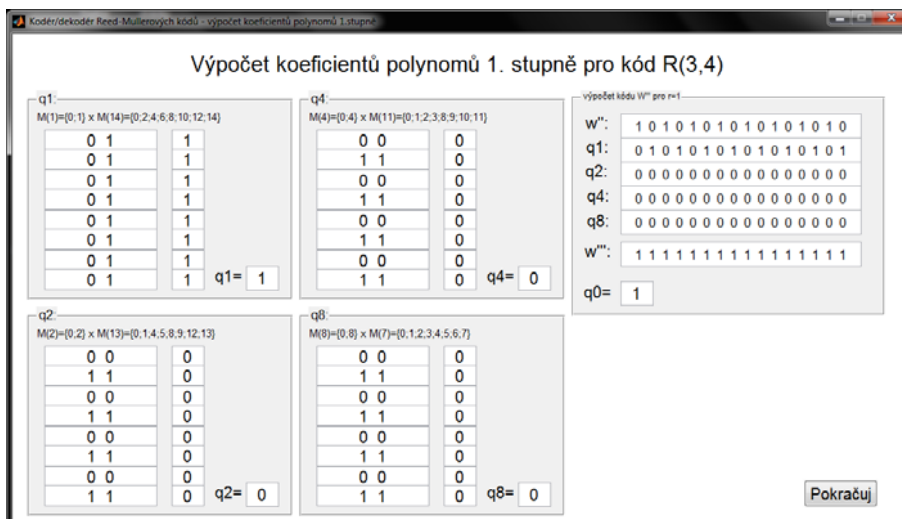
w:	1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1
q7:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
q11:	0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1
q13:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1
q14:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
w':	1 0 1 1 1 1 1 0 1 1 1 0 1 0 1 1

Obrázek 11 - panel výpočtu slova W'

V prvním řádku je zapsané slovo w . V dalších řádcích jsou postupně zapsány polynomy vynásobené odpovídajícími koeficienty. Po binárním sečtení všech řádků nám vyjde výsledné slovo w' , které budeme používat při dalším dekódování u polynomů nižších řádů (Obr. 12 a Obr. 13).



Obrázek 12 - okno "Výpočet hodnot koeficientů polynomů 2. stupně"



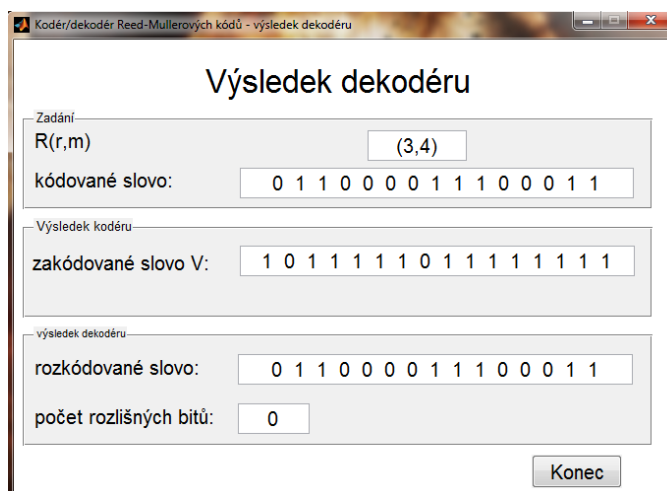
Obrázek 13 - okno "Výpočet hodnot koeficientů polynomů 1. stupně"

Při počítání koeficientů polynomů 1. stupně nakonec vyjde slovo w'' . Z toho slova se zjistí pomocí většinového pravidla poslední koeficient q_0 .

Další práce s koeficienty je rozebrána v následující kapitole.

5.9 Okno výsledek dekodéru

V tomto okně dojde k celkovému seznámení o proběhnutém kódování a dekódování (Obr. 14).



Obrázek 14 - okno "Výsledek dekodéru"

V panelu „Zadání“ je zapsána složitost kódu, který byl na začátku zvolen, a vlastní kód I .

V panelu „Výsledek kodéru“ je obsaženo zakódované slovo v , které se přenáší mezi vysílačem a přijímačem signálu.

V poslední části okna je vypsán výsledek dekódování a počet bitů, ve kterých se liší výsledek od zadání.

Závěr

Náplní bakalářské práce bylo vytvoření kodéru/dekodéru vybraných Reed-Mullerových kódů.

Výstupem bakalářské práce je vytvoření programu, který dokáže kódovat a dekódovat informace pomocí tohoto druhu kódování. Program bude využit jako názorná učební pomůcka pro předmět Teorie informace. Program umožňuje zavedení chyby vzniklé při přenosu v přenosovém kanálu. Lze zavést libovolný počet chyb a tím názorně ověřit parametry vybraného kódu. Software umožňuje zobrazit jednotlivé kroky kódování/dekódování v samostatných oknech, aby byl celkový proces přehledný a srozumitelný.

Teoretická část práce se nejdříve zabývá vysvětlení vybraných pojmů z oblasti teorie informace, které budou využity v dalších částech bakalářské práce. Součástí této kapitoly je i popis booleovských funkcí, které jsou stěžejní pro vytváření Reed-Mullerova kódu. Vlastní Reed-Mullerovo kódování je popsáno v Kapitole 3, kde jsou uvedeny vlastnosti kódování, jeho složitost a vlastní proces kódování/dekódování.

V následující kapitole je ukázkový příklad vytvoření RM kódu, zavedení chyb a následný proces dekódování, který přispěje pro hlubší porozumění principu vlastního kódu.

Kapitola 5 obsahuje vlastní popis programu. Postupně je zde zobrazen postup kódování vstupního informačního slova pomocí generující matice a vytvoření zakódovaného slova. Dále je možné zavést chybu při přenosu mezi kodérem a dekodérem pomocí chybového vektoru. Dekódování probíhá od polynomů nejvyšších řádů k polynomům 1. řádu. Každý koeficient polynomu je vypočítán jednotlivě a poté je zobrazen jeho výsledek, ovlivňující další část dekódování. Každý výpočet polynomů je zobrazen v samostatném okně. V poslední části programu je zobrazeno informační slovo na vstupu, výsledek kodéru a dekódované slovo.

Vytvořený program je modulární, a lze jej rozšířit o další druhy kódování i o složitosti kódů.

Podle mého názoru jsou nejlépe využitelné Reed-Mullerovy kódy $R(r, m)$ u nichž je $m \gg r$ v porovnání s BCH kódy. Tyto kódy jsou schopny odhalit a opravit větší množství chyb, jelikož mají vysokou redundanci a tím pádem se při kódování zavádí více nadbytečných bitů. Tyto kódy jsem kvůli nepřehlednosti při zobrazování do programu nezařadil.

Do programu je navíc přidána funkce pro export výsledků. Tato funkce vypíše do textového souboru vstupní informační slovo, výsledek jeho kódování a postupné vyjádření průběhu dekódování po jednotlivých stupních polynomů, které kódu náleží.

Literatura

- [1] ADÁMEK, Jiří. Kódování. Praha: SNTL – Nakladatelství technické literatury, 1989.
- [2] VLČEK, Karel. Kompresce a kódová zabezpečení v multimediálních komunikacích. 2. vyd. Praha: BEN - technická literatura, 258 s. ISBN 80-730-0134-9.
- [3] JIANG, Yuan. A practical Guide to Error-Control Coding Using MATLAB. London: Artech House, 2010. ISBN 13:978-1-60807-088-6.
- [4] LIN, Shu a Daniel J. COSTELLO. Error Control Coding, 2/E. Prentice Hall, 2005. ISBN-10:0130426725.
- [5] *MathWorks* [online]. [cit. 2013-12-11]. Dostupné z: <http://www.mathworks.com/>
- [6] *Wikipedia*. [online]. [cit. 2014-03-23]. Dostupné z: <http://cs.wikipedia.org>
- [7] *AmaPro*. [online]. [cit. 2014-03-23]. Dostupné z: <http://amapro.cz>
- [8] NĚMEC, Zdeněk. Přenos dat a informací – přednášky. UPCE. 2013.
- [9] PIDANIČ, Jan. Teorie informace – přednášky. UPCE. 2014.
- [10] COOKE, Ben. Reed-Muller Error Correcting Codes. MIT Under-graduate Journal of Mathematics Volume 1, MIT Department of Mathematics, 1999.

Přílohy

Zdrojový kód programu na CD.