

STUDIJU DARBS

mācību disciplīnā

Informācijas pārraides un ciparu sakaru sistēmu pamati

Mārtiņš Dundurs

rect0 grupa, apl. nr.

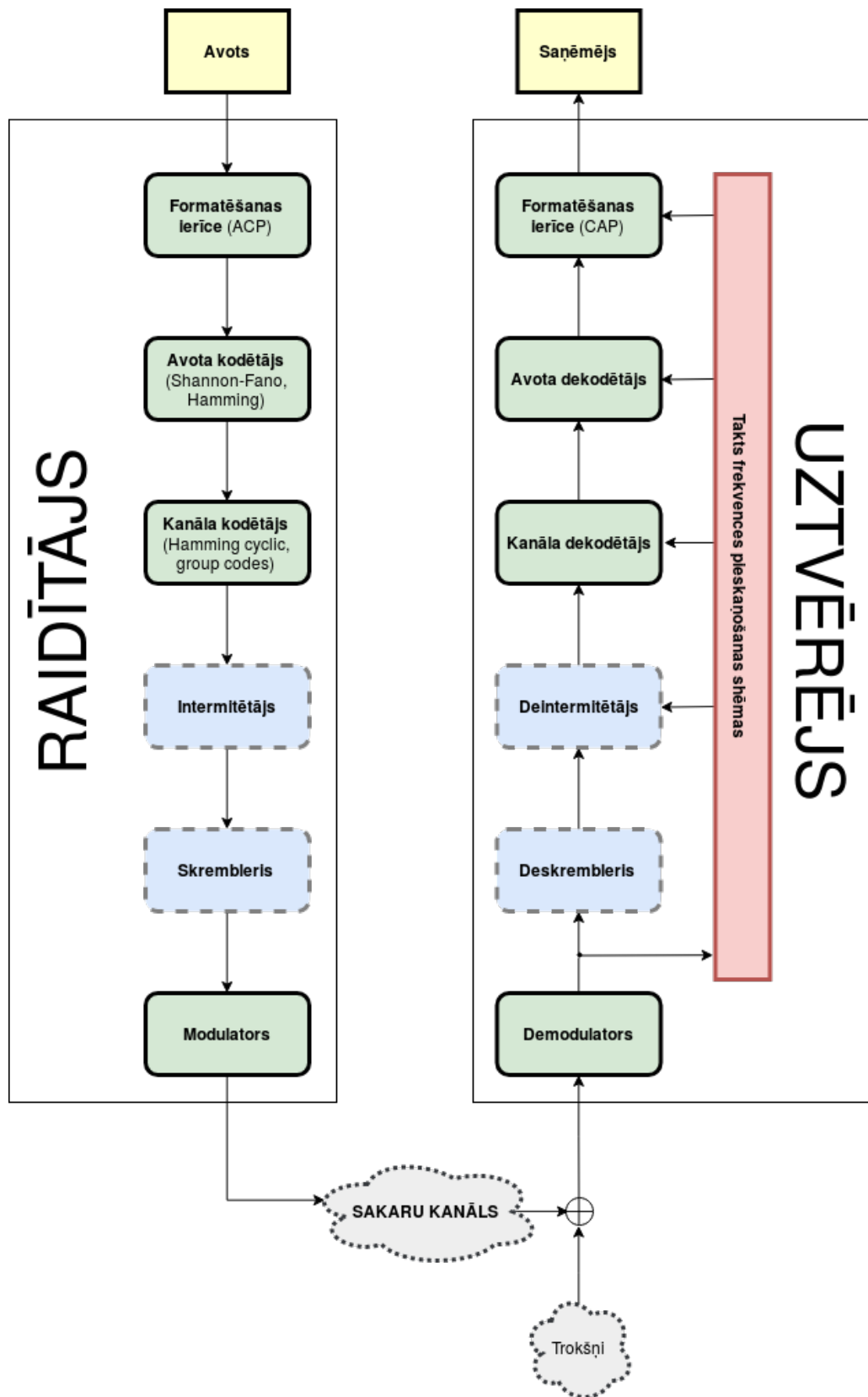
2018. g. 7. jūnijā

1. DARBAM UZDOTIE PARAMETRI

Simbolu ilgums: $\tau = 100 \mu s$;
Signāla-trokšņu attiecība: $(E/N_0)_{dB} = 7.5 \text{ dB}$;
Ekonomiskā avota kodēšana: Šanona-Fano kods;
Koriģējošā kodēšana: sistemātisks Heminga grupas kods;
Koriģējošā koda parametri: izejot no $m = 31$;
Nesošā frekvence: $f_c = 3.9 \text{ MHz}$;
Manipulācija: BPSK;
Fāzes nobīde: $\phi = \pi \text{ rad.}$
Alfabēts:

b_i	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}	b_{11}	b_{12}
$W(b_i)$	0.01	0.01	0.02	0.02	0.05	0.05	0.1	0.15	0.2	0.25	0.06	0.08

2. DISKRĒTU SIGNĀLU PĀRRAIDES SISTĒMAS PA CIPARU KANĀLU STRUKTŪRAS SHĒMA



3. INFORMĀCIJAS AVOTA LIETDERĪBAS, ENTROPIJAS UN REDUNDANCES ATRAŠANAS PROGRAMMA

Izmantojot MatLab, izstrādājam programmu, kas saņem avota alfabētu csv formātā un izvada trīs vērtības: avota entropiju, lietderību un redundanci. Avota alfabēts csv formātā tiek uzdots kā vektors ar garumu N (burtu skaitu) nu vērtībām, kas ir attiecīgo burtu varbūtības. Faila nosaukumam jābūt 'alphabet.csv'. Avota redundanci aprēķināsim pēc formulas (1).

$$H(A) = - \sum_{i=1}^N p_i \log_2 p_i = 3.033 \text{ [bit]} \quad (1)$$

Avota lietderību aprēķināsim pēc (2), kur n ir avota alfabēta burtu skaits.

$$L = \frac{H(A)}{\log_2 n} = 0.846 \quad (2)$$

Redundance ir pretējais lietderībai (3).

$$\rho_a = 1 - \frac{H(A)}{\log_2 n} = 0.154 \quad (3)$$

MatLab kods izrādās ļoti vienkāršs:

```
clc;
W = csvread('alphabet.csv'); %Attiecīgo alfabēta burtu varbūtība
H = -sum(W.*log2(W)) %Avota entropija
L = H/log2(length(W)) %Avota lietderība
reds = 1-H/log2(length(W)) %Avota redundance
```

4. EKONOMISKĀ KODĒŠANA

Mēs veicam avota alfabēta ekonomisko kodēšanu ar Šanona-Fano ekonomisko kodu.

b_i	$W(b_i)$		b_i	$W(b_i)$						ŠF kods	m_i	m_{1i}	m_{0i}	
b_1	0.01		b_{10}	0.25	1	1				11	2	2	0	
b_2	0.01		b_9	0.2	1	0				10	2	1	1	
b_3	0.02		b_8	0.15	0	1	1			011	3	2	1	
b_4	0.02		b_7	0.1	0	1	0			010	3	1	2	
b_5	0.05		b_{12}	0.08	0	0	1	1		0011	4	2	2	
b_6	0.05	→	b_{11}	0.06	0	0	1	0		0010	4	1	3	
b_7	0.1		b_5	0.05	0	0	0	1	1	00011	5	2	3	
b_8	0.15		b_6	0.05	0	0	0	1	0	00010	5	1	4	
b_9	0.2		b_3	0.02	0	0	0	0	1	1	000011	6	2	4
b_{10}	0.25		b_4	0.02	0	0	0	0	1	0	000010	6	1	5
b_{11}	0.06		b_1	0.01	0	0	0	0	0	1	000001	6	1	5
b_{12}	0.08		b_2	0.01	0	0	0	0	0	0	000000	6	0	6

Koda kombinācijas vidējo garumu mēs varam noteikt ar (4).

$$\bar{m} = \sum_{i=1}^N m_i W(b_i) = \sum_{i=1}^{12} m_i W(b_i) = 3.07 \text{ [bit]} \quad (4)$$

Redzam ka vidējais kombinācijas garums ir nedaudz lielāks par entropiju. Tas bija mūsu mērķis, jo tas nozīmē ka mūsu burti ir kļuvuši informatīvāki, proti sūtītā informācija ir kļuvusi neparedzamāka. Lai noteiktu simbolu '1' un '0' parādīšanās varbūtības nokodētajā ziņojumā izmantosim nulļu un vieninieku vidējo daudzumu katra burtā kodā - (5) un (6).

$$\bar{m}_0 = \sum_{i=1}^N m_{0i} W(b_i) = \sum_{i=1}^{12} m_{0i} W(b_i) = 1.53 \text{ [bit]} \quad (5)$$

$$\bar{m}_1 = \sum_{i=1}^N m_{1i} W(b_i) = \sum_{i=1}^{12} m_{1i} W(b_i) = 1.54 \text{ [bit]} \quad (6)$$

Attiecīgi '0' un '1' varbūtības būs (7) un (8).

$$W(0) = \frac{\bar{m}_0}{\bar{m}} = 0.4984 \quad (7)$$

$$W(1) = \frac{\bar{m}_1}{\bar{m}} = 0.5016 \quad (8)$$

Redzam ka varbūtības ir gandrīz vienādas, bet '1' tomēr sastopami nedaudz biežāk. Tas atkal skaidrojams ar to ka nebija iespējams izpildīt Šanona-Fano algoritmu perfekti.

Mēs varam tagad salīdzināt mūsu tagad veidojamā kanāla redundanci (9) ar oriģinālo iepriekš aprēķināto avota redundanci.

$$\rho_k = 1 - \frac{H(A)}{\bar{m}} = 0.0121 \quad (9)$$

Redzam, ka kanālā esam redundanci ievērojami samazinājuši par aptuveni 92%.

5. KANĀLA CAURLAIDES SPĒJAS NOTEIKŠANA

Binārās informācijas pārraides sistēmas kanāla caurlaides spēju kanālā ar trokšņiem mēs varam noteikt pēc (10) formulas.

$$C_{kan} = \frac{1 + P_{err} \log_2 P_{err} + (1 - P_{err}) \log_2 (1 - P_{err})}{\tau} = 9.9996 \left[\frac{kbit}{s} \right] \quad (10)$$

Bitu kļūdas varbūtību P_{err} fāzes manipulācijai (PM) mēs noteicām pēc (11) formulas, izmantojot kļūdas funkciju. Tā atkarīga arī no signāls-trokšņi attiecības E/N_0 , kas tika dota. Es pieņemu šeit, ka tā ir dota deciBelos un to apzīmēsim kā $(E/N_0)_{dB}$.

$$P_{err_{PM}} = 1 - \Phi\left(\sqrt{2 \cdot \frac{E}{N_0}}\right) = 1 - \Phi\left(\sqrt{2 \cdot 10^{(E/N_0)_{dB}/10}}\right) \quad (11)$$

$$P_{err_{PM}} = 1 - \frac{1}{2} \left(1 + \operatorname{erf}\left(\sqrt{10^{(E/N_0)_{dB}/10}}\right) \right) = \frac{1}{2} - 0.5 \cdot \operatorname{erf}\left(\sqrt{10^{(E/N_0)_{dB}/10}}\right) \approx 0.4 \cdot 10^{-3}$$

Šī vērtība mums ir vajadzīga, lai mēs varētu pārbaudīt vai izpildās Šenona teorēmas nosacījums. Lai to izdarītu, mums vajadzīgs arī aprēķināt avota veikspēju R_a (12). Ja kanāla caurlaides spēja C_{kan} izrādīsies lielāka par avota veikspēju, tad pēc Šenona teorēmas mēs varēsim pārraidīt caur šo kanālu mēs varēsim pārraidīt ciparu signālu ar pēc patikas mazu kļūdas varbūtību.

$$R_a = \frac{H_m(A)}{\bar{m}\tau} = \frac{H(A)}{\bar{m}\tau} = 9.8794 \left[\frac{kbit}{s} \right] \quad (12)$$

Redzam, ka mūsu gadījumā Šenona teorēmas nosacījums izpildās.

6. KORIGĒJOŠĀ KODĒTĀJA UN DEKODĒTĀJA SHĒMAS UN TO DARBĪBA

6.1. Sistemātiskā Heminga grupas koda kodētāja izstrāde

Mums ir izvirzīts nosacījums, ka kopējais koda garums ir $m = 31$. Pēc (13) mēs atradīsim nepieciešamo redundances bitu skaitu r .

$$r = \log_2(m + 1) = \log_2(31 + 1) = \log_2(32) = 5 [bit] \quad (13)$$

Tātad vienā mūsu koda vārdā būs $k = m - r = 31 - 5 = 26$ lietotāja informācijas biti.

1. un 2. solis Sastādam kļūdas vektoru tabulu un piešķiram atpazīnējus binārajā kodā.

Kļūdainā kārta	Kļūdas vektors						Atpazīnējs
	a_1	a_2	a_3	...	a_{30}	a_{31}	
a_1	1	0	0	...	0	0	00001
a_2	0	1	0	...	0	0	00010
...			
a_{31}	0	0	0	...	0	1	11111

3., 4. un 5. solis Pēc šīs tabulas mēs varam sastādīt 5 vienādojumu sistēmu (14) - piecu kontroles bitu vērtību aprēķināšanas formulas. To mēs daram izejot no 5 atpazīnējus veidojošajām kolonnām. Pirmajā vienādojumā ietilps visas tās kārtas, kurām 1. kolonnā ir '1'. Otrajā vienādojumā ietilps visas tās kārtas, kurām 2. kolonnā ir '1', utt.

$$\begin{cases} a_{16} = a_{17} \oplus a_{18} \oplus a_{19} \oplus a_{20} \oplus a_{21} \oplus a_{22} \oplus a_{23} \oplus a_{24} \oplus a_{25} \oplus a_{26} \oplus a_{27} \oplus a_{28} \oplus a_{29} \oplus a_{30} \oplus a_{31} \\ a_8 = a_9 \oplus a_{10} \oplus a_{11} \oplus a_{12} \oplus a_{13} \oplus a_{14} \oplus a_{15} \oplus a_{24} \oplus a_{25} \oplus a_{26} \oplus a_{27} \oplus a_{28} \oplus a_{29} \oplus a_{30} \oplus a_{31} \\ a_4 = a_5 \oplus a_6 \oplus a_7 \oplus a_{12} \oplus a_{13} \oplus a_{14} \oplus a_{15} \oplus a_{20} \oplus a_{21} \oplus a_{22} \oplus a_{23} \oplus a_{28} \oplus a_{29} \oplus a_{30} \oplus a_{31} \\ a_2 = a_3 \oplus a_6 \oplus a_7 \oplus a_{10} \oplus a_{11} \oplus a_{14} \oplus a_{15} \oplus a_{18} \oplus a_{19} \oplus a_{22} \oplus a_{23} \oplus a_{26} \oplus a_{27} \oplus a_{30} \oplus a_{31} \\ a_1 = a_3 \oplus a_5 \oplus a_7 \oplus a_9 \oplus a_{11} \oplus a_{13} \oplus a_{15} \oplus a_{17} \oplus a_{19} \oplus a_{21} \oplus a_{23} \oplus a_{25} \oplus a_{27} \oplus a_{29} \oplus a_{31} \end{cases} \quad (14)$$

Tā kā tika izvirzīta sistemātiska koda prasība, apzīmēsim ar jauniem apzīmējumiem katru kārtu. Šos vienādojumus mēs izmantosim kodētājā, kurš atradīs lietotāja informācijai atbilstošo redundanci.

$$\begin{cases} e_5 = c_{12} \oplus c_{13} \oplus c_{14} \oplus c_{15} \oplus c_{16} \oplus c_{17} \oplus c_{18} \oplus c_{19} \oplus c_{20} \oplus c_{21} \oplus c_{22} \oplus c_{23} \oplus c_{24} \oplus c_{25} \oplus c_{26} \\ e_4 = c_5 \oplus c_6 \oplus c_7 \oplus c_8 \oplus c_9 \oplus c_{10} \oplus c_{11} \oplus c_{19} \oplus c_{20} \oplus c_{21} \oplus c_{22} \oplus c_{23} \oplus c_{24} \oplus c_{25} \oplus c_{26} \\ e_3 = c_2 \oplus c_3 \oplus c_4 \oplus c_8 \oplus c_9 \oplus c_{10} \oplus c_{11} \oplus c_{15} \oplus c_{16} \oplus c_{17} \oplus c_{18} \oplus c_{23} \oplus c_{24} \oplus c_{25} \oplus c_{26} \\ e_2 = c_1 \oplus c_3 \oplus c_4 \oplus c_6 \oplus c_7 \oplus c_{10} \oplus c_{11} \oplus c_{13} \oplus c_{14} \oplus c_{17} \oplus c_{18} \oplus c_{21} \oplus c_{22} \oplus c_{25} \oplus c_{26} \\ e_1 = c_1 \oplus c_2 \oplus c_4 \oplus c_5 \oplus c_7 \oplus c_9 \oplus c_{11} \oplus c_{12} \oplus c_{14} \oplus c_{16} \oplus c_{18} \oplus c_{20} \oplus c_{22} \oplus c_{24} \oplus c_{26} \end{cases} \quad (15)$$

6. solis Tagad uzrakstīsim vienādojumu sistēmu sindromu aprēķināšanai, kas būs nepieciešams dekodēšanai. Sindroms parādīs kurā kārtā ir kļūda, ja būs pielaista viena kļūda. * norāda ka atbilstošā kārta ir caur kanālu saņemtais bits, kas var nesakrist ar oriģinālo, sūtīto bitu.

$$\begin{cases} S_5 = c_{12}^* \oplus c_{13}^* \oplus c_{14}^* \oplus c_{15}^* \oplus c_{16}^* \oplus c_{17}^* \oplus c_{18}^* \oplus c_{19}^* \oplus c_{20}^* \oplus c_{21}^* \oplus c_{22}^* \oplus c_{23}^* \oplus c_{24}^* \oplus c_{25}^* \oplus c_{26}^* \oplus e_5^* \\ S_4 = c_5^* \oplus c_6^* \oplus c_7^* \oplus c_8^* \oplus c_9^* \oplus c_{10}^* \oplus c_{11}^* \oplus c_{19}^* \oplus c_{20}^* \oplus c_{21}^* \oplus c_{22}^* \oplus c_{23}^* \oplus c_{24}^* \oplus c_{25}^* \oplus c_{26}^* \oplus e_4^* \\ S_3 = c_2^* \oplus c_3^* \oplus c_4^* \oplus c_8^* \oplus c_9^* \oplus c_{10}^* \oplus c_{11}^* \oplus c_{15}^* \oplus c_{16}^* \oplus c_{17}^* \oplus c_{18}^* \oplus c_{23}^* \oplus c_{24}^* \oplus c_{25}^* \oplus c_{26}^* \oplus e_3^* \\ S_2 = c_1^* \oplus c_3^* \oplus c_4^* \oplus c_6^* \oplus c_7^* \oplus c_{10}^* \oplus c_{11}^* \oplus c_{13}^* \oplus c_{14}^* \oplus c_{17}^* \oplus c_{18}^* \oplus c_{21}^* \oplus c_{22}^* \oplus c_{25}^* \oplus c_{26}^* \oplus e_2^* \\ S_1 = c_1^* \oplus c_2^* \oplus c_4^* \oplus c_5^* \oplus c_7^* \oplus c_9^* \oplus c_{11}^* \oplus c_{12}^* \oplus c_{14}^* \oplus c_{16}^* \oplus c_{18}^* \oplus c_{20}^* \oplus c_{22}^* \oplus c_{24}^* \oplus c_{26}^* \oplus e_1^* \end{cases} \quad (16)$$

Rezultātā mums tagad ir izstrādāts mūsu korigējošā koda matemātiskais modelis. Nākamais solis ir to realizēt.

6.2. Sistemātiskā Heminga grupas koda kodētāja realizācija

Realizācijas shēma ir pielikumā ar nosaukumu 'Sistemātiska Heminga grupas koda kodētājs'. Tā darbības princips ir sekojošs. Lietotāja informācijas 26 biti tiek ievadīti 28 bitu reģistrā (DD1-DD7). Piekļuve šiem bitiem visām turpmākajām darbībām ir caur kopni BUS. Galvenais kodētāja uzdevums ir aprēķināt kontroles bitu vērtības pēc (15) formulas. Tā kā katra kontroles bita aprēķinam ir jāizdara XOR operācija ar 15 bitiem, tad mums kopumā nepieciešamas 14 XOR ierīces. Mēs izmantojam XOR ierīces, kurās ir iestrādāti uzreiz 4 XOR vārti. Lai izdarītu vienu XOR operāciju priekš 5 kontroles bitiem, mums vajadzēs 2 šādas ierīces, no kurām otrai paliks neizmantoti 3 vārti. Mums tāpat būs vajadzīgi 14 šādi XOR ierīču pāri (DD8-DD35). Pirmajam pārim visas ieejas tiek ņemtas no kopnes, bet visām nākamajām viena vārtu ieeja nāk no iepriekšējās XOR kaskādes, bet otra no kopnes. Izejot visu kaskādi tiek realizēta XOR operācija 15 ieejām. Pēdējai kaskādei ir 5 izejas, kuras ir attiecīgie kontroles biti.

Tā kā mums ir nepieciešams sistemātisks kods, tad mēs iegūtās vērtības sakārtojam izejas reģistrā (DD36-DD43) sistemātiski, proti, 1-26 reģistra ieejā tiek ievadīti lietotāja 26 biti no kopnes, bet vēl 5 ieejās - iegūtos kontroles bitus secībā no 1 līdz 5. Kopumā reģistram ir 31 ieeja, tāpēc tiek izmantots 32 bitu reģistrs, kas sastāv no 8 4-bitu reģistriem. Viena ieeja un viena izeja tam netiek izmantota.

6.3. Sistemātiskā Heminga grupas koda dekodētāja realizācija

Dekodētāja shēma dota pielikumos ar nosaukumiem 'Sistemātiskā Heminga grupas koda kodētājs a' un 'Sistemātiskā Heminga grupas koda kodētājs b'. Dekodētāja galvenais uzdevums ir izpildīt (16) formulas, aprēķinot sindroma 5 kārtas. Tas parādīts (a) shēmā. Formulas ir ļoti līdzīgas kā kodētājā, tikai jāpieskaita pēc mod2 vēl uztvertie kontroles biti. Līdz ar to ir vajadzīgas 15 kaskādes, kuras veiks XOR operāciju 16 saskaitāmajiem. Pēdējā kaskādē tiks pieskaitīti kontroles biti. Ja kļūdu nav, tad visi sindroma biti būs nulles.

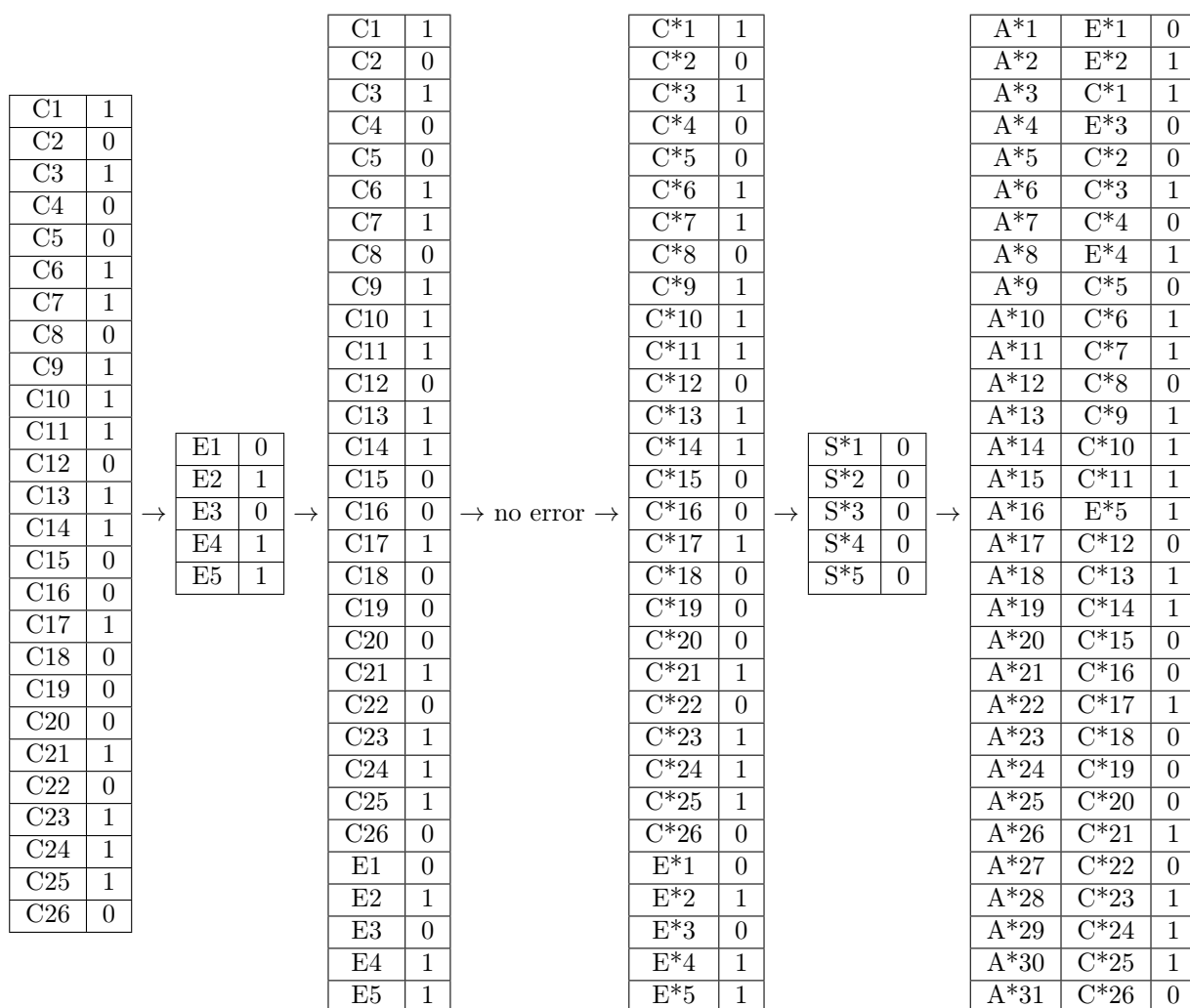
Otrs dekodētāja uzdevums ir veikt kļūdu labošanu. Ja sindroms $S^* \neq 0$, tad tā binārā vērtība norāda kurā pēc kārtas kārtā ir notikusi kļūda (vienas kļūdas gadījumā). Lai tieši šajā kārtā nomainītu simbolu uz pretējo, mums pieciek šī simbola vērtību saskaitīt pēc mod2 (XOR) ar '1'. Mēs izmantosim dekodera ierīci, kura nolasa

bināru skaitli un padod '1' tajā izejā, kuras kārtas numurs atbilst šim skaitlim. Tā kā mums ir 26 informācijas biti, tad mums vajadzēs vismaz 5x32 dekoderi. Tādas ierīces tirgū nav, bet to var realizēt kombinējot mazākus dekoderus. Es izvēlējos attēlot abstraktu iedomātu ierīci - Dekoderi 5x32 ar 5 ieejām un 32 izejām no Y0 līdz Y31. Tā ieejās es padošu sindromu. Mums nevajadzēs izmantot visas tā izejas - tikai 26, proti ja kļūda ir kādā no kontroles bitiem, tad mums tie nav jālabo, jo tālāk tie netiks izmantoti. Tāpat mums nevajadzēs arī izeju Y0, kur tiks padots '1', ja $S^* = 0$. Ja sindroms ir nulle, tad dekoderim jāuzskata ka kļūdas nav un nekādi labojumi nav jāveic. Jāpiezīmē, ka lai gan mūsu pārraidītais kods bija sistematiskais, sindroms norādis kārtu nesistematiskajā kodā. Tāpēc mums dekodera DD1 izejas ir jāorganizē nesistematiski. Neizmantotās kontroles bitu izejas būs Y1, Y2, Y4, Y8 un Y16. Uzskatāmības pēc visas izejas ievadam kopnē BUS, kurā ievadīti ir arī mūsu uztvertie biti (C^*1 līdz C^*26). Tālāk no BUS izvelkam katram XOR nepieciešamo bitu pāri. Ja pārraide kļūdu nebija, tad XOR operācijas izejā būs tās pašas C^* bitu vērtības.

6.4. Kodētāja un dekodētāja pārbaude

Lai pārbaudītu mūsu sistēmas darbību, izmantosim sekojošu datu bloku, kas sastāv no 26 bitiem. Vispirms šie dati (C1-C26) tiks ievadīti reģistrā (DD1-DD7). Pēc tam XOR ierīču kaskādē (DD8-DD35) tiks aprēķināti pieci kontroles biti (E1-E5). Pēc tam šie kontroles biti tiks salikti vienā reģistrā (DD36-DD43) ar datu bitiem. Ja netiek izmantota ne intermitēšana, ne skremblēšana, tad šie biti ir gatavi padošanai uz modulētāju. Vispirms apskatīsim gadījumu, kad sakaru kanālā kļūdas nerodas.

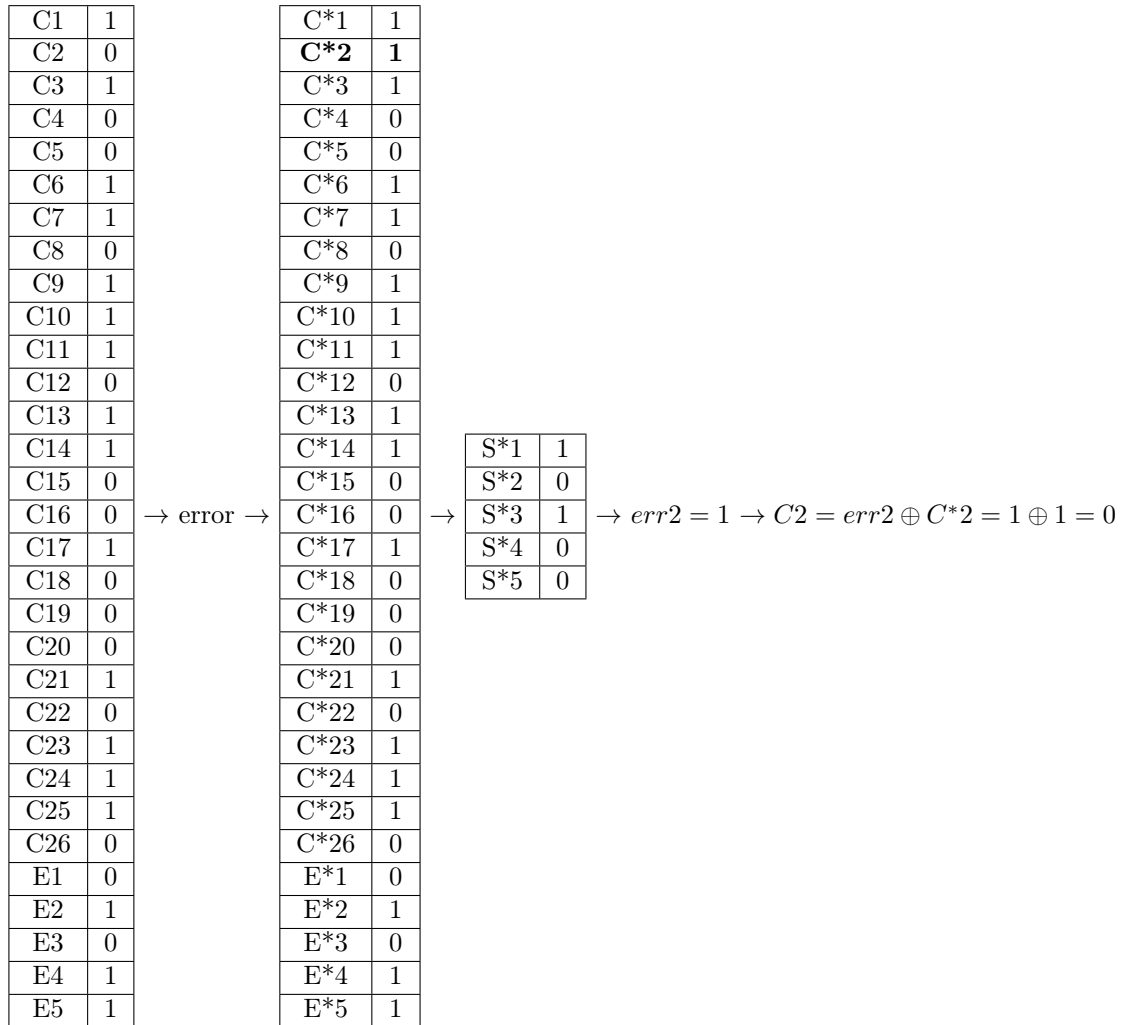
Dekodera sākuma struktūra gandrīz identiska koderim. Atšķirība ir tā, ka sākuma reģistrs (DD44-DD51) satur papildus ierīci, jo saņemts tiek 31 bits. Pie tam tie tiek saņemti nesistematiskā kodā. Praksē tas sagādā problēmas, jo shēmā (PCB vai IC) ir sarežģīti savienot vadus, bet mūsu gadījumā mēs visus bitus ievadīsim kopnē BUS, no kuras mēs ērti varēsim izmantot, kad mums tos vajadzēs. Tālāk shēmā seko tāda pati kaskāde kā koderī līdz posmam DD78-DD79. Tā kā kļūda nav pielasta, tad šī posma izejā mēs iegūsim tos pašus kontroles bitus kā koderī. Nākamais posms (DD80-DD81) šiem kontroles bitiem pieskaitīs identiskus kontroles bitus. Tā kā $x \oplus x = 0$, tad visi sindroma biti būs nulles.



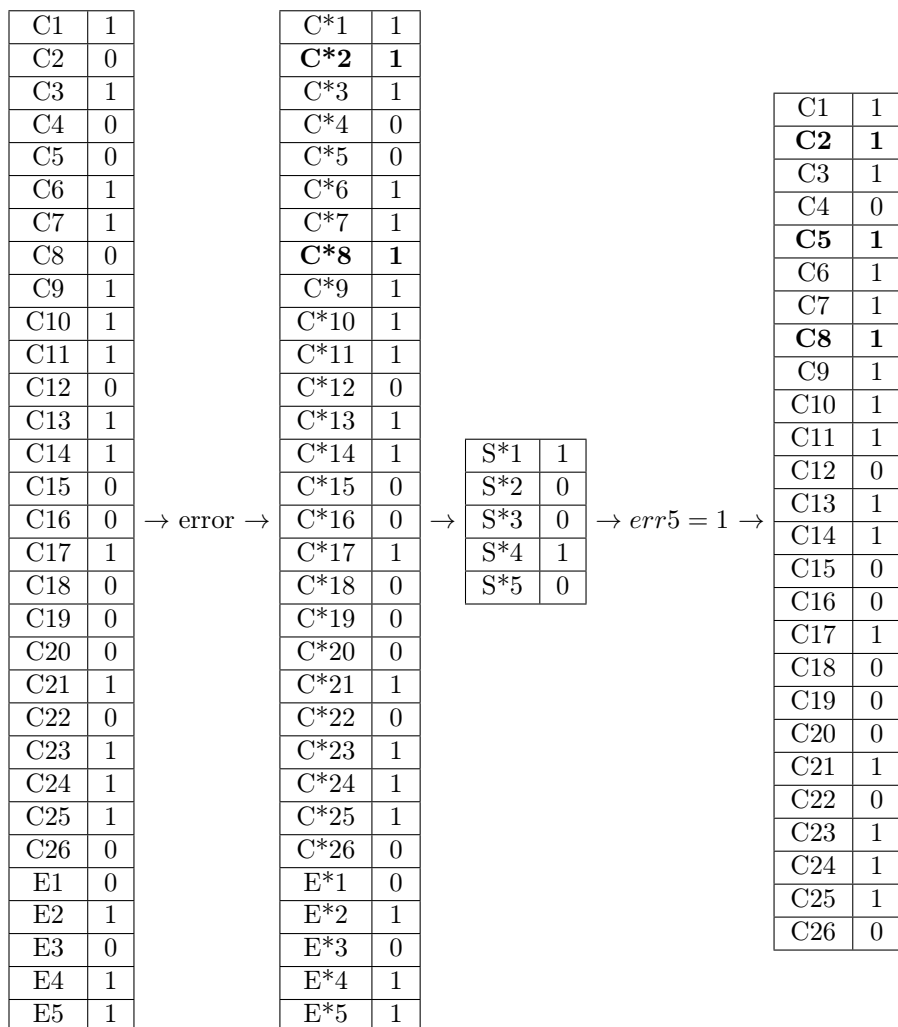
Tā kā visi sindromi ir '0', tad dekoderis (DD82), kurš tos saņem, aktivizēs savu izeju Y0. Bet šo izeju mēs nekur neizmantojam. Visās pārējās izejās (Y1-Y31) tiks padota '0'. No šīm izejām mēs visas neizmantojam - mēs

izmantojam Y3, Y5-Y7, Y9-Y15, Y17-Y31. Šīs izejas tiek savienotas ar attiecīgajiem mezgliem err1, err2-err4, err5-err11, err12-err26, kurus mēs pievienojam dekodera kopnei BUS. No kopnes pēc tam uz XOR ierīcēm DD83-DD89 tiek padoti pa pāriem err1 un C*1, err2 un C*2 utt. Tā kā visi err mezgli ir ar '0', tad tie, tos saskaitot ar C*, nemainīs C* vērtību. Rezultātā DD83-DD89 izejās mēs iegūstam nemainītas saņemtās informācijas vērtības. Un tās arī nebija jāmaina, jo kļūdas netika pielaistas. Nākamais, mēs apskatīsimies kas notiek, ja pieļauta viena kļūda.

Kodešanas process ir tāds pats, tikai šoreiz pārraidē rodas viena kļūda bitā C*2. Tagad aprēķinātais sindroms nav nulle, bet gan $S^* = 00101_2 = 5_{10}$. Tātad šāda sindroma padeve dekoderī DD82 aktivizēs tā piekto izeju Y5, kas savukārt pievienota mezglam err2. Tālāk XOR ierīces DD83 otrie vārti veiks operāciju $err2 \oplus C^*2 = 1 \oplus 1 = 0$. Rezultātā izejā, kur mēs gaidām informācijas bitu C2, mēs saņemsim '0', kas arī bija tā sūtītā vērtība, neskatoties uz to, ka pārraides procesā notika kļūda. Nākamais, paskatīsimies kas notiek, ja rodas divas kļūdas.



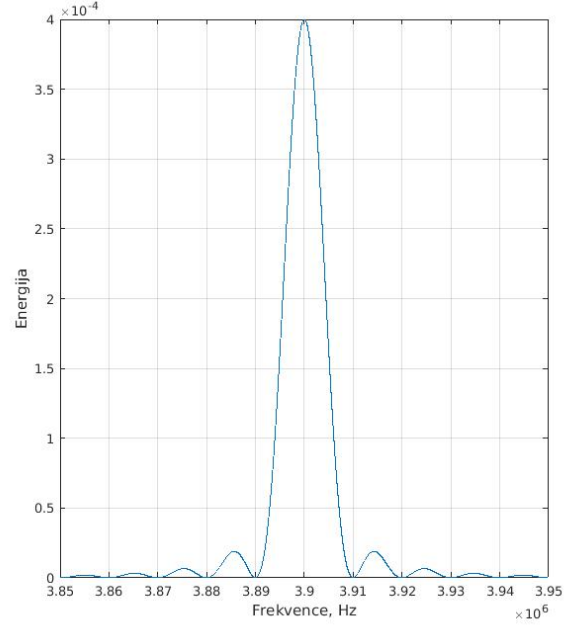
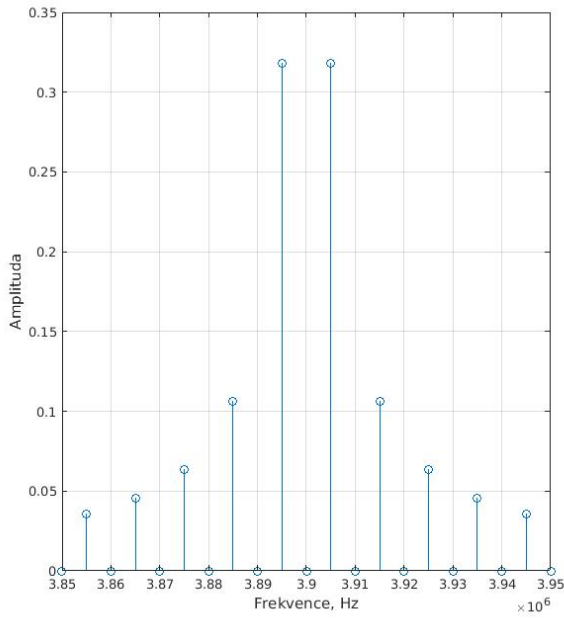
Tagad apskatīsimies kas notiek, ja rodas divas kļūdas. Ja notiek divas kļūdas bitos C*2 un C*8, tad aprēķinātais sindroms mainās uz $S^* = 01001_2 = 9_{10}$ un tas aktivizēs dekodera DD82 Y9 izeju, kas savienota ar err5 mezglu. Līdz ar to XOR ierīces DD84 pirmie vārti izdarīs operāciju $C5 = err5 \oplus C^*5 = 1 \oplus 0 = 1$. Bet C*5 kļūda netika pieļauta. Rezultātā mūsu kopējā dekodera izejā tagad ir trīs kļūdas.



Pastāv likumsakarība, ka, pieļaujot 2 kļūdas sindroms nekad nebūs nulle. Tātad vienmēr varēs saprast, ka kļūda pieļauta. Tomēr atklāt kur kļūda ir notikusi nevarēs. Tāpēc praksē bieži vien šāda veida Heminga grupas kodētāju apvieno ar pārības pārbaudes bitu. Tas pievieno vēl viena bita redundanci, bet dos mums ziņu, vai oriģinālā pārība ir saglabāta. Ja pārība ir saglabāta, bet sindroms nav '0', tad tas nozīmē, ka kļūdu skaits ir lielāks par 1 un ziņojums ir jāpārsūta.

7. SIGNĀLA SPEKTRS MODULATORA IZEJĀ

Mums bija jāizmanto binārā fāzes modulācija (BPSK). Par amplitūdu uzskatām $U = 1\text{ V}$. Simbola ilgums $\tau = 100\text{ }\mu\text{s}$. Nesošā frekvence $f_c = 3.9\text{ kHz}$.



Jāatzīmē ka enerģētiskajā spektrā pašā nesošajā frekvencē ir delta funkcija.

8. TROKŠNOTURĪBAS UZLABOŠANĀS

Mēs varam novērtēt par cik labāka trokšņu noturība mums ir, ja korigējot kļūdas vienā bitā tās vairs nav jāņem vērā. Mēs izmantosim formulu (6.46) no Sllara grāmatas¹.

$$P_B = \frac{1}{n} \sum_{j=t+1}^n j \binom{n}{j} p^j (1-p)^{n-j} \quad (17)$$

Ievietosim šajā formulā mūsu apzīmējumus: $t = g$, $n = m$.

$$P_{err_dec} = \frac{1}{m} \sum_{j=2}^m j C_m^j P_{er}^j (1 - P_{er})^{m-j}$$

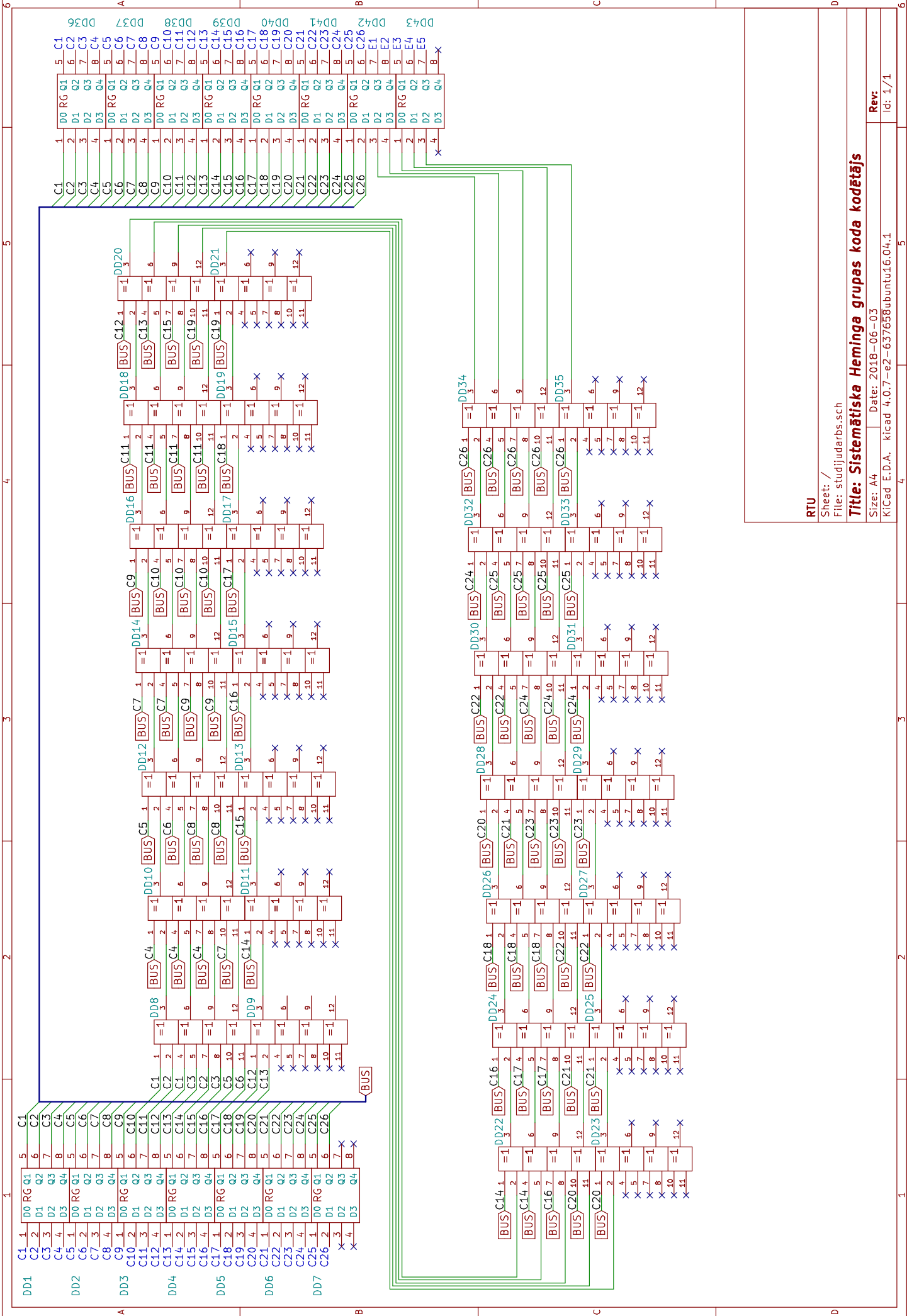
Mums ir dots, ka $SNR = 7.5 \text{ dB}$ un mēs 5. nodaļā jau aprēķinājām ka $P_{er_{PM}} \approx 0.4 \cdot 10^{-3}$. Tā kā korigējamo kļūdu skaits ir $g = 1$ un $m = 31$, tad

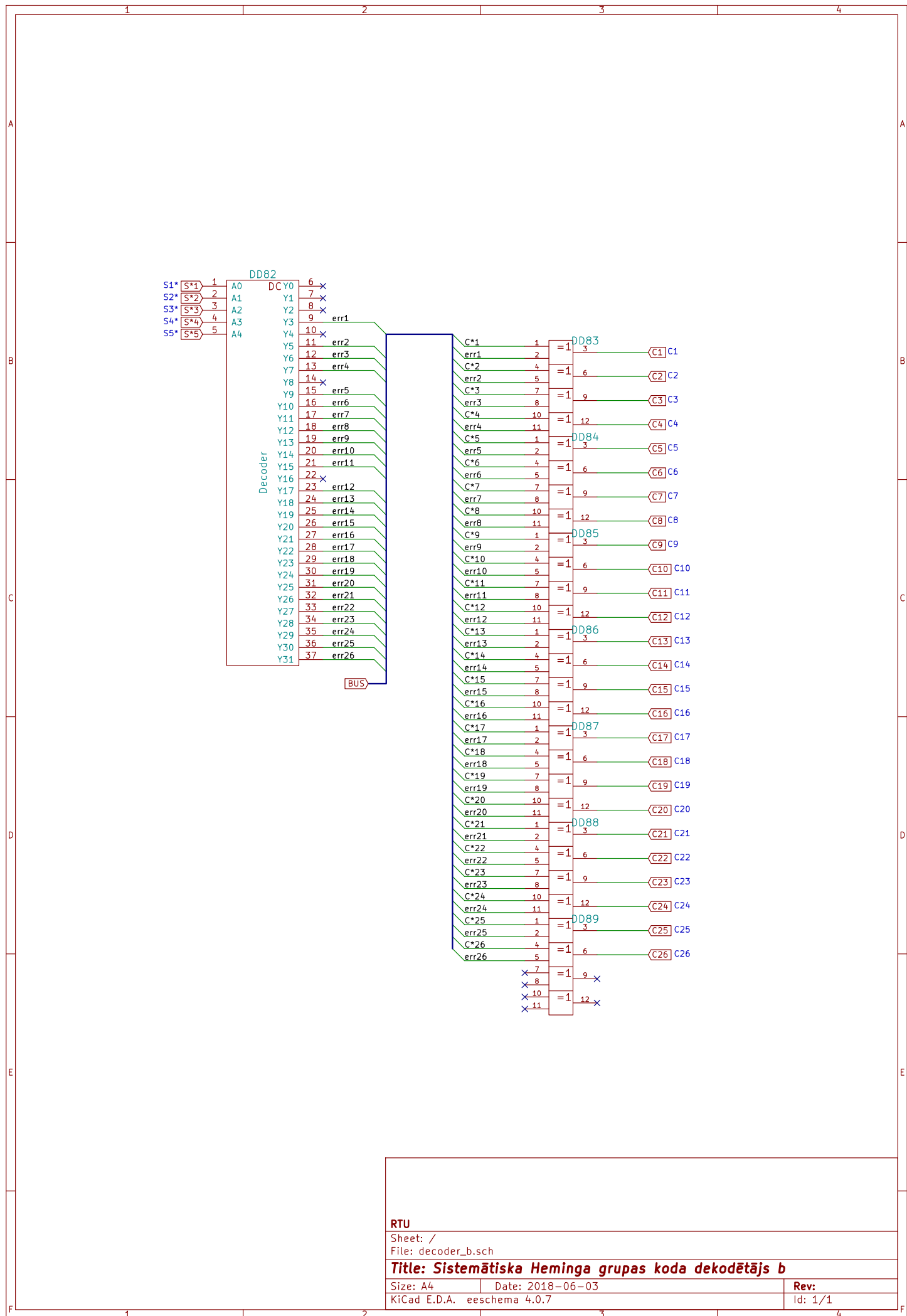
$$P_{err_dec} = \frac{1}{31} \sum_{j=2}^{31} j C_{31}^j P_{er}^j (1 - P_{er})^{31-j} \approx 4.7437 \cdot 10^{-6}$$

Tātad izrādās ka mūsu korigējošais kods kļūdas varbūtību samazina par gandrīz 99%, kas ir ievērojams uzlabojums.

9. PIELIKUMS

¹Bernard Sklar, "Digital Communications", 2014, 2nd ed., p.345.





RTU

Sheet: /

File: decoder_b.sch

Title: Sistēmātiska Heminga grupas koda dekodētājs b

Size: A4

Date: 2018-06-03

Rev:

KiCad E.D.A. eeschema 4.0.7

Id: 1/1