Sekvence za razdvajanje u celobrojnim kodovima

Dragana Bajić
Departman za energetiku, elektroniku i telekomunikacije
Fakultet tehničkih nauka
Novi Sad, Srbija
dragana.bajic@gmail.com

Nikola Zogović, Goran Dimić
Računarski sistemi, Telekomunikacije
Institut Mihajlo Pupin
Beograd, Srbija

nikola.zogovic@pupin.rs, goran.dimic@pupin.rs

Sažetak—**U ovom radu opisuje se kôd definisan na konačnom prstenu $\mathbb{Z}_{p_M}$, gde je moduo $p_M = 2^m - 1$ Mersenov prost broj a m dužina elemenata prstena u binarnim jedinicama. Kôd je zasnovan na sekvenci za razdvajanje $\mathcal{S}$ definisanoj za multiplikativni skup $\mathcal{E} = \{\pm 2^0, \pm 2^1, \dots, \pm 2^{m-1}\}$ čiji elementi predstavljaju vrednosti (težine) onih paterna grešaka koje je moguće ispraviti. Sekvenca za razdvajanje deli kodnu reč u podreči, pa je od toga potekao naziv razdeljeni kôd. Kôd ima osobine skraćivanja kojom se, bez izmene u proceduri dekodovanja, povećava kapacitet ispravljanje grešaka.**

Ključne riječi-sekvenca za razdvajanje, Mersenov broj, celobrojni zaštitni kodovi, DB-SpC kôd

I. UVOD

Klasična teorija zaštitnog kodovanja zasnovana je na postulatima diskretne algebre. Sekvence za razdvajanje su manje poznat koncept diskretne algebra koji je u ovom radu primenjen u zaštitnom kodu definisanom nad konačnim skupovima celih brojeva po modulu Mersenovog broja.

Razdvajanje je proces koji se koristi i definiše u kontekstu Abelovih grupa. Pretpostavimo da je $\mathcal{E}$ konačan skup celih brojeva po modulu i da je $\mathcal{G}$ Abelova grupa. Ako postoji podskup $\mathcal{S} \subset \mathcal{G}$ čiji se svaki nenulti element $g \in \mathcal{G}$ može da prikaže kao proizvod elemenata $\varepsilon \cdot s$ gde element $\varepsilon \in \mathcal{E}$ i gde element $s \in \mathcal{S}$, uz uslov $1 \in \mathcal{E}$, tada skup $\mathcal{E}$ razdvaja skup $\mathcal{G}$ pomoću skupa za razdvajanje $\mathcal{S}$. Skup $\mathcal{E}$ se naziva 'multiplikativni skup', ili 'multiplikativno zatvoren skup', a skup za razdvajanje $\mathcal{S}$ se češće naziva 'sekvenca za radzvajanje', $s_1, s_2, \dots$ [1]. U trivijalnom slučaju skupovi $\mathcal{S}$ i $\mathcal{G}$ se poklapaju, $\mathcal{S} = \mathcal{G}$, i tada je $\mathcal{E} = \{1\}$. Diskretna algebra u prebrojivo beskonačnom prstenu celih brojeva $\mathbb{Z}$ definiše $\{a^0, a^1, a^1, \dots\}, a \in \mathbb{Z}$ kao primer muliplikativnog skupa.

Nema mnogo radova koji se bave ovom značajnom, ali izrazito matematičkom problematikom. Počeci se vezuju za rad G. Hajósa iz 1942 [2] i na docniju generalizaciju u vezi sa grupama koje nisu Abelove [3]. U radu [4] razmatra se faktorizacija semigrupa celih brojeva po modulu u skupove $\mathcal{E}$ i $\mathcal{S}$, gde je $\mathcal{E}$ ravno ili {1, 2, …, k}, ili { ±1, ±2, …, ±k} a isti multiplikativni skupovi se analiziraju i u [5].

Tek je teorija zaštitnog kodovanja omogućila inženjersku primenu sekvenci za razdvajanje. Na primer, u radu [6] razmatra se kôd za korekciju grešaka nastalih usled jednog preskoka (ili ubacivanja) takta, a u radu [7] analiziraju se multiplikatorski skupovi $\{1, a, \dots, a^r, b, \dots, b^s\}$ i $\{\pm 1, \pm a, \dots, \pm a^r, \pm b, \dots, \pm b^s\}$ predviđeni za veće preskoke. Čak je i inicijalni teorijski rad [1], uz razvoj opšte teorije razdvajanja i prateće dokaze, prikazao kao ilustrativni primer ispravljanje asimetričnih grešaka sa organičenom amplitudom karakterističnih za klasične memorije sa jednostrukim upisom. Uprkos tome, fokus radova je na apstraktnim teorijskim konceptima diskretne algebre: kodovi imaju prelepu algebarsku strukturu, ali ispravljaju specifične i retke greške nesvojstvene realnim telekomunikacionim sistemima, i to bez osvrta na ne-matematičke trivijalnosti kao što su potrošnja i brzina realizacije. Nedavno se pojavio sveobuhvatan i detaljan teorijski osvrt sa pregledom mogućnosti za primenu [8].

Sekvence za razdvajanje se nisu koristile u borbi protiv aditivnog Gausovog šuma sve do zaštitnog kôda jednostavne realizacije i niske potrošnje opisanog u [9], [10] i patentiranog u [11], [12] ali bez algebarskog tumačenja i terminologije. Splet okolnosti je odložio teorijsko tumačenje [13], [14] sve do rada [15], kada su kodovi nazvani DB-SpC ('DB splitting codes').

U ovom radu se najpre objašnjava korišćenje sekvence za razdvajanje u prstenu celih brojeva po modulu Mersenovog broja, kao i sama struktura SpC-DB kodova. U jednom pasusu opisuje se hibridna trostepena inkrementalna ARQ ('automatic repeat request') procedura uvedena u [15]. Treće poglavlje opisuje 'teleskopsku skalabilnost' koja je karakteristična isključivo za SpC-DB kodove: skraćivanje kodova bez izmene procedura za kodovanje i ispravljanje grešaka rezultuje povećanjem broja grešaka koje mogu da se isprave bez izmena u koderu i dekoderu. Ova osobina (uz jednostavnu realizaciju, nisku potrošnju i specifične inkrementalne ARQ mogućnosti) predstavlja osnovnu karakteristiku SpC-DB kodova. Četvrto poglavlje je posvećeno rezultatima, zaključnim razmatranjima i mogućnostima za dalju razradu i primenu.

II. KODOVI ZASNOVANI NA SEKVENCAMA ZA RAZDVAJANJE

Svrha SpC-DB kodova je ispravljanje jedne ili više bitskih grešaka u kodnoj reči koja je organizovana kao niz celobrojnih simbola (bajta) iz prstena celih brojeva po modulu $2^m - 1$, a svaki simbol je dužine m bita. Odgovarajući multiplikativni skup je $\mathcal{E} = \{\pm 2^0, \pm 2^1, \dots, \pm 2^{m-1}\}$ a njegovi elementi imaju inženjerski korisnu osobinu da se dobijaju udvostručavanjem prvog elementa skupa, $(\pm 2^0)$, što je ekvivalentno cikličnom pomeraju sadržaja registra u kojem je simbol (bajt) smešten.

Svaki element $\varepsilon_j = \pm 2^j \in \mathcal{E},\ j = 0, \dots, m-1$ predstavlja celobrojnu 'težinu' bidirekcione greške po bitu koja se javi na poziciji $(j+1)$ unutar pogrešno primljenog simbola (bajta). Znak ε_j predstavlja smer greške: pozitivne, 0→1, gde je nula pogrešno detektovana kao jedinica, i negativne, 1→0, gde je jedinica pogrešno detektovana kao nula. Eksponent j prikazuje poziciju pogrešnog bita unutar bajta.

SpC-DB kodovi se definišu nad konačnim prstenom celih brojeva $\mathbb{Z}_{x_M} = \mathbb{Z}_{2^m-1}$ gde moduo x_M predstavlja Mersenov broj $2^m - 1$ [16] a binarni ekvivalenti elemenata prstena sadrže m bita. Mersenov broj može da bude prost, $x_M = p_M$, kompozitan ako je m prost broj a $x_M = c_M$ nije. Ako ni m ni $x_M = n_M$ nisu prosti, Mersenov broj je običan. Treba naglasiti da prema nekim tumačenjima broj može da bude Mersenov (prost ili kompozitan) isključivo ako je eksponent m prost (tj. ne postoje 'obični' Mersenovi brojevi). Dodatno, ako je , $m = 2^p - 1$ gde je p prost broj, i ako je $p_M = 2^{2^p-1} - 1$ prost broj, tada se naziva 'dvostruki Mersenov broj'.

Zaštitni kodovi definisani nad prstenom celih brojeva $\mathbb{Z}_{2^m}$, tj. po modulu 2^m, daleko su brojniji, ali nemaju jednostavnost realizacije, osobine, niti dužine SpC-DB kodova.

Osnovna verzija SpC-DB kôda definiše se za Mersenove proste brojeve p_M, gde je prsten $\mathbb{Z}_{p_M}$ istovremeno polje GF(p_M), a odgovarajuća aditivna Abelova grupa ciklična. Svaki od nenultih elemenata prstena $z_k \in \mathbb{Z}_{p_M}$ predstavlja generišući element prstena jer je aditivni red svakoga od njih ravan p_M.

Kardinalnost multiplikativnog skupa $\mathcal{E} = \{\pm 2^0, \pm 2^1, \dots, \pm 2^{m-1}\}$ koji, kako je već rečeno, odgovara skupu težina jedne bitske greške unutar sumbola iznosi $|\mathcal{E}| = 2 \cdot m$. Pošto je $|\mathbb{Z}_{p_M} \backslash \{0\}| = 2^m$-2, sledi da je kardinalnost skupa za razdvajanje (tj. dužina sekvence za razdvajanje) ravna $|\mathcal{S}| = \frac{|\mathbb{Z}_{p_M} \backslash \{0\}|}{|\mathcal{E}|} = \frac{2^{m-1}-1}{m}$. Prema Maloj Fermaovoj teoremi, ako je m prost broj, tada je razlomak koji definiše $|\mathcal{S}|$ ceo broj, to jest važi $|\mathbb{Z}_{p_M} \backslash \{0\}| = |\mathcal{E}| \cdot |\mathcal{S}|$. Isti zaključak važi i za kompozitne Mersenove brojeve i prsten $\mathbb{Z}_{c_M}$. Drugim rečima, skup $|\mathbb{Z}_{p_M} \backslash \{0\}|$ ima osobinu savršene razdeljenosti. To je važna osobina za SpC-DB kôd koju kodovi definisani nad $\mathbb{Z}_{2^m}$ nemaju. Za ilustraciju, broj informacionih simbola koji se štiti sa dva kontrolna simbola je $|\mathcal{S}| \cdot (2^m$-2): ako je $m = 7$, broj zaštićenih simbola iznosi $9 \cdot 126 = 1134$.

Pošto je $\mathbb{Z}_{p_M}$ konačan prsten celih brojeva, $z_k = k \in \mathbb{Z}_{p_M} \backslash \{0\}, k = 1, \dots, 2^m - 2$. Nadalje, $\mathcal{s}_i \in \mathcal{S}, i = 1, \dots, |\mathcal{S}|$, i $\varepsilon_j \in \mathcal{E}, j = 1, \dots, 2 \cdot m$. Indeksi $k, i,$ i j rezervisani su resprektivno za simbol, element sekvence za razdvajanje i grešku. Množenje po modulu p_M svakog od simbola $z_k \in \mathbb{Z}_{p_M} \backslash \{0\}$ sa greškama ε_j ima kao rezultat različitu permutaciju niza simbolâ z_k. Simboli na istoj poziciji u različitim permutacijama međusobno su različiti usled već pomenutog maksimalnog aditivnog reda elemenata prstena $z_k \in \mathbb{Z}_{p_M} \backslash \{0\}$. Kao posledica, za svaku moguću kombinaciju elementa sekvence za razdvajanje i, težine greške j i informacionog simbola k gde je $i = 1, \dots, |\mathcal{S}|,\ j = 1, \dots, 2 \cdot m$ i $k = 1, \dots, 2^m - 2$, par $(\mathcal{s}_i \cdot \varepsilon_j,\ z_k \cdot \varepsilon_j)$ ima jedinstvenu vrednost. Dokaz je jednostavan i izložen je u radu [15] čime se pokazuje da kôd sa sindromom $(\mathcal{s}_i \cdot \varepsilon_j, z_k \cdot \varepsilon_j)$ može da ispravi grešku težine $\varepsilon_j \in \mathcal{E},\ j = 1, \dots, 2 \cdot m$ na bilo kom bajtu kodne reči.

Pretpostavimo da je informaciona reč sastavljena od $|\mathbb{Z}_{p_M} \backslash \{0\}| \cdot |\mathcal{S}|$ informacionih simbola označenih sa a i da može da se razdeli na $|\mathcal{S}|$ pod-reči u kojima su informacioni simboli označeni sa a_{ik}. Indeksi simbola a_{ik} označavaju k-ti informacion simbol u okviru i-te pod-reči. Iz toga sledi da, zahvaljujući sekvenci za razdvajanje, svaka kodna reč može da se izdeli na $|\mathcal{S}|$ pod-reči dužine $|\mathbb{Z}_{p_M} \backslash \{0\}|$. Kodovanje i formiranje sindroma rade se na sledeći način:

$$C_1 = -\sum_{i=1}^{|\mathcal{S}|} \mathcal{s}_i \cdot \sum_{k=1}^{2^m-2} a_{ik}, \quad (1)$$

$$C_2 = -\sum_{i=1}^{|\mathcal{S}|} \sum_{k=1}^{2^m-2} (2^m - 1 - k) \cdot a_{ik}, \quad (2)$$

$$S_1 = \sum_{i=1}^{|\mathcal{S}|} \mathcal{s}_i \cdot \sum_{k=1}^{2^m-2} \hat{a}_{ik} + \hat{C}_1, \quad (3)$$

$$S_2 = \sum_{i=1}^{|\mathcal{S}|} \sum_{k=1}^{2^m-2} (2^m - 1 - k) \cdot \hat{a}_{ik} + \hat{C}_2, \quad (4)$$

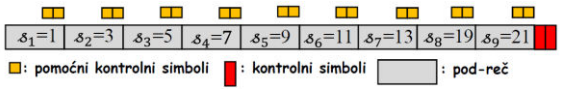
gde su C_1 i C_2 kontrolni simboli dodati informacionoj reči na predaji, "^" označava procenjenu vrednost simbolâ na prijemu a S_1 i S_2 su sindromi. Iz sindroma se izvlači informacija o težini greške ε_j, njenoj poziciji u okviru pod-reči ($z_k = k$), i o pod-reči u okviru koje se greška desila ($\mathcal{s}_i$):

$$(S_1, S_2) = (\mathcal{s}_i \cdot \varepsilon_j, (2^m - 1 - k) \cdot \varepsilon_j) =$$
$$= (\mathcal{s}_i \cdot \varepsilon_j, -k \cdot \varepsilon_j) = (\mathcal{s}_i \cdot \varepsilon_j, -z_k \cdot \varepsilon_j). \quad (3)$$

Međutim, potrebne su tri informacije za ispravljanje greške, ε_j, $\mathcal{s}_i$ i $z_k = k$ a na raspolaganju su samo dva sindroma. Treća informacija je sakrivena u već pomenutoj osobini aditivnih Abelovih grupa da svaka težina greške ε_j predstavlja zatvorenu cikličnu permutaciju težine greške $\pm 2^0 = \pm 1$. Uzastopno množenje težine greške $\cdot \varepsilon_j$ sa dva, ili, ekvivalentno, ciklično pomeranje ('šiftovanje') udesno binarne vrednosti greške na kraju dovodi do vrednosti ± 1. Iz toga sledi da je moguće da se iz broja cikličnih pomeraja $j \in \{0, 1, \cdots, m-1\}$ koji dovode prvi sindrom na vrednost $\pm 1 \cdot \mathcal{s}_i$ odredi težina greške $\cdot \varepsilon_j$. Nakon toga je lako odrediti raspodeljnu pod-reč i u okviru kodne reči, i poziciju k –tog pogrešnog simbola u okviru pod-reči: $\mathcal{s}_i = S_1/\varepsilon_j$ i $k = -S_2/\varepsilon_j$, pri čemu je važno naglasiti da je deljenje modularno.

Iako tema ovog rada nije utrošak procesorskog vremena, na osnovnu jednačina (2) i (4) moglo bi da se zaključi da unutrašnje sume drugog kontrolnog simbola i drugog sindroma zahtevaju $2^m - 2$ sabiranja i $2^m - 2$ množenja. Realna inženjerska implementacija, međutim, zahteva samo $2^m - 2$ sabiranja bez množenja. To je posledica Flečerovog algoritma [17] i predstavlja izuzetnu prednost imajući u vidu da se kodovanje i formiranje sindroma izvode pri svakom prenosu a da kodne reči mogu da budu izuzetno dugačke.

Iz formula se takođe vidi da unutrašnje sume u stvari predstavljaju pomoćne kontrolne simbole i pomoćne sindrome, što je takođe jedinstvena karakteristika. Upravo je na osnovu pomoćnih kontrolnih simbola koji se uvek izračunavaju, nezavisno od toga da li se koriste ili ne, uvedena hibridna trostepena inkrementalna ARQ procedura koja u prvom koraku



Slika 1. Kodna reč za m=7 sa naznačenim elementima multiplikativnog skupa $\mathcal{S}_i$, $i = 1, \dots, 9$. Svaka pod-reč može da ima različitu dužinu; maksimalna dužina svake pod-reči je $2^m - 2 = 126$ simbola (simbol za m=7 je sedmobitni bajt).

prenosi celu kodnu reč, u slučaju detektovane greške prenosi pomoćne kontrolne simbole, a u trećem koraku prenosi samo one delove kodne reči koji su označeni kao neispravni. Dodatno, u svakom od pomenutih koraka može da se isključi retransmisija i radi ispravljanje grešaka, bilo na celoj kodnoj reči, bilo na pod-rečima [15].

Na Sl. 1 prikazana je kodna reč maksimalne dužine za $m = 7$ i $\mathcal{S} = \{1, 3, 5, 7, 9, 11, 13, 19, 21\}$. Naznačeni su i pomoćni kontrolni simboli koji nisu sastavni deo kodne reči.

III. TELESKOPSKO SKRAĆIVANJE SPC-DB KÔDA

A. Prost Mersenov broj

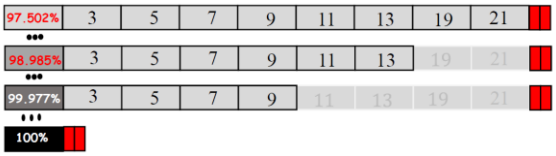
Pun kapacitet kôda se postiže kada je moduo p_M prost Mersenov broj. Za razliku od klasičnih kôdova gde skraćivanje uz nepromenjen dekoder omogućava mogućnost pojačane detekcije, ovde je omogućena jača korekcija grešaka bez ikakvih, ili uz minimalne izmene u dekoderu. U nastavku teksta prikazaćemo nekoliko mogućnosti.

1) *Ispravljanje više grešaka bez skraćenja kôda*

Najčešće greške su usamljene i statistički nezavisne, bilo zato što ih unosi kanal, bilo zato što su 'paketi' grešaka razbijeni interlivingom pa je tako izabran i inicijalni multiplikativni skup. Međutim, može se desiti da neki paterni grešaka postanu veća smetnja od usamljenih grešaka. U takvim slučajevima multipikativni skup može da se promeni u neku od preostalih mogućih vrednosti $\mathcal{E}_i = \{\pm 2^0 \cdot \mathcal{s}_i, \pm 2^1 \cdot \mathcal{s}_i, \dots, \pm 2^{m-1} \cdot \mathcal{s}_i\}$, $i = 1, \cdots, |\mathcal{S}|$. Vodeći element $\mathcal{s}_i$ se bira na osnovu sličnosti sa paternima grešaka koje se javljaju na kanalu. Time se broj bitskih grešaka koje se ispravljaju povećava jer binarni ekvivalenti elemenata novog multiplikativnog skupa sadrže više jedinica. Cena ovakve izmene je izgubljena mogućnost za ispravljanje usamljenih (jednostrukih) grešaka po bitu.

2) *Asimetrično ispravljanje grešaka eliminacijom pod-reči*

Ako se iz 'običnog' kôda eliminiše proizvoljna pod-reč X (Sl. 2), skup prvih sindroma $(\mathcal{s}_X \cdot \varepsilon_j)$ ostaje neiskorišćen i vrednosti $\mathcal{s}_X \cdot \varepsilon_j$ postaju nove težine greške koje mogu da se isprave. U tom slučaju je drugi sindrom $(-\mathrm{z}_k \cdot \mathcal{s}_X \cdot \varepsilon_j)$. Ako se greška sa težinom $\mathcal{s}_X \cdot \varepsilon_j$ desi na prvoj pod-reči posmatrane kodne reči za koju je $\mathcal{s}_X = 1$, tada neizmenjena procedura za ispravljanje grešaka na prvoj pod-reči ispravlja obe težine grešaka $-$ i ε_j i $\mathcal{s}_X \cdot \varepsilon_j$ $-$ uz praktično neizmenjenu proceduru ispravljanja grešaka [15]. Ispravljanje grešaka postaje asimetrično u odnosu na pod-reči jer prva reč postaje zaštićenija. U opštem slučaju bilo koja pod-reč može da postane više zaštićena, ali realizacija je najjednostavnija za prvu. Sl. 2 prikazuje primere skraćivanja i povećanje procenta ispravljenih grešaka pod pretpostavkom Gausovog šuma. Cena ovog postupka je smanjenje kodnog količnika.



Slika 2. Skraćivanje kodne reči za m=7 i rast procenta asimetrično ispravljanih grešaka na prvom bajtu. U prvom redu je neskraćena kodna reč, u četvrtom redu je kodna reč svedena na jednu pod-reč.

3) *Ispravljanje svih težina grešaka na simbolu*

Kada se sve pod-reči eliminišu sem prve, vrednosti $\mathcal{s}_X \cdot \varepsilon_j$ pokrivaju se moguće vrednost greške koje mogu da se dese i kôd dobija sposobnost da ispravi sve greške koje se dese na jednom simbolu (bajtu) kodne reči (prikazano u četvom redu Sl. 2). U tom smislu kôd je poredljiv sa Rid-Solomonovim (RS) kôdom koji ispravlja jedan pogrešan simbol. Jedina razlika je u tome što je RS kôd $(2^m - 1, 2^m - 3)$ i definisan je nad GF(2^m), a maksimalno skraćeni SpC-DB kôd je $(2^m, 2^m - 2)$ i definisan je nad GF($2^m - 1$), uz značajno jednostavniji postupak ispravljanja greške [15]. Cena postupka je smanjenje kodnog količnika u odnosu na izvorni SpC-DB kôd.

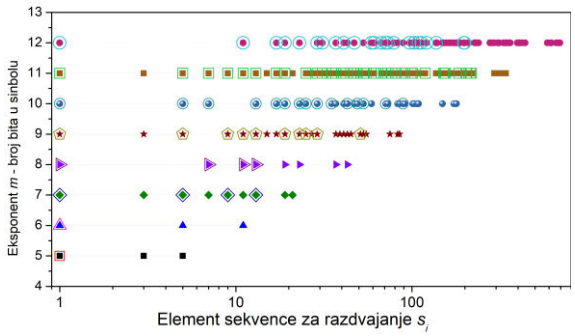
4) *Produžavanje kodne reči*

Svaka pod-reč ima svoje pomoćne kontrolne simbole koji, kada bi se prenosili, mogu da isprave grešku na jednom celom bajtu pod-reči. Na osnovu toga sledi da kodna reč može i da se produži, a ne samo skrati, i to bez dodatno utrošenog procesorskog vremena na predaji: dovoljno je uz kodnu reč dodati i pomoćne kontrolne simbole. U prijemniku se tada 'obični' kontrolni simboli koriste za detekciju greške, i tek ako se greška detektuje, pomoćni kontrolni simboli se koriste za testiranje pod-reči sa mogućnošću da se u okviru svake pod-reči ispravi po jedan pogrešan bajt. Alternativno, moguće je uraditi dodatni prenos pomoćnih kontrolnih simbola ako se u prijemiku detektuje greška. Cena je opet smanjenje kodnog količnika.

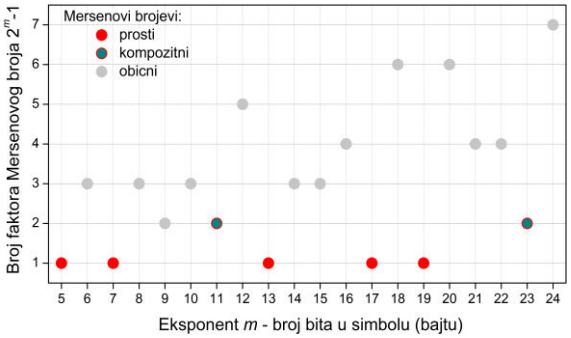
5) *Povećane kardinalnosti multiplikativnog skupa*

Multiplikativni skup može da se udvostruči, utrostruči … Inicijalni multiplikativni skup podešen je da odgovara težinama jedne greške po bitu u okviru simbola, ali može da dobije i oblik: $\mathcal{E}_{1i} = \{\pm 2^0, \pm 2^1, \dots, \pm 2^{m-1}, \pm \mathcal{s}_i \cdot 2^0, \pm \mathcal{s}_i \cdot 2^1, \dots, \pm \mathcal{s}_i \cdot 2^{m-1}\}$.

U tom slučaju mora da se uradi pretraga koja bi odredila novi skup za razdvajanje a kôd gubi osobinu asimetrične savršenosti pokazanu u [15]. Ako elementi skupa za razdvajanje kojima se proširuje multiplikativni skup imaju netrivijalne zajedničke delioce sa Mersenovim brojem, potrebno je proveriti da li pod-reč može da bude maksimalne dužine $2^m - 2$, ili mora da se skrati. Na Sl. 3 zaokruženi su elementi sekvence za radvajanje koji se zadržavaju nakon povećanja multiplikativnog skupa.



Slika 3. Elementi sekvence za razdvajanje za multiplikativne skupove kardinalnosti $2m$ i $4m$ (zaokruženi), prikazani za sve Mersenove brojeve eksponenta do 12



Slika 4. Faktorizacija Mersenovih brojeva

B. .Kompozitni i obični Mersenovi brojevi

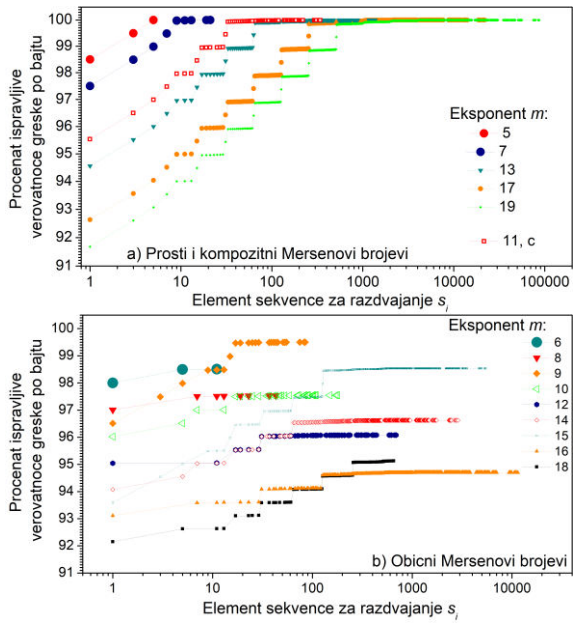
Moduo ne mora da bude prost Mersenov broj, mada kodovi definisani nad običnim i kompozitnim Mersenovim brojevima imaju nešto slabije performanse. Ako je Mersenov broj kompozitan, tada je eksponent m (dužina simbola u bitima) prost. Fermaova mala teorema i dalje važi, tako da je prsten celih brojeva po modulu savršeno razdeljen. Međutim, neki od elemenata sekvence za razdvajanje (i celog prstena) nemaju maksimalni red pa u odgovarajućim pod-rečima drugi sindromi ne moraju da budu jedinstveni. Takve pod-reči moraju da se eliminišu i kôd se skraćuje. Pored toga, kôd skraćen na jednu kodnu reč ne može da ispravi sve greške na bajtu, a tu mogućnost ne pružaju ni kontrolni simboli na pod-rečima. Međutim, budući da svaka pod-reč može da ima različitu dužinu, po potrebi se pod-reči mogu skratiti tako da kod zadrži i pod-reči koje bi se inače izbacile. Skraćivanjem prve pod-reči se omogćava i ispravljanje svih grešaka na bajtu. Cena je dodatno skraćenje kôda.

Obični Mersenovi brojevi nisu prosti a nemaju ni prost eksponent pa, kao i za kompozitne brojeve, neki od elemenata sekvence za razdvajanje nemaju maksimalan red. Fermaova mala teorema ne važi, pa proizvod elemenata sekvence za razdvajanje i elemenata multiplikativnog skupa ne mora da bude jedinstven (postoje ponavljanja u multiplikativnom skupu). Iz tog razloga se dodatno izbacuju pod-reči i kodni količnik se još više smanjuje. Podrazumeva se da ne postoji mogućnost ispravljanja jednog celog bajta. Sve ostale osobine kodova definisanih nad prostim Mersenovim brojevima važe.

IV. REZULTATI I ZAKLJUČAK

SpC-DB kôd može da se koristi za sve Mersenove brojeve, ali performance zavise od vrste Mersenovog broja, odnosno od broja netrivijalnih faktora na koje je Mersenov broj moguće razložiti, što je prikazano na Sl. 4. Za kôd je već rečeno da ispravlja težine grešaka, tako da za svaki bajt može da se izračuna procenat ispravljivih verovatnoćâ grešaka. Na Sl. 5 je prikazano povećanje ovog procenta omogućeno sukcesivnim izbacivanjem pod-reči. Na apscisi je navedena vrednost elementa sekvence za razdvajanje koji odgovara izbačenoj pod-reči.

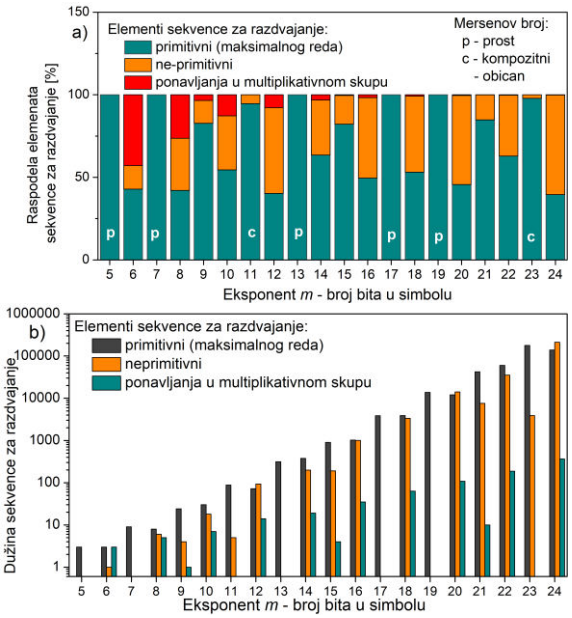
Sl. 5a prikazuje verovatnoće za proste i kompozitne Mersenove brojeve, a Sl. 5b za obične Mersenove brojeve. Može se uočiti da kodovi definisani i nad prostim i nad kompozitnim



Slika 5. Procenat ispravljivih verovatnoćâ greške po bajtu. a) Prosti i kompozitni Mersenovi brojevi; b) Obični Mersenovi brojevi

Mersenovim brojevima imaju sličan obrazac ponašanja i da sa određenim procentom odbacivanja pod-reči može da se postigne skoro stopostotna zaštita bajta određene pod-reči. Sl. 5b pokazuje da kodovi nad običnim Mersenovim brojevima to ne postižu. Dobre performanse ima kôd za m=9, moguće jer moduo 511 ima samo 2 netrivijalna faktora (7x73). Međutim, kodovi sa m=14 i m=15 oba imaju isti broj faktora, a performanse im se razlikuju.

Moguće tumačenje različitog ponašanja ova dva kôda prikazano je na Sl. 6a i Sl. 6b koje prikazuju relativnu i apsolutnu raspodelu broja različitih elemenata raspodeljene sekvence. Sl. 6a pokazuje da je za m=15 broj elemenata koji se ponavljaju množenjem sa multiplikativnim skupom procentualno zanemariv, a da je za m=14 taj broj značajno veći. Dodatno, m=14 ima i veći procenat ne-primitivnih elemenata. To je lakše uočiti iz raspodele u procentima (Sl. 6a), nego iz raspodele u apsolutnim brojevima (Sl. 6b).



Slika 6. Raspodela elemenatat sekvence za razdvajanje na primitivne, neprimitivne i sa ponavljanjem. a) procentualna, b) apsolutna

Navedena tumačenja su opisna. Sledeći korak, i dalji rad, usmereni su ka formalnim teorijskim tumačenjima performansi raspodeljenih sekvenci, i optimizaciju rada kôda za m=8, 16, 24 i 32 što odgovara realnim dužinama (proširenih) bajta.

Realizacija ovog kôda je veoma jednostavna a potrošnja minimalna, tako da omogućava velike uštede, a to je nekoliko puta i pomenuto u ovom radu. Međutim, o tome ne pišemo jer je objavljeno i patentirano pa ne predstavlja nove rezultate [15].

ZAHVALNICA

Ovaj rad je podržan od strane Fakulteta tehničkih nauka u Novom Sadu, Departmana za energetiku elektroniku i telekomunikacije, u okviru realizacije projekta u 2025. godini pod nazivom: "Razvoj i primena savremenih alata i metoda istraživanja u energetici, elektronici i telekomunikacijama".

LITERATURA

[1] S. Buzaglo, T. Etzion, "Tilings with n-dimensional chairs and their applications to asymmetric codes", IEEE Trans. Inform. Theory, vol. 59, pp. 1573–1582, 2013.

[2] G. Hajos, "Über einfache und mehrfache bedeckung des n-dimensionalen raumes mit einem wurfelgitter". Math. Z. vol. 47, pp 427–467, 1942.

[3] A.D. Sands, "On the factorization of finite groups". J. Lond. Math. Soc. vol. 7, pp. 627–631, 1974.

[4] S. Stein, "Factoring by subsets". Pac. J. Math. vol. 22, pp. 523–541. 1967.

[5] S. Stein, "Tiling, packing, and covering by clusters". Rocky Mt. J. Math. vol. 16, pp. 277–322, 1986.

[6] V.I. Levenshtein, A.J.H. Vinck, "Perfect (d; k)-codes capable of correcting single peak shifts". IEEE Trans. Inform. Theory, vol. 39, pp. 656–662, 1993.

[7] U. Tamm, "Splittings of cyclic groups and perfect shift codes". IEEE Trans. Inform. Theory 1998, vol. 44, pp. 2003–2009.

[8] Zhao, K. "The complete splitting of finite abelian groups". arXiv 2020, arXiv:2003.13290.

[9] D. Bajic, A. Burr: "A Simple Suboptimal Integer Code", International Symposium on Information Theory and its Applications, ISITA2004, Parma, Italy, pp 1315-1320, October 10–13, 2004.

[10] D. Bajic, C. Stefanovic: "Low Power Consuming, FEC Scheme" Proceedings of International Workshop on Optimal Codes and Related Topics 2005, 17.-23. 06 2005, pp 07-13, Pamporovo, Bulgaria, June 2005.

[11] D. Bajic: "Postupak za ispravljanje grešaka na združenim paketima nejednake dužine celobrojnim kodom niske potrošnje". Registarski broj 54806, Broj i datum rešenja o priznanju prava 2016/8250 od 01.08.2016, Zavod za intelektualnu svojinu Republike Srbije ("veliki patent"), isteklo 2025.

[12] D. Bajic: "Hibridni postupak ispravljanja grešaka i selektivne retransmisije pri združenom paketskom prenosu paketa nejednake dužine korišćenjem celobrojnog koda niske potrošnje". Registarski broj 54807, Broj i datum rešenja o priznanju prava 2016/8251 od 01.08.2016, Zavod za intelektualnu svojinu Republike Srbije ("veliki patent"), isteklo 2025.

[13] D. Bajic, A. Burr, "Comments on "Integer SEC-DED codes for low power communications"", Information Processing Letters ISSN: 0020-0190 Vol. 111, No. 9, Str. 414-415, 2011.

[14] https://drive.google.com/file/d/19120D_pylff8qB1sUJw91zirrM9Wb2s-/view?usp=drive_link

[15] D. Bajic, G. Dimic, N. Zogovic, "Splitting Sequences for Coding and Hybrid Incremental ARQ with Fragment Retransmission" Mathematics Vol. 9, No. 20, 22 pages, ISSN: 2227-7390, Section Network Science, Special Issue Advanced Coding and Stochastic Signal Processing in Dense Communication Networks, 2021.

[16] Mersenne Primes: History, Theorems and Lists. Available online: https://primes.utm.edu/mersenne/ (pristup 5. decembra 2024).

[17] J.G. Fletcher, "An arithmetic checksum for serial transmission". IEEE Trans. Commun. vol. 30, pp. 247–252, 1982.

ABSTRACT

This paper proposes a code defined on a finite ring $\mathbb{Z}_{p_M}$, where $p_M = 2^m - 1$ is a Mersenne prime, and m is a binary size of ring elements. The code is based on a splitting sequence (splitting set) $\mathcal{S}$, defined for the given multiplier set $\mathcal{E} = \{\pm 2^0, \pm 2^1, \ldots, \pm 2^{m-1}\}$. The elements of $\mathcal{E}$ correspond to the weights of binary error patterns that can be corrected, with the bidirectional single-bit error being the representative that occurs the most. The splitting set splits the code-word into sub-words, which inspired the name splitting code.

Splitting sequences in integer codes

Dragana Bajic, Nikola Zogovic, Goran Dimic