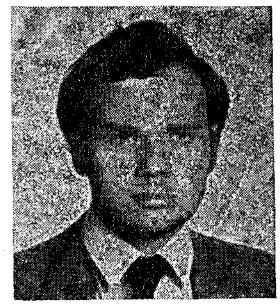
Jelsorozatok szinkronizálása vesszőmentes kódokkal



DR. HUSZTY GÁBOR

Posta Kísérleti Intézet

ÖSSZEFOGLALÁS

A digitális jelsorozatok átviteléhez szükséges keretezési, keretszinkronizálási eljárások összefoglalása után a cikk elemzi a szinkronizálható kódok egy lényeges csoportjának, a vesszőmentes tulajdonságú kódok csoportjának tulajdonságait. Megadja az újonnan bevezetett szegmens kódok néhány jellemzőjét. Részletesen elemzi a gyakorlat szempontjából előnyös redukált szegmens kódok tulajdonságait. Egzisztencia tételeket állít fel a kód létezésére vonatkozóan és bemutatja a kód előállításának lehetőségeit is. A cikket a redukált szegmens kódok gyakorlati alkalmazási példája, a budapesti távközlő hálózatban üzemelő távfelügyeleti adatgyűjtő rendszer néhány jellemzőjének összefoglalása zárja.

DR. HUSZTY GÁBOR

A BME Híradástechnikai Szakán végzett 1976-ban, azóta a Posta Kísérleti Intézetben dolgozik. PCM rendszerek és digitális hálózatok kérdéseivel, távközlő hálózatok központosított felügyeleti rendszereinek problémáival foglalkozott. 1978-ban Pollák-Virág díjat kapott. 1979 óta képviseli a Magyar Postát a CCITT XVIII., a digitális hálózatokkal foglalkozó Tanulmányi Bizottságában. 1985-ben a Budapesti Műszaki Egyetemen műszaki doktori címet szerzett. Szakmai érdeklődési körébe az Integrált Szolgáltatású Digitális Hálózatok és a központosított felügyeleti rendszerek kérdései tartoznak.

1. Bevezetés

A digitális távbeszélő átkérő hálózat létesítése során a Posta Kísérleti Intézet létrehozta a Magyar Posta első, központosított felügyeleti célú adatgyűjtő hálózatát. A különleges peremfeltételek között működő adatgyűjtő hálózatban egyszerű, de nagy megbízhatóságú adatátviteli módszert alkalmaztunk. A módszer alapja az itt részletesen vizsgált redukált szegmens kódnak nevezett, igen előnyösnek bizonyuló, vesszőmentes tulajdonságú kód.

A továbbiakban át kívánjuk tekinteni azokat a különleges kódolási eljárásokat, melyek előnyösen segítik a továbbítandó jelsorozat (keret-) szinkronizált átvitelét. Részletesen elemezzük az itt bevezetett újszerű, ún. redukált szegmens kódok jellemzőit, létrehozásuk módszereit. Bemutatjuk a kódok gyakorlati alkalmazási lehetőségeit. A jobb áttekinthetőség érdekében a kapcsolódó tételek bizonyításait a Függelékben adjuk meg.

2. Szinkronizálás és keretezés

Vizsgáljuk meg, hogyan lehet a továbbítandó információt, tehát az üzenetforrások szimbólumsorozatait olyan egységekbe, keretekbe szervezni, melyek a vételi oldalon egyértelműen szétbonthatók, tehát megállapítható a keretek eleje és vége. Ebből a szempontból az ismert megoldások lényegében két csoportba sorolhatók:

A) a továbbítandó információt kiegészítő algoritmusok (protokollok) és redundáns szimbólumok segítségével eredeti formájukban továbbító (pl. PCM rendszerek);

B) az információt átkódoló, a csatornához illesztett kódot alkalmazó, és így az információt nem eredeti formájukban továbbító eljárások.

Az „A" csoport a szisztematikus eljárásokat, a „B" a nem szisztematikusokat tartalmazza. Úgy tűnhet, hogy a szisztematikus eljárások szükségképpen feltételeznek valamilyen kiegészítő algoritmust is, tehát ez az üzenetforrás szimbólumainak eredeti formában való továbbításának az ára.

A gyakorlat szempontjából igen előnyös lehet olyan szisztematikus kódolási eljárások keresése, amelyeknél kihasználható a szimbólumok változatlan formájú átvitelének előnye és a kódolási-dekódolási folyamatnak a kiegészítő algoritmusokkal szembeni egyszerűsége is. Látni fogjuk, hogy a vesszőmentes tulajdonságú kódok egyes csoportjai éppen ilyenek.

2.1. A szinkronizálásról általában

A szinkronizálás fogalmának említésekor a digitális technika kérdéskörében három nagy témakörre szokás gondolni. A bit (vagy óra) szinkronizáció kérdéseire, a karakter-, szó-, byte-, keret-, blokkszinkronizáció, tehát itt használt terminológiánk szerint az információ keretezés tárgykörére, végül a távközlő hálózatokban üzemelő digitális berendezések együttműködését biztosító hálózatszinkronizáció területére.*

A továbbiakban tárgyalásainkat az információ keretezés témakörére korlátozzuk, a másik két tárgykört illetően a széles körű irodalomra utalunk [1], [2], [3], [4], [5], [6], [7], [8].

2.2. Az információ keretezéséről

Célkitűzésünknek megfelelően rövidebben szólunk a markert használó információ keretezési eljárásokról, és részletesebben vizsgáljuk a különleges, tehát a markert nem tartalmazó, illetve speciális blokk kódolást alkalmazó eljárásokat.

Beérkezett: 1985. XII. 2. (□)

* Az analóg átviteltechnika vivőszinkronizálási kérdéseit itt nem vizsgáljuk.

A markert alkalmazó eljárások közé kell sorolni az elterjedten használt adatátviteli rendszerekben alkalmazott módszerek legtöbbjét, ahol az egyébként speciális tulajdonságokkal felruházott, kódolt információt (ciklikus kódok, konvolúciós kódok stb.) egy megfelelő jelzősorozat (marker, flag stb.) periodikus beiktatásával vagy keret kezdési-lezárási funkcióval szinkronizálják (pl. HDLC) [9], [10], [11].

Speciális markert alkalmaznak a főként távbeszélő célú digitális multiplex és PCM multiplex berendezéseknél is. Kétféle marker használata terjedt el: az európai (2048 kbit/s-os) hierarchia rendszereinél a marker a jelfolyamba (egy vagy több helyen) beiktatott néhány bitből álló minimális utánzási tulajdonságú sorozat, más néven blokkszinkronszó [1]. A blokkszinkronizálás másik módszerét, egyenletesen elosztott szinkronbitek beiktatását a jelfolyamba, az amerikai—japán (1544 kbit/s) hierarchiában alkalmazzák [12]. A PCM jelfolyamok ezektől eltérő, a jelstatisztika ismert tulajdonságain alapuló, a gyakorlatban végül is nem alkalmazott módszerét Gray és Pan tárgyalta egy korai munkájában [13].

A PCM és digitális multiplexerek keret-(blokk) szinkronizációs tulajdonságait a Markov-láncok elméletén alapuló vizsgálatok tárgyalják [14], [15], [16], [17].

A markert nem használó, a szinkronizálható blokk kódokkal való információ keretezési eljárások irodalma szerteágazó.

A vesszőmentes kódok (definíciót ld. 5.1. pontban) matematikai megalapozása Golomb és társai nevéhez fűződik, akik 1958-ban négyféle nukleotid láncbakapcsolódásából kialakult 20-féle nukleinsav kódolásáról igazolták, hogy azok valóban vesszőmentes kódot alkotnak, és ezzel biztosítják a kromoszómák genetikus információinak egyértelmű dekódolhatóságát [18].

A vesszőmentes kódok továbbfejlesztéseként Kendall és Reed dolgozatában [19] szerepelt elsőként a vonalinvariáns vesszőmentes kódok fogalma (ld. 2. ábra), és a kód néhány lényeges tulajdonságának (szavak száma, kódgenerálás) elemzése.

Gilbert 1960-ban közölte alapvető dolgozatát a bináris üzenetek szinkronizálásáról [20]. A vesszőmentes kódok számos előnyös tulajdonságuk mellett igen körülményesen generálhatók, ezért Gilbert olyan módszert javasolt, amellyel a lényegesebb jó tulajdonságok megtartása mellett egyszerűbb módon építhetők fel ilyen kódok. A kód, mely prefix kód néven is ismertté vált, a keretezési eljárások egyik jól használható eszköze lett. Megjegyezzük, hogy a kód más neveken is előfordul, pl. [21] prefixált vesszőmentes kódnak nevezi. Gilbert említett munkájában [20] felvetette egy olyan kód gondolatát is, mely a vesszőmentes tulajdonságok, ill. a prefix kód tulajdonságok megtartásával tovább egyszerűsíti és szisztematikussá teszi a kódot. Ez a kód, melyet itt Gilbert- kódnak nevezünk, a most bevezetendő ún. szegmentált kódok alapja.

Nagyjából ebben az időben publikálta Mühlrád és Dénes azt a cikkét, melyben magyar nyelven a legelsők között vizsgálta a vesszőmentes kódokat [22], [23].

A marker nélküli keretezési módszerek blokk kódokra kiterjesztett változatát Stiffler vizsgálta részletesen [24], [25]. Az ún. PSK szinkronizálható blokk kódokkal Eastman és Even foglalkozott [26], [27].

A vesszőmentes, vonalinvariáns vesszőmentes, ill. vesszős kódok tulajdonságait Golomb és Gordon elemezték [28], vezették be a szinkronizációs késleltetés és a véges szinkronizációs késleltetésű kódok fogalmát is, mely mint általános érvényű tulajdonsággal rendelkező kód-osztály, valójában magában foglal gyakorlatilag minden (nem végtelen késleltetéssel) szinkronizálható kódot.

A vesszőmentes kódok egy lehetséges előállításának algoritmusát, páratlan számú szimbólumból álló kódszavak esetére Eastman adta meg [29].

A szinkronizálható kódok — ill. véges szinkronizációs késleltetésű kódok — előállításához Scholtz javasolt egy ún. szuffix konstrukciós eljárást [30] és [31], melyet Stiffler [3] prefix konstrukciós eljárásnak nevez.

Ezekkel az erőfeszítésekkel egy időben egy másik irányvonal is kialakult: a már régebbről ismert előnyös tulajdonságú kódok olyan átalakítását alkalmazták, melyekkel a kedvező tulajdonságok megtartása mellett igen jó szinkronizációs viselkedést tudtak elérni. Ennek a munkának az eredménye nyomán alakultak ki a hibajavító szinkronizálható kódok, illetve a szinkronhibát is javító kódok [11], [32], [33], [34], [35], [36], [37], [38]. Ezt a kérdéskört itt nem elemezzük.

3. Jelsorozat keretezési eljárások

Tekintsünk elsőként egy olyan egyszerű példát, mely alkalmas a felmerülő kérdések, problémák szemléltetésére.

Az adó eszköz (továbbiakban csak az adó, ill vevő kifejezéseket használjuk) továbbítson 2 bitből álló sorozatokat, melyeket az alábbi szótárból lehet válogatni:

Keret jele	Sorozat elemei
k	00
l	01
m	10

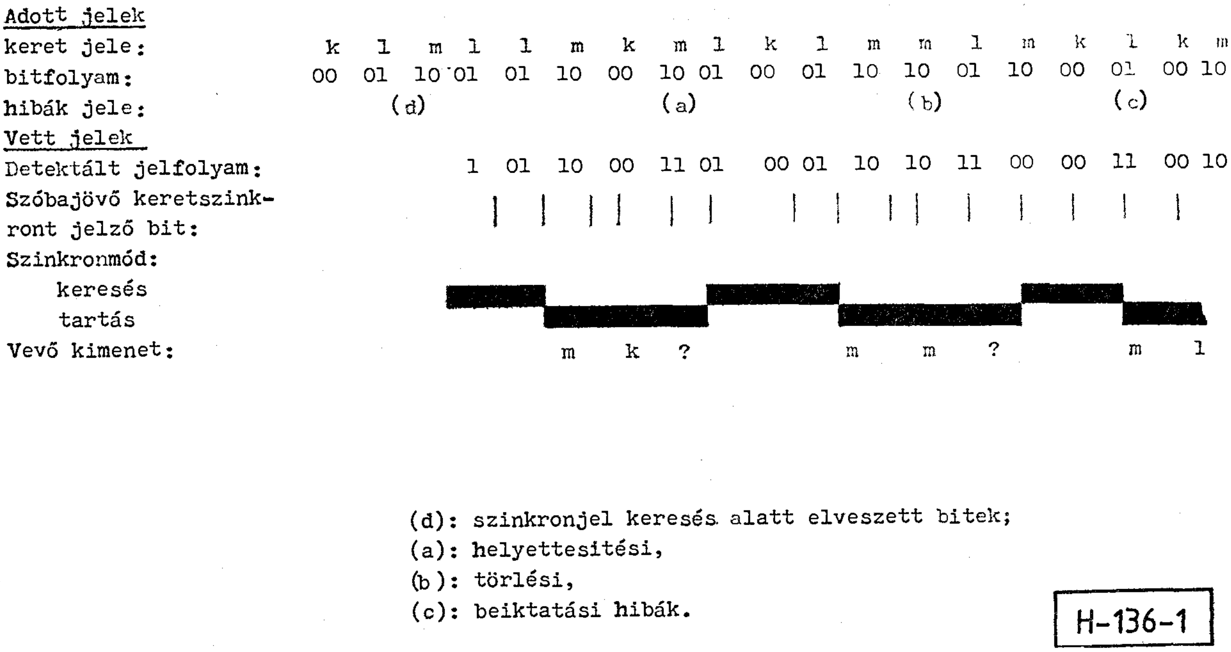
Az 11 sorozatot kizárjuk a választható szavak közül, mert ezt kívánjuk felhasználni a keretezéshez.* Észre kell viszont vennünk, hogy az 11 részsorozat mégis előállhat, ha az „*l*" és az „*m*" sorozat egymás mellé kerül.

Válasszunk most valamilyen egyszerű keretszinkronizálási algoritmust, mely alkalmas a viszonyok szemléltetésére: a keretszinkronizmust két szomszédos „1" értékű bit jelzi, a szinkronizált keret elejét, ill. a végét a második „1" előtti, ill. utáni óraimpulzus jelzi.

Ha az impulzus sorozatok zajos környezetben jutnak el a vevőhöz, úgy a zaj többféle hibát is okozhat a szinkronizmusban:

a) helyettesítési hibát
(például 0 helyett 1-et vesz a vevő);
b) törlési hibát
(például 01 helyett csak 0-t vesz a vevő);
c) beiktatási hibát
(például 1 helyett 11-et vesz a vevő).

* Látható, hogy a keretezés igénye máris csökkentette a felhasználható jelelemek számát: a továbbítható információ mennyiségét, vagyis növelte a redundanciát.



1. ábra. Keretszinkronizálási példa különféle hibákkal

Az 1. ábrán bemutatunk néhány olyan hibakombinációt, mely láthatóan lehetetlenné teszi a helyes vételt.

Vizsgáljuk most meg, hogyan csoportosíthatjuk a jelsorozat keretezési eljárásokat [39].

Az első nagy csoport azokat az eljárásokat tartalmazza, melyeknél a jelfolyamba iktatott speciális elemek segítségével teszik azonosíthatóvá a keretszerkezet felismerését. Ezeket az eljárásokat, melyek magukba foglalják a gyakorlatban alkalmazott csaknem valamennyi adatátviteli és egyéb digitális távközlési átviteli eljárást (pl. PCM—TDM) közös névvel markert alkalmazó jelsorozat keretezési eljárásoknak nevezzük. A marker alkalmazásán túl e módszerek közös jellemzője az is, hogy legtöbbször valamilyen alkalmasan választott algoritmus lehetővé teszi, hogy a kereten belül az információt hordozó digitek eredeti struktúrájukat megtartva (szisztematikusan) szerepeljenek.

A második nagy csoportba azokat a módszereket soroljuk, melyek különleges kódolási eljárások alkalmazásával úgy teszik lehetővé a keretszerkezet felismerését, hogy nem használnak külön keretező elemeket. Ezeket az eljárásokat közös névvel vesszőmentes tulajdonságú kódokkal való jelsorozat keretezési eljárásoknak nevezzük.

A 2. ábrán összefoglaltuk a lényegesebb jelsorozat keretezési eljárásokat. Az ábrán külön nem tüntettük fel, de megemlítjük, hogy a vesszőmentes tulajdonságú kódolási módszerek legtöbbje hibajelző, ül. hibajavító tulajdonságúvá is tehető.

A továbbiakban a markert alkalmazó eljárásokkal nem foglalkozunk.

4. A kódok szinkronizálása

A keretezési módszerek tényleges vizsgálata előtt áttekintünk néhány, a továbbiakban felhasznált kódoláselméleti kérdést.

4.1. A dekódolhatóság

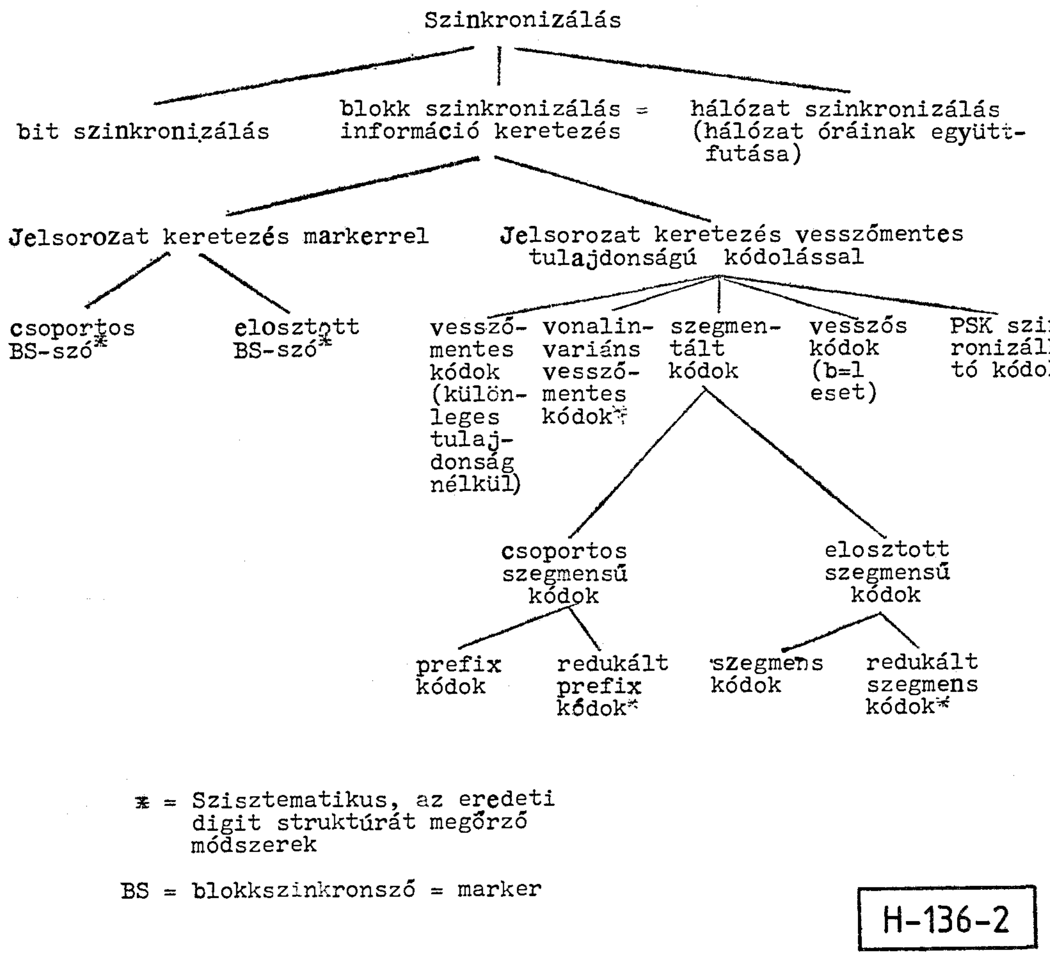
A továbbításra szánt információkat az üzenetforrás szimbólumai hordozzák. Az üzeneteket a csatorna továbbítja a rendeltetési helyre. A továbbiakban a csatorna fogalmán mindig a véges számú be- és kimeneti állapottal jellemezhető diszkrét csatornát fogjuk érteni. A csatornaszimbólumokat a csatornaforrás állítja elő. Az üzenetforrás szimbólumsorozatainak és a csatornaforrás szimbólumsorozatainak összerendelése a kódolás, a folyamat eredménye a kód (kód abc, szótár). A csatornaforrás szimbólumainak sorozata a kódszavak halmaza [4]. Egy kód akkor szisztematikus, ha a kódszavak az üzenetforrás szimbólumait változatlanul tartalmazzák, de minden kódszó tartalmaz(hat) redundáns szimbólumokat is (pl. védelem, szinkronizáció stb. céljából). Ha az üzenetforrás szimbólumait nem tartalmazzák a kód szavai, akkor a kód nem szisztematikus. Az olyan kódokat, melyekben az üzenetforrás egyes szimbólumaihoz a csatornaforrás szimbólumainak sorozatát rendeljük, blokk kódoknak nevezzük [3], [11].

Valamely kód szavainak dekódolhatósága feltételezi, hogy a szavakból álló sorozat kezdő kódszavának első szimbólumát ismerjük.

Szimbólumok (egy kód abc elemei) tetszőleges sorozata *akkor dekódolható,* ha érvényes kódszavak sorozatára bontható. A $\{k_i\} \in K$ kódszavakból álló szótárt akkor nevezzük *egyértelműen dekódolhatónak,* ha a kód szavaiból képzett sorozat csak egyféleképpen bontható fel érvényes kódszavakra [3].

A $K=\{k_i\}$ kódszótár p jelű, egy vagy több elemből álló sorozatát a k kódszó prefix-ének nevezzük, ha k felírható $k=ps$ alakban, ahol s egy vagy több elemből álló sorozat. Hasonló módon az s sorozatot a k kódszó szuffix-ének nevezzük.

A $K=\{k_i\}$ kódszótárt akkor nevezzük prefix tulajdonságú kódnak (vagy irredubicilis kódnak), és



2. ábra. Szinkronizáció és jelsorozatok keretezése

akkor jelöljük K_p-vel, ha egyetlen $\{k_i\} \in K_p$ kódszó sem prefixe valamely másik $\{k_i\} \in K_p$ kódszónak.

Ha egy prefix tulajdonságú kód bármely kódszavának összes betűjét vette a vevő, akkor a kódszó dekódolható. Ezért az ilyen kódokat azonnal dekódolhatónak, vagy nulla késleltetéssel dekódolhatónak, ill. pillanatnyi kódnak nevezzük (szokás még promt-dekódolhatónak is nevezni) [44]. Ezzel szemben léteznek olyan egyértelmű dekódolható, természetesen nem prefix tulajdonságú kódok is, melyek csak bizonyos számú kódszó vétele után dekódolhatóak.

A dekódolható K_d szótár szavaiból álló azon leghosszabb sorozat betűinek d_c számát, mely sorozat egészében szükséges a sorozat első kódszavának egyértelmű meghatározásához, a K_d szótár dekódolási késleltetésének nevezzük.

Ha létezik olyan χ pozitív egész szám, hogy $d_c = \chi$, akkor a szótár véges (vagy korlátos) dekódolási késleltetésű.

Ha nem létezik olyan χ pozitív egész szám, hogy $d_c \leq \chi$ legyen, akkor a szótár végtelen dekódolási késleltetésű. A dekódolhatóság kérdéseit részletesen tárgyalja pl. [3] és [44].

4.2. A szinkronizálhatóság

Valamely kód szavainak dekódolhatósága azt követeli meg, hogy a szótár szavaiból összeállított sorozatot egyértelműen szét lehessen bontani az egyedi kódszavakra, feltéve, hogy a sorozat kezdő szimbólumát előzetesen meghatároztuk.

A gyakorlatban a kezdő szimbólum általában előre nem ismert, ugyanis a vevő többnyire csak az üzenettovábbítás kezdete után kapcsolódik csak be, vagy mert valamely átviteli hiba folytán a jelfolyamban csúszás (szlip) keletkezik.

Ilyen esetekben a szótár szinkronizálhatóságát is meg kell követelni, tehát azt, hogy a vételi oldalon a sorozat elejét fel tudja ismerni a vevő.

Azt mondhatjuk, hogy a K szótár szavaiból álló sorozat d_s (szinkronizációs) késleltetéssel szinkronizálható, ha a sorozat egy kódszavának felismeréséhez a megelőző sorozat d_s számú szimbólumát (is) meg kell figyelni.

Így a K_s kód akkor szinkronizálható d_s késleltetéssel, ha dekódolható és a $\{k_i\} \in K_s$ szavakból álló sorozatok d_s, vagy kisebb késleltetéssel szinkronizálhatók.

Természetesen a d_s számú betű vétele során is dekódolhat üzeneteket a vevő, de lehet, hogy ezeket helytelenül dekódolja, vagy az is lehet, hogy az első néhány szimbólumot nem tudja értelmezni.

Például, ha a $K_s = \{01; 100; 101; 1101\}$ kód szavaiból az (100) (1101) sorozatot adjuk, és a vevő az első bit elvesztése után kapcsolódik csak fel, akkor a vett sorozat (00=hiba) (1101) formában dekódolható. Ugyanakkor az első két bit elvesztése után (01) (101) formában lehet dekódolni az üzenetet. Láthatóan mindkét esetben helyes szinkronizmusba került a vevő. A szinkronizálhatóság eldöntésére több tétel ismeretes, ezeket [3] részletezi, [45] idézi.

4.3. Önszinkronizáló kódok

Az előbbiekben már bemutattuk, hogy ha a vett jelsorozatból hiányzik az adott jelsorozat első néhány betűje, akkor elvileg két úton juthat el a vevő a helyes szinkronizmus felismeréséig:

a) az első néhány betű vétele után detektálja, hogy a vett részsorozat nem kódszó, tehát szinkronhiba van, és ezért a helyes kódszókeresést egy következő betűtől kezdve megismétli stb., vagy
b) az első néhány betűből álló részsorozat kódszó, így a helyes szinkronizmus beáll, bár a vett szó (szavak) az adott szavaktól eltér (eltérhet).

Tehát ezekben az esetekben a vevő — zajmentes csatorna feltételezése esetén — minden, a szinkronizmusra vonatkozó előzetes információ nélkül a vett jelfolyam véges számú elemének kiértékelése alapján meg kell tudja állapítani a helyes szinkronizmust.

Azokat a sorozatokat, melyek azzal a tulajdonsággal rendelkeznek, hogy a vevő a kód valamely szuffixével kezdődő sorozat vétele után automatikusan szinkronba kerül, önszinkronizáló sorozatoknak nevezzük.

Az automatizmus azt jelenti, hogy a vevő a sorozat első betűjétől kezdődően értelmezhető kódszavakat detektál, és meghatározott számú helytelen detekció után beáll a helyes szinkronizmus: a vevőt a sorozat mintegy „rávezeti" a helyes szinkronra.

Teljesen önszinkronizáló kódoknak nevezzük azokat a dekódolható kódokat, melyeknek tetszőleges szavaiból a kód bármely szuffixével alkotott sorozata önszinkronizáló sorozat. Ha az előbbi feltétel csak bizonyos sorozatokra teljesül, akkor a kód részben önszinkronizáló.

Ha a kód szavaiból az előbbiek szerint létrehozott egyetlen sorozat sem önszinkronizáló, akkor a kódot nem önszinkronizálónak nevezzük.

A viszonyokat jól szemléltetően azt mondhatjuk, hogy egy önszinkronizáló kód szavaiból álló sorozatba bárhol belépve a vevő mindig kódszavakat detektál (de a szinkronizációs késleltetési idő alatt helytelenül, tehát nem a ténylegesen adott szavakat detektálja), míg egy nem önszinkronizáló kód szavaiból álló sorozatba, nem valamely szó kezdeténél belépve, a vevő soha nem detektálhat azonnal kódszót. (Egyszerű teljesen önszinkronizáló kód pl. a $K=\{0, 10, 110, 1110\}$ kód.)

Az önszinkronizáló sorozatokra vonatkozóan számos eredmény és tétel ismeretes az irodalomban, többek között választ lehet kapni arra a kérdésre is, hogy egy adott kód önszinkronizáló tulajdonságának milyen feltételei vannak, és ezen feltételek teljesülését hogyan célszerű vizsgálni [3].

5. Vesszőmentes tulajdonságú szinkronizálható blokk kódok

A gyakorlat szempontjából kiemelt fontosságúak a változó szóhosszúságú kódszavak helyett azonos hosszúságú szavakat alkalmazó blokk kódok.

Ezek a blokk kódok nyilvánvalóan dekódolhatóak, de természetesen nem (teljesen) önszinkronizálók, ugyanakkor lehetnek szinkronizálhatók.

Vizsgáljuk meg elsőként, mi módon lehet a szinkronizációs késleltetést előírt értéknél kisebbre beállítani.

5.1. Vesszőmentes kódok

Legyenek a K blokk kód k_i $(i=1\ldots N)$ szavai a következők:

$$k_i = x_0^i x_1^i x_2^i \ldots x_{n-1}^i.$$

A következő n elemű sorozatot a k_i és k_j szavak t rendű átlapolásának vagy t rendű átlapolt szónak nevezzük:

$$k_{ijt} = x_{t+1}^i x_{t+2}^i \ldots x_{n-1}^i x_0^j x_1^j \ldots x_t^j,\ 1 \leq t \leq n-2$$

(felső indexben jelöltük, hogy a részsorozat eredetileg melyik kódszó része).

A vesszőmentes kódokat egzaktul az alábbiak szerint definiáljuk.

A K_m kód akkor és csak akkor nevezhető vesszőmentesnek, ha $\{k_i\}\in K_m$ $(i=1\ldots N)$ esetén

$$\{k_{ijt}\}\notin K_m;\quad i=1\ldots N,\quad j=1\ldots N,\quad t=1\ldots n-2,$$

tehát egyetlen (n elemű) átlapolt szó sem lehet a kód szava, pl. [18].

Itt kell kitérnünk a vesszőmentességi index fogalmára. Azokat a kódokat, melyeknél a lehetséges összes átlapolt kódszó Hamming távolsága valamennyi kódszóra nézve legalább u, és van olyan kódszó, melynél a Hamming távolság pontosan u, u indexű vesszőmentes tulajdonságú kódoknak nevezzük.

A definícióból következően egy vesszőmentes kód szinkronizációs késleltetése

$$d_s \leqq 2n-1,$$

hiszen a $(2n-1)$ elemből álló sorozat bizonyosan tartalmaz egy kódszót.

A vesszőmentes kód szavainak számára adható felső korlát [3], [18], [28]:

$$N(n, r) = \frac{1}{n} \sum_{d|n} \mu(d) \cdot r^{n/d},$$

ahol r a szimbólumok száma, $\mu(d)$ a Möbius függvény, melyre:

$$\mu(d) = \begin{cases} 1, \text{ ha} \quad d=1 \\ (-1)^k, \text{ha } d=p_1 \cdot p_2 \cdots p_k \text{ ahol } p_1, p_2, \ldots, p_k \\ \quad \text{különböző prímszám} \\ 0 \text{ egyébként.} \end{cases}$$

A vesszőmentes kódok előállításának teljes érvényességű szabályai az irodalomból nem ismeretesek. Néhány speciális eset mellett a páratlan számú betűkből álló kód előállítása ismert: ezt vesszőmentes prefix konstrukciós eljárásnak nevezik [3].

5.2. Prefix kódok

Ha alkalmazott kódunk olyan, hogy az n elemű szó első m betűjéből álló prefix máshol nem fordul elő a kódszóban, illetve a kódszó szuffixének és prefixének tetszőleges átlapolásából nem állhat elő a prefix, akkor a kódot prefix kódnak nevezzük [3], [20].

A prefix kód és a prefix tulajdonságú kódok — vagy irreducibilis kódok — fogalma láthatóan nem azonos. Ugyanakkor nyilvánvaló, hogy a prefix kódok prefix tulajdonságúak is, hiszen blokk kódok.

Természetes, hogy a definícióból következően a prefix kódok vesszőmentes tulajdonságú kódok is.

Szinkronizálásuk egyszerű, hiszen az m hosszúságú prefix vétele a szinkronizmust jelzi. Szinkronizációs késleltetésük: $d_s \leq 2n-1$.

Mindennek ára a kódolás-dekódolás bonyolultsága.

A viszonyok szemléltetéséhez tekintsünk egy n bitből álló sorozatot, m bitből álló prefixszel. A kódszó fennmaradó n—m bitjét úgy kell kódolni, hogy egyetlen m bitből álló kódolt részsorozat se legyen azonos a prefixszel.

Példaként 1010 prefixet választva, 1010101100 megengedett kódszó, azonban 1010010101 már nem az. A prefix kódok generálása igen bonyolult, ezért gyakorlati alkalmazása nem terjedt el.

Megemlítjük még, hogy a vesszőmentes tulajdonságú kódokhoz tartoznak még az ún. vonalinvariáns vesszőmentes kódok, a vesszős kódok és a PSK szinkronizálható blokk kódok, melyeket az irodalom több helyen is részletesebben tárgyal [3], [45], [46], [47], [48].

6. *Szegmentált kódok*

Az előzőekben bemutatott vagy említett vesszőmentes tulajdonságú kódok legtöbbje nem szisztematikus kód volt, mert az üzenetforrás szimbólumait a kódok nem tartalmazták.

Vezessük most be a következő definíciókat, amelyek egységessé teszik a további tárgyalást.

1. definíció: Szegmensnek nevezzük az n elemű r-áris jelsorozat (pl. kódszó) rögzített, nem szükségképpen egymás melletti pozícióiban található, rögzített elemeinek sorozatát. A szegmens az elemek értékének és helyének, tehát a szegmens pozícióknak megadásával egyértelműen meghatározott.

2. definíció: Szegmentált kódoknak nevezzük azokat a vesszőmentes tulajdonságú kódokat, amelyeknek valamennyi kódszavában megtalálható ugyanaz a legalább két digitből álló szegmens.

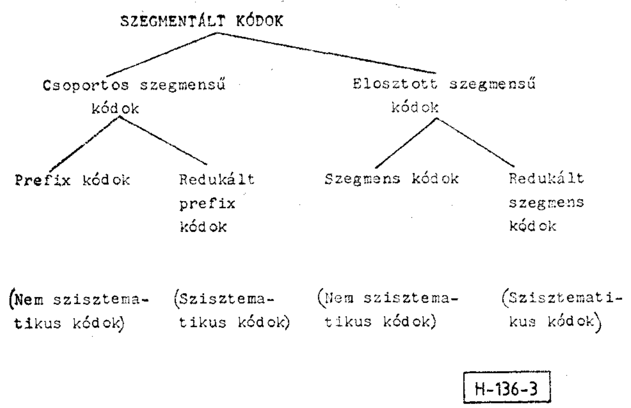
3. definíció: Csoportos szegmensű szegmentált kódoknak nevezzük azokat a szegmentált kódokat, melyek szegmense csak egymás melletti jelelemeket tartalmaz.

4. definíció: Elosztott szegmensű szegmentált kódoknak nevezzük azokat a szegmentált kódokat, melyek szegmense nemcsak egymás melletti jelelemeket tartalmaz.

5. definíció: Prefix kódoknak nevezzük a csoportos szegmensű nem szisztematikus szegmentált kódokat.

Ebből következően a szegmensen kívüli szimbólumok nem választhatók ki tetszőlegesen, hiszen nem szisztematikus a kód. A definíció tartalmában azonos Gilbert eredeti, [20] szerinti megfogalmazásával, ugyanakkor annál egy fokkal általánosabb, hiszen a „prefixnek" már nem feltétlenül kell prefix pozícióban állni.

6. definíció: Redukált prefix kódoknak nevezzük a csoportos szegmensű szisztematikus szegmentált kódokat, melyek az összes lehetséges kódszót tartalmazzák.*



3. ábra. A szegmentált kódok javasolt csoportosítása

A 6. definíció szerinti kódok szegmensen kívüli része tetszőlegesen kódolható, de a csoportos szegmens hossza $(n/2)+1$. A redukált jelzőt az indokolja, hogy ugyanolyan szóhossz esetén a redukált prefix kód szavainak száma jóval kevesebb, mint a prefix kódé.

7. definíció: Szegmens kódoknak nevezzük az elosztott szegmensű nem szisztematikus szegmentált kódokat.

8. definíció: Redukált szegmens kódoknak nevezzük az elosztott szegmensű szisztematikus szegmentált kódokat, melyek az összes lehetséges kódszót tartalmazzák.* Láthatóan a redukált szegmens kódok szegmensen kívüli része tetszőlegesen kódolható.

Azt, hogy a fenti definíciók szerinti kódok léteznek, a továbbiakban igazolni fogjuk.

Jól látható, hogy a most bevezetett definíciók szerinti kódok kapcsolata a 3. ábra szerinti. A továbbiakban az irodalomban más néven említett, a szegmentált kódokhoz sorolható eseteket is megvizsgáljuk, és részletesen tárgyaljuk a kódcsalád tulajdonságait.

6.1. *Csoportos szegmensű szegmentált kódok*

A szegmentált kódok teljesebb körű vizsgálata érdekében térünk ki — röviden — e kód-csoportra.

A prefix kódokról már szóltunk az előzőekben. A redukált prefix kódok két speciális tulajdonságát emeljük ki. A 6. definíció alapján nyilvánvaló, hogy ezeknél a kódoknál meglehetősen hosszú szegmensre van szükség, hiszen — ha a szegmens prefix — a szuffix csak akkor kódolható tetszőlegesen, ha n elemű kódszavak és m elemű prefix esetén $n<2m$. Ebből azonnal következik, hogy a kód redudanciája meglehetősen nagy $(R>0{,}5)$.

A redukált prefix kódok tekinthetők olyan vesszős kódoknak is, melyeknél az $(n-m)$ elemű kódszavakat m elemű vessző választja el, és a vesszőzöttség $b=1$ (ti. a vesszőt minden kódszó elé tesszük). Ugyanakkor lényeges, hogy a redukált prefix kódok itt bevezetett definíciója megengedi, hogy a szegmens ne prefix legyen.

* Pl. az n betűs, s szegmens bitet tartalmazó bináris kód szavainak száma 2^{n-s} lehet.

6.2. *Elosztott szegmensű szegmentált kódok*

A kódcsoportot három részre bontva mutatjuk be. Elsőként a gyakorlat szempontjából kisebb fontosságú szegmens kódokról szólunk, majd történeti okokból külön tárgyaljuk a redukált szegmens kódok közé sorolható Gilbert kódokat. A redukált szegmens kódok optimális változatait fontosságuk miatt a 7. fejezetben külön elemezzük.

6.2.1. *Szegmens kódok*

A prefix kódokat az jellemezte, hogy a szegmens a kódszóban egymás melletti jelelemekből áll. Ennek alapján felmerül a gondolat, hogy a szegmens esetleg elosztott formában is elhelyezhető: néhány jelelem a szó elején, néhány a szó belsejében. Erre a szegmensre is kiköthetjük az átlapolódási illetőleg a vesszőmentességi tulajdonságot, és akkor a szegmens kódok halmazát kapjuk.

A szegmens kódok kódszavainak előállítása még a prefix kódok eseténél is bonyolultabb, hiszen itt a szegmens elemei közé kódolható jelelem helyek is esnek.

Az áttekintett irodalom a szegmens kódokhoz hasonló kódokat nem tárgyal, így az ehhez tartozó kódolási módszer sem ismert.

További tárgyalásunkban nem vizsgáljuk a szegmens kódok lehetséges kódolási szabályait, mert a gyakorlat szempontjából jóval előnyösebbek a redukált szegmens kódok.

6.2.2. *A Gilbert kódok, mint a redukált szegmens kódok előzményei*

Történeti okokból elsőként és külön tárggyaljuk a redukált szegmens kódok e csoportját, melyeket 1960-ban Gilbert — nem ilyen néven — javasolt. A prefix kódok, mint már láttuk, nem szisztematikusak, hiszen az információt hordozó szuffix részt a prefix felépítésének függvényében kell kódolni, többnyire bonyolult eljárások igénybevételével.

Az első olyan kísérletet, hogy a prefix kód, vagyis a vesszőmentes kódolás előnyös tulajdonságait megtartva, szisztematikus kódot eredményező eljárást alkalmazzunk, Gilbert tette [20]. Gilbert módszerét csak bináris ($r=2$) esetre mutatjuk be.

Az eljárást [49] nyomán itt Gilbert kód néven tárgyaljuk. A kód egyébként, mint látni fogjuk valójában a redukált szegmens kódokhoz sorolható.

Gilbert módszerének lényege, hogy a prefix tiszta 1 sorozatból áll (m db 1 értékű bit) és az m bites P prefixet olyan jelsorozat követi, melyre teljesül, hogy:

$$x_1 = x_{m+1} = x_{2m+1} = \ldots = x_{n-m} = 0$$

és

$$n = m^2+1,$$

a többi elem tetszőleges.

Például $P=1111$ prefix esetén a kódszavak lehetséges halmaza a következő sorozatból határozható meg:

11110XXX0XXX0XXX0,

ahol X tetszőleges értékű lehet.

A javaslat leglényegesebb előnye, hogy az X helyeken közvetlenül, mintegy bitsorozat-függetlenül lehet továbbítani a bináris üzeneteket. Ennek ára a megnövekedett redundancia, melynek értéke:

$$R = \frac{2m}{m^2+1} = \frac{2\sqrt{n-1}}{n}, \tag{1}$$

mert az $n=m^2+1$ számú bitből $2m$ számú bitet rögzítettünk. Definiáljuk a redundanciát úgy, hogy:

$$R = \tag{2}$$

$$= \frac{F}{n} = \frac{\text{Nem szabadon felhasználható jelelemek száma}}{\text{összes jelelem száma}}$$

Ekkor a Gilbert kód redundanciája:

$$R_G = 2 \cdot n^{-1/2} + O(n^{-3/2}), \tag{3}$$

ahol $O(x)$ a nagy ordó jele [59].

Nyilvánvaló, hogy a kód egyértelműen dekódolható, prefix tulajdonságú (irreducibilis) és az is, hogy a Gilbert kód redukált szegmens kód.

Az is igazolható [45], hogy a Gilbert kód rögzített m mellett az eredeti szuffix konstrukció megtartásával tetszőlegesen n hosszúságig kiterjeszthető úgy, hogy a kód tulajdonságai változatlanok maradnak, de redundanciája csökken.

Itt szeretnénk hangsúlyozni, hogy a továbbiakban hacsak külön nem emeljük ki, a vesszőmentességi indexet mindig a szegmensre értjük, tehát a kódszó tényleges indexe a szegmens indexénél nagyobb is lehet.

A később bevezetendő eszközök segítségével igen egyszerűen bizonyítható, hogy a Gilbert kód és kiterjesztése is $\mu=2$ vesszőmentességi indexű.

A Gilbert kódok kiterjeszthetők r szintű esetre is, azonban ezek kicsiny gyakorlati fontossága miatt ezekkel itt most nem foglalkozunk, de utalunk [45]-re

7. *Redukált szegmens kódok*

A 6. pontban megmutattuk, hogy a Gilbert, de még a kiterjesztett Gilbert kódok redundanciája is nagy, ezért a kód hatásfoka alacsony. A következőkben főként olyan redukált szegmens kódokkal foglalkozunk, melyek hatékonysága optimális abban az értelemben, hogy a kódszavakban található szegmens a lehető legrövidebb.

A redukált szegmens kódok kifejlesztésének alapgondolata az, hogy míg az eredeti prefix kódokat — természetesen nem egyszerű kódolással — $\mu=1$ vesszőmentességi indexszel (a szegmensre) is fel lehet építeni, addig a Gilbert féle konstrukció *a priori* $\mu=2$ indexet eredményez. Ennek oka az a követelmény, hogy Gilbert módszere a prefixet az elnevezéshez ragaszkodva csak a szavak elején álló elemekből képezi. Ha beleértjük a prefixbe a kódszó belsejében elosztott biteket is, és az így keletkezett, most már elosztott kombinációt, vagyis szegmenset használjuk, akkor a viszonyok bizonyosan javulni fognak.

7.1. Redukált szegmens kódok tulajdonságai

Ha egy K_s redukált szegmens kód szavai w számú elemből álló szegmenst tartalmaznak, akkor a kódot w szegmensű redukált szegmens kódnak nevezzük. Így az alábbiakat mondhatjuk:

Minden bináris Gilbert kód redukált szegmens kód, $2m$ szegmenssel.

Minden kiterjesztett bináris Gilbert kód redukált szegmens kód, $m+k$ szegmenssel.

Az állítások triviálisak, mégis lényegesek, mert a Gilbert féle konstrukció kódjait a szegmentált kódok családjába illesztik.

Definiáljuk most az optimális redukált szegmens kódot a következőképpen:

9. definíció: A K_{RS} redukált szegmens kódot akkor nevezzük K_{ORS} optimális redukált szegmens kódnak, ha *a)* a szegmens vesszőmentességi indexe $\mu=1$, és *b)* az $R=w/n$ hányados értéke a lehető legkisebb, továbbá a minimálást az n betűs szavakra értjük, ahol n a kódszó, w a szegmens elemeinek száma.

A K_{ORS} redundanciája nyilván a lehetőleg kisebb lesz az adott szóhosszúság esetén, hiszen ha a rögzített elemek számát csökkentjük (hogy csökkentsük a redundanciát is), akkor $\mu<1$ lesz, ami viszont azt jelenti, hogy a kód már nem vesszőmentes tulajdonságú. Az optimális redukált szegmens kódok létezésére vonatkozik a következő:

1. Tétel: Annak szükséges feltétele, hogy az r szimbólumból álló n betűs szavakat tartalmazó K kód optimális redukált szegmens kód legyen az, hogy

$$n \leq 2\cdot[u_0\cdot(u_1+u_2+\ldots+u_{r-1})+u_1\cdot(u_2+\ldots+u_{r-1})+ +\ldots+u_{r-2}\cdot u_{r-1}]+1$$

alakban legyen felírható, úgy hogy

$$|u_i-u_j|\leq 1,\ \forall i,j\in\{0,r-1\} \text{ esetén,}$$

ahol u_i az i értékű szegmens digitek számát jelöli.

Tételünket a Függelékben bizonyítjuk.

A továbbiakhoz szükségünk lesz a következő műveletre, melyet az alábbiak szerint definiálunk.

10. definíció: Legyen a, b és r tetszőleges nem negatív egész szám. Ekkor:

$$a\oplus b=\begin{cases}1, \text{ ha } a<r \text{ és } b<r \text{ és } a\neq b\\ 0 \text{ egyébként.}\end{cases} \qquad (4)$$

A művelet némileg hasonlít a „kizáró vagy" műveletre, ezért — önkényesen — a (4) szerinti művelet eredményét a és b deviációjának fogjuk nevezni.

Legyen most d_0 tetszőleges, $x_0x_1\ldots x_{n-1}$ elemű sorozat, ahol $x_i\leq r$, $(i=0\ldots n-1)$ Ez a sorozat egyértelműen meghatározza a **D** ciklikus mátrixot.

Képezzük most azt a **D*** mátrixot, melynek minden sorában a d_0 sorozat található. A (4) szerinti müveletet általánosítva a **D** és a **D*** mátrixok deviációja:

$$\mathbf{D}\oplus\mathbf{D}^*=\mathbf{\Delta} \text{ úgy, hogy}$$

$$\mathbf{D}=[d_{ik}],\quad \mathbf{D}^*=[d^*_{ik}] \quad \text{és} \quad \mathbf{\Delta}=[\delta_{ik}] \text{ esetén}$$

$$\delta_{ik}=d_{ik}\oplus d^*_{ik}$$

A $\mathbf{\Delta}$ mátrixot a d_0 sorvektor deviációs mátrixának nevezzük.

Az r szimbólumból álló r-áris kódok esetén a $\mathbf{\Delta}$ mátrix létrehozásakor a szegmens-pozíciókban szereplő szimbólumok értéke $0\ldots r-1$ lehet. Az adatátvitelhez szabadon felhasználható kódszóbeli pozíciókat, tehát a nem szegmensbeli pozíciókat pontosan az r szimbólummal töltjük fel. Ennek oka az, hogy így a $\oplus$ művelet célszerű definiálásával szegmensbeli és információs elem, illetve két információs elem deviációja 0 lesz.

A $\mathbf{\Delta}$ mátrix segítségével a redukált szegmens kódok két lényeges, általános érvényű tulajdonságát jellemezhetjük:

2. Tétel: Az n szimbólumból álló r-áris K kód akkor és csak akkor redukált szegmens kód, ha a $\{k_i\}\in K$ kódszavakból képzett $\mathbf{\Delta}_i$ deviációs mátrixok minden $j>1$ sora tartalmaz legalább egy 1 értékű elemet a kód összes $k_i\in K$ szavára vonatkozóan.

1. Állítás: Ha a K_S redukált szegmens kód szegmensének vesszőmentességi indexe $\mu=M$, akkor a kód szavaiból képzett $\mathbf{\Delta}_i$ deviációs mátrix minden sorában M számú 1 értékű elem van. Belátható, hogy az állítás megfordítása is igaz.

Ezt az állítást a definíciók segítségével láthatjuk be. A $\mathbf{\Delta}$ mátrix tulajdonsága, hogy pontosan ott van 1 értékű elem, ahol az átlapolt kódszó a szegmens pozíciókban eltér az eredetitől. A szegmens vesszőmentességi indexe viszont éppen azt adja meg, hogy hány helyen különbözik az átlapolt és az eredeti kódszó. Ez az eredmény azért fontos, mert kapcsolatot teremt a $\mathbf{\Delta}$ mátrix és a vesszőmentességi index között.

A 2. tételt a függelékben később bizonyítjuk.

Az általunk optimális redukált szegmens kódoknak nevezett kódcsoport egy alosztályának tulajdonságait elemezte Clague [49] bináris esetekre. Eredményeit az alábbiak szerint általánosítottuk:

3. Tétel: Az n betűs szavakból álló r-áris K szegmentált kód, melynek k szegmense az $a^p_{i_p}$ pozíciókban rögzített értékű p elemekből áll, ahol $p=0\ldots r-1$, $i_p=1\ldots u_p$, akkor és csak akkor redukált szegmens kód $\mu=M$ vesszőmentességi indexszel, ha a

$$\{\pm(a^p_{i_p}-a^q_{i_q})\} \text{ halmaz minden } p\neq q, \text{ és}$$

$(p, q)\in(0,1\ldots r-1)$ esetén teljes M-szeres nem nulla maradékrendszert alkot, modulo n.

A 3. tétel két fogalmát részletesebben is megvilágítjuk. Az $a^p_{i_p}$ jelöléssel jelöltük a kódszó szegmens pozícióit, úgy, hogy pl. $a^0_{i_0}$ a szegmens 0 értékű helyeit jelöli, és $i_0=1\ldots u_0$, tehát pl. bináris esetben $i_0=1\ldots u$, ill. $i_1=1\ldots v$.

A $\{\pm(x-y)\}$ mod n úgy értendő, hogy a különbségképzésben mind a kivonandóra, mind a kisebbítendőre teljesül, hogy $x<n+1$, ill. $y<n+1$, hiszen x és y a szegmens pozíciókat jelöli. Ha $x>y$, akkor $(x-y)_{\text{mod}\,n}=x-y$, de ha $x<y$, akkor $(x-y_{\text{mod}\,n}=$ $=n+x-y$. A teljes nem nulla maradékrendszer mod n az $1, 2\ldots n-1$ értékű elemek halmazát jelöli. M-szeres a nem nulla teljes maradékrendszer, ha minden teljes maradékrendszer M-szer szerepel [50].

A 2. és 3. tétel ugyanannak az állításnak két megfogalmazása. A 3. tételt a függelékben bizonyítjuk.

Az 1. és 2. tételekből következik a gyakorlat szempontjából lényeges

1. Korollárium: Ha az n betűs bináris szegmens kód szegmensére $2 \cdot uv < n-1$, akkor a kód nem lehet redukált szegmens kód, és $\mu < 1$, ahol u az 1-esek, v a 0-k száma (vagy fordítva).

Az 1. tétel függelékben részletezett bizonyításának következménye az alábbi, a gyakorlat szempontjából igen hasznos.

2. Korrollárium: Ha az r szimbólumból álló n betűs szavakat tartalmazó kódra $n = r \cdot (r-1) \cdot u^2 + 1$, úgy hogy u egész szám, akkor lehet olyan szegmenst létrehozni, mellyel a kód optimális redukált szegmens kód és a szegmens $r \cdot u$ elemből áll. Ennél rövidebb szegmenssel a kód nem hozható létre.

Utóbbi állításunkat az 1. tétel bizonyítása során láttuk be.

7.2. *Redukált bináris szegmens kódok előállítása*

A prefix szinkronizált kódoknál láttuk, hogy a nehézséget végül is a kódolás-dekódolás okozza.

A redukált szegmens kódoknál a szegmens optimális megválasztása jelenti a fő feladatot.

Az előző szakasz eredményeiből már tudjuk, hogy — bináris esetre vizsgálva — az u és v számú szegmens bitekre, illetve a kódszó hosszára: $n-1=2uv$ esetben, ha még $|u-v| \leq 1$, akkor optimális lehet a kódszó (1. tétel). Ez azt is jelenti, hogy olyan n-esetén is tudunk megfelelő szegmenst találni, melyre $n \neq 2 \cdot x^2 + 1$, úgy, hogy x egész szám.

A 3. tétel felveti a kérdést: hogyan lehet olyan módon kiválasztani az 1, 2, ... n számokból a_i-ket és b_j-ket, ahol $i = 1 \ldots u$, $j = 1 \ldots v$, hogy teljesüljön a maradékrendszer követelmény?

Ugyanezt a kérdést az előzőekben bevezetett Δ mátrix terminológiájával is fel lehet tenni, hiszen a teljes nem nulla maradékrendszer egy-egy értelmű kapcsolatban van a Δ mátrix sorainak sorszámával.

További kérdés lehet az is, hogy hány különböző előállítása lehet a szegmensnek.

A választások számával kapcsolatban a következőket állíthatjuk.

4. Állítás: Ha egy n elemű bináris sorozat olyan szegmenst tartalmaz, mellyel a kód redukált szegmens kód, akkor ennek a sorozatnak a reverzált (fordított sorrendű) felírása vagy az eredeti, illetve a reverzált sorozat negált értékű változata, illetőleg ezek bármely ciklikusan eltolt változata is redukált szegmens kódot határoz meg.

Az állítás természetesen azonnal következik, például a Δ mátrix negálásra, ciklikus eltolásra, reverzálásra való érzéketlenségéből.

A 4. állításból viszont következik, hogy egy n elemű alkalmas konstrukció azonnal további $2 \cdot 2(n-1)$ ekvivalens konstrukciót ad.

Az alábbiakban megadunk három lehetséges alapkonstrukciót, melyek közül kettő az irodalomban is megtalálható.

I. Konstrukció [49]

Legyen $n > 1$. Tekintsük azt a p számot, melyre $p \geq \sqrt{2(n-1)}$. Legyen u az a legkisebb egész szám, melyre $u \geq \frac{1}{2} p$ és $v = p - u$. A szegmens legyen olyan, hogy az első u pozícióban állnak a 0 értékű elemek, az 1 értékűek pedig az n, $n-u$, $n-2u$, ... $n-(v-1) \cdot u$ pozíciókban.

II. Konstrukció [46]

Az n, p, u és v értékei ugyanazok itt is, mint az I. Konstrukciónál.

A konstrukciónál az első u pozícióban állnak a 0 értékű elemek. Az 1 értékű elemek az $u+1$, $2u+1$, ... $(v-1) \cdot u + 1$ pozíciókban találhatók.

III. Konstrukció

Az n, p, u és v értékei ugyanazok, mint az I. Konstrukciónál.

Az első pozícióban található az első 0 elem. A további pozíciókban, tehát a 2,3, ... $(v+1)$ pozíciókban találhatók az 1 értékű elemek, ezt u elem kihagyása után követi a második 0 értékű, majd $u-1$ elem kihagyásával a harmadik 0 értékű elem stb.

További konstrukcióra az alábbiakban adunk példát, ahol mind a Δ mátrix, mind a maradékrendszer számításához szükséges alapokat is megadtuk $n=9$ esetére.

A példaként választott esetben $n=9$, így $u=2$, $v=2$ és az optimális eset $n = 2 \cdot 2^2 + 1$ alapján létrehozható.

Az előbbiek szerinti konstrukció számozást betartva megadjuk 4 lehetséges kódszó felépítését. A kódszavakat **D** mátrixukkal jellemezzük, úgy, hogy a $\mu = 1$ vesszőmentességet biztosító pozíciót aláhúzással jelöljük meg. A **D** mátrixok alatt felírtuk azokat az a_i és b_j értékeket, melyekkel képezhető a $\pm(a_i - b_j)$ teljes nem nulla maradékrendszer, modulo 9.

Az alapkonstrukciók a bal oldalon láthatók. Mellettük állnak a reverzált kódszavakon alapuló **D** mátrixok. A negált változatokat nem adtuk meg.

I. konstrukció

0 0XXXX 1X 1
$\underline{1}$ 0 0XXXX 1X
$\bar{X}\underline{1}$ 0 0XXXX 1
$\underline{1}\bar{X}$ 1 0 0XXXX
$\bar{X}\underline{1}$X 1 0 0XXX
X$\bar{X}$ 1X 1 0$\underline{0}$XX
XXX 1X 1$\underline{\bar{0}}$ 0X
XXXX 1X$\bar{1}$ 0$\underline{0}$
0XXXX 1X 1$\underline{\bar{0}}$

$a_1=1$ $a_2=2$
$b_1=7$ $b_2=9$

I. reverzált konstrukció

1X 1XXXX 0 0
$\underline{0}$ 1X 1XXXX 0
$\underline{\bar{0}}$ 0 1X 1XXXX
$\bar{X}$ 0$\underline{0}$ 1X 1XXX
XX$\underline{\bar{0}}$ 0 1X 1XX
XX$\bar{X}$ 0 0 1X$\underline{1}$ 0
XXXX 0 0 1$\bar{X}\underline{1}$
1XXXX 0 0$\underline{1}\bar{X}$
X 1XXXX 0$\bar{0}\underline{1}$

$a_1=8$ $a_2=9$
$b_1=1$ $b_2=3$

II. konstrukció

0 0 1X 1XXXX
X 0$\underline{0}$ 1X 1XXX
XX$\underline{\bar{0}}$ 0 1X 1XX
XX$\bar{X}$ 0$\underline{0}$ 1X 1X
XXXX$\underline{\bar{0}}$ 0 1X 1
$\underline{1}$XXX$\bar{X}$ 0 0X 1
$\bar{X}\underline{1}$XXXX 0 0 1
$\underline{1}\bar{X}\bar{1}$XXXX 0 0
$\bar{0}\underline{1}$X 1XXXX 0

$a_1=1$ $a_2=2$
$b_1=3$ $b_2=5$

II. reverzált konstrukció

XXXX 1X 1 0 0
0XXXX 1X$\underline{1}$ 0
0 0XXXX 1$\bar{X}\underline{1}$
1 0 0XXXX$\underline{1}\bar{X}$
X 1 0 0XXX$\bar{X}\underline{1}$
1X 1 0$\underline{0}$XXX$\bar{X}$
X 1X 1$\underline{\bar{0}}$ 0XXX
X 1X 1$\underline{\bar{0}}$ 0XXX
XX 1X$\bar{1}$ 0$\underline{0}$XX
XXX 1X 1$\underline{\bar{0}}$ 0X

$a_1=9$ $a_2=8$
$b_1=7$ $b_2=5$

III. konstrukció

0 1 1XX 0XXX
X0 1 1XX 0XX
XX0 1 1XX 0X
XXX 0 1 1XX 0
0XXX 0 1 1XX
X0XXX 0 1 IX
XX0XXX 0 i i
iXX 0XXX 0 i
1 iXX 0XXX 0

$a_1=1$ $a_2=6$
$b_1=2$ $b_2=3$

III. reverzált konstrukció

XXX 0XX 1 1 0
0XXX 0XX 1 1
1 0XXX 0XX 1
1 1 0XXX 0XX
X 1 1 0XXX 0X
XX I 1 0XXX 0
X 0XX 1 1 0XX
XX 0XX 1 1 0X

$a_1=4$ $a_2=9$
$b_1=7$ $b_2=8$

IV. konstrukció

0 1X 1X 0XXX
X0 IX IX 0XX
XX 0 1X 1X 0X
XXX0 1X 1X 0
0XXX 0 1X 1X
X0XXX 0 1X 1
1X 0XXX 0 IX
X 1X 0XXX 0 1
1X 1X 0XXX 0

$a_1=1$ $a_2=6$
$b_1=2$ $b_2=4$

IV. reverzált konstrukció

XXX 0X 1X 1 0
0XXX 0X 1X 1
1 0XXX 0X IX
X I 0XXX 0X 1
IX 1 0XXX 0X
X 1X 1 0XXX 0
0X IX 1 0XXX
X 0X 1X 1 0XX
XX 0X 1X 1 0X

$a_1=4$ $a_2=9$
$b_1=6$ $b_2=8$

8. *A redukált szegmens kódok alkalmazása a budapesti távközlő hálózatban*

Ebben a fejezetben néhány részletet kívánunk bemutatni a kódcsoport gyakorlati alkalmazásáról. A gyakorlati alkalmazás számos kérdését [51], [52], [53], [54], [55], [56] tárgyalja.

Ismeretes, hogy az egyre bonyolultabbá váló távközlési berendezések fenntartásához a postaigazgatóságok centralizált fenntartási szervezetek létrehozásán fáradoznak.

A centralizált fenntartási szervezet hatékony működésének feltétele egy olyan centralizált felügyeleti rendszer kialakítása, mely megfelelő adatokat szolgáltat a rendszerbe kapcsolt berendezések állapotáról.

A budapesti hálózatban a digitális átviteltechnikai berendezéseket kellett bevonni a központosított fenntartásba, melynek irányelveit a CCITT G.803 ajánlása rögzíti [12].

A felügyelt berendezések néhány (1—3) riasztási kimenetének állapotáról kell tájékoztatni a felügyelő személyzetet. Az állapotok átjelzésére több lehetőségünk adódik, például:

a) a teljes hálózatban minden figyelt riasztási kimenethez egy bitet rendelünk és ezt folyamatosan, a kimenet állapotától függetlenül jelezzük át a központ felé, vagy

b) a riasztási pontok ciklikus letapogatása után csak a hibás berendezéseket jelző pontok kódolt információját jelezzük át.

A rendelkezésre álló adatátviteli utak tulajdonságai, a lassan változó (ti. riasztási) információk alapján az a) változat került bevezetésre.

Tekintettel arra, hogy felügyeleti célokra elegendő kb. 1200 bps átviteli sebesség [57] és figyelembe véve a primer PCM rendszerek keretszervezéséből eredő lehetőségeket és kötöttségeket, jelen megvalósításban a 8 kbit/s-os átviteli sebességet biztosító szerviz (hulladék vagy háztartási) bitek kerültek hasznosításra.

A budapesti távbeszélő-hálózat PCM összeköttetései szövevényes digitális átviteli hálózatot alkotnak. A szervizbitek alkalmazásával így egy 8 kbps-os szolgálati csatornákból álló szövevényes hálózat, bizonyos, célszerűen kiválasztott útjait lehetett alkalmazni. A megvalósult hálózat topológiai kérdéseivel részletesebben foglalkozik pl. [58], e tárgykört itt nem elemezzük.

8.1. *Az üzemfelügyeleti jelek kódolása*

Olyan kódolásra volt szükség, mely:

1. költségkímélő, tehát egyszerű,
2. folyamatos adási üzem megvalósítására használható,
3. mert ritkán változó adatokat továbbít (ti. riasztásokat),
4. egyszerű multiplexálásra, felfűzésre alkalmas,
5. zajok ellen kismértékben védett (mert az átviteli csatorna igen jó minőségű),
6. a szokásos kódolási eljárásoknál esetleg nagyobb redundanciát is tartalmazhat (mert az átviteli csatorna igen nagy sebességű),
7. nagykapacitású és szisztematikus kódolást tesz lehetővé (mely kezdeti kiépítésekor teljes kapacitásának csak töredékével fog működni),
8. az adatok csomagok, blokkok formájában továbbíthatók, melyek egymástól függetlenek (mert a folyamatos adás miatt időlegesen bármelyik blokk „elveszhet", de ez nem hathat egy másik blokkra).

Anélkül, hogy a fenti követelményeket tételesen vizsgálnánk, modhatjuk, hogy várhatóan a redukált szegmens kódok megfelelő felépítésű változata eleget fog tenni a fenti előírásoknak.

8.2. *A felhasznált redukált szegmens kód tervezése*

A szóhossz (n) meghatározásánál vegyük figyelembe az 1. Korolláriumot, mely szerint bináris esetben

$$2uv \geq n-1$$

kell legyen, és optimális esetben $|u-v| \geq 1$, ilyenkor tehát a szegmens vesszőmentességi index $\mu=1$.

A tényleges megvalósításnál digitális áramköröket, léptető regisztereket, mikroprocesszort stb. fogunk alkalmazni, melyeknél az alapegység a byte, vagyis 8 bit.

A hálózatot a későbbi bővítések figyelembevételével mintegy 80 000 kódolatlan jelzés átvitelére kívánjuk felhasználni (hiszen riasztási kimenetek periodikusan letapogatott állapotait jelezzük át a központba). Legyen $n=48$. Ekkor $u=v=5$, mert

$$2 \cdot nv = 2 \cdot 5 \cdot 5 = 50, \quad n-1 = 48-1 = 47 \qquad (5)$$

úgy, hogy a szegmensre $\mu=1$.

Megjegyezzük, hogy az 1. tétéi értelmében $u=v=5$ esetében $n_1=51$ is lehetne.)

Az (5) eredmény azt jelenti, hogy $\mu=1$ szegmens vesszőmentességi indexszel $I=38$ bit áll a bitsorozat-független átvitel céljaira rendelkezésre. Ha ezt a 38 bites egységet tovább osztjuk például $I_1=24$ információs és $C=14$ címbitre, akkor ezzel

$$L = 2^{14} \cdot 24 = 393216 \qquad (6)$$

számú riasztási kimenet állapotának átjelzésére alkalmas kódot építhetünk fel.

Az előző, (6) szerinti kapacitás meghaladja a szükséges mennyiséget, ezért lehetőségünk van arra, hogy $\mu>1$ választással kódunkat a zajokkal szemben érzéketlenebbé tegyük. Legyen most

$$n=48, \quad u=7, \quad v=8, \quad I=33, \quad C=12,$$
$$I_1=20, \quad P=1,$$

ahol u és v a szegmens bitek, I_1 az információs bitek (riasztásokat átjelző bitek), C a címbitek jele és P a cimbiteket védő paritás bitet jelöli.

Látható, hogy a kód redundanciáját jelentősen növeltük, hiszen

$$R = \frac{u+v}{n} = \frac{15}{48} = 0{,}3125$$

ugyanakkor

$$L = 2^{12} \cdot 20 = 81920$$

számú riasztási kimenet állapotát tudjuk átjelezni, és még a címbiteket is védtük egy paritás bittel, továbbá 4096 címzett blokkot tudunk képezni.

A 7. pont eredményeire építve heurisztikus módszerrel egészíthetjük ki a konstrukciónkat úgy, hogy az eljárás a lehető legnagyobb μ vesszőmentességi indexet eredményezze. A 4. ábrán bemutatjuk a Budapesten

```
0 0 0 0 0 0 0 1 0 X X X 1 X X X X X X 1 X X X X X X 1 X X X X X P 1 C C C C C C 1 C C C C C C 1
1 0 0 0 0 0 0 0 1 0 X X X 1 X X X X X X 1 X X X X X X 1 X X X X X P 1 C C C C C C 1 C C C C C C
C 1 0 0 0 0 0 0 0 1 0 X X X 1 X X X X X X 1 X X X X X X 1 X X X X X P 1 C C C C C C 1 C C C C C
C C 1 0 0 0 0 0 0 0 1 0 X X X 1 X X X X X X 1 X X X X X X 1 X X X X X P 1 C C C C C C 1 C C C C
C C C 1 0 0 0 0 0 0 0 1 0 X X X 1 X X X X X X 1 X X X X X X 1 X X X X X P 1 C C C C C C 1 C C C
C C C C 1 0 0 0 0 0 0 0 1 0 X X X 1 X X X X X X 1 X X X X X X 1 X X X X X P 1 C C C C C C 1 C C
C C C C C 1 0 0 0 0 0 0 0 1 0 X X X 1 X X X X X X 1 X X X X X X 1 X X X X X P 1 C C C C C C 1 C
C C C C C C 1 0 0 0 0 0 0 0 1 0 X X X 1 X X X X X X 1 X X X X X X 1 X X X X X P 1 C C C C C C 1
1 C C C C C C 1 0 0 0 0 0 0 0 1 0 X X X 1 X X X X X X 1 X X X X X X 1 X X X X X P 1 C C C C C C
C 1 C C C C C C 1 0 0 0 0 0 0 0 1 0 X X X 1 X X X X X X 1 X X X X X X 1 X X X X X P 1 C C C C C
C C 1 C C C C C C 1 0 0 0 0 0 0 0 1 0 X X X 1 X X X X X X 1 X X X X X X 1 X X X X X P 1 C C C C
C C C 1 C C C C C C 1 0 0 0 0 0 0 0 1 0 X X X 1 X X X X X X 1 X X X X X X 1 X X X X X P 1 C C C
C C C C 1 C C C C C C 1 0 0 0 0 0 0 0 1 0 X X X 1 X X X X X X 1 X X X X X X 1 X X X X X P 1 C C
C C C C C 1 C C C C C C 1 0 0 0 0 0 0 0 1 0 X X X 1 X X X X X X 1 X X X X X X 1 X X X X X P 1 C
C C C C C C 1 C C C C C C 1 0 0 0 0 0 0 0 1 0 X X X 1 X X X X X X 1 X X X X X X 1 X X X X X P 1
1 C C C C C C 1 C C C C C C 1 0 0 0 0 0 0 0 1 0 X X X 1 X X X X X X 1 X X X X X X 1 X X X X X P
P 1 C C C C C C 1 C C C C C C 1 0 0 0 0 0 0 0 1 0 X X X 1 X X X X X X 1 X X X X X X 1 X X X X X
X P 1 C C C C C C 1 C C C C C C 1 0 0 0 0 0 0 0 1 0 X X X 1 X X X X X X 1 X X X X X X 1 X X X X
X X P 1 C C C C C C 1 C C C C C C 1 0 0 0 0 0 0 0 1 0 X X X 1 X X X X X X 1 X X X X X X 1 X X X
X X X P 1 C C C C C C 1 C C C C C C 1 0 0 0 0 0 0 0 1 0 X X X 1 X X X X X X 1 X X X X X X 1 X X
X X X X P 1 C C C C C C 1 C C C C C C 1 0 0 0 0 0 0 0 1 0 X X X 1 X X X X X X 1 X X X X X X 1 X
X X X X X P 1 C C C C C C 1 C C C C C C 1 0 0 0 0 0 0 0 1 0 X X X 1 X X X X X X 1 X X X X X X 1
1 X X X X X P 1 C C C C C C 1 C C C C C C 1 0 0 0 0 0 0 0 1 0 X X X 1 X X X X X X 1 X X X X X X
X 1 X X X X X P 1 C C C C C C 1 C C C C C C 1 0 0 0 0 0 0 0 1 0 X X X 1 X X X X X X 1 X X X X X
X X 1 X X X X X P 1 C C C C C C 1 C C C C C C 1 0 0 0 0 0 0 0 1 0 X X X 1 X X X X X X 1 X X X X
X X X 1 X X X X X P 1 C C C C C C 1 C C C C C C 1 0 0 0 0 0 0 0 1 0 X X X 1 X X X X X X 1 X X X
X X X X 1 X X X X X P 1 C C C C C C 1 C C C C C C 1 0 0 0 0 0 0 0 1 0 X X X 1 X X X X X X 1 X X
X X X X X 1 X X X X X P 1 C C C C C C 1 C C C C C C 1 0 0 0 0 0 0 0 1 0 X X X 1 X X X X X X 1 X
X X X X X X 1 X X X X X P 1 C C C C C C 1 C C C C C C 1 0 0 0 0 0 0 0 1 0 X X X 1 X X X X X X 1
1 X X X X X X 1 X X X X X P 1 C C C C C C 1 C C C C C C 1 0 0 0 0 0 0 0 1 0 X X X 1 X X X X X X
X 1 X X X X X X 1 X X X X X P 1 C C C C C C 1 C C C C C C 1 0 0 0 0 0 0 0 1 0 X X X 1 X X X X X
X X 1 X X X X X X 1 X X X X X P 1 C C C C C C 1 C C C C C C 1 0 0 0 0 0 0 0 1 0 X X X 1 X X X X
X X X 1 X X X X X X 1 X X X X X P 1 C C C C C C 1 C C C C C C 1 0 0 0 0 0 0 0 1 0 X X X 1 X X X
X X X X 1 X X X X X X 1 X X X X X P 1 C C C C C C 1 C C C C C C 1 0 0 0 0 0 0 0 1 0 X X X 1 X X
X X X X X 1 X X X X X X 1 X X X X X P 1 C C C C C C 1 C C C C C C 1 0 0 0 0 0 0 0 1 0 X X X 1 X
X X X X X X 1 X X X X X X 1 X X X X X P 1 C C C C C C 1 C C C C C C 1 0 0 0 0 0 0 0 1 0 X X X 1
1 X X X X X X 1 X X X X X X 1 X X X X X P 1 C C C C C C 1 C C C C C C 1 0 0 0 0 0 0 0 1 0 X X X
X 1 X X X X X X 1 X X X X X X 1 X X X X X P 1 C C C C C C 1 C C C C C C 1 0 0 0 0 0 0 0 1 0 X X
X X 1 X X X X X X 1 X X X X X X 1 X X X X X P 1 C C C C C C 1 C C C C C C 1 0 0 0 0 0 0 0 1 0 X
X X X 1 X X X X X X 1 X X X X X X 1 X X X X X P 1 C C C C C C 1 C C C C C C 1 0 0 0 0 0 0 0 1 0
0 X X X 1 X X X X X X 1 X X X X X X 1 X X X X X P 1 C C C C C C 1 C C C C C C 1 0 0 0 0 0 0 0 1
1 0 X X X 1 X X X X X X 1 X X X X X X 1 X X X X X P 1 C C C C C C 1 C C C C C C 1 0 0 0 0 0 0 0
0 1 0 X X X 1 X X X X X X 1 X X X X X X 1 X X X X X P 1 C C C C C C 1 C C C C C C 1 0 0 0 0 0 0
0 0 1 0 X X X 1 X X X X X X 1 X X X X X X 1 X X X X X P 1 C C C C C C 1 C C C C C C 1 0 0 0 0 0
0 0 0 1 0 X X X 1 X X X X X X 1 X X X X X X 1 X X X X X P 1 C C C C C C 1 C C C C C C 1 0 0 0 0
0 0 0 0 1 0 X X X 1 X X X X X X 1 X X X X X X 1 X X X X X P 1 C C C C C C 1 C C C C C C 1 0 0 0
0 0 0 0 0 1 0 X X X 1 X X X X X X 1 X X X X X X 1 X X X X X P 1 C C C C C C 1 C C C C C C 1 0 0
0 0 0 0 0 0 1 0 X X X 1 X X X X X X 1 X X X X X X 1 X X X X X P 1 C C C C C C 1 C C C C C C 1 0
```

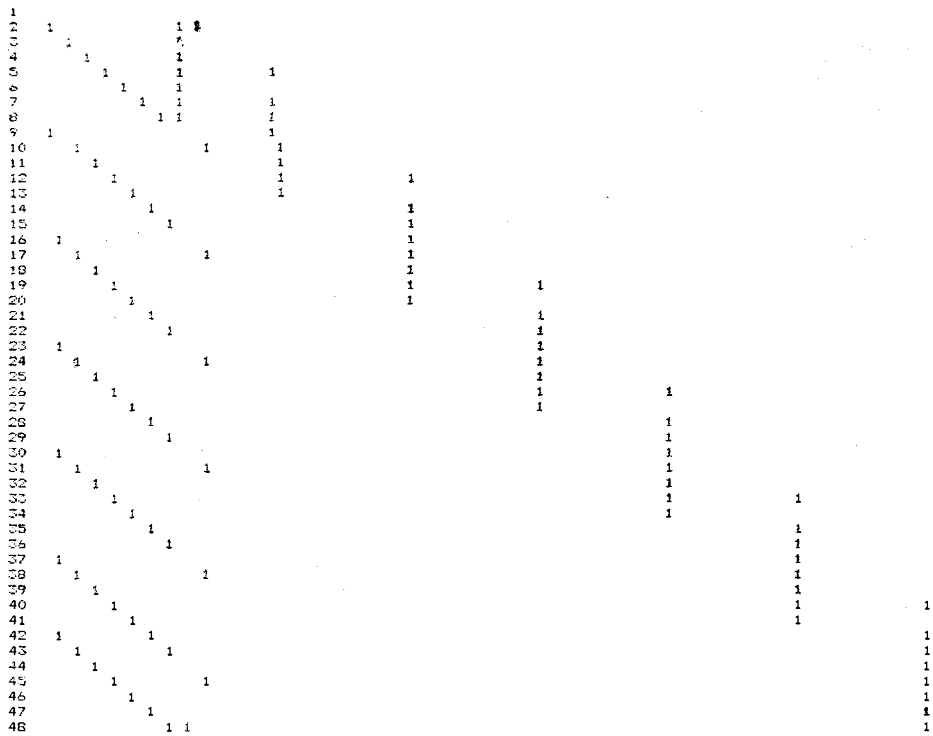
a_i = 1, 2, 3, 4, 5, 6, 7, 9.

b_j = 8, 13, 20, 27, 34, 41, 48.

Megj.: a definició szerinti D mátrixban X, P, C helyén 2 olvasandó.

H-136-4

4. ábra. A budapesti üzemfelügyeleti hálózat redukált szegmens kódjának **D** mátrixa



H-136-5

5. ábra. A budapesti üzemfelügyeleti hálózat redukált szegmenskódjának Δ mátrixa

alkalmazott redukált szegmens kód **D** mátrixát, úgy hogy az információ átvitelre felhasznált bitek helyén X-et, a címbitek helyére C-t írtunk, a paritásbit helyét P jelöli, és megadjuk azokat az a_i és b_j értékeket, melyekkel ellenőrizhető az M-szeres teljes nem nulla maradékrendszer feltétel kielégítése. Az 5. ábrán bemutatjuk a kód **Δ** mátrixát, melyben a jobb áttekinthetőség érdekében a nullák helyét üresen hagytuk. Látható, hogy a vesszőmentességi index $2 \leq \mu_e \leq 3$.

8.3. Néhány különleges tulajdonság: hibavédelem és felfűzés

A szegmensbitek hibáiról szólva μ_e értékből az következik, hogy $H_e = 1$ szegmensbit hibáig nem jöhet létre meghibásodott kódszó vétel, hiszen a szegmens biteket figyelő, a vevő áramkörben levő léptető regiszter megfelelő párhuzamos kimenetein ilyenkor nem találhatók meg a várt szegmens bitjei, tehát a vevő nem veszi a kódszót.

Tekintettel arra, hogy a hibaarány 10^{-8} körüli érték, a 48 bites kódszót vizsgálva közelítőleg minden 10^{+8}-adik szó fog meghibásodni.

Azért, hogy az információs biteket még jobban védjük, a ténylegesen megvalósított rendszerben még további két egyszerű módszert alkalmaztunk a hibás információ kiszűrésére. Elsőként a már említett cimbitek paritás bitjét is ellenőrizzük, és ha a paritásbit nem felel meg a cimbitek értékének, akkor a vett blokkot (kódszót) figyelmen kívül hagyjuk.

Második ellenőrzésként azt a módszert alkalmaztuk — valójában ez már egyszerű adatátviteli protokoll — hogy minden kódszót egymás után kétszer adunk, úgy, hogy először minden I_1 jelű bit az eredeti értelmű (ponált), majd ezeket rendre negáljuk. A módszer természetes következményeként

a) meg kellett egyszer s mindenkorra határozni, hogy az elsőként adott kódszó tartalmazza a ponált I_1 biteket,

b) egy vonalon egymás után legalább két különböző kódszót kell küldeni (ellenkező esetben nem tudni, melyik az „első", hiszen folyamatos adás üzemmódban dolgoznak az áramkörök).

A központi vevőáramkörökre lehet bízni a paritásbit ellenőrzését és célszerűen a feldolgozó egység előfeldolgozó processzorai végezhetik a ponált-negált pár ellenőrzését. Eltérés esetén a kódszót figyelmen kívül kell hagyni.

Látszólag igen sok esetben „eldobjuk" a vett kódszavakat, valójában az igen jó minőségű csatorna és a lassan változó bemenő adatok következtében semmilyen gondot nem okoz, ha egy riasztást a keletkezés után csak néhány 100 msec múlva észlel a központ. Arról viszont gondoskodunk, hogy átviteli hibák miatti hamis riasztások ne keletkezzenek.

Az $n = 48$ bit hosszúságú adatblokkok (kódszavak) után lehetőség van egy bizonyos Z hosszúságú (pl. 16 bites) tiszta 0 sorozat fűzésére, és ez nem változtatja meg az alkalmazott kód tulajdonságait. Ezt az állítást igen könnyű belátni, hiszen így a gyakorlatilag $48+Z$ elemből álló kódról kell igazolni a szegmens, azaz vesszőmentességi tulajdonságot, ami nyilvánvalóan megmarad, hiszen csak a kód redundanciáját növeltük.

A most már $48+Z$ bites kódszavak segítségével továbbított információ nyilvánvalóan nem sérül meg, ha a kódszó utolsó, legfeljebb Z bitjét az átvitel során „elveszítjük". Észre kell vennünk, hogy módszerünk tulajdonképpen egy bitbeékelési eljárás speciális esetének tekinthető.

A felügyeleti adatgyűjtő hálózat valamennyi összeköttetése névlegesen 8 kbit/s sebességű. A tényleges sebesség a 2048 kbit/s$\pm$100 bit/s sebesség változásának megfelelően alakul. Ahhoz, hogy biztosan ne jöhessen létre információvesztés, azt kell meggondolnunk, hogy minden adattranzitálási pontban át kell emelni az egyik átviteli útról (f_1 bit/sec sebesség) a másik, továbbmenő átviteli útra (f_2 bit/sec) az adatokat. Szélső esetben az eltérés

$$\Delta f_c = 2 \cdot \frac{100}{256} \text{ bit/sec lehet}$$

A hálózat útkiválasztási stratégiája olyan, hogy legfeljebb 6 ilyen adatátemelés jöhet létre, amiből

$$f_d \cong 5 \text{ bit/sec.}$$

A biztonság érdekében célszerű ennél kissé nagyobbra választani a minden adatblokk után csatolt 0-k számát, például $Z=8$ szerint. A módszer természetesen csökkenti a csatorna kapacitását, de esetünkben ez nem elsőrendű kérdés. Ezeket a nullákat az adatblokk (kódszó) keletkezési helyén kell a szó után fűzni.

Az adatátemelések e nullák számát növelik vagy csökkentik a pillanatnyi órafrekvencia eltérések szerint. Az eljárás legnagyobb előnye, hogy semmiféle különleges bitbeékelés vezérlést nem igényel, a felesleges nullákat a központi vevőáramkörök figyelmen kívül hagyják.

Látható, hogy ezt az előnyt a redukált szegmens kódok tulajdonságai hordozzák magukban.

8.4. *A megvalósítás néhány rendszertechnikai jellemzője*

Az adatgyűjtő hálózat csomópontjaiban található kihelyezett egység elvi felépítését a 6. ábra mutatja. Az ábra szerinti vezérlő áramkör a visszirányú csatornák jelzései alapján kiválaszt egy utat, és az adóáramkör segítségével adatait ezen az úton továbbítja, ezenkívül sorban kijelöli adásra az $1, 2 \ldots N$ illesztőket és együttműködik az $1, 2 \ldots M$ vevőkkel.

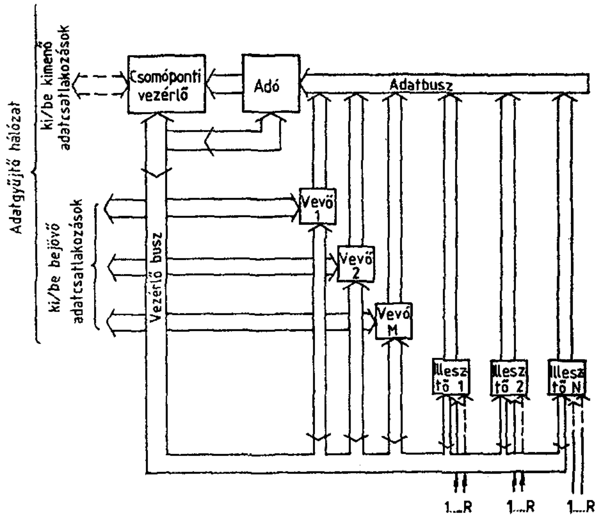
Az adatgyűjtő hálózat berendezéseinek részleteit másutt már ismertettük [45], [55], [56], itt csak néhány, a redukált szegmens kódokhoz közvetlenül kötődő részletet kívánunk kiemelni.

Az illesztő áramkörök feladata a felügyelt berendezések riasztási kimenetei (illetve általánosságban az információ-források) és az adatgyűjtő hálózat közötti kapcsolat megteremtése, beleértve a jelszintek elektromos konverzióját (pl. −48 V-ról TTL szintre) és a riasztási bemeneteken jelenlevő jelek multiplexelését a belső sínre.

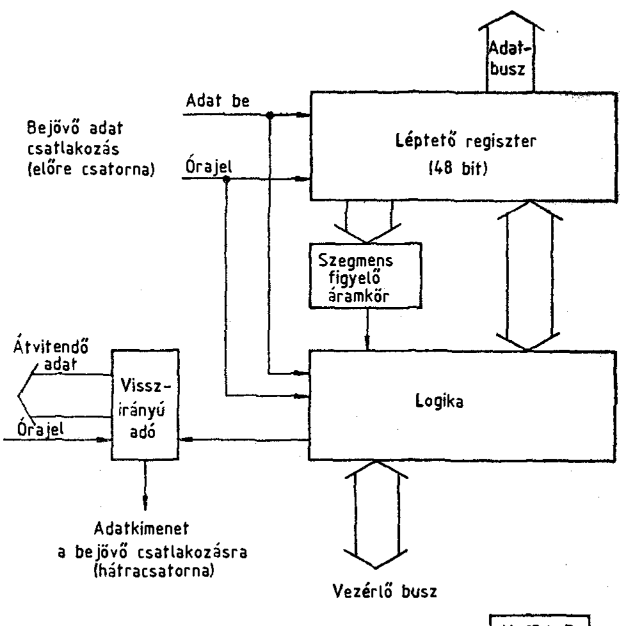
A kihelyezett egység vezérlője ciklikusan letapogatja az illesztők segítségével az egyes riasztási kimenetek állapotát, és továbbítja a kiválasztott adatátviteli útra. Így ezeken az utakon állandóan van forgalom, aminek természetes következménye az, hogy a tranzitpontokon mindig csak egy kimenő útra történik az adatok továbbítása, míg a tranzitálandó adatok egy időben több úton is érkezhetnek. Jelen alkalmazás szempontjából azonban ez semmilyen problémát nem okoz, mert valamennyi kihelyezett egység folyamatosan működik, és így valamennyi adat előre meghatározható időn belül eljut a központba.

Gyakorlatilag ezt a rendszertechnikai felépítést a redukált szegmens kódok felfűzhetősége teszi lehetővé.

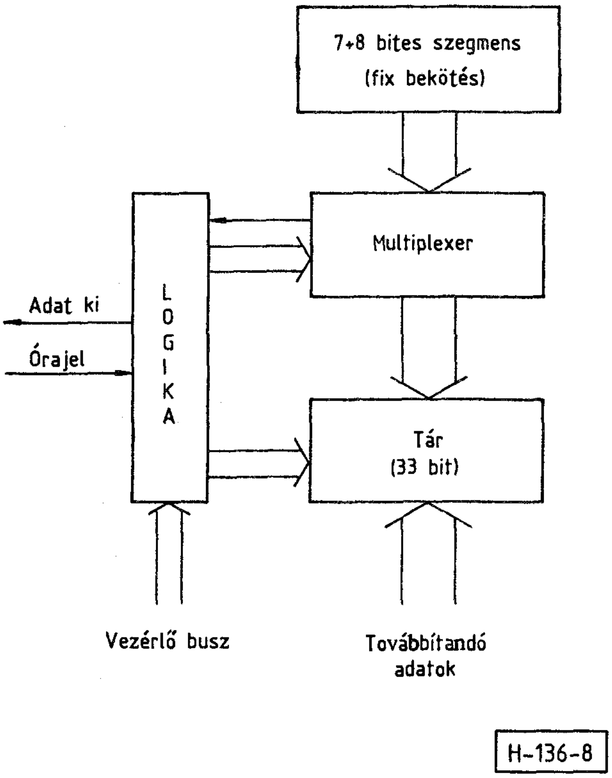
A kihelyezett egység adó áramkörének vázlatát a 7. ábra, a vevő áramkörét a 8. ábra mutatja. A vevő lényegében soros-párhuzamos átalakítást, az adó párhuzamos-soros átalakítást végez. Mindkét áramkörhöz egy minimális intelligenciájú logika is tartozik, melynek a vezérlővel történő együttműködésen kívül fel-



6. ábra. A kihelyezett egység rendszertechnikája



7. ábra. Adó áramkör blokkvázlata redukált szegmens kód előállításához



8. ábra. Vevő áramkör blokkvázlata redukált szegmens kód vételéhez

adata a hibák (pl. órajelkimaradás) detektálása, és az áramkör állapotára (aktív, passzív, hibajelzés) vonatkozó információk továbbítása a vezérlő felé.

A blokkvázlatokról leolvasható a redukált szegmens kódot előállító igen egyszerű adó áramkör és az önszinkronizáló tulajdonságot kihasználó vevő áramkör szerkezete.

A szinkronizált vételt a vevőben úgy lehet megvalósítani, hogy a szegmens figyelő — egyszerű kombinációs áramkör —, mely a léptető regiszter megfelelő megcsapolásaihoz kapcsolódik, a helyes szegmens detektálása esetén beíró impulzust ad a tárolókat tartalmazó logikának, melybe ekkor átírhatók a kódszó informatív bitjei.

A budapesti távbeszélőhálózat 12 központját látta el a Posta Kísérleti Intézet ilyen, a redukált szegmens kódokat felhasználó adatgyűjtő állomással, melyek a mintegy 2 éves üzemidő alatt beváltották a hozzájuk fűzött reményeket.

A vizsgált időszakban hamis riasztást egyetlen alkalommal sem észlelt a központi feldolgozó számítógép. Az adatátvitel gyorsaságára jellemző, hogy a rendszer segítségével sikerült behatárolni olyan, néhányszor 10 msec-ideig fennálló és pergő jellegű berendezéshibákat, melyeket egyébként gyakorlatilag lehetetlen lett volna azonosítani.

A rendszer, és így a redukált szegmens kódok alkalmazásának eredményességét az is jól jelzi, hogy eddig a fenntartó személyzet a hibák keletkezését követően (kisebb megbízhatósággal) átlagosan csak mintegy 180 perc múlva kapott pontos értesítést, így a jelenlegi néhány 100 msec-os késleltetés gyakorlatilag 3 órával csökkenti a berendezések kiesési idejét, növelve ezzel ezek használhatósági értékét.

A gyakorlati alkalmazás részleteit, a tapasztalatok elemzését [58] tárgyalja.

Függelék

A 7. fejezetben a redukált szegmens kódokra kimondott három tétel az alábbiak szerint bizonyítható.

Az 1. Tétel bizonyítása

A szóban forgó kód akkor optimális redukált szegmens kód, ha a szegmens vesszőmentességi indexe $\mu=1$.

Ez akkor teljesül, ha a kódszavak Δ deviációs mátrixainak minden sorában pontosan egy 1 értékű elem van, ahol a Δ képzéséhez a kódszó nem szegmens elemeit r-rel kell helyettesíteni. Ez valójában a 3. Állításból következik.

A Δ mátrix szerkezetének vizsgálatából rögtön látszik, hogy ott találhatók 1 értékű elemek, ahol két egymástól eltérő értékű szegmensbeli elem került egymás alá. Ebből, és a ciklikus tulajdonságból az következik, hogy a kódszó szegmensében (az egyenlőre nem definiált helyekre) kiválasztva két egymástól eltérő elemet, a mátrixban két 1 értékű pozíciót hoztunk létre. Nevezzük optimális konstrukciónak azt a kiválasztást, melynél rendre $u_0, u_1, u_2 \ldots u_{r-1}$ számú elemet jelölünk ki a szegmensben úgy, hogy u_0 a 0 értékű, u_{r-l} az $r-1$ értékű jelek száma, és minden szegmensbeli jel úgy van elhelyezve, hogy bármely másik jellel két különböző Δ-beli sorban generál egy 1-est, továbbá, hogy minden ilyen jel-pár különböző sorban hoz létre egy 1-et. Vagyis ekkor az optimális konstrukció összesen

$$E = 2\cdot[u_0\cdot(u_1+u_2+\ldots+u_{r-1})+u_1\cdot(u_2+\ldots+u_{r-1})+ +(\text{F.1.})+\ldots+u_{r-2}\cdot u_{r-1}] \qquad (\text{F.1.})$$

számú eltérést, tehát Δ beli 1 elemet kell, hogy létrehozzon, melyek a Δ mátrix különböző $n-1$ sorában találhatók, tehát

$$E = n-1.$$

A tétel állításainak bizonyításához azt is ki tudjuk még használni, hogy az $u_0+u_1+\ldots u_{r-1}$ összeget, vagyis a szegmens hosszát minimalizálnunk kell, természetesen úgy, hogy közben E a lehető legnagyobb legyen.

A tagok szorzata nyilvánvalóan úgy lesz maximális, és összegük minimális, ha

$u_0=u_1=u_2=\ldots u_{r-1}$, vagy ha ez nem lehet, akkor $|u_i-u_j|=1$, $\forall(i,j)\in\{0, r-1\}$ esetén, amiből az is következik, hogy egyenlőség esetén

$$n-1 = 2\frac{r\cdot(r-1)}{2}\cdot u_0^2, \text{ vagyis}$$

$$n = r\cdot(r-1)\cdot u_0^2+1.$$

Ez azt is jelenti, hogy ha $n\neq r\cdot(r-1)\cdot u^2+1$ alakú, akkor az ehhez szükséges szegmens hosszabb n_1 sorozat átvitelére is alkalmas, mert ha $u_i\neq u_j$ minden i, j-re, akkor az (F. 1.) szorzat olyan u_l értékekre teljesül, melyekre

$$n-1 < E = n_1-1$$

lesz.

A 2. Tétel bizonyítása

A 2. definíció értelmében az ugyanolyan szegmenseket tartalmazó kódszavakból álló kódot akkor nevezzük szegmentált kódnak, ha vesszőmentes tulajdonságú. A vesszőmentes tulajdonság azt követeli meg, hogy minden kódszótól legalább egy pozícióban eltérjen minden átlapolt kódszó ($\mu \geq 1$). A Δ mátrix megfelelő létrehozása (nem szegmens elemek értéke=r) pontosan azokat a pozíciókat jelöli 1 értékű elemmel, melyeknél eltérés van az eredeti, és az eltólt, átlapolt kódszavak között. Ha minden sorban van legalább egy 1, akkor $\mu \geq 1$, a szegmensre (a kódszó vesszőmentességi indexe a tényleges adatelemek értékétől függően ennél nagyobb lehet), tehát a kód valóban vesszőmentes tulajdonságú, ezért redukált szegmens kód.

A 3. Tétel bizonyítása

A bizonyításhoz vizsgáljuk meg, hogy mi az $(x-y)_{\mathrm{mod}\, n}$ tartalma esetünkben. Tekintve, hogy x és y is a szegmens valamely olyan pozícióját jelöli, melyben két egymástól eltérő elem van, $x>y$ esetén $(x-y)_{\mathrm{mod}\, n}=$ $=x-y$ pontosan azt adja meg, hogy az $(x-y)-$ szorosan ciklikusan átlapolt szóban lesz egy eltérő elem. Ha $x<y$, akkor $(x-y)_{\mathrm{mod}\, n}=\mathrm{n}+x-y$ azt adja meg, hogy az $(\mathrm{n}+x-y)$-szoros átlapolt szóban lesz egy eltérő elem.

Ha tehát a maradékrendszer nem teljes (vagy nem M-szeresen teljes), akkor a hiányzó szám azt jelzi, hogy a megfelelő ciklikusan átlapolt szó nem tér el (nem tér el M-szeresen) az eredeti szótól a kijelölt szegmens helyeken, így feltételünk valóban szükséges.

Ha a maradékrendszer teljes (M-szeres), akkor viszont éppen 1 (éppen M) eltérés lesz a szegmens pozíciókban az eredeti és az átlapolt szavak között. A feltétel tehát elégséges is.

IRODALOM

[1] *Lajkó S., Lajtha Gy. (szerk.)*: PCM a távközlésben. Műszaki Könyvkiadó Budapest, 1978.
[2] *Huszty G., Takács Gy., Sallai Gy., Wiener J.*: Digitális távközlő hálózatok. KÖZDOK Budapest, 1981.
[3] *Stiffler J. J.*: Theory of Synchronous Communication Prentice Hall. Englewood Cliffs, New Jersey, 1971.
[4] *Lindsey, W. C.*: Synchronization Systems in Communication and Control. Prentice Hall. Englewood Cliffs, New Jersey, 1972.
[5] *Bylanski P., Ingram D. G. W.*: Digital Transmission Systems. Peter Peregnius Ltd. England. 1976.
[6] *Gordos G., Varga A.*: Adatátvitel és adatfeldolgozás. Tankönyvkiadó Budapest, 1975 (kézirat).
[7] *Inose H.*: Introduction to Digital Integrated Communications Systems. Peter Peregnius Ltd. England, 1981.
[8] *Földes A.*: A PCM hálózatok szinkronizálása. Egyetemi doktori értekezés. Budapesti Műszaki Egyetem, 1972. (kézirat)
[9] CCITT, Yellow Book Vol. VIII. UIT, Geneva, 1985.
[10] *Rét A., Svéd J.*: Távadatfeldolgozó rendszerek. Műszaki Könyvkiadó Budapest, 1983.
[11] *Lucky, R. W., Salz J., Weldon E. J.*: Adatátvitel. Műszaki Könyvkiadó Budapest, 1983.
[12] CCITT, Yellow Book Vol. III. UIT, Geneva, 1985.
[13] *Gray J. R., Pan J. W.*: Using Digit Statistics to Word-Frame PCM Signals. BSTJ. Nov. 1964.
[14] *Zsiga Á.*: Blokkszinkron program. Diplomaterv. Budapesti Műszaki Egyetem, 1977 (kézirat).
[15] *Zsiga Á.*: Blokkszinkron rendszerek szinkronizációs időinek számítása. Híradástechnika XXX. évf. 1979. jan.
[16] *Haberle H.*: Frame Synchronizing PCM System. Electrical Communication. Vol. 44. No. 4. 1969.
[17] *Brugia O., De Seta D., Maggi W., Rossi C., Wolfowicz W.*: Design Criteria for Determining the Frame Format and the Frame Alignment Strategy Parameters of Digital Multiplex Systems. Alta Frequenza, Vd. XLVII. No. 7. 1978.
[18] *Golomb S. W., Gordon B., Welch L. R.*: Comma-Free Codes. Canadian J. Math. Vol. 10. 1958. pp. 202—209.
[19] *Kendall W. B., Reed I. S.*: Path-Invariant Comma-Free Codes. IRE Trans. on Information Theory. Oct. 1962.
[20] *Gilbert E. N.*: Synchronization of Binary Messages. IRE Trans. on Information Theory. Sept. 1960.
[21] *Ramamoorthy C. V., Tufts D. W.*: Reinforced Prefixed Comma-Free Codes. IEEE Trans. on Inf. Theory. Vol. IT—1.3 No. 3. 1967.
[22] *Mühlrád A., Dénes J.*: A fehérjeszintézis információelméleti vonatkozásai. Biológiai Közlemények X. kötet. 2. füzet, 1962.
[23] *Dénes J., Radó T.*: Véges struktúrák és digitális áramkörök kapcsolata. I. MTA Matematikai Kutató Intézetének Közleményei. IX. évf. B. sorozat, 4. füzet. 1964.
[24] *Stiffler J. J.*:Synchronization Methods for Block Codes. IRE Trans. on Information Theory. Vol. IT—8. Sept. 1962.
[25] *Stiffler J. J.*:Synchronization of Codes IRE Transactions on Space Electronics and Telemetry. Jun. 1962.
[26] *Eastman W. L., Even S.*: On Synchronizable and PSK—Synchronizable Block Codes. IEEE Trans. on Information Theory. Oct. 1964.
[27] *Eastman W. L., Even S.*: Some Further Results on Synchronizable Block Codes. IEEE Trans. on Inf. Theory. July 1966.
[28] *Golomb S. W., Gordon B.*: Codes with Bounded Synchronization Delay. Information and Control. 8. 1965. pp. 355—372.
[29] *Eastman W. L.*: On the Construction of Comma-Free Codes. IEEE Trans. on Inf. Theory. April 1965.
[30] *Scholtz R. A.*: Codes with Synchronization Capability. IEEE Trans. on Inf. Theory Vol. IT—12. no. 2. 1966.
[31] *Scholtz R. A.*: Maximai and Variable Word-Length Comma-Free Codes. IEEE Trans. on Inf. Theory Vol. IT—15. No. 2. 1969.
[32] *Tong S. Y.*: Synchronization Recovery Techniques for Binary Cyclic Codes. BSTJ. April 1966.
[33] *Levy J. E.*: Self-Synchronizing Codes Derived from Binary Cyclic Codes. IEEE Trans. on Inf. Theory. Vol. IT—12. No. 3. 1966.
[34] *Sellers F. F.*: Bit Loss and Gain Correction Code. IRE Trans. on Inf. Theory. Jan. 1962.
[35] *Ullman J. D.*: Near Optimal, Single-Synchronization-Error-Correcting Code. IEEE Trans. on Inf. Theory. Vol. IT—12. No. 4. 1966.
[36] *Ullman J. D.*: On the Capabilities of Codes to Correct Synchronization Errors. IEEE Trans. on Inf. Theory. Vol. IT—13. No. 1. 1967.
[37] *Calabi L., Hartnett W. E.*: A Family of Codes for the Correction of Substitution and Synchronization Errors. IEEE Trans. on Inf. Theory. Vol. It—15. No. 1. 1969.
[38] *Stiffler J. J.*: Comma-Free Error-Correcting Codes. IEEE Trans. on Information Theory. Jan. 1965.
[39] *Scholtz R. A.*: Frame Synchronization Techniques. IEEE Trans. on Communication. Vol. COM—28. No. 8. 1980.
[40] *Tanaka E., Kasai T.*: Synchronization and Substitution Error-Correcting Codes for the Levenshtein Metric. IEEE Trans. on Inf. Theory. Vol. IT—22. No. 2. 1976.
[41] *Iizuka I., Kasahara M., Namekawa T.*: Block Codes Capable of Correcting both Additive and Timing Errors. IEEE Trans. on Inf. Theory. Vol. IT—26. No. 4. 1980.
[42] *Siso L. V., Garcia J. B. R.*: On Detection of a Class of Synchronization Errors. IEEE Trans. on Inf. Theory. Vol. IT—27. No. 6. 1981.
[43] *Capocelli R. M.*: A Note on Uniquely Decipherable Codes. IEEE Trans. on Inf. Theory. Vol. IT—25. No. 1. 1979.
[44] *Bende S.*: Változó szóhosszúságú kódolás. Egyetemi matematikai doktori értekezés. Budapesti Műszaki Egyetem 1967 (kézirat).
[45] *Huszty G.*: Digitális jelsorozatok keretezési eljárásai hírközlő rendszerekben. Egyetemi doktori értekezés. Budapesti Műszaki Egyetem 1984 (kézirat).

[46] *Imai H.:* A Construction Method for Path-Invariant Comma-Free Codes. IEEE Trans. on Communications. July 1974.
[47] *Artom A.:* Choice of Prefix in Self-Synchronizing Codes. IEEE Trans. on Communications. April 1972.
[48] *Levitt* B. K.: Long Frame Sync Words for Binary PSK Telemetry. IEEE Trans. on Communications. Nov. 1975.
[49] *Clague* D. *J.:* New Classes of Synchronouos Codes. IEEE Trans. on Electronic Computers. Vol. EC—16. No. 3. 1967.
[50] *Niven J., Zuckerman H. S.:* Bevezetés a számelméletbe. Műszaki Könyvkiadó, Budapest 1978.
[51] *Huszty G., Wiener J.:* Data Network for Maintenance. Proc. of the VII th Colloquium on Microwave Communication. Budapest, 1982. pp. 56—59.
[52] *Wiener J., Huszty G.:* Hungary's Maintenance Data Network. Telephony, International Issue. Vol. 204. No. 26. June 1983.
[53] *Huszty G., Wiener J.:* Primer PCM rendszerek szervizcsatornáin kiépített adatgyűjtő hálózat fenntartási célokra. V. Országos Elektronikus Műszer és Méréstechnikai Konferencia kiadványa. Budapest, 1980.
[54] *Huszty G., Wiener J.:* A budapesti trönkhálózat központosított felügyelete. A Posta Kísérleti Intézet Tudományos Napjai 1980. c. kiadványa. Budapest, 1980.
[55] *Huszty G., Wiener J.:* Centralized Supervisory System for the Budapest Junction Network. Budavox Telecommunication Review. No. 3. 1982.
[56] *Huszty G.:* Digital Transmission Systems Centralized Supervisory. Proc. of the „Telecom '83" Conference, Brno, Czechoslovakia, 1983.
[57] *Sakurai K., Yoshida N., Yoshiska Z., Tsukui A., Akai Y.:* New Centralized Supervisory System for Transmission Lines. NEC Research and Development. No. 48. 1978.
[58] *Huszty G., Wiener J.:* Networking Aspects and Operational Results on the Budapest Trunk Maintenance Network. Budavox Telecommunication Review No. 3. 1985.
[59] *Korn G. A., Korn T. M.:* Matematikai kézikönyv. műszakiaknak, Műszaki Kiadó, Bp. 1975.