

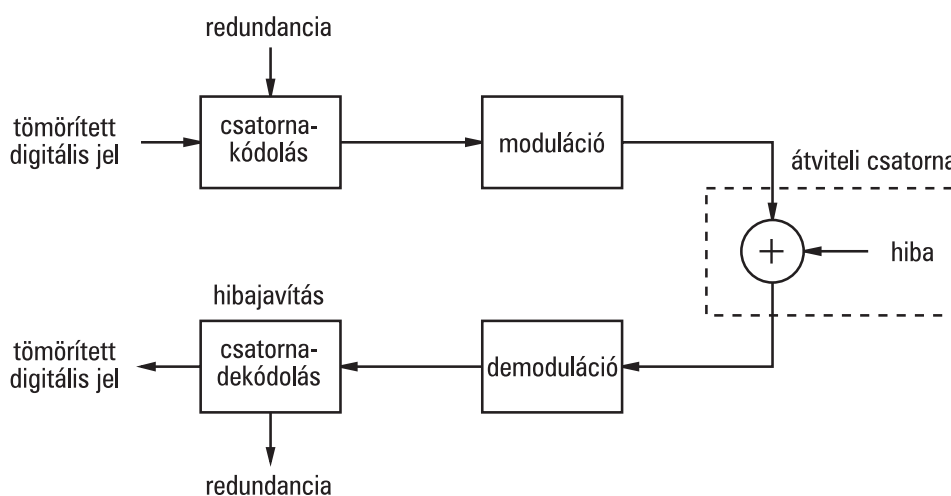
6 Digitális tv-műsorszórásban alkalmazott hibajavítási módok

Az MPEG szerinti forráskódolás során (3. és 4. fejezet) a digitális jelből redundanciát távolítottunk el, ami miatt azonban érzékenyebbé vált az átviteli út zavaival szemben. Ha egy digitális, nem forráskódolt jelet pl. 270 Mbit/s-mal viszünk át, és egy bit meghibásodik, az a legtöbb esetben a képen csak egyetlen képpontban lesz észrevehető. Ha ellenben egy MPEG-kódolású képet viszünk át, akkor egy bithiba legalább egy makroblokkot meghamisít.

Bithiba keletkezhet, ha a demodulátorban a jel a zajtól torzulva hibás bitkombinációként értelmeződik (lásd 7. fejezet). Ezért kell csatornakódolást alkalmazni, az ún. Forward Error Correction-t (FEC), magyarul előre irányban ható hibajavítást, amely hibajavító bitek segítségével képes a keletkezett hibák korrigálására. (Az ún. Backward Error Correction esetében egy vissz irányú csatornán az adó parancsot kap a hibásan átvitt adatsor újbóli átküldésére. A vissz irányú csatorna szükségessége miatt a műsorszórásban nem alkalmazható). Ebben a fejezetben a digitális tv-jelátvitelben releváns két hibajavító eljárással foglalkozunk.

6.1 Általános alapfogalmak

A hibajavító kódolás elvét a 6.1. ábra szemlélteti:



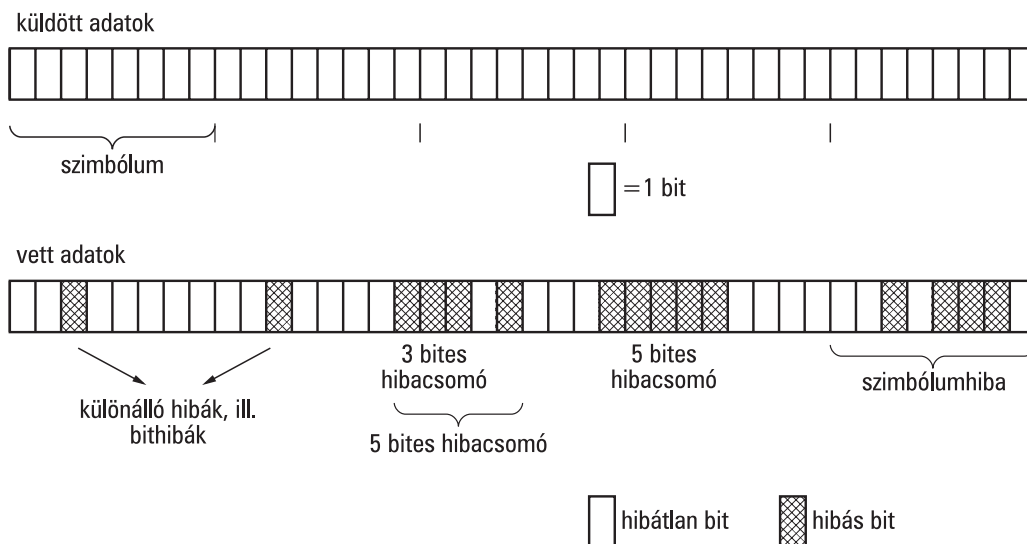
6.1. ábra
Átvitel hibajavító kódolással

A forráskódolással ellentétben a digitális forráskódolt jel redundanciáját az adási oldalon szándékosan megnöveljük, ami a vevő csatornadekódereben lehetővé teszi a keletkezett hibák korrekcióját. A redundancia megnövelésével, azaz további, a forráskódolt jelben nem lévő adatok átvitelével megnövekszik a továbbítandó adatmennyiség is, amit a modulációs mód megválasztásakor figyelembe kell vennünk, nehogy túllépjük a csatorna sávszélességét.

Ha a digitálisan modulált jel az átvitel során meghibásodik (mert pl. zaj adódik hozzá), akkor ez egy vagy több bit megváltozását okozza: az „1”-esekből „0”-ák lesznek, vagy fordítva. Csatornakódolás esetén a vevő dekódere a digitális jelhez adott redundancia

alapján meghatározza a hibás bitek pozícióját, majd invertálja azokat. Utána természetesen a redundanciát a jelből eltávolítjuk.

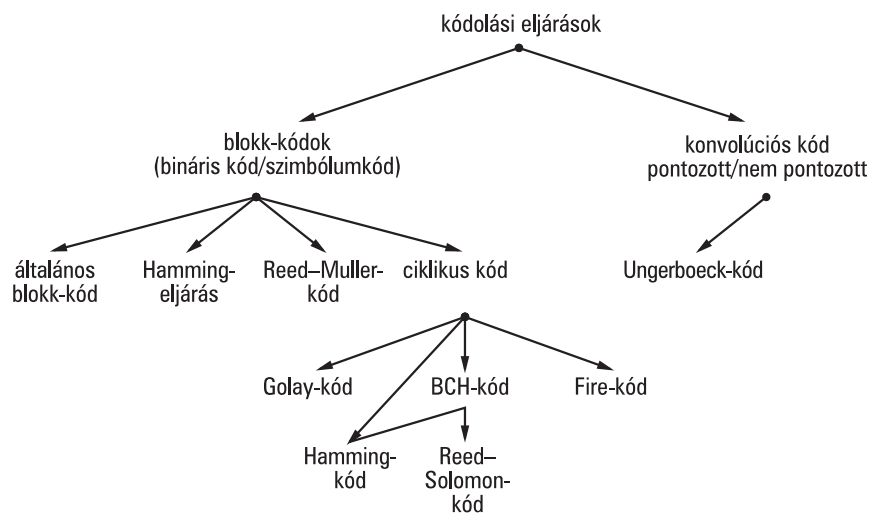
A 6.2. ábra a különböző jellegű hibákat tünteti fel. Az adatfolyam hibáit bithibának hívjuk. Ha egy n bitből álló bloknak legalább az első és az utolsó bite hibás, akkor n bites hibacsomóról vagy hibaburstról beszélünk.



6.2. ábra
Lehetséges hibafajták

A blokkon belül nem kell feltétlenül minden bitnek hibásnak lennie. Szimbólumhiba esetén a szimbólum hibásodik meg, ami pl. a 6.2. ábrán 8 bitből (1 byte) áll. A szimbólumon belül akár 8 bithiba is lehet.

Egy adott csatornában a ténylegesen fellépő hibafajták ismeretében különféle kódok konstruálhatók, amelyek a gyakran fellépő hibafajtákat különösen jól, míg másfélüket, pl. a kevésbé gyakoriakat kevésbé hatásosan korrigálják. A 6.3. ábra a leggyakrabban alkalmazott kódosztályokról ad áttekintést.

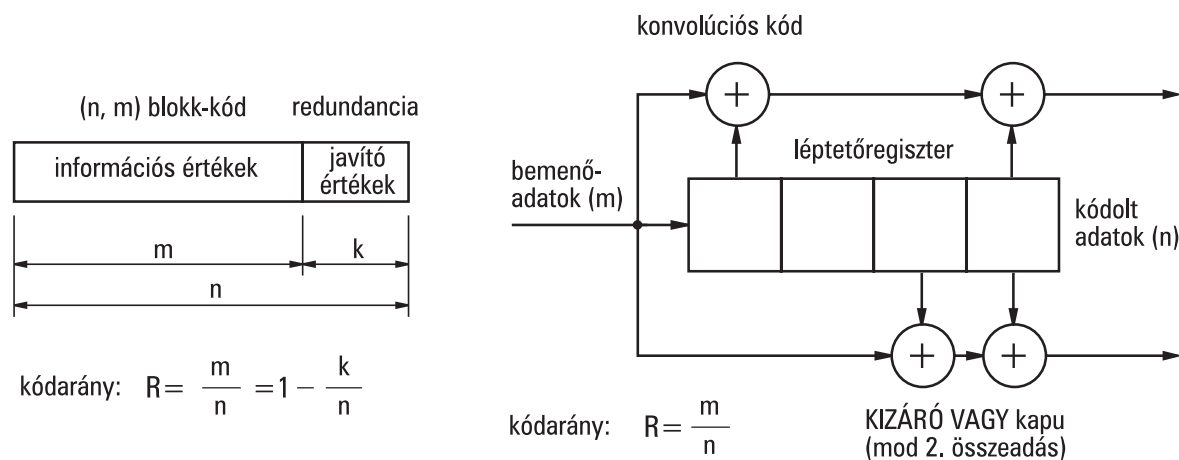


6.3. ábra
Hibajavító kódok osztályozása

A felsorolás természetesen nem teljes, és minden felsorolt kóddal nem is foglalkozunk majd.

Két nagy csoportot különböztethetünk meg, a blokk-kódokat és az ún. konvolúciós kódokat. Blokk-kódok esetében a bemeneti adatfolyamot rögzített m hosszúságú blokkokra osztjuk, ahol m a blokkban lévő szimbólumok számát jelenti. Egy szimbólum egyetlen bitből is állhat, vagy mint a 6.2. ábra példájában, akár többől is. A megfelelő kódokat az első esetben binér (vagy bináris) kódoknak, míg a második esetben többértékű vagy szimbólumkódnak hívjuk. Többértékű kódokkal nagyon jól lehet szimbólumhibákat javítani, és teljesen mindegy, hogy a szimbólum melyik bitjei hibásak. A hibajavítás ilyenkor nem csak a hibás szimbólum megtalálásából áll, hanem a szimbólum eredeti értékének meghatározásából is, amíg a bináris kódnál a hibás bitet csak invertálni kell. Blokk-kódoknál a kiszámított redundancia k számú javítási értéke a tulajdonképpen m számú információs érték után helyezkedik el, és így végül egy $n = m + k$ hosszúságú blokkot kell továbbítani. A kódarány az információs értékek számának (m) és az összes érték számának (n) a hányadosa.

A blokk-kódtól eltérően konvolúciós kódnál, amely itt mindig bináris kód, nem osztjuk fel a bemeneti adatfolyamot rögzített hosszúságú szegmensekre (szakaszokra), hanem sokkal inkább a bemeneti információval több kimeneti adat értékét befolyásoljuk. Ez a „befolyásolás” megvalósítható a bemeneti adatoknak léptetőregiszterben való tárolásával és a különböző leágazások értékeinek kombinálásával. Az ily módon történő kódolással a konvolúciós kódok az egyedi bithibák javítására predesztináltak. A kódarányt itt az egyszerre beolvasott és az egyszerre kiolvasott bitek számának hányadosaként definiáljuk. Úgy, ahogy a blokkkódoknál, a kódarány ez esetben is kisebb, mint 1, mert különben nem volna járulékos redundancia. A 6.4. ábra a blokk-kódok és konvolúciós kódok elvét ábrázolja.

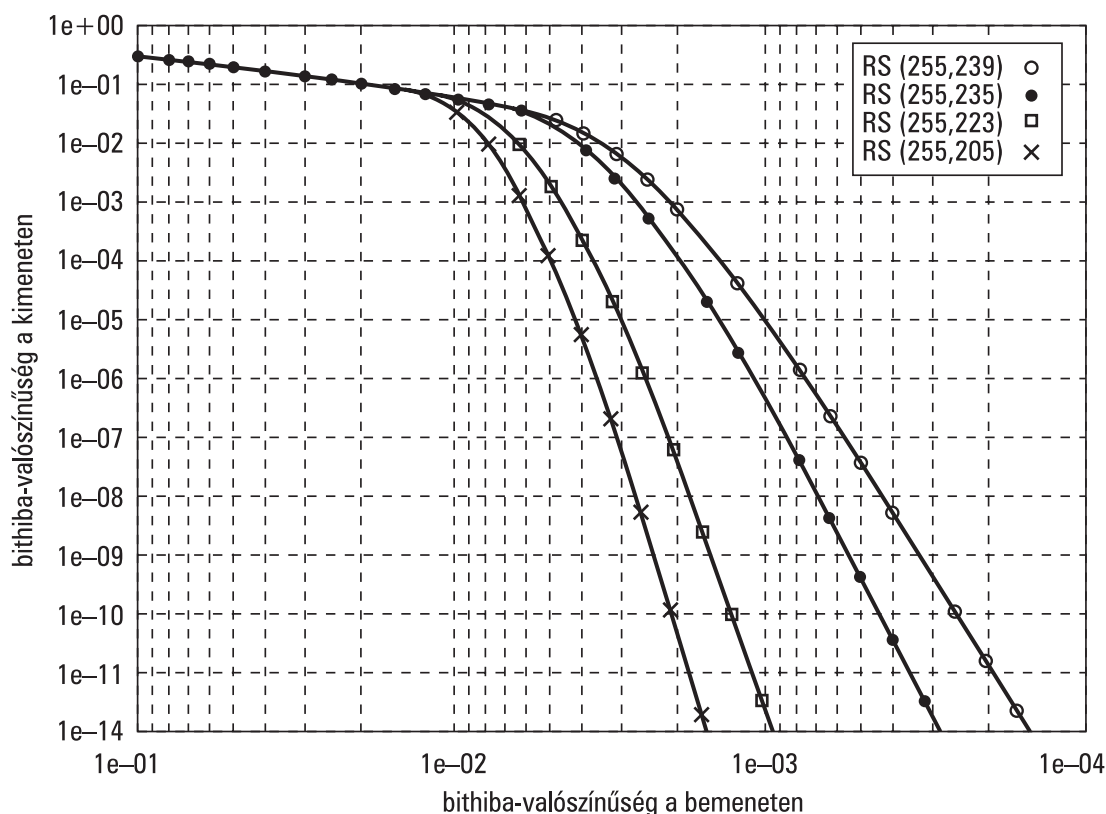


6.4. ábra
Blokk- és konvolúciós kódok

Végül ismerkedjünk meg a bithibaarány fogalmával. A bithibaarány (Bit Error Rate, BER) meghatározott időtartam alatt a hibásan vett bitek és az összes vett bit számának viszonya. Hasonlóan defináljuk a szimbólumhibaarányt (Symbol Error Rate, SER) is, amely a hibásan vett szimbólumok és az összes vett szimbólum viszonya.

6.2 Reed–Solomon-kódok

A Reed–Solomon-kód un. szimbólumorientált blokk-kód, azaz a hibajavításnak fel kell ismerni, hogy az n hosszúságú blokk melyik szimbóluma hibás, valamint meg kell határozni a hibátlan szimbólumot is. Egy szimbólum 8 bitből áll. Az RS-kódolás és -dekódolás számítási igénye jelentős.



6.5. ábra

Különböző Reed–Solomon-kódok bithiba-valószínűsége

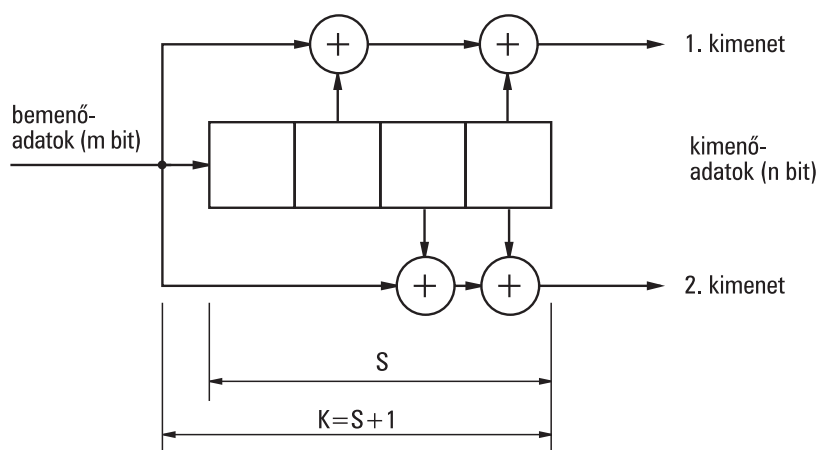
A 6.5. ábra az un. maradék bithiba valószínűség-görbét szemlélteti, azaz a dekóder kimenetén lévő bithiba-valószínűséget a bemeneti bithiba-valószínűség függvényében különböző Reed–Solomon-kódoknál. A bithiba-valószínűség a bithibaarány határértéke, amelyet a bithibaarány annál jobban megközelít, minél hosszabb a mérési idő, azaz minél több adatból történik meghatározása. Jóllehet a Reed–Solomon-kódok szimbólumorientáltak, a teljesítőképesség analízisékor mégis a bithibaarányt vizsgáljuk. Feltételezve, hogy a bithibák eloszlása egyenletes, akkor ebből a szimbólumhiba-arányra, azaz a kód hibajavító-képességére is következtethetünk, ami természetesen a javítószimbólumok számával nő. Így az RS(255,205)-kódnál (ahol a zárójelben lévő első szám n értékét, a második pedig m értékét adja meg) a maradék (kimeneti) bithiba-valószínűség kb. $1 \cdot 10^{-10}$, miközben a bemeneti $2 \cdot 10^{-3}$, ami 10^7 kódnyereségnek felel meg. Ezzel szemben az RS(255,239)-kódnál ugyanakkora ($2 \cdot 10^{-3}$) bemeneti bithiba-valószínűség esetén a kimeneti $9 \cdot 10^{-4}$, azaz a kódnyereség alig nagyobb 0,5-nél. A javítási határt akkor érjük el, ha a kimeneten a bithiba-valószínűség a bemenetivel egyenlő, vagy annál nagyobb. Az RS(255,205)-kódnál ez kb. $7 \cdot 10^{-3}$, az RS(255,239)-kódnál pedig $2 \cdot 10^{-3}$. Ettől a határtól kezdve több hiba lép fel, mint amit a kód javítani képes. A következmény járulékos, a javítóalgoritmus által keltett

hibák megjelenése a kimeneti adatfolyamban, ami ahhoz vezet, hogy a dekóder kimenetén a bemeneténél nagyobb lesz a bithiba-valószínűség.

A DVB-szabványban az 51 byte-tal rövidített RS(255,239)-kódot alkalmazzák, amelyik lehetővé teszi, hogy $2 \cdot 10^{-4}$ bemeneti bithiba-valószínűségnél a maradék-bithiba-valószínűség csak kb. $1 \cdot 10^{-11}$ lesz. Ezzel a kóddal blokkonként 8 szimbólumhiba javítható. Az RS(255,239)-kód 51 byte-tal rövidített változata az RS(204,188)-kód. A rövidítés azt jelenti, hogy a 188 üzenetbyte-hoz 51 db 0 értékű byte-ot teszünk, majd elvégezzük az RS-kódolást. Kódolás után az 51 db 0 byte-ot eltávolítjuk, és a 188 üzenetbyte után közvetlenül a generált redundancia-szimbólumokat tesszük.

6.3 Konvolúciós kódok

Ahogy már a 6.1 fejezetben láttuk, a konvolúciós kódok bináris kódok, amelyeknél az információt több átvíendő jelre „szétterítjük”. A konvolúciós kód tehát mindig bitorientált. A kódolásnál az információt, amely az egyes biteket jelenti, léptetőregiszterbe tápláljuk. A léptetőregiszter különböző leágazásainak kombinációjából kapjuk meg az átvíendő jelet. A 6.6. ábra szerinti relatíve egyszerű konvolúciós kódolásnál mindig egy bitet táplálunk a léptetőregiszterbe, míg a kimeneten két bit jelenik meg (kimenetenként egy-egy).



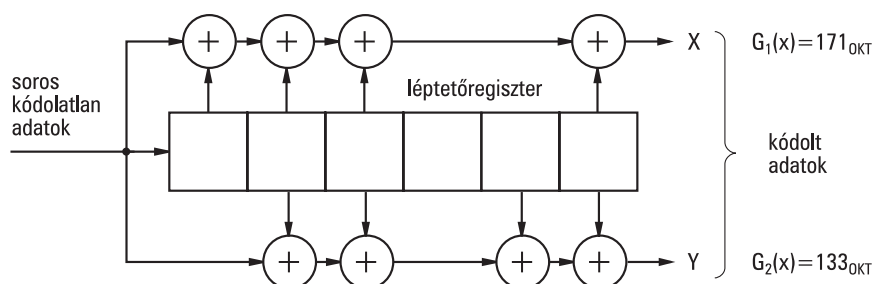
6.6. ábra
Példa konvolúciós kódolóra

A bemenőkeret hossza $m = 1$ bit, a kimeneti keret mérete $n = 2$ bit, így a kódarány $R = m/n = 1/2$. A kódoló emlékezetén, azaz tárolási mélységén az éppen aktuális bit kódolásához felhasznált korábbi bitek számát értjük. Ebben az esetben ez $Sm=4$, ahol S a léptetőregiszter hossza ($S=4$). Az ún. hatáshossz vagy kényszerhosszúság (angolul constraint length) K ellenben a kódolási folyamatban résztvevő összes bit számát jelenti. Példánkban $K=(S+1)m = 5$. Ha a konvolúciós kódolót mint egy automatát vizsgáljuk, akkor jellemezhetjük a lehetséges belső állapotainak számával is. Ennek értéke függ a tárolóhelyek, ill. a lehetséges 1/0 kombinációk számától. Az ábrázolt kódolónál a lehetséges belső állapotok száma $2^{Sm} = 16$. Végül is a konvolúciós kódolót a léptetőregiszter leágazásainak elrendezésével és számával jellemezzük. Ez gyakran az ún. G generátorpolinomok segítségével történik, amelyeknek együtthatói 1 vagy 0, attól függően, hogy az adott helyen van vagy nincs leágazás. Az együtthatókat csoportosítva, oktálszámokkal is megadhatjuk a konvolúciós kódolót. Akár az egyik, akár a másik megoldást választjuk, azt szem előtt kell tartani, hogy mindkét kimenethez tartozik polinom, ill. oktálszám. Foglalkozunk össze a vizsgált konvolúciós kódoló legfőbb adatait:

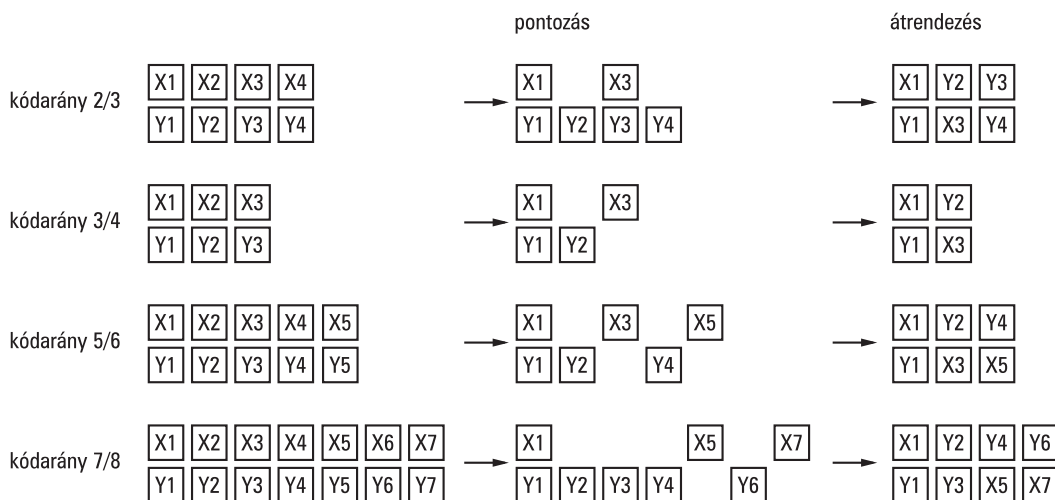
Bemeneti kerethosszúság:	$m = 1$	
Kimeneti kerethosszúság:	$n = 2$	
Kódarány:	$R = m/n = 1/2$	
Emlékezet:	$S \cdot m = 4$	
Lehetséges állapotok száma:	$2^{S \cdot m} = 16$	
Kényszerhossz:	$K = (S + 1)m = 5$	
Generátorpolinom 1:	$G_1 = 1 + x^2 + x^4$	(25 _{OKT})
Generátorpolinom 2:	$G_2 = 1 + x^3 + x^4$	(23 _{OKT})
		$\begin{array}{c c} 10 & 011 \\ \hline 2 & 3 \end{array}$

A konvolúciós kódolásnál megtehetjük, hogy a kimeneti jelszekvencia egyes bitjeit meghatározott séma szerint elhagyjuk vagy más szóval pontozzuk. Így különböző kódarányokat hozhatunk létre, mindig a megkívánt védelmi követelményeknek megfelelően. Természetesen a vevőnek ismernie kell a sémát, amit számára ún. pontozási vektor vagy pontozási mátrix formájában közlünk. A változó kódarányú kódolás ilyen megvalósításának előnye, hogy a dekódoló az összes lehetséges kódarányhoz alkalmazható. A pontozott (elhagyott) bitek helyére a vevő dekódere helyettesítő biteket iktat, aminek révén visszaáll az eredeti órafrekvencia, ill. sebesség. Minél több a kipontozott bit, a hibajavítási hatékonyság természetesen annál kisebb.

A 6.7. ábra a DVB-szabvány szerinti konvolúciós kódolót, a 6.8. ábra pedig a különböző kódarányok pontozási sémáját szemlélteti.



6.7. ábra
DVB-szabvány szerinti konvolúciós kódoló

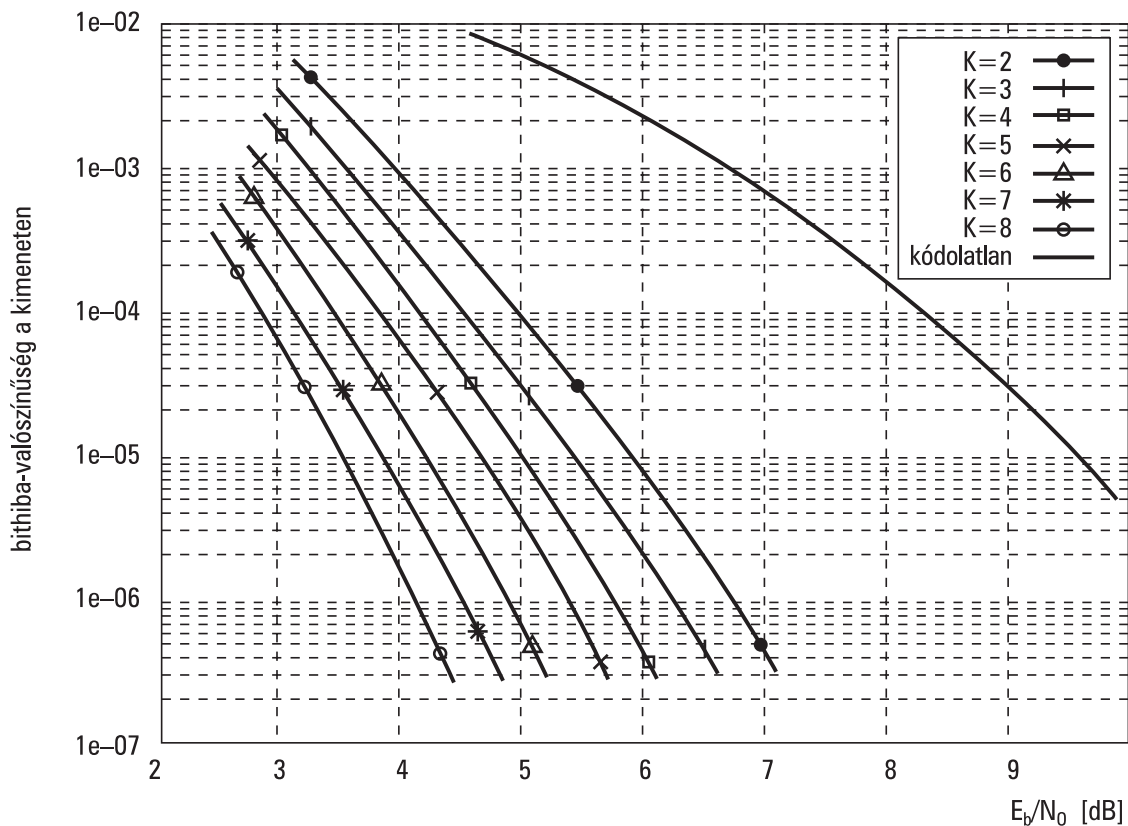


6.8. ábra
Pontozási séma a különböző kódarányokhoz

Az $\frac{1}{2}$ -es kódarányt pontozás nélkül kapjuk. A pontozás után megmaradt biteket át kell rendezni, hogy a 6.7. ábra szerinti két ágon történő adattovábbítás továbbra is megmaradjon. Pl. műholdas átvitelnél a QPSK-modulátor I és Q moduláló jele ez az átrendezett, két ágon érkező jelfolyam lesz.

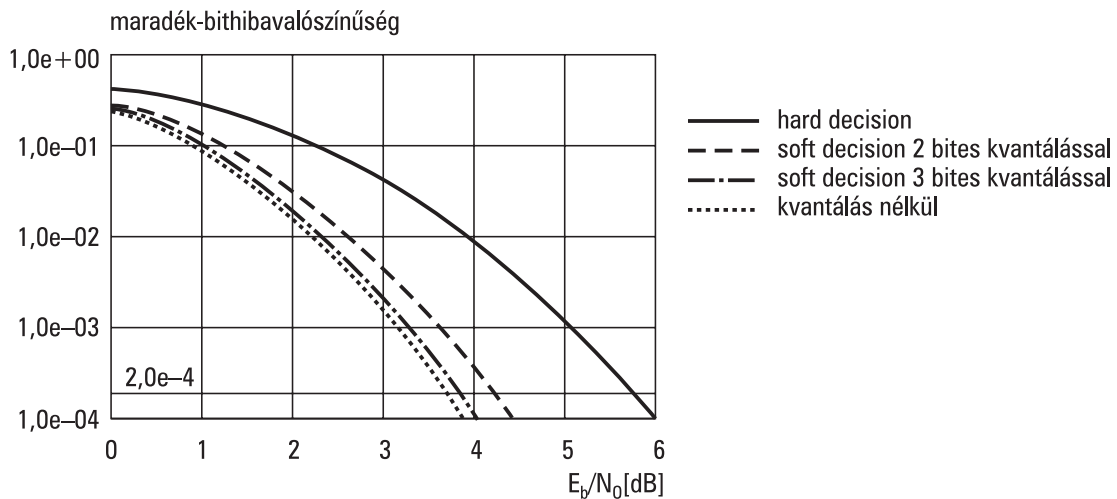
A dekóder feladata a meghibásodott bitek felismerése, ill. kijavítása, ami pl. a Viterbi-féle algoritmussal megvalósítható. Ehhez ismerni kell a kódoló belső állapotait, azaz a léptetőregiszter lehetséges tartalmait, és az állapotok közötti átmeneteket. Az átviteli hibát úgy fedezi fel a dekóder, hogy a hibás bitsorozat nem megengedett állapotátmenethez vezet.

A 6.9. ábra a konvolúciós kódolással elérhető maradék-bithibavalósínűséget adja meg $R = \frac{1}{2}$ -es kódarányánál E_b/N_0 függvényében, ahol E_b a bitenként átvitt energia, N_0 pedig a jelhez adódó Gauss-eloszlású fehérzaj kétoldalas spektrumának teljesítménysűrűsége.



6.9. ábra
 $R = \frac{1}{2}$ -es kódarányú konvolúciós kód
 maradék-bithibavalósínűsége QPSK-modulációnál

A moduláció QPSK, K pedig a Constraint Length, azaz a hatáshossz. A hibajavítás teljesítőképessége, ahogy azt várjuk is, az alkalmazott kódoló hatáshosszával nő. Ha pl. a QPSK-demodulátor bemenetén E_b/N_0 5 dB, akkor $K=2$ -nél a maradék-bithibavalósínűség alig lesz kisebb a kódolatlan átvitelhez tartozó érték 100-ad részénél, míg $K = 6$ esetén az arány már 10^4 -szeres.

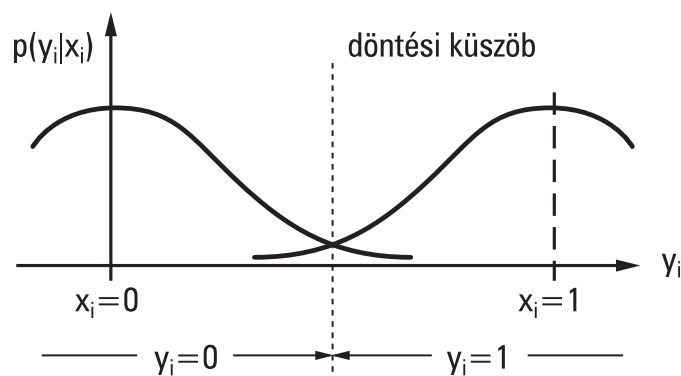


6.10. ábra
DVB konvolúciós kódjának maradék-bithibavalósínúsége
QPSK-modulációnál, ha $R = \frac{1}{2}$

A DVB-szabvány a 6.7. ábra szerinti konvolúciós kódolót írja elő, amelynek teljesítőképességét ugyancsak $\frac{1}{2}$ -es kódaránynál, QPSK-moduláció esetén a 6.10. ábra érzékelteti. Attól függően, hogy a demodulátorban hard vagy soft decisiont alkalmazunk (bővebben a 6.4 pont alatt), különböző görbéket kapunk. $E_b/N_0 = 4$ dB és soft decision esetén a maradék-bithibavalósínúség $2 \cdot 10^{-4}$, amely a Reed–Solomon-dekóder számára a még megengedhető maximális érték, és amivel az végül $1 \cdot 10^{-11}$ -nél kisebb maradék-bithibavalósínúséget biztosít.

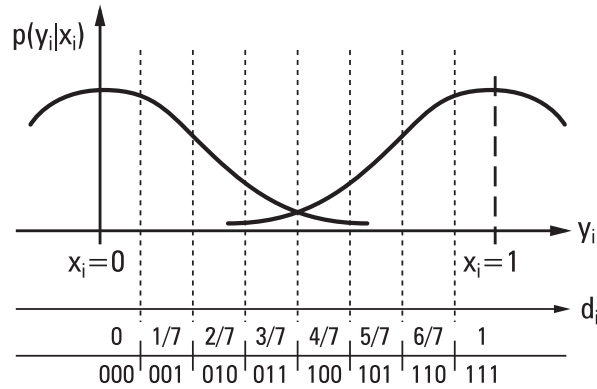
6.4 Hard és soft decision

Vizsgáljuk az egyszerűség kedvéért egy kétértékű digitális jel átvitelét. Az adó által küldött jel $x_i=0$, vagy $x_i=1$. Vételi oldalon ez a két diszkrét érték azonban a jelhez adódó zaj miatt tág értékhatárok között változhat. A vett jelértékeket valószínűségi változónak tekintve, a sűrűségfüggvények a 6.11. ábra szerinti lesznek. A sűrűségfüggvény definíciója szerint $p(y_i|x_i)dy_i$ megadja, hogy az y_i jel mekkora valószínűséggel vesz fel valamely y_i és $(y_i + dy_i)$ közötti értéket, ha a küldött jel x_i (feltételes valószínűség).



6.11. ábra
Sűrűségfüggvények bináris jelátvitelnél

Hard decisionnál (keménydöntésnél) az y_i értéktartományt egy döntési küszöbvel két részre osztjuk, és a küszöb feletti y_i értékekhez „1”-et, a küszöb alattiakhoz pedig „0”-t rendelünk. Soft decision esetén nem csak azt vizsgáljuk, hogy a vett jel a küszöb melyik oldalára esik, hanem még azt is megnézzük, hogy mennyire tér el a küszöb feszültség értékétől. Ezzel a döntés minőségét is meghatározzuk. A minőséginformációt – a vett analóg jelnek a döntési küszöbtől való távolságát – 3 vagy 4 bittel kvantáljuk, azaz y_i értéktartományát 8 vagy 16 részre osztjuk, és 3, ill. 4 bit segítségével megadjuk, hogy a jel mekkora értékű. (6.12. ábra).



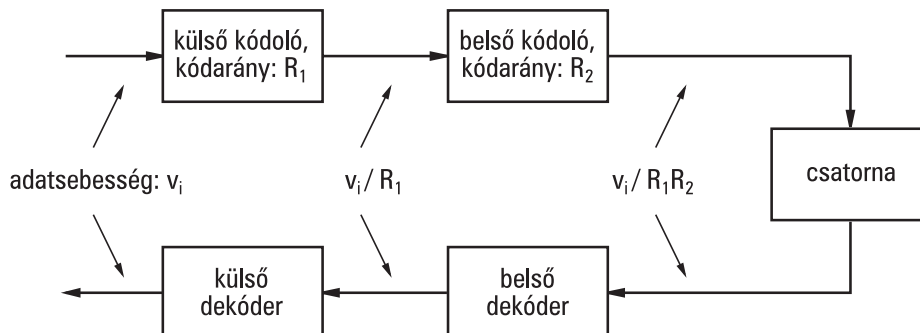
6.12. ábra
 y_i értéktartományának felosztása soft decisionnál

Ha a hibadetektálás során azt tapasztaljuk, hogy pl. egy kódszó valamelyik bite meghibásodott, akkor nagyon valószínű, hogy a döntési küszöböt átlépő és így hibát okozó jel nem sokkal tért el a küszöb értékétől, tehát a minőséginformációja 011, ill. 100 lehetett. Ezzel a hibás bitet lokalizáltuk, invertálásával a hibát ki is javíthatjuk.

A Viterbi-algoritmusba bevonható a soft decision minőséginformációja, és ezáltal a konvolúciós kód még hatékonyabbá válik, ahogy azt a 6.10. ábrán is látjuk.

6.5 Kódok összekapcsolása

Különböző kódok együttes alkalmazása révén (concatenation) a hibajavítás teljesítő-képessége tovább fokozható, ami azonban az átvitel adatsebességének növekedését is okozza. A 6.13. ábra két kód, ill. kódoló egymás utáni kapcsolását szemlélteti. Az átvendő információ adatsebessége v_i , az elsőként alkalmazott kód kódaránya R_1 . Ezt hívjuk külső kódnak, mert a kóder és a dekóder az átviteli lánc legelején, ill. a legvégén található.

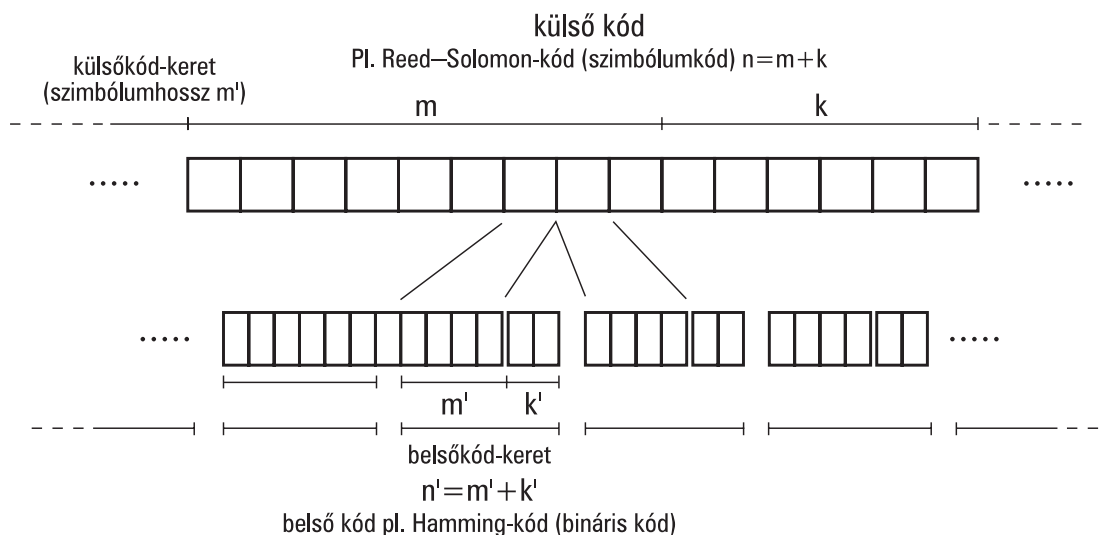


6.13. ábra
Kódok együttes alkalmazása (concatenation)

A második R_2 kódarányú kódot belső kódnak hívjuk. Összességében így az átviteli adatsebesség v_i/R_1R_2 lesz. A kódok együttes alkalmazásának előnyét két blokk-kód egymás után „kapcsolásának” példáján a következő szakaszban vizsgáljuk.

6.5.1 Blokk-kódok egymásba illesztése

A 6.14. ábra példája azt szemlélteti, hogy mit értünk két blokk-kód egymásba illesztésen.

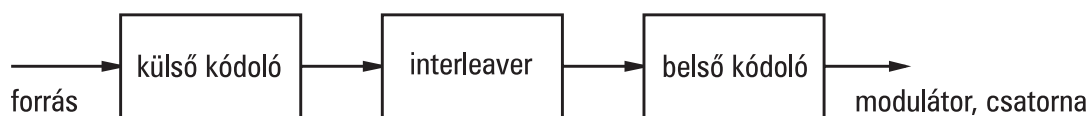


6.14. ábra
Blokk-kódok egymásba illesztése

A külső kód most Reed–Solomon-kód, amelynél az m információsztimbólumhoz k redundanciasztimbólum tartozik. A külső kód kerethossza így $n = m + k$. Mindegyik szimbólum m' bithől áll. A belső kód ez esetben egy Hamming-kód, amely az egyes szimbólumok m' bitjéhez k' redundanciabitet csatlakoztat hibavédelem céljából. A dekóderben egyetlen bithiba a belső kóddal javítható, és így a hozzátartozó szimbólum hibátlan lesz, azaz a külső kódolót a bithiba nem fogja terhelni. Feltéve, hogy a belső kód mindegyik keretében csak egy bithiba fordul elő, és az a belső kóddal javítható, akkor a külső kód egy keretén belül n bithiba léphet fel anélkül, hogy a külső dekódernek hibát kellene javítania. Belső kód nélkül a külső dekóder felmondaná a szolgálatot. Ha egynél több bithiba lépne fel a belső kód néhány keretében, akkor ezt a belső kóddal nem lehet javítani. A külső kód vonatkozásában így csak a több hibát tartalmazó szimbólumok maradnak hibásak, amit viszont a külső dekóder ki tud javítani. A 6.14. ábra szerinti kétszeres kódolás tehát kombinálja az egyes kódok előnyeit, azaz a belső kód jó bithibajavító képességét a külső kód jó bursthibajavító képességével.

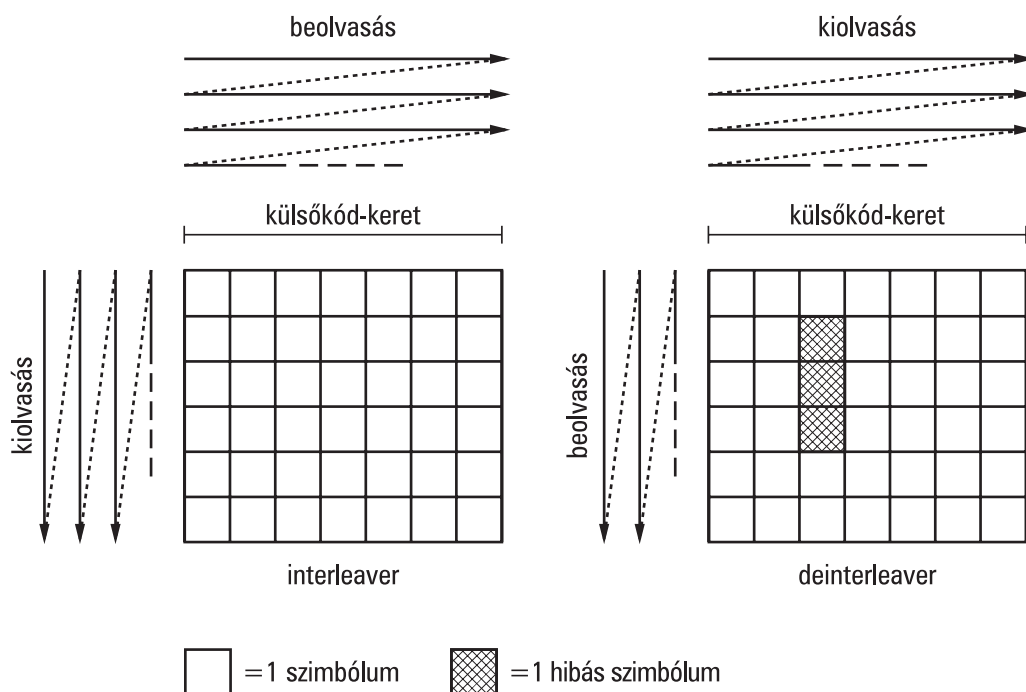
6.6 Interleaving

Annak érdekében, hogy a bithibák és a rövid bursthibák mellett a hosszú bursthibák is javíthatók legyenek, a külső és a belső kódoló közé a 6.15. ábra szerint egy interleavert kapcsolunk.



6.15. ábra
Interleaving csatorkódolásnál

Az interleaver nem végez további hibajavító kódolást, hanem csupán átrendezi, ill. egymásba szövi a külső kódoló szimbólumait. Az elvet a 6.16. ábra szemlélteti. A külső kódoló által létrehozott szimbólumokat (itt 1 szimbólum = 1 byte) soronként a blokkinterleaver mátrix-tárolójába tápláljuk, majd a mátrixot oszloponként kiolvassuk, és a szimbólumokat a belső kódolóba vezetjük. A külső kódoló által generált két egymás melletti szimbólum a kiolvasás után pontosan olyan távol lesz egymástól, ahány sorból áll a mátrix.



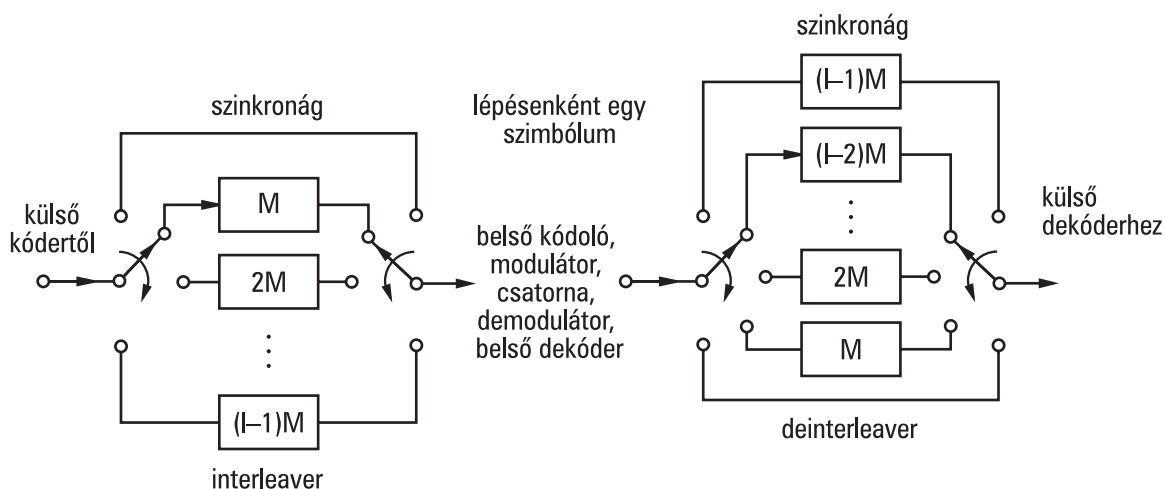
6.16. ábra
Blokinterleaver elve és működése

Ezzel a számmal jellemezhetjük az interleavert. Elnevezése interleavingmélység vagy átszövési mélység, jelölése I . Példánkban $I=6$.

Vételi oldalon az átvitel során esetleg meghibásodott szimbólumokat oszloponként olvassuk be a deinterleaver mátrixtárolójába. Egy hosszabb bursthiba így egy oszlopban jelenik meg, amint az a 6.16. ábrán is látható (szürke négyzetek). Soronként kiolvasva a mátrixot, a külső kód kereteiben csak egy hiba lép fel. Interleaver és deinterleaver nélkül a burst által okozott összes szimbólumhiba a külső kód egyetlen keretében lenne, aminek következtében a dekóder felmondaná a szolgálatot.

A blokkinterleaver hibája, hogy azok a periódikus zavarok, amelyek mindegyik oszlopban az ugyanabban a sorban lévő szimbólumot hamisítják meg, a külső dekóder működésképtelenné válását okozzák, mert ilyenkor valamennyi szimbólumhiba a deinterleaver egy sorában található. Ezenkívül hátrányos a viszonylag nagy tárigény, és az,

hogy a dekóder un. kétdimenziós szinkronizálást igényel, azaz nem csak a külsőkód-szavak elejét kell megtalálni, hanem még az első külsőkód-keretet is, amely a deinterleaver első sorában van. Ezekről a hátrányokról mentes a 6.17. ábra szerinti Forney-féle konvolúciós interleaver.



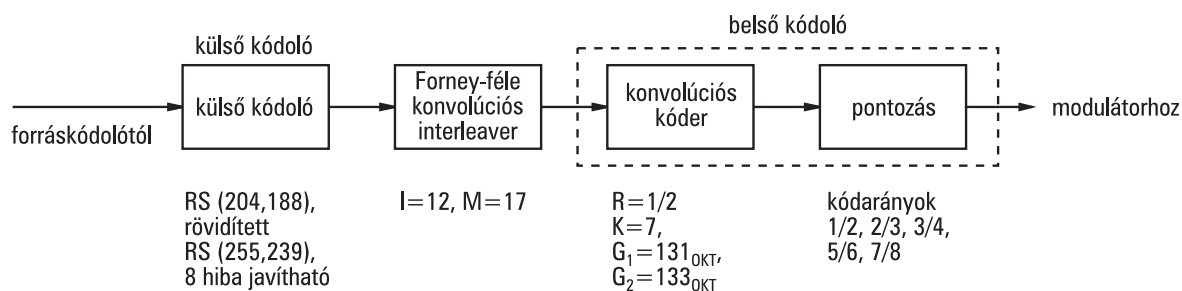
6.17. ábra
Konvolúciós interleaver elve és működése

A konvolúciós interleaver $(I-1)$ First-In-First-Out (FIFO) léptetőregiszterből áll, amelyek hossza $M, 2M, \dots, (I-1)M$, valamint a megfelelő multiplexerből és demultiplexerből (itt kapcsolókkal ábrázolva), amelyek mindig egy léptetőregisztert iktatnak a bemenet és a kimenet közé. I most is az átszövési mélységet jelenti, M az un. báziskésleltetés, amelyre érvényes, hogy $M = n/I$, ha n továbbra is a külső kód kerethosszúsága. A multiplexerek és demultiplexerek minden lépésnél a következő be-, ill. kimenetre kapcsolnak tovább. Ilyenkor egy szimbólum az éppen aktuális léptetőregiszterbe beíródik, a kimeneten pedig egy az éppen aktuális léptetőregiszterből a beérkezésnek megfelelő sorrendben kiolvasható. Ha az interleaverben pont a felső un. szinkronág az aktív, akkor a bemenet a kimenettel közvetlenül össze van kapcsolva. Az interleaver tehát biztosítja, hogy a bemeneten szomszédos két szimbólum között $M \cdot I$ további szimbólum legyen átvihető.

A deinterleaver úgy van felépítve, hogy benne az interleaver nem késleltetett szimbólumai lesznek a maximális késleltetésűek. Így a teljes késleltetés minden szimbólum esetében $M(I-1)$. Minden egyes külsőkód-keret kezdetekor a multiplexernek és a demultiplexernek kiindulási helyzetben kell lennie, azaz csak „egydimenziós” szinkronizációra van szükség. A külső kód n kerethossza az I átszövési mélység egész számú többszöröse. Ebben az elrendezésben az átszövési mélységnek nagyobbnak vagy egyenlőnek kell lennie az interleavert követő konvolúciós kódoló kényszerhosszával.

6.7 Hibajavítás a műholdas, a földfelszíni és a kábeles átvitelnél

A digitális tv-jel műholdas és földfelszíni sugárzásánál a 6.18. ábra szerint Reed–Solomon- és konvolúciós kódolást alkalmaznak interleaver közbeiktatásával. A Reed–Solomon-kód szimbólumhossza 8 bit. A szabvány az RS(255,239)-kódot írja elő, amelynél a 239 szimbólumból álló blokkhoz 16 redundanciaszimbólum tartozik, és ezzel 8 szimbólumhiba javítható.

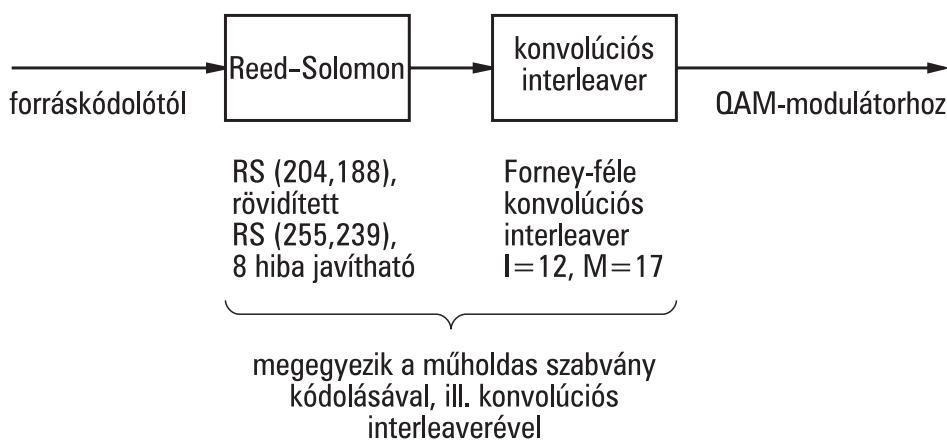


6.18. ábra

Hibajavító kódolás digitális tv-jelek műholdas és földfelszíni sugárzásánál

Mivel egy MPEG-2 transzportcsomag 188 byte hosszúságú, ezért az RS(255,239)-es kód rövidített változatát, az RS(204,188)-as kódot alkalmazzák. A külső kódolást követi a Forney-féle interleaving. Az átszövési mélység $I=12$. A külső kód $n=204$ kerethosszúságával a báziskésleltetés $M=n/I=17$. Végül az átrendezett szimbólumok konvolúciós kódolása következik, amelynél a kódarány $R=1/2$ és a hatáshossz (constraint length) $K=7$. A léptetőregiszter leágazásainak helye a két generátorpolinommal adható meg: $G_1=171_{\text{OKT}}$, $G_2=133_{\text{OKT}}$. Választható pontozással $2/3$, $3/4$, $5/6$ és $7/8$ kódarányok lehetségesek.

A szélessávú kábelhálózatokon történő átvitelhez hasonló kódolást alkalmaznak (6.19. ábra), csupán a konvolúciós kódolást hagyják el, mivel a jel-zaj viszony a kábelben egy nagyságrenddel nagyobb, mint a műholdas sugárzásnál.



6.19. ábra

Hibajavító kódolás digitális tv-jelek műsorelosztó kábelhálózatokon történő átvitelénél

A következő fejezetben a rendszerek teljes kialakításával és teljesítőképességükkel foglalkozunk.