A CRC kódolás

(szájbarágósan)

Mielőtt, a konkrét algoritmus tárgyalásához kezdenénk, tekintsünk át a GT(2) testalgebrát és e test feletti polinomok ábrázolását. A ***test*** szó az algebrában használatos fogalmat jelöl. Nevezetesen, egy alaphalmazt és a rajta értelmezett négy műveletet jelenti, azok sajátságaival együtt (pl.: összeadás és a szorzás itt kommutatív stb.). Az általános iskolában tanult összeadás, kivonás, szorzás, osztás (a négy alapművelet) is *test* algebrát alkot, a valós számok felett (azokon értelmezve). Hasonló, mégis más jellegű, a *csoportalgebra* (egy műveletes), más a *gyűrűalgebra* (amit vektor és mátrix algebraként ismerünk, itt pl. a szorzás ált. nem kommutatív. Az osztás nincs is értelmezve, ezért kell a mátrix inverziót megtanulni, ezzel már szorzásra visszavezethető).

1. A Mod(2) algebráról

Mod(2) algebrán, olyan algebrai műveleteket értünk, melyek a GT(2) un. ***Galois test*** feletti műveletekből áll. GT(2)-n egy alaphalmazt értünk, melynek elemei: {0,1}. A műveletekre érvényes a zártság, ami azt jelenti, hogy bármely művelet eredménye nem vezethet ki az alaphalmazból. Az értelmezett műveletek: a négy ismert alapművelet. A műveleteket helyértékenként végezzük de átvitelek nélkül. A helyértékeken keletkezett eredmény mindig a mod(2) osztálybeli maradék (ami csak 0 vagy 1 lehet).

Pl: összeadás, kivonás:

A	B	Y=A±B
0	0	0
0	1	1
1	0	1
1	1	0

Tehát itt 1+1=0 és az 1-1=0. Az 1-0=1 vagy az 1+0=1 de 0-1=1 is azt adja.

A táblázatból látható, hogy a fenti műveleteket a logikai algebrából jól ismert EX-OR (antivalencia) művelettel analóg. Ilyen kapuval elvégezhető.

Szorzás, osztás: a logikai És-műveletnek felel meg, amint azt a következő példában (CRC algoritmus) látjuk.

2. GT(2)-beli polinomok ábrázolása:

Legyen pl: *P(x)* polinom

$$P(x) = x^{12} + x^{9} + x^{4} + x^{3} + x^{2} + 1$$

Ez megfelel egy *n* bites bináris sorozatnak, ahol a legmagasabb hatványkitevőjű tag kitevője: *n-1*. Itt most, *n-1=12*, tehát a polinom egy *n=13* bites sorozat. Egyeseket azokra a helyekre írunk, mely tagok polinombeli együtthatója nem zérus, a legmagasabb helyérték mindig ilyen. Így:

$$P(x) = 1001000011101$$

3. A kódolás algoritmusa:

Írjuk fel az elküldendő bináris kódot M(x)-et, mint egy polinomot, amelynek a fokszáma a kódban lévő bitek száma mínusz egy. K=b-1
Legyen egy előre megadott, un. generátor-polinom P(x) , fokszáma: n, (így P(x) n+1 bites bináris sorozat) . A P(x) polinom különleges sajátosságú, un. *irreducibilis polinom*. Ez azt jelenti, hogy 1-en és önmagán kívül nem osztható maradék nélkül más polinommal (mint a prímszámok), azaz prím-modulusú polinom.

<u>Adó oldali algoritmus:</u>
A küldendő információt egészítsük ki a generátor-polinom P(x) fokszámának (n)

$$M(x) * x^{n}$$

megfelelő számú zérussal. Ezt algebrailag egy x^n-el való szorzással kapjuk. Ezután az így kapott polinomot (hosszban megnyújtott n-db nullával megtoldott bináris kódot) a generátor-polinommal, P(x)-el elosztjuk

$$\frac{M(x) * x^{n}}{P(x)} = Q(x) + \frac{R(x)}{P(x)}$$

Az így kapott R(x) maradék-polinomot, az eltolt M(x)* x^n végén lévő zérusok helyére beillesztjük. Algebrailag ez összeadást jelent:

$$T(x) = M(x) * x^{n} + R(x)$$

A kapott T(x) lesz az üzenet polinom. Ez kerül a csatornára.

<u>Vevő oldali algoritmus:</u>
A beérkezett T'(x) polinomot osszuk el a generátor-polinommal:

$$\frac{T'(x)}{P(x)}=\frac{M(x)*x^{n}+R(x)}{P(x)}=\frac{M(x)*x^{n}}{P(x)}+\frac{R(x)}{P(x)}=Q(x)+\frac{R(x)}{P(x)}+\frac{R'(x)}{P(x)}=Q(x)+\frac{R(x)+R'(x)}{P(x)}$$

Ahol: *R(x)* az adó oldalon, *R´(x)* a vevő oldalon számított maradék polinom.
Végül, a Mod(2) összeadás szabályai szerint, ha *R=R´*, az
adó és a vevő oldalon képzett maradékok egyezőek, akkor:

$$\frac{R(x)+R'(x)}{P(x)}=0$$

/ két azonos polinom mod(2) összege zérus, azaz a számláló zérus /

Így ezt az eredményt kaptuk, ami azt jelenti: ha az átvitel során nem sérült meg az üzenet (*T'(x)=T(x)*), akkor a teljes algoritmus végén: *<u>zérus maradékot</u>* kapunk.

$$\frac{T'(x)}{P(x)}=\frac{T(x)}{P(x)}=Q(x)$$

Ha az üzenet bármely része (akár az információs rész, akár a maradék rész) megsérül (ami *R≠R´*-ben nyilvánul meg), a vevőoldali algoritmus zérustól különböző maradékot ad eredményül. Azaz a vevőoldali nem fog egyezni az adóoldali maradékkal. Így a mod(2) összegük sem lesz zérus. Ezzel jelzi, hogy az átvitel során hiba keletkezett.

4. *Példa CRC algoritmus végrehajtására:*
A könnyebb érthetőség kedvéért vegyünk egy rövid információs sorozatot *M(x)*-et amit továbbítani szeretnénk majdan a T(x) üzenetbe beépítve.
Legyen az adat: 1101, ennek megfelelően a polinom alak:

$M(x)=1*x^3+1*x^2+0*x^1+1*x^0 = x^3+x^2+1$, azaz : 1101

Legyen a generátor-polinom $P(x)= x^3+x+1$, a bináris képe: 1011

A megoldás lépései:

– A generátor polinom fokszáma n=3, így az M(x) polinomot szorozni kell x^3-onnal, ami három helyértékkel való balra shiftelést jelent, a kisebb helyértékeken ezek helyébe nullák lépnek.

$$M(x) = x^3 + x^2 + 1 \quad \rightarrow \quad 1101$$

$$P(x) = x^3 + x + 1 \quad \rightarrow \quad 1011 \quad \rightarrow x^n = x^3 \quad \rightarrow n = 3$$

$$x^n * M(x) = x^3 * (x^3 + x^2 + 1) = 1000 * 1101 = 1101000$$

– Az így kapott értéket el kell osztani a generátor polinommal, hogy megkapjuk az R(x) maradék polinomot.

$$\frac{x^n * M(x)}{P(x)} = Q(x) + \frac{R(x)}{P(x)}$$

$$\cdot\cdot \quad 1101000 : 1011 = 1111 + \frac{R(x)}{P(x)} = \underbrace{1111}_{Q(x)} + \frac{1}{x^3 + x + 1}$$

$$\begin{array}{l}
\oplus 1011 \downarrow \\
\hline
\cdot 01100 \\
\oplus 1011 \downarrow \\
\hline
\cdot\cdot 01110 \\
\cdot\cdot \oplus 1011 \downarrow \\
\hline
\cdot\cdot\cdot 01010 \\
\cdot\cdot\cdot \oplus 1011 \downarrow \\
\hline
\cdot\cdot\cdot\cdot 01010 \\
\cdot\cdot\cdot\cdot \oplus 1011 \\
\hline
R(x) : 0001
\end{array}$$

Megjegyzés: a számolásnál látható pontocskák, csupán a helyi értékek megfelelő egymás alá való pozícionálását szolgálja (a képletszerkesztő hibáját korrigálja). A karikába írt pluszjel a mod(2) összeadást szimbolizálja, a lefelé mutató nyíl pedig az osztandó polinom megfelelő helyértékű bitjének a lehozatalát jelenti.

Figyelem: a harmadik és negyedik osztási ciklusban az is látható, hogy az 1010:1011 esetén az osztandó számértéke kisebb, mint az osztóé (1010<1011), mégis megvan benne egyszer, hiszen a visszaszorzás és a mod2 kivonással még maradékot is kapunk. Egyébként minden olyan esetben egy az osztás eredménye, ha az előző visszaszorzás maradéka kiegészítve a lehozatallal, eggyel kezdődik. Ha nullával kezdődne, akkor nullaszor volna meg benne.

– Tehát R(x) maradé-kpolinom bitnáriskép: 001. Az üzenet polinomot T(x)-et úgy kapjuk, hogy az n-el eltolt információs bitek végére, a toldalék nullák helyére beírjuk a maradék-polinom bitjeit, azaz az utolsó n=3 biten mod(2) összeadást végzünk:

$$T(x) = x^n * M(x) + R(x)$$
$$T(x) = 1101000 + 001 = 1101001$$

– Ez kerül a csatornára és megérkezik, mint T'(x).
A vétel helyén, most tételezzük fel, hogy- az üzenet hibátlanul megérkezik
Végezzük el a vételoldali algoritmust.

– Mint látjuk, a maradék zérus. Tehát a vétel hibátlan.

$$\begin{array}{l}
\cdots 1101001 : 1011 = 1111 \\
\underline{\;1011\downarrow\;} \\
\cdot 01100 \\
\underline{\oplus 1011\downarrow} \\
\cdot 01110 \\
\underline{\cdot\oplus 1011\downarrow} \\
\cdots 01011 \\
\underline{\cdot\cdot\oplus 1011} \\
\cdots\cdot\cdot 0000 \\
R(x) = 0
\end{array}$$

Tegyük fel, hogy az üzenet a csatornán megsérül. Ekkor T'(x)≠T(x), a vételi algoritmustól most azt várjuk, hogy R(x) ≠0-t hozzon ki.
Legyen a vett üzenetben egy hiba, azaz a második legnagyobb helyértéken 1-es helyett 0.

$$T(x) = 1101001$$
$$T'(x) = 1001001$$
$$P(x) = 1011$$
$$\frac{T'(x)}{P(x)} = ?$$

$$
\begin{array}{l}
1001001 : 1011 = 1010 \\
\underline{1011\downarrow} \\
\;00100 \\
\underline{\oplus\, 0000\downarrow} \\
\quad 01000 \\
\underline{\quad \oplus 1011\downarrow} \\
\quad\; 00111 \\
\underline{\quad\; \oplus\, 0000} \\
\qquad 0111 \\
R(x) = 111 \neq 0
\end{array}
$$

Valóban, most a maradék-polinom nem zérus.

Legyen most két bit hiba az átvitel után. Legyenek a legnagyobb helyérték felől a másodi és a harmadik bitek hibásak.
Ekkor T’’(x)=1011001. Végezzük el a vevő oldali műveletet:

$$
\begin{array}{l}
T(x) = 1101001 \\
T''(x) = 1011001 \\
1011001 : 1011 = 1000 \\
\underline{1011\downarrow} \\
\;00000 \\
\underline{\oplus\, 0000\downarrow} \\
\quad 00000 \\
\underline{\quad \oplus\, 0000\downarrow} \\
\quad\; 00001 \\
\underline{\quad\; \oplus\, 0000} \\
\qquad 0001 \\
R(x) = 001 \neq 0
\end{array}
$$

Tehát a 2 bit hibát, mint hibacsomót is képes a kód detektálni, hiszen R(x) ≠0-t kaptunk.

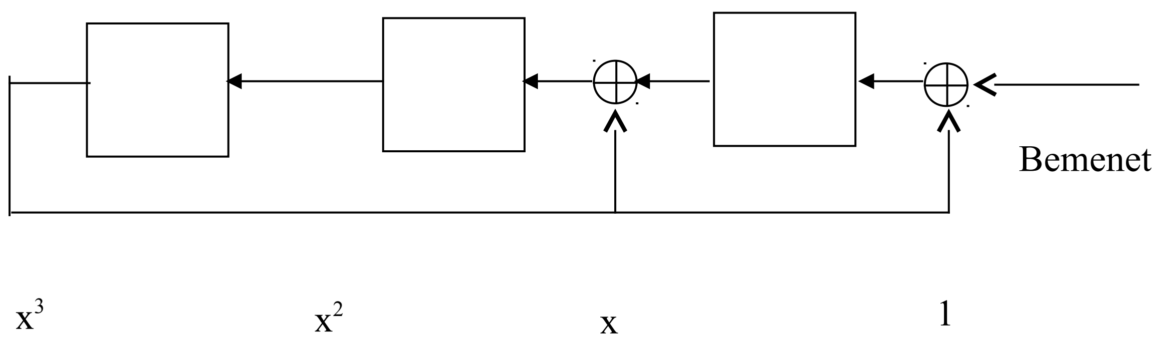
Összefoglalva, az algoritmust lépésekre bontva:

- Az elküldendő információs biteket /M(x) -et /, a generátor polinom /P(x)/ fokszámával (n-el) egyező bitnyi eltolást végzünk balra.

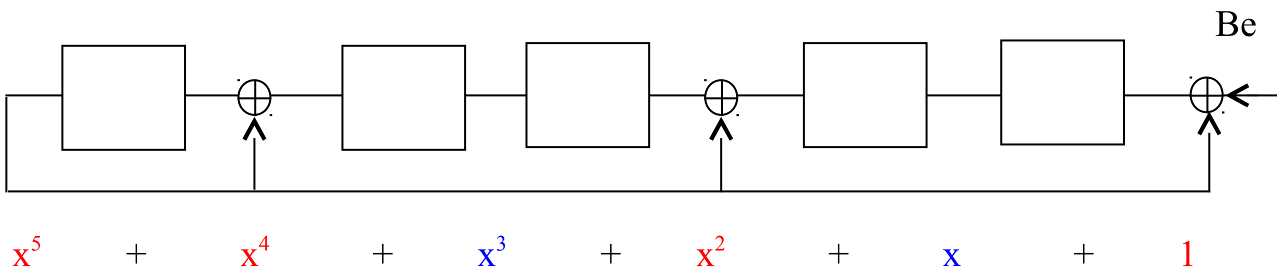
- Az így eltolt polinomot elosztjuk a generátor-polinommal, eredményül kapjuk egész-polinom részként Q(x)-et, maradék- polinomként pedig R(x)-et.
- A kapott R(x) bitjeit beillesztjük, az eltol információs polinom jobboldalán lévő toldalék zérus bitek helyébe. Így kaptuk meg a T(x) üzenet-polinomot, amit a csatornán továbbítunk.
- A vétel helyén a kapott T'(x)-et elosztjuk P(x)-el, a generátor-polinommal.
- Megvizsgáljuk a maradék-polinom bináris értékét, ha R(x) ≠0-t kaptunk, a közlés során hiba lépett fel. Ismétlés szükséges.

5. *Áramköri realizáció:*

Az áramköri megvalósítás léptetőregiszterrel történik. Ennek tartalma induláskor nulla. A regiszter hossza a generátor-polinom fokszámával azonos. Az információ a legkisebb helyértéken lép be. A legnagyobb helyértékről egy-egy EX-OR kapuval visszacsatoljuk minden olyan helyértékhez, amely a generátor polinomban egyesként szerepel.
Legyen az előzőekben használt generátor-polinom $P(x)= x^3+x+1$, a bináris képe: 1011. Realizálás:

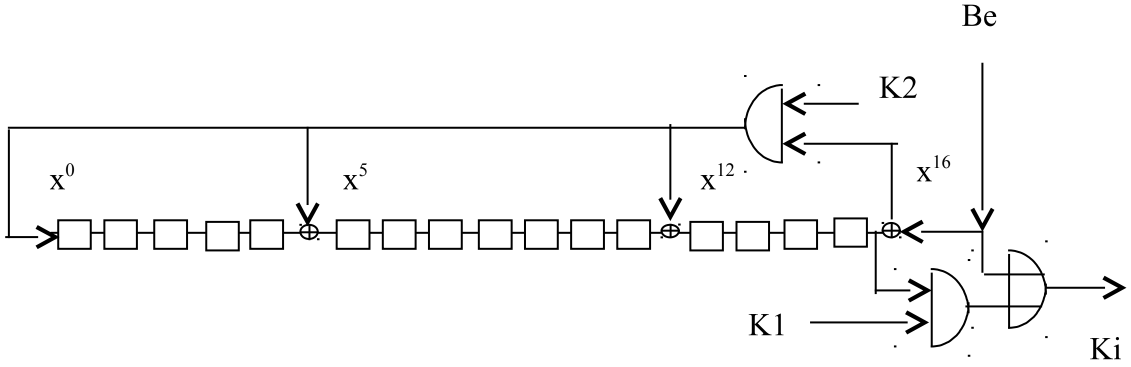


Az utolsó adatbit beléptetése után a regiszter tartalmazni fogja a maradék polinomot (annak együtthatóit).
Most egy másik példaként legyen $P(x)=x^5+x^4+x^2+1=110101$



A valóságban ezek lényegesen hosszabb regiszterek. A generátor-polinomok a CCITT ajánlása szerintiek, melyek hossza 16 vagy 32 bit szokott lenni. Rendszerint ugyan az a regiszter működik adás és vétel esetén, mind a kódolást, mind a dekódolást ugyanaz a regiszter végzi.

A CCITT V.41-es ajánláson alapuló adó-vevő kapcsolás.
$P(x)=x^{16}+x^{12}+x^5+1$



Adáskor K2 elejétől bekapcsolt állapotban van, amíg az üzenet maradék-polinom része nem kerül adásra. Ez idő alatt K1 kikapcsolt. Mikor a maradék-rész kiléptetése következik (ekkor a bemeneten tizenhat 0 sorakozik be) K1 bekapcsol, K2 kikapcsol. A maradék polinom együtthatói ekkor kerülnek a kimenetre.
A léptetőregiszter kezdeti nullázását, az utolsó 16 nulla bit beléptetése alatt, a K2 kikapcsolt állapota biztosítja.

Vételkor ugyanez történik, csak a maradék résznél (az utolsó 16 bitnél) most nem feltétlenül nullák jönnek. A K2 végig nyitva a K1 csak az utolsó 16 bit kiléptetése esetén van nyitva. Ha ez a kilépő 16-os sorozat nem zérus, akkor hiba lépett fel.

6. Felhasználási területek:

Ez a hibadetektálási módszer rendkívüli jó tulajdonságokkal rendelkezik. Általában soros adattovábbításnál alkalmazzák. Ilyen lehet a telex vonal, a soros MODEM, de ilyen CRC kód jelenik meg a mágneslemez, winchester track-en, a szektorokba írt adatblokkok végén. Régebben a mágnesszalagos technikában is használták, a hosszanti paritás után következtek a CRC bájtok, a szalag mindegyik sávjára. Egy 32 bites CRC kód, a hosszú adatblokkokban (pl.: 4k-s) p=99.9998 % valószínűséggel mutatja ki a hibát, de még a 2-3 bites hibacsomókat is.

Javasolt irodalom:

[1.] G. Birkhoff, T.C.Bartee:A modern algebra a számítógép tudományban (Műszaki Könyvkiadó, Budapest, 1974)
[2.] Hardy Zs, dr. Sólyom M: Út a modern algebrához (Tankönyv Kiadó Budapest 1975)
[3.] Sz.V. Jablonszkij és O.B.Lupanov: Diszkrét matematika a számítástudományban. (Műszaki Könyvkiadó, Budapest, 1980)
[4.] Lucky. Salz. Weldon: Adatátvitel (MK Budapest, 1973)
[5.] Dr Varga A: Adatátvitel (BME jegyzet:J5-934) (Tankönyv Kiadó Budapest 1975)
[6.] Tóth M. Janovics S.: Digitális rendszertechnika (BME jegyzet:J5-673) (Tankönyv Kiadó Budapest 1973)
[7.] J. Wakerly: Hibajavító kódok Önellenörző áramkörök (Műszaki Könyvkiadó, Budapest, 1984)
[8.] Jákó P. Digitális hangtechnika (Budapest, 2000)