Počítačové siete
Kontrola chýb – CRC

Cyclic Redundancy Code

- *m*-bitovú správu zakódujeme ako polynóm M(x) stupňa (nanajvýš) *m*-1 nad (Z_2 , +, .)
  - operácia + je XOR, -*a* = *a*, operácie + a – sú rovnaké
- jednotlivé bity správy budú koeficientami
  - napr.: 101101 -> $x^5 + x^3 + x^2 + x^0$
- majme *generujúci polynóm* G(x) stupňa *r*
- nájdime P(x) a R(x) také, že
  - $x^r M(x) = P(x)G(x)+R(x)$, stupeň(R(x)) < *r*
    - R(x) je zvyšok po delení $x^r M(x)$ polynómom G(x)

Cyclic Redundancy Code

- nech $T(x) = x^r M(x) + R(x)$
  - $T(x) = P(x)G(x)$
- odvysielame koeficienty T(x)
- príjmeme koeficienty $T'(x) = T(x) + E(x)$
  - E(x) je polynóm reprezentujúci chybu
- vydelíme $T'(x)/G(x)$: $T'(x) = P'(x)G(x) + R'(x)$
  - ak $R'(x) = 0$, považujeme prenos za bezchybný
    - bity správy sú koeficienty $T'(x)$ pri x^{m+r-1} až x^r
  - ak $R'(x) \neq 0$, došlo pri prenose k chybe

Cyclic Redundancy Code

- chyba bude odhalená práve vtedy, keď E(x) nie je deliteľný G(x)
- vhodnou konštrukciou G(x) vieme zabezpečiť
  - odhalenie všetkých jednobitových chýb (x^i)
    - stačí, aby G(x) mal aspoň 2 členy
  - odhalenie všetkých dvojbitových chýb
  - odhalenie všetkých chýb s nepárnym počtom 1
  - odhalenie všetkých chýb v podobe súvislého bloku 1 dĺžky max. *r*

Cyclic Redundancy Code

- dvojbitové chyby
  - $E(x) = x^i + x^j$, $i > j$
  - $E(x) = x^j(x^{i-j}+1)$
  - ak G(x) nie je deliteľný x a (x^k+1) nie je deliteľné G(x) pre žiadne k=1, ..., m+r-1, tak E(x) nie je deliteľné G(x)
  - napr. $x^{15} + x^{14} + 1$ nedelí x^k+1 pre k<32768

Cyclic Redundancy Code

- $x \nmid G(x) \wedge G(x) \nmid A(x) \Rightarrow G(x) \nmid xA(x)$
  - $A(x) = P(x)G(x) + R(x)$
    - $st(R(s)) < st(G(x)), R(x) \neq 0$
  - $xA(x) = xP(x)G(x) + xR(x) = P'(x)G(x) + R'(x)$
    - $st(R'(x)) < st(G(x))$
  - ak $st(R(x)) < st(G(x)) - 1$
    - $st(xR(x)) < st(G(x)), R'(x) = xR(x) \neq 0, G(x) \nmid xA(x)$
  - ak $st(R(x)) = st(G(x)) - 1$
    - $st(xR(x)) = st(G(x)), R'(x) = xR(x) - G(x) \neq 0, G(x) \nmid xA(x)$
- $G(x) \nmid x^k A(x)$ indukciou

Cyclic Redundancy Code

- chyby s nepárnym počtom 1
  - žiadny polynóm s nepárnym počtom jednotkových koeficientov nie je deliteľný (x+1)
    - E(1) = 1
    - nech E(x) = P(x)(x+1)
    - E(1) = P(1)(1+1) = 0, spor.
  - ak G(x) je deliteľné (x+1), tak E(x) nie je deliteľné G(x)

Cyclic Redundancy Code

- súvislý blok 1
  - $E(x) = x^i(x^{k-1} + x^{k-2} + ... + 1), k \leq r$
  - ak G(x) nie je deliteľné x, tak E(x) nemôže byť deliteľné G(x)

Cyclic Redundancy Code

- príklad:
  - G = 10011 $G(x) = x^4 + x + 1$
  - M = 11011001 $M(x) = x^7 + x^6 + x^4 + x^3 + 1$
    - $(x^{11} + x^{10} + x^8 + x^7 + x^4) : (x^4 + x + 1) = x^7+x^6+x^3+x^2$

      $x^{11} + x^8 + x^7$

      $x^{10} + x^4$

      $x^{10} + x^7+x^6$

      $x^7+x^6+x^4$

      $x^7 + x^4+x^3$

      $x^6 + x^3$

      $x^6 + x^3+x^2$

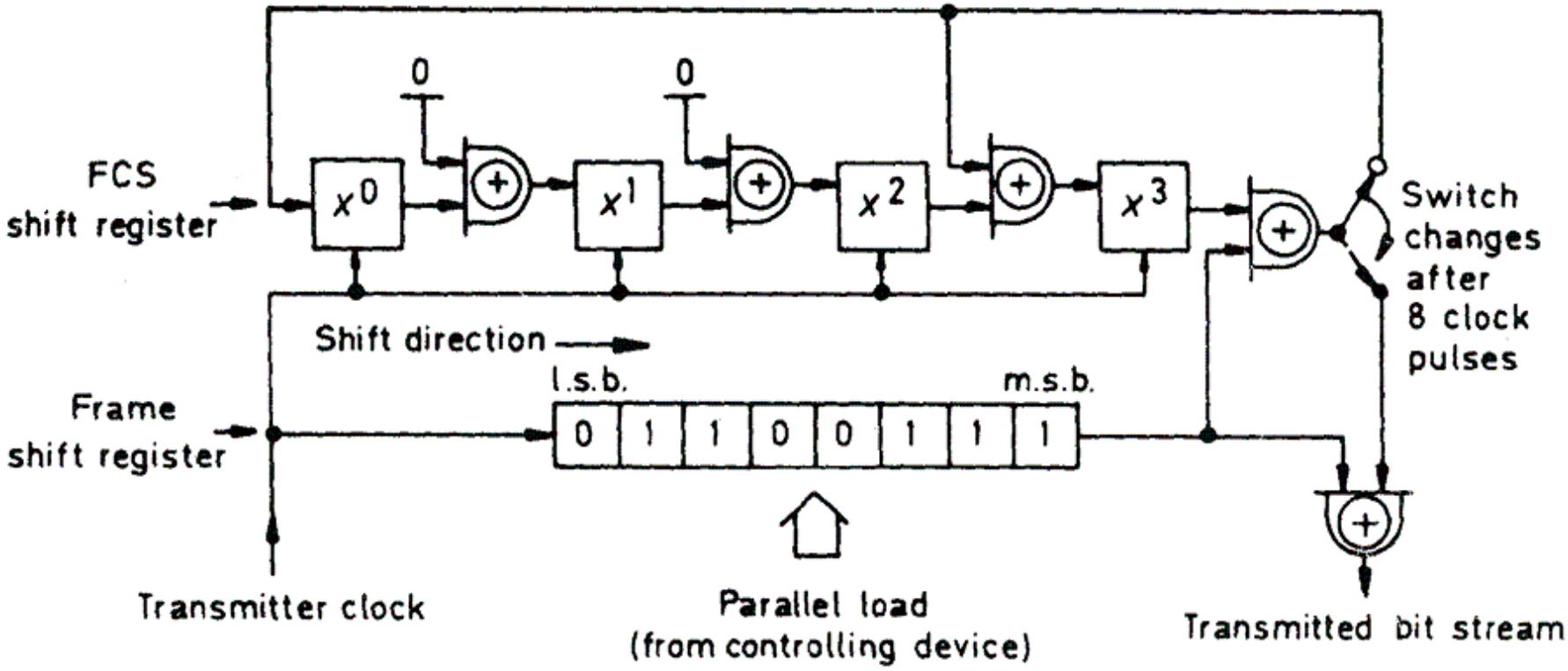
      x^2
    - $R(x) = x^2$, $T(x) = x^{11} + x^{10} + x^8 + x^7 + x^4 + x^2$
    - T = 11011001**0100**

Cyclic Redundancy Code

- príklad
  - G = 10011
  - M = 11011001
  - 11011 0010000 : 10011 = 11001100
    10011
    01000 0010000
     10011
     00011 0010000
      00110 10000
       01101 0000
        11010 000
         10011
         01001 000
          10011
          00010 0
           00100
           00100 T = 11011 00 10100

Cyclic Redundancy Code

Generator: $x^4 + x^3 + 1$
Active bits: x^3, x^0
Frame contents: 11100110



Cyclic Redundancy Code

- CRC-16-CCIT
  - x16+x12+x5+1
  - 1000100000100001
  - 0x1021
- CRC-32
  - $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^{8}+x^{7}+x^{5}+x^{4}+x^{2}+x^{1}+1$
  - 10000010011000001000111011011011 1
  - 0x04C11DB7

Internet Checksum

- používaný v IP, TCP, …
- z jednotlivých oktetov sa vytvoria 16-bitové slová (ak je nepárny počet oktetov, doplní sa 0 na konci)
- kontrolný súčet je 16-bitový jednotkový doplnok k sume v jednotkovom doplnkovom kóde 16-bitových slov
  - pri sčitovaní sa vždy ešte pripočíta carry bit
  - jednotkový doplnok je bitová negácia
- kontrola
  - spočíta sa kontrolný súčet rovnakým spôsobom (vrátane políčka s kontrolným súčtom)
  - výsledok musí byť -0 = 0xFFFF

Internet Checksum

- sčitovanie v jednotkovom doplnkovom kóde
  - je nezávislé na poradí byte-ov
    - teda netreba riešiť rozdielne podľa LSB/MSB-first architektúry (okrem zarovnania na konci)
  - dá sa rozdeliť na časti
    - [A,B] +' [C,D] +' [E,F] = [A,B] +' [C,0] +' [0,D] +' [E,F]
  - dá sa paralelizovať
    - napr. počítaním 32-bitovej sumy a potom spočítanie hornej a dolnej časti (plus carry)
  - dá sa odložiť pripočítavanie carry bitov
    - napr. spočítavaním 16-bit slov do 32-bit sumy a následne spočítanie hornej a dolnej časti (plus carry)