Codificator si decodificator de mesaje folosind paritate

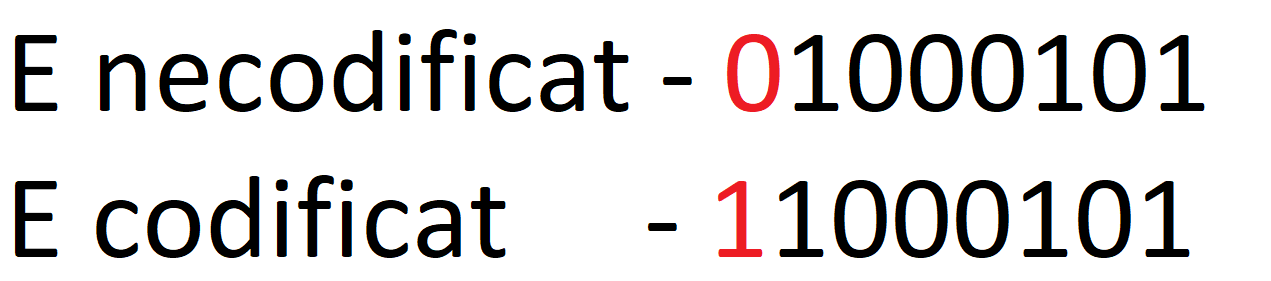
Macarie Razvan Cristian, 322CB

Se face precizarea ca trimitem mesaje text, ASCII pe 7 biti (toate cifrele,

literele si simbolurile uzuale). Se transforma un octet in decimal si din decimal

in array de boolene si se stocheaza in al 8-lea bit un bit de control, ce reprezinta

rezultatul operatiei XOR aplicata cumulativ peste cei 7 biti ai caracterului.



Decodificarea se face setand bitul de semn 0, iar verificarea de erori se

face facand XOR peste toti cei 8 biti din mesaj. Daca rezultatul operatiei XOR

pe toti bitii este diferit de 0 inseamna ca mesajul este corupt. Daca facem

presupunerea ca mesajul a fost codificat corect, atunci singura posibilitate de

coruptie apare in transmisie.

Decodificarea si codificarea se fac "in place". Se modifica intr-un array

initial valoarea deja existenta.

Aceasta codificare ne garanteaza ca trebuie sa avem un numar par de biti de 1 intr-un octet, astfel daca un bit face "flip" in timpul transimisei schema

de codificare o sa prind eroarea. Singura problema apare atunci cand se schimba

2 biti, verificarea nu o sa prinda eroarea insa deobicei in transmisie se schimba

un singur bit (si asta destul de rar).

Adica ar fi bine sa folosim aceasta metoda de detectare de erori cand

informatiile primite sunt de la o sursa conectata local (nu ar fi fezabil pentru

o retea mai mare cum ar fi internetul) care trimite deobicei informatii la

frecventa relativ mica.

87.5% din informatia totala este utila, restul fiind folosita pentru codificare.

Astfel schema de codificare este relativ eficienta si pentru mesaje de lungime

mica cat si pentru mesaje de lungime mare.

Pentru a realiza proiectul am creat 3 instrumente virtuale:

- Codificator : seteaza in bit-ul de semn rezultatul operatiei xor aplicat

pe toti ceilalti biti.

- Verificator : se verifica daca rezultatul operatiei xor pe toti bitii

dintr-un octet este 0.

- Decodificator : se seteaza bitul de semn 0 la fiecare octet din mesaj.

Informatia este transmisa printr-un text-box si afisata tot intr-un textbox.

Daca verificatorul detecteaza o eroare, va trimite un semnal boolean True la un

bec led.

**Despre proiect:**

Nu am folosit din librariile deja existente lucruri deja implementate. Am

folosit doar while si for loops cu shift registers, porti logice si constante true si false. Am folosit in exemple, totusi, din libraria TCP a labview dar nu sunt in proiectul final.

Intrumentul final este facut din sub-instrumente, fiecare are doar un singur

input si output. Doua dintre ele au string -> string si unul are strin -> bool.

Daca mesajul este detectat ca fiind corupt se opreste executia decodificatorului

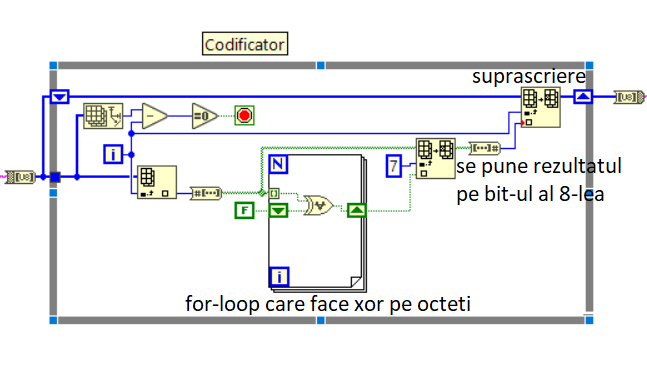
si se scrie garbage (nu stiu daca a fost o idee buna asta, ca macar erau

caractere ASCII printabile).

Codificatorul ia un string, il imparte in octeti, imparte octetii in array de

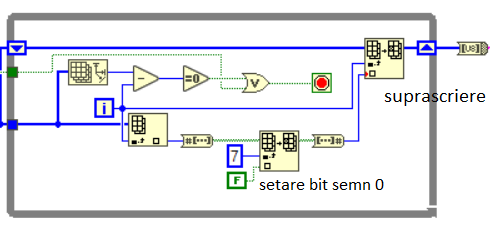
boolene, calculeaza XOR-ul, seteaza bitul de semn, suprascrie fosta valoare si

intoarce string.



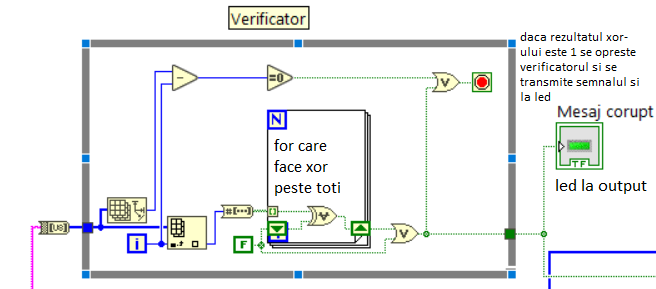
Decodificatorul ia un string, il imparte in octeti, imparte octetii in array de

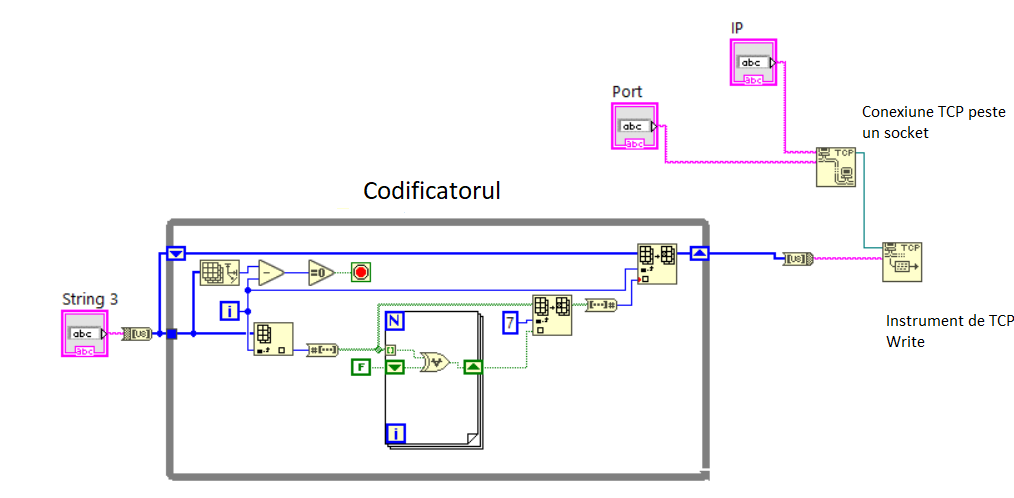
boolene, seteaza 0 in bitul de semn, suprascrie fosta valoare si intoarce string.

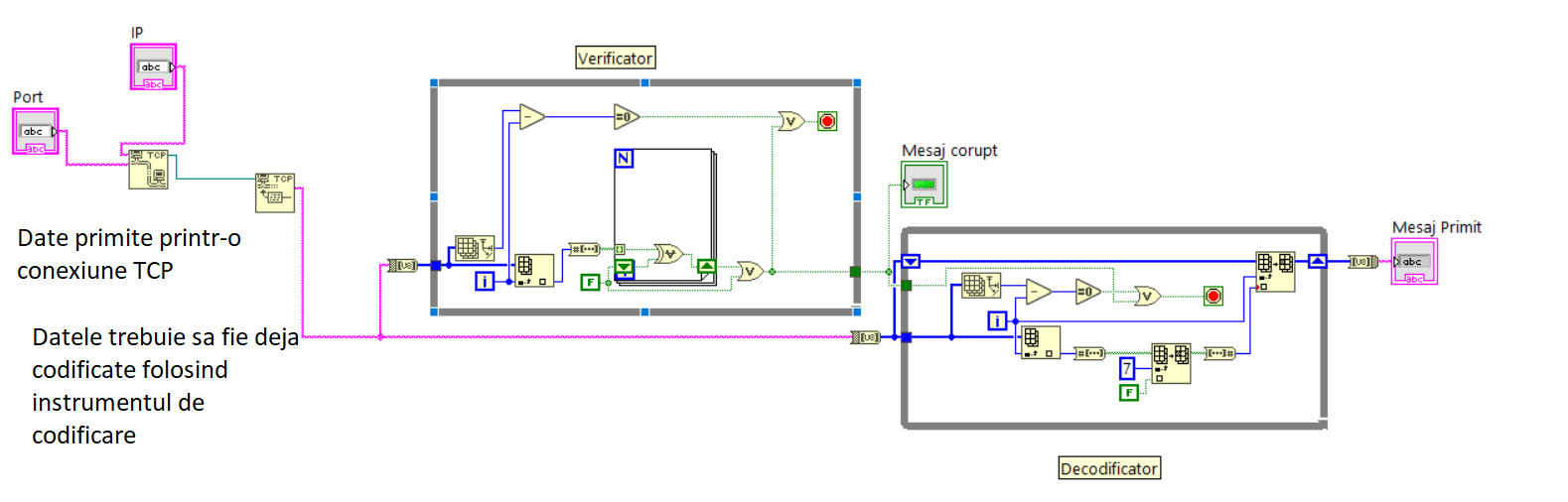


Verificatorul ia un string, il imparte in octeti, imparte octetii in array de

boolene, calculeaza XOR-ul pe toti si daca acesta nu este 0 scoate un semnal

boolean de true, in rest e tot timpul 0.

Putem transmite date intre sursele noastre folosind API-ul de Berkeley Sockets.



Daca datele nu sunt codificate de noi in prealabil cel mai probabil vor fi detectate ca fiind corupte.

**Input:**

* String in format ASCII pe 7 biti (asemanator protocolului de E-mail).
* Numerele sunt codificate tot ca stringuri (de exemplu numarul 100 este codificat pe 3 octeti (“1”, “0”, “0”).

**Output:**

* String ASCII pe 7 biti.

Am pornit initial de la idea ca nu o sa copiez ca toata lumea ceasul cu 7

segmente si ca o sa ma descurc cu o idee pe care chiar vreau sa o fac. Mi s-a

parut interesanta idea codurilor Hamming (despre care am invatat la Protocoale)

si a detectiei de erori in general si am vrut sa fac initial codificare si

decodificare cu coduri Hamming. M-am chinuit ceva timp dar nu am reusit sa-l fac

sa mearga cum trebuie, asa ca decat sa nu trimit nimic am incercat sa fac ceva

mai usor. Spre surprinderea mea chiar functioneaza, si se poate verifica

trimitand dupa la verificator un numar cum ar fi 0b11001101. Mi-am dat seama in

timp ce scriam aceste randuri ca schema asta de codificare este cam redundanta

pe un singur octet, pentru ca se poate seta bitul de semn 1 pentru caractere ASCII

ce au reprezentarea in decimal impara si se ajunge la acelasi efect.