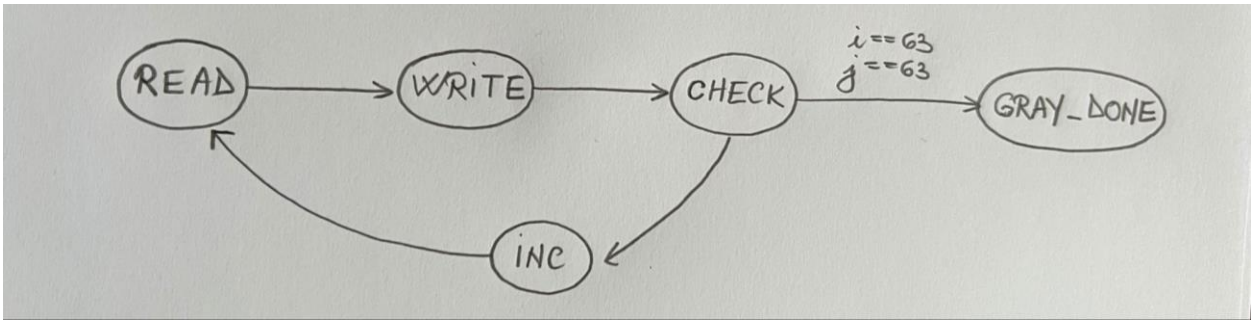
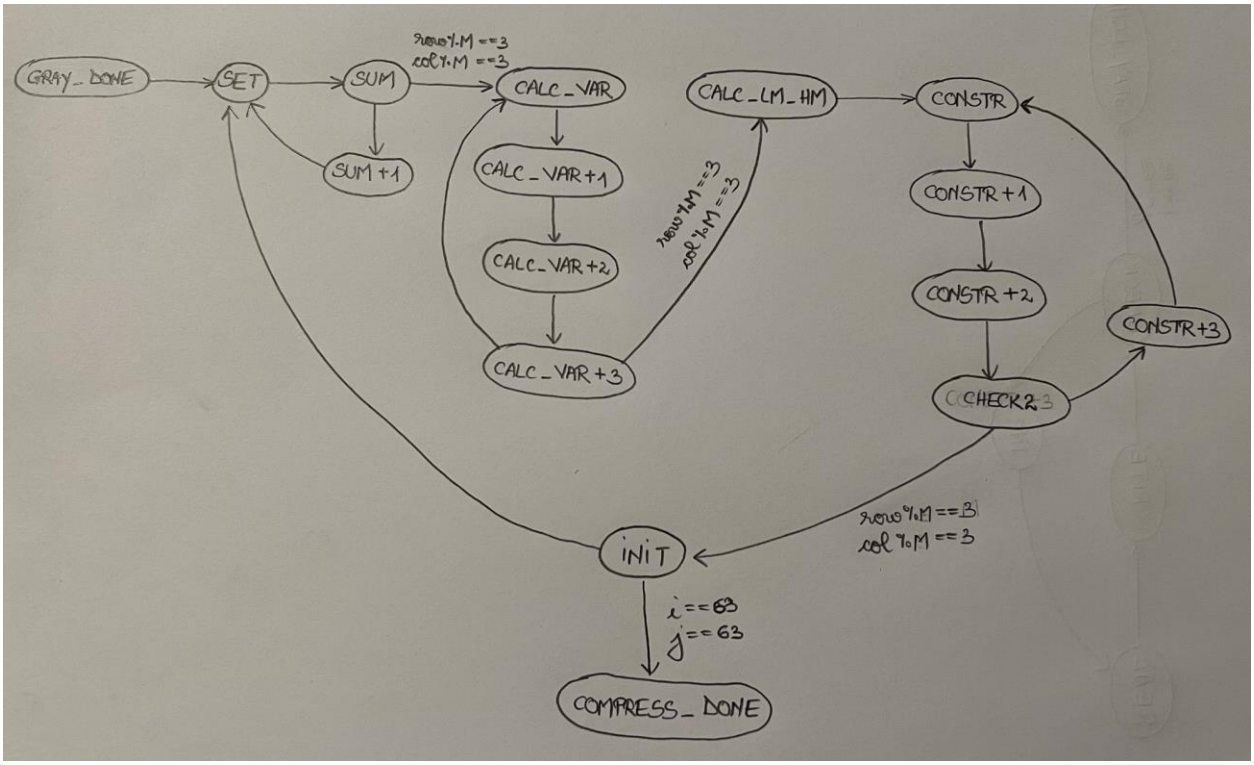
Boboc Talida Ștefania
Grupa 332AB, subgrupa 1

Pentru a rezolva tema, am urmarit sa tratez separat cele 3 etape de prelucrare a imaginii. Astfel, pentru prima etapa de conversie a imaginii din RGB in grayscale m-am gandit la urmatorul automat:



In starea **READ**, am setat linia si coloana (cu i si j) si am aflat maximul si minimul din cele 3 canale ale lui in_pix. In starea **WRITE** am calculat media celor 2 valori aflate anterior si am setat-o ca valoarea din canalul `G`, iar canalele `R` si `B` le-am setat cu 0. In starea **CHECK** verific daca s-a ajuns la ultimul pixel prin i == 63 si j == 63, daca da, atunci trec in starea **GRAY_DONE** care marcheaza sfarsitul procesului grayscale. Daca nu s-a ajuns la ultimul pixel, atunci trec in starea **INC** unde incrementez i si j. Mai intai incrementez j, pentru a parcurge matricea pe linii, cand j ajunge la 64, se va trece la linia urmatoare (adica i = i + 1) si j va deveni 0 pentru a incepe parcurgerea de la inceputul liniei. Din aceasta stare, ma intorc in **READ**.

Pentru a rezolva partea a2-a m-am gandit la urmatorul automat:



In starea **SET**, le-am dat lui row si col valorile lui i si j. In starea **SUM** adaug la suma in_pix curent. Daca s-a ajuns cu row si col la sfarsitul blocului 4x4, calculez media si trec in starea **CALC_VAR**, altfel trec

in starea **SUM+1** unde incrementez i si j. In starea **CALC_VAR** calculez deviatia standard si beta, aici m-am folosit de 4 stari pentru a putea parcurge din nou blocul 4x4. Daca se ajunge la conditia de sfarsit de bloc, trec in starea **CALC_LM_HM**, altfel ma intorc in **CALC_VAR**. In **CALC_LM_HM** calculez valorile pentru Lm si Hm, dupa care trec in starea **CONSTR**, unde voi folosi alte 4 stari pentru a scrie in out_pix Lm sau Hm in functie de medie. Aici am folosit o stare de verificare **CHECK2**. Daca nu este indeplinita conditia de sfarsit de bloc 4x4 ma duc **CONSTR+3** si ma intorc in **CONSTR**. Daca este indeplinita conditia merg in starea **INIT** in care trec la blocul urmator si reinitializez suma, suma_var, beta si avg cu 0 si ma intorc in **SET**. Daca cumva am ajuns la sfarsitul imaginii, atunci ma duc in starea **COMPRESS_DONE** care marcheaza sfarsitul procesului de compresie. Am folosit cate 4 stari pentru **CALC_VAR** si **CONSTR**: una pentru marcarea inceputului blocului, una pentru setarea lui row si col, una pentru incrementare si alta pentru calculul efectiv.

Pentru a realiza partea de incapsulare a mesajului secret am facut urmatorii pasi: in starea **INIT2** m-am pozitionat la inceputul imaginii, apoi am trecut in starea **PRIMAPOZ** unde salvez valoarea pixelului de pe prima pozitie din blocul curent totodata cu indicii acesteia (ma folosesc de i1 si j1), dupa care trec in starea **ADOUAPOZ** unde salvez urmatoarea valoare diferita de cea salvata anterior totodata cu indicii acesteia (ma folosesc de i2 si j2). Tratez separat cazul in care nu exista decat o singura valoare in blocul curent cu ajutorul unui contor nr. Daca nr ajunge la 16 atunci i2 va fi i1 si j2 va fi j1+1. Dupa ce am gasit aceste 2 pozitii merg in starea **CONV** unde se realizeaza conversia din baza 2 in baza 3 a cate 16 pixeli din hiding_string. Doar atunci cand done devine 1 mergem in starea **SETRC** unde setez row si col, altfel ma intorc in **CONV**. Din **SETRC** merg in **ENCODE** unde modific pixelii in functie de algoritmul dat. Mai intai verific daca am ajuns la sfarsitul blocului sau la sfarsitul imaginii prin **CHECK3** si apoi ma folosesc de starile **ENCODE+1**, **ENCODE+2**, **ENCODE+3** pentru a parcurge blocul curent. Atunci cand ma aflu la sfarsitul unui bloc, ma intorc in starea **PRIMAPOZ**. In caz ca am ajuns la sfarsitul imaginii, trec in starea **ENCODE_DONE** care marcheaza sfarsitul procesului de codificare.