# ARHITECTURA SISTEMELOR DE CALCUL

-Document explicativ-

Sd. cap. Avram Mara-Luiza

Grupa C112-B

# TO DO 1(funcția SEED):

* Am folosit întreruperea 2CH pentru a stoca ora actuală a sistemului astfel **CH** = **ORA**, **CL** = **MINUTUL**, **DH** = **SECUNDA**, **DL** = **SUTIMILE DE SECUNDĂ**

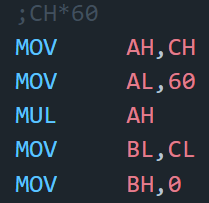
**Formula:**

x0 = ((CH ∗3600 + CL ∗60 + DH) ∗100 + DL) mod 255

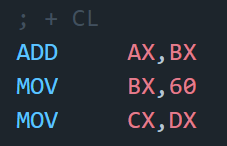
* Pentru ușurința calculului am dat factor comun 60

(CH\*60+CL)\*60

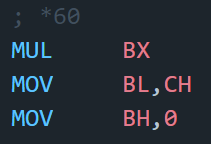
* Am folosit instructiunea MUL pentru a efectua operația 60\*CH



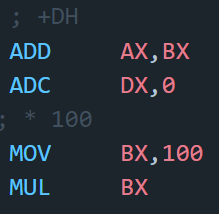
* Am adunat la rezultatul înmulțirii CL



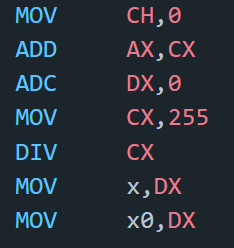
* Am înmulțit cu 60 rezultatul operației anterioare



* Am adunat DH și am înmulțit rezultatul cu 100



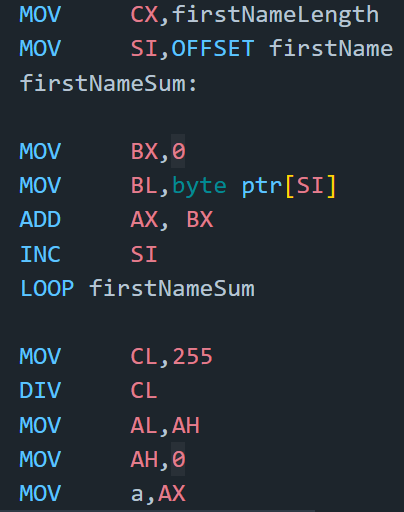
* Am efectuat operația DIV, restul salvându-se în registrul DX și atribuind rezultatul variabilelor x și x0



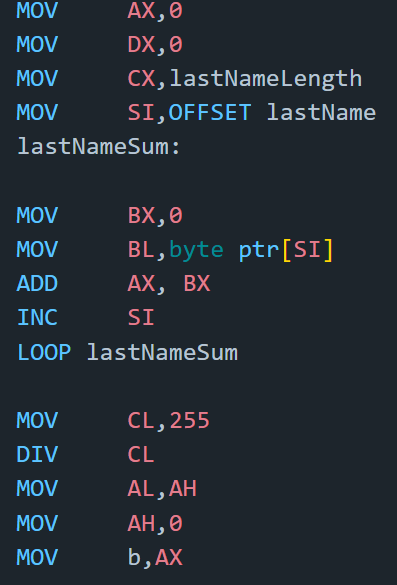
# TO DO 2(funcția RAND):

Această funcție calculeaza variabilele a și b în funcție de string-urile din variabilele firstName și lastName.

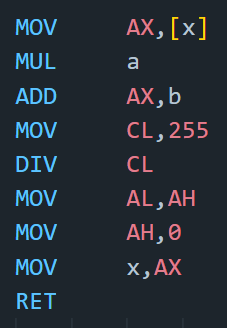
* În **LOOP firstNameSum** se calculeaza suma caracterelor din variabila **firstName**, după care se folosește instrucțiunea **DIV** pentru a calcula **restul împărțirii sumei obținute la 255**, ulterior rezultatul se salvează in variabila **a**



* Folosind aceleași instrucțiuni am calculat suma caracterelor din variabila **lastName**, **în LOOP lastNameSum** și am împarțit rezultatul la 255, în final rezultatul se salveaza în variabila **b**



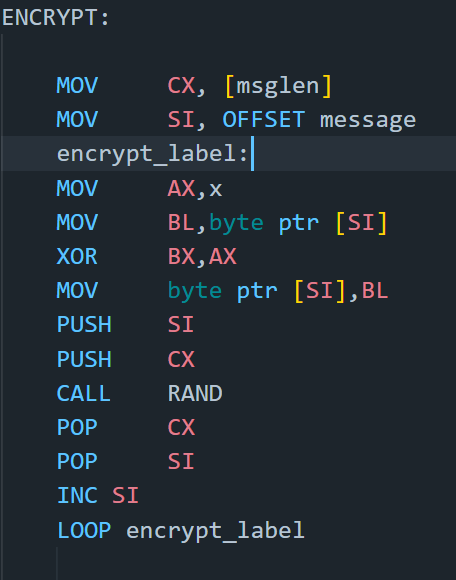
* Folosind formula **xn= (a ∗xn−1 + b) mod 255** am calculat xn în funcție de x, la fiecare apelare a funcției în **x era salvata valoarea lui xn-1**, la final în **x se salveaza valoarea lui xn**



# TO DO 3(funcția ENCRYPT):

Această funcție **codifică mesajul** din variabila **message**, etapele codficării fiind:

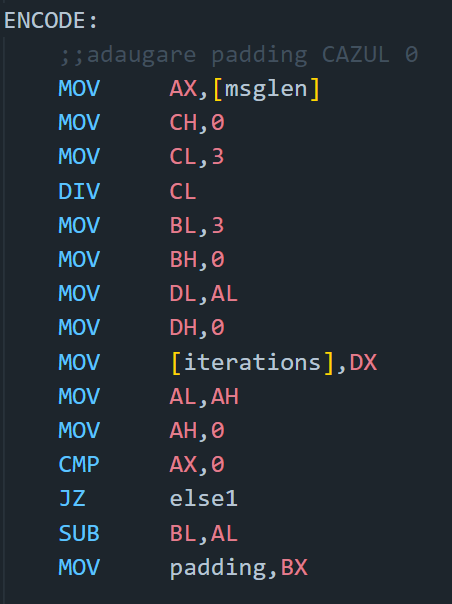
* Se salvează pe rând caracterele din **message** în registrul BL, se aplică intrucțiunea **XOR pe BX ,AX**, în AX fiind salvată valoarea lui x, în funcție de poziția caracterului din message
* Se apelează funcția **RAND** pentru a **genera x-ul** aferent poziției pe care ne aflăm în message



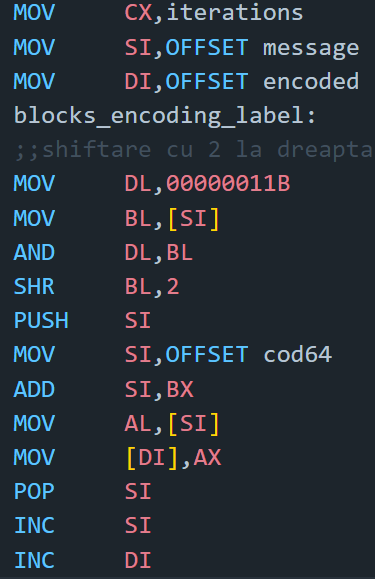
# TO DO 4(funcția ENCODE):

Funcția ENCODE codifică mesajul criptat în COD64, urmărind regulile specificate în cerința temei.

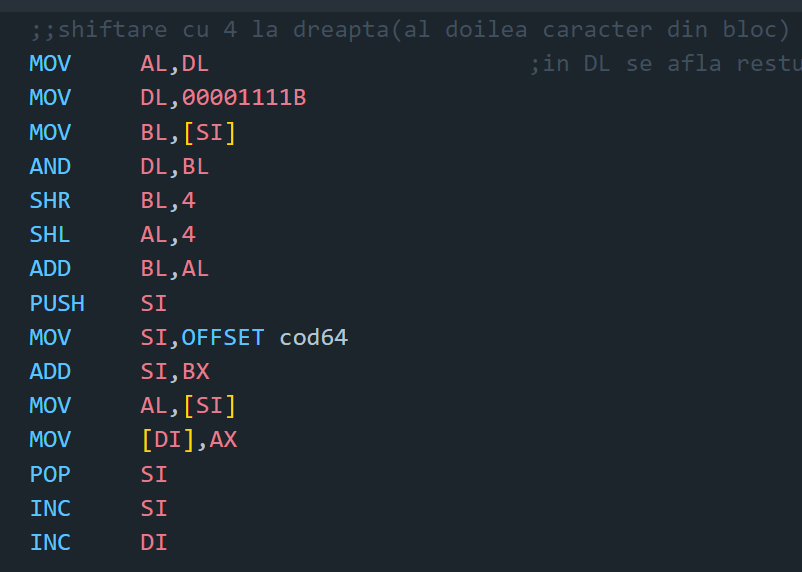
* Se calculeaza restul împărțirii lui msglen la 3, pentru a obtine câte blocuri de 24 de biți conține mesajul și câți biți rămân în plus. Câtul se salveaza în iterations și restul in padding.



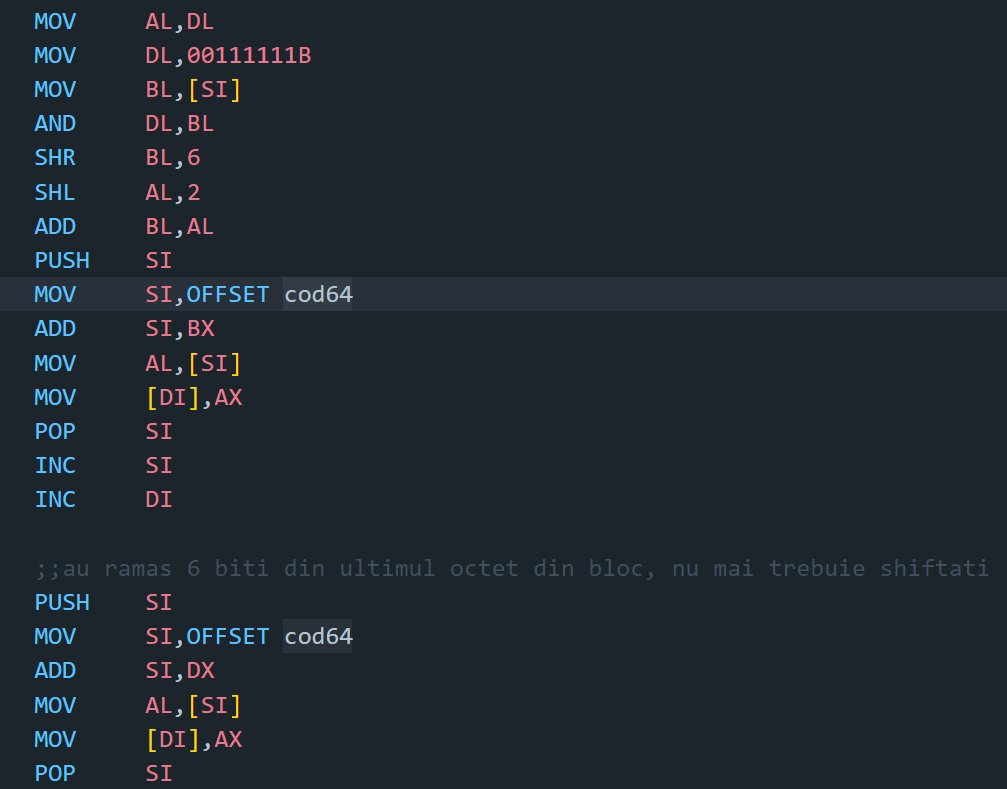
* Se shiftează cu 2 la dreapta primul octet din bloc, cei doi biți shiftați se încarcă in registrul DL, folosind instrucțiunea AND între caracterul din encoded și masca 00000011b, apoi rezultatul shiftării indexează un anumit caracter din COD64, acesta fiind pus pe poziția asociată din encoded.



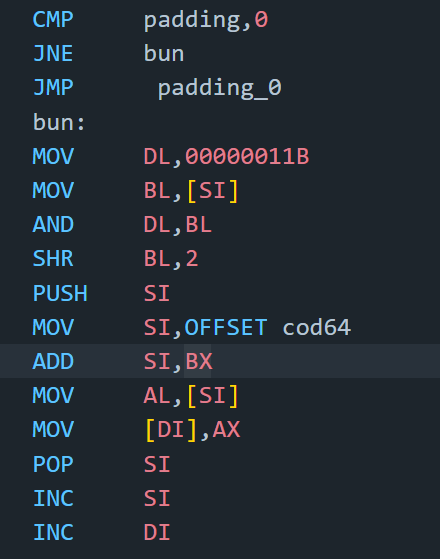
* Se repetă aceeleași operații pentru al doilea octet din bloc, cu mențiunea că se shifteaza 4 biți și se adaugă la cei 4 biți rămași cei 2 biți salvați de la primul octet.

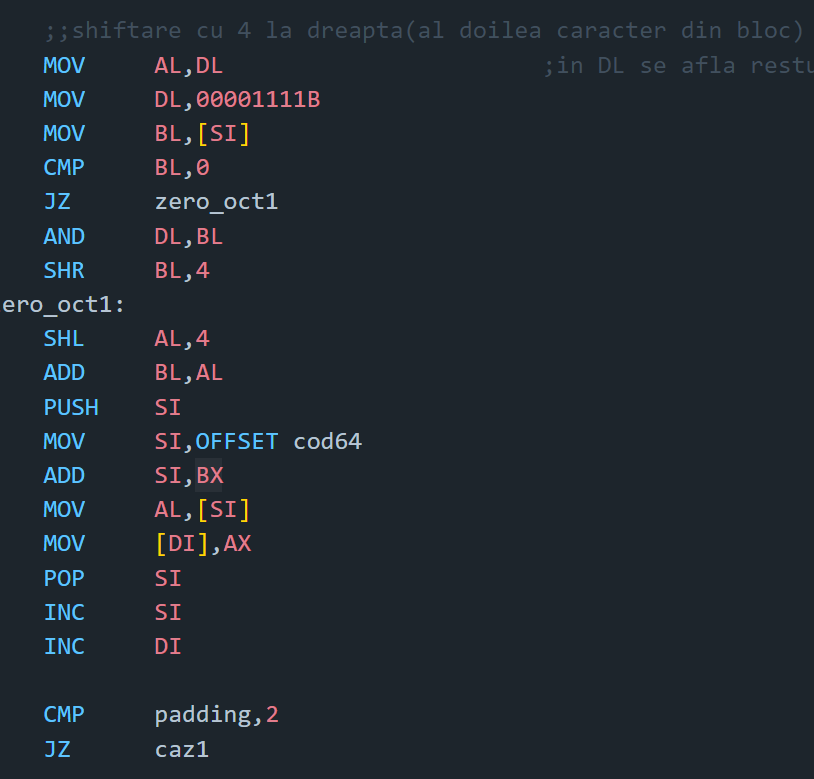


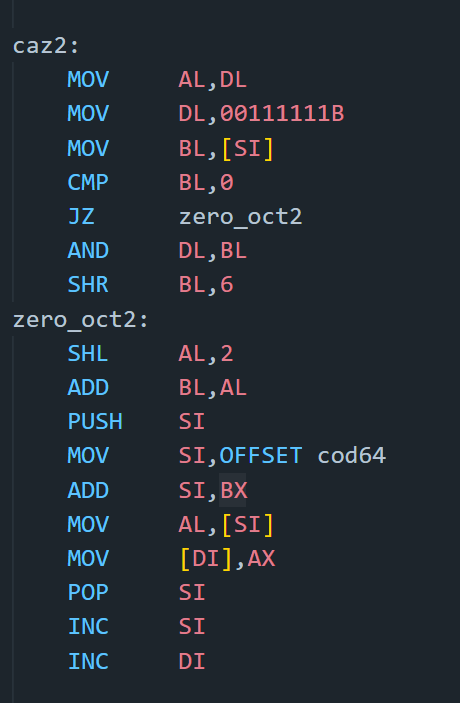
* Se repetă aceleași operații pentru ultimul octet din bloc, se shiftează 6 biți la dreapta, se adaugă în față cei 4 biți rămași de la octetul anterior și se indexează in COD64, se pune pe poziția asociată din encoded. La acest pas, cei 6 biți shiftați din ultimul octet din bloc se indexează in COD64, fară a mai adăuga altceva în față.



* Următoarele operații dupa **ieșirea din LOOP blocks\_encoding\_label**, se fac pentru unul sau doi octeți în funcție de caz (**octeții care nu au putut fi încadrați într-un bloc**).







* Se **adaugă octeții de padding** în funcție de numărul din variabila padding.

