

Wyższa Szkoła Ekonomii i Informatyki w Krakowie

Łukasz Gancarczyk

Nr. Albumu: 12846

Grupa ćwiczeniowa: cw1

Sprawozdanie z projektu symulatora procesora Intel 8086

2021

Spis treści

[Ogólna konstrukcja symulatora. 2](#_Toc70770097)

[1. Na jaka ocenę symulator był projektowany. 2](#_Toc70770098)

[2. Wygląd i sposób używania. 2](#_Toc70770099)

[3. Język programowania i sposób wykonania. 3](#_Toc70770100)

[Przykładowe polecenia 4](#_Toc70770101)

[1. MOV i XCHG bez kontaktu z pamięcią procesora 4](#_Toc70770102)

[2. MOV i XCHG z kontaktem z pamięcią 5](#_Toc70770103)

[3. Operacje PUSH i POP 7](#_Toc70770104)

[4. Clear Memory, Clear Stack 9](#_Toc70770105)

# Ogólna konstrukcja symulatora.

## Na jaka ocenę symulator był projektowany.

Symulator zaprojektowany jest z myślą o ocenie bardzo dobrej, czyli stworzony jest w trybie graficznym i można wykonywać w nim takie operacje jak MOV, EXCHG, MOV z pamięcią procesora, XCHG z pamięcią procesora, polecenia PUSH i POP.

## Wygląd i sposób używania.

Rozwijana lista wyboru adresowania z Pamięci do rejestru i odwrotnie.

Rozwijana lista trybów adresowania, która przy wyborze wpływa na wyświetlane przyciski radiowe.

Przyciski Random i Reset, służące kolejno do wypełnienia i zerowania poniżysz pól z wyjątkiem SP

Obraz zawierający stół

Opis wygenerowany automatycznie

Przyciski radiowe, służące do ustalenia rejestrów.

Operacje MOV i XCHG z pamięcią procesora, oraz operacje PUSH i POP.

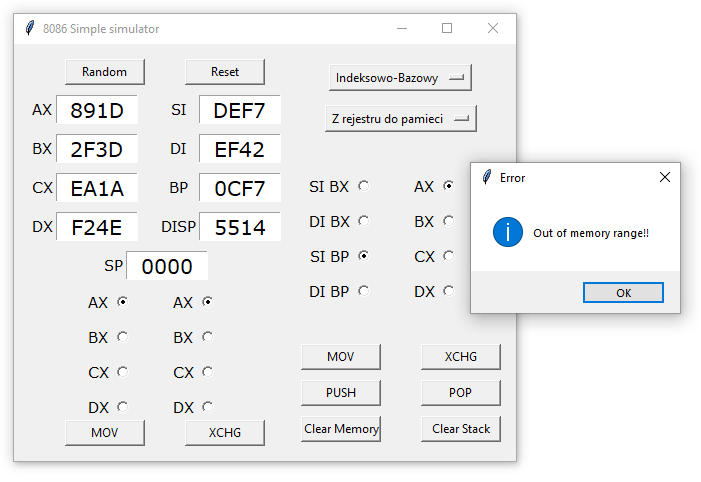
Przyciski radiowe, służące do ustalenia rejestru, przy operacjach PUSH, POP, XCHG z pamięcią procesora i MOV z pamięcią procesora.

Operacje MOV i XCHG bez kontaktu z pamięcią procesora. procesora.

Przyciski służące do zerowania pamięci procesora i stosu.

W przypadku adresu komórki pamięci procesora przekraczającej wielkość pamięci, jest wyrzucany błąd z informacją. Podobnie jest w przypadku próby ściągnięcia wartości z pustego stosu procesora.

Np.

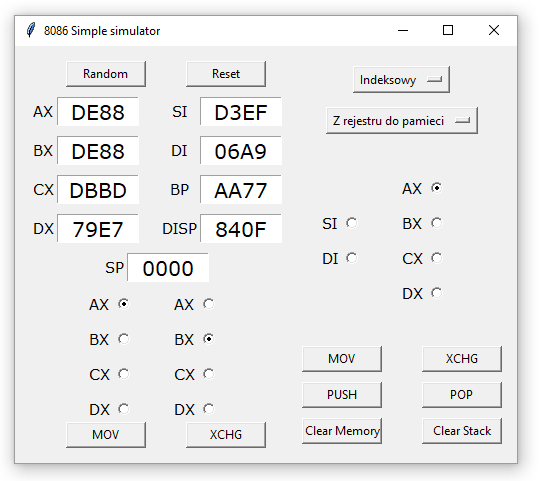
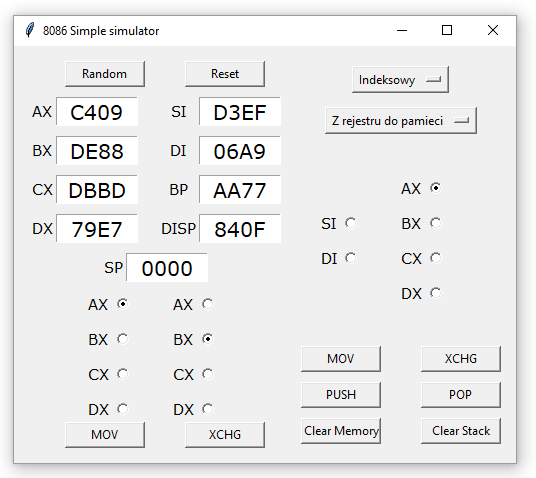


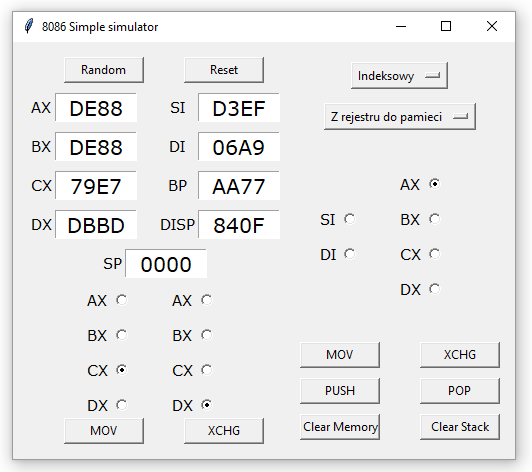
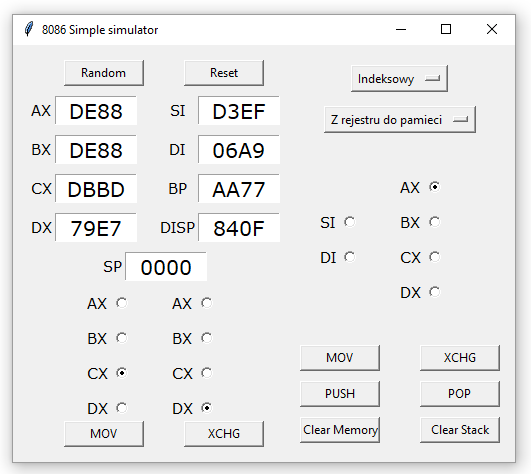
## Język programowania i sposób wykonania.

Program został napisany w języku Python 3.9, przy użyciu biblioteki Tkinter.

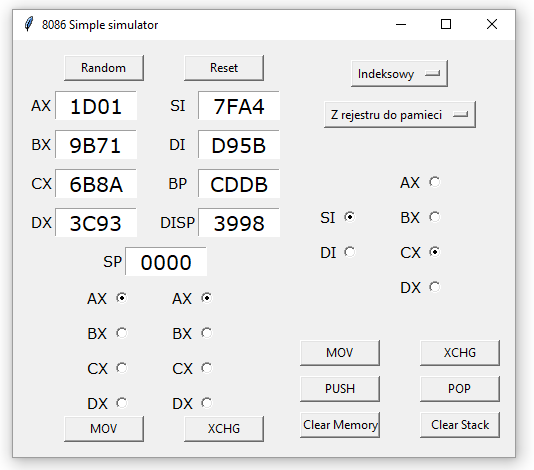
# Przykładowe polecenia

## MOV i XCHG bez kontaktu z pamięcią procesora

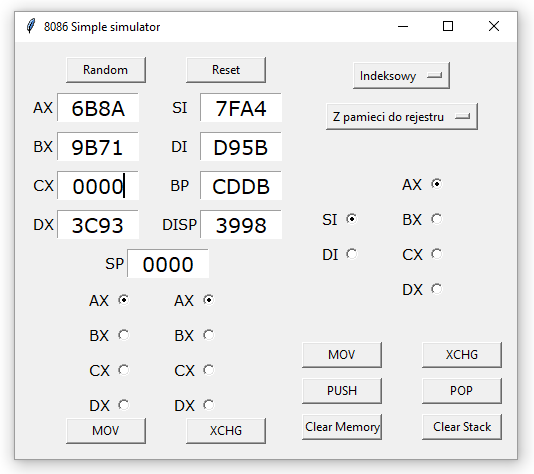
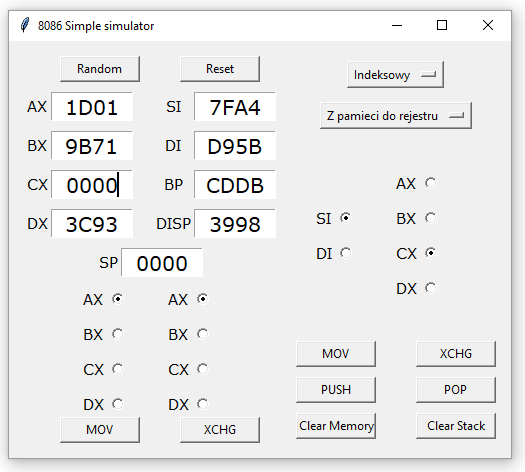
Np. MOV AX, BX  


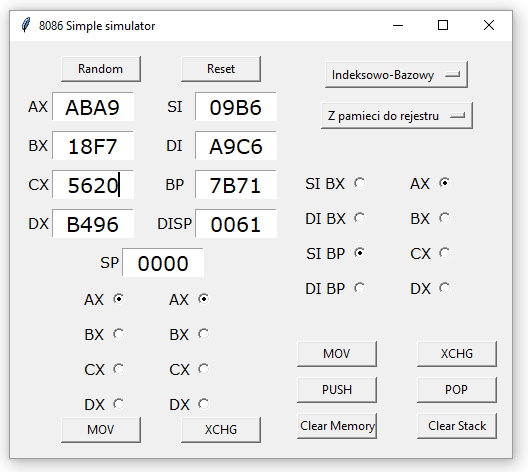
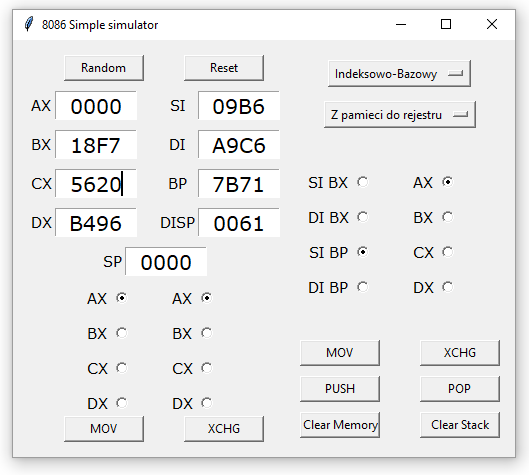
Np. XCHG CX, DX  


## MOV i XCHG z kontaktem z pamięcią

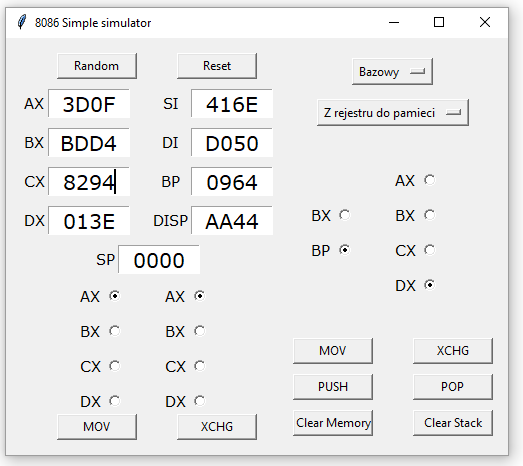
Np. MOV [ SI + DISP ], CX  
Wysyłamy teraz zawartość komórki CX pod wskazany adres pamięci (osobno starszy i młodszy bajt)

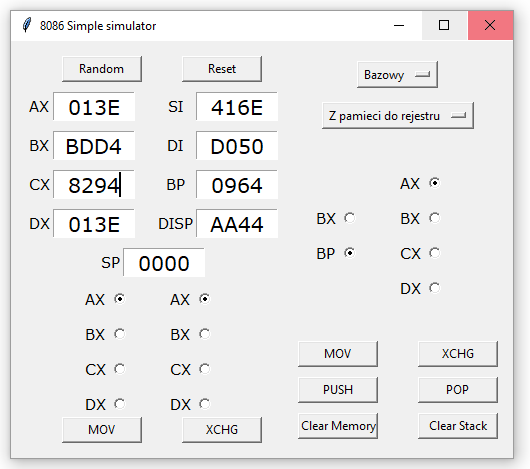
Możemy teraz dla potwierdzenia wyzerować komórkę CX i spróbować wysłać zapisaną wartość do komórki AX   
  
Np. MOV AX, [SI+DISP]



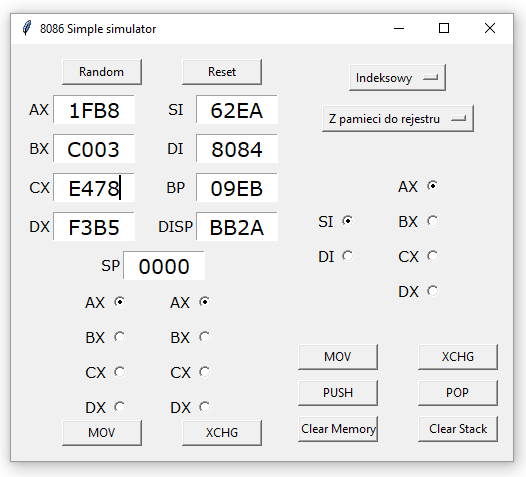
Np. XCHG [SI + BP + DISP], AX  
 

Ponieważ dana komórka nie była przez nas wcześniej nadpisana, otrzymaliśmy zera.

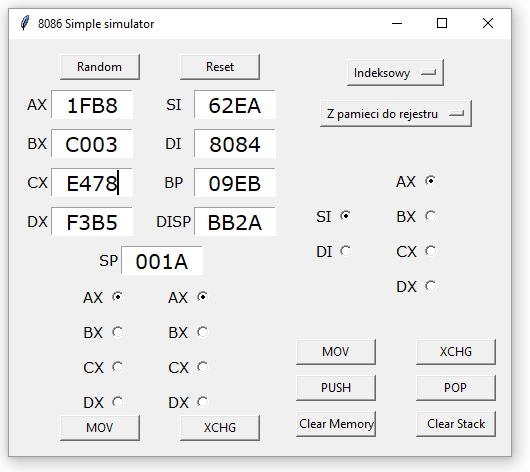
Np. MOV [ BP + DISP ], DX  


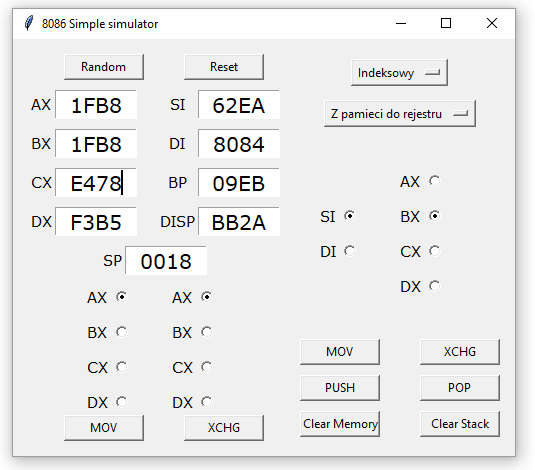
Teraz te same komórki pamięci wyślijmy do rejestru AX  
MOV AX, [ BP + DISP ]  


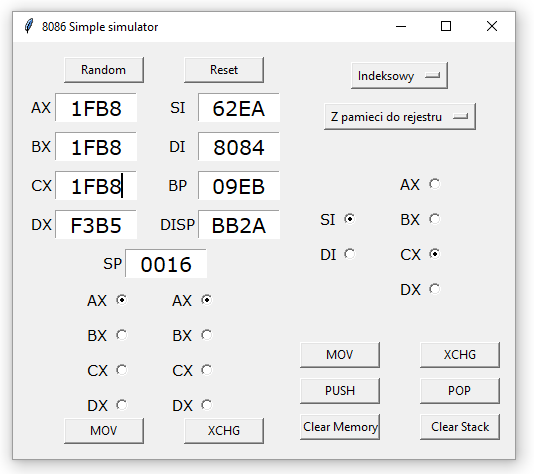
## Operacje PUSH i POP



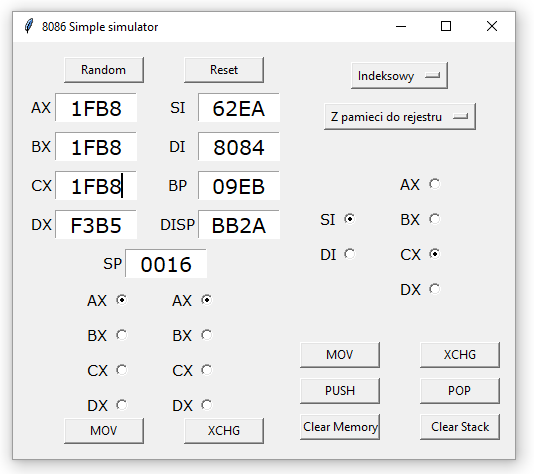
Wykonajmy operację PUSH AX 13 razy.

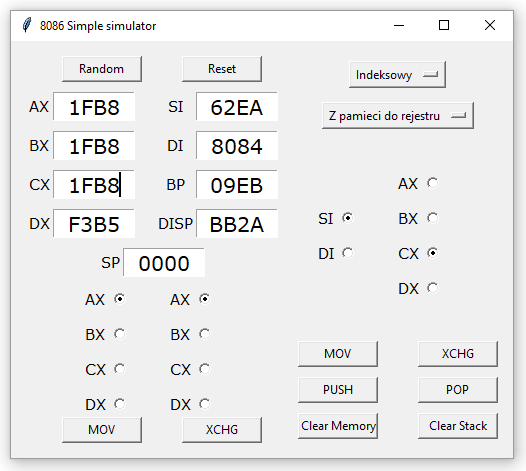


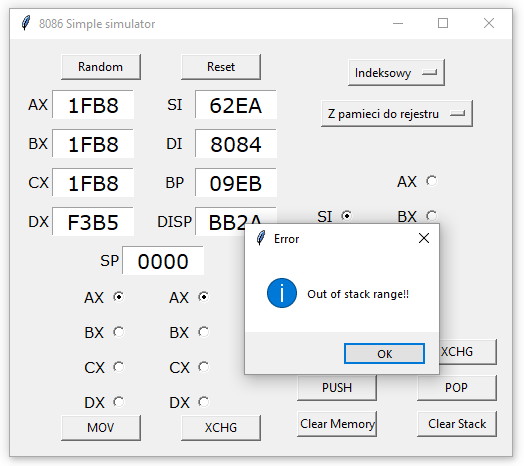
Wykonajmy teraz operację POP BX  


Teraz POP CX   


## 4. Clear Memory, Clear Stack



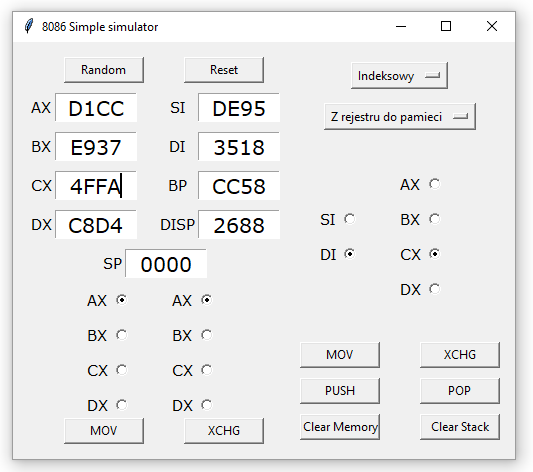
Teraz wciśnijmy przycisk Clear Stack.  


Spróbujmy wykonać operację POP AX  


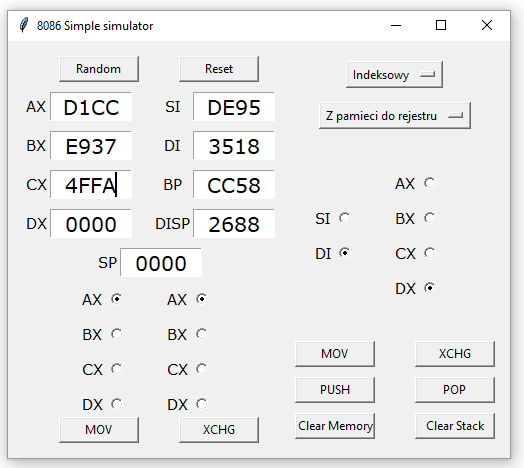
Jak widać Clear Stack zadziałało poprawnie.

Wypróbujmy teraz Clear Memory

Wykonajmy MOV [ DI + DISP ], CX



Teraz Wykonajmy Clear Memory i spróbujmy polecenie MOV DX, [ DI + DISP ]



Jak widać Clear Memory zadziałało poprawnie.

# Kod w języku Python.