

Intel 8086 Simulator

Projekt wykonał:

Franciszek Rybski

Informatyka i Ekonometria I rok

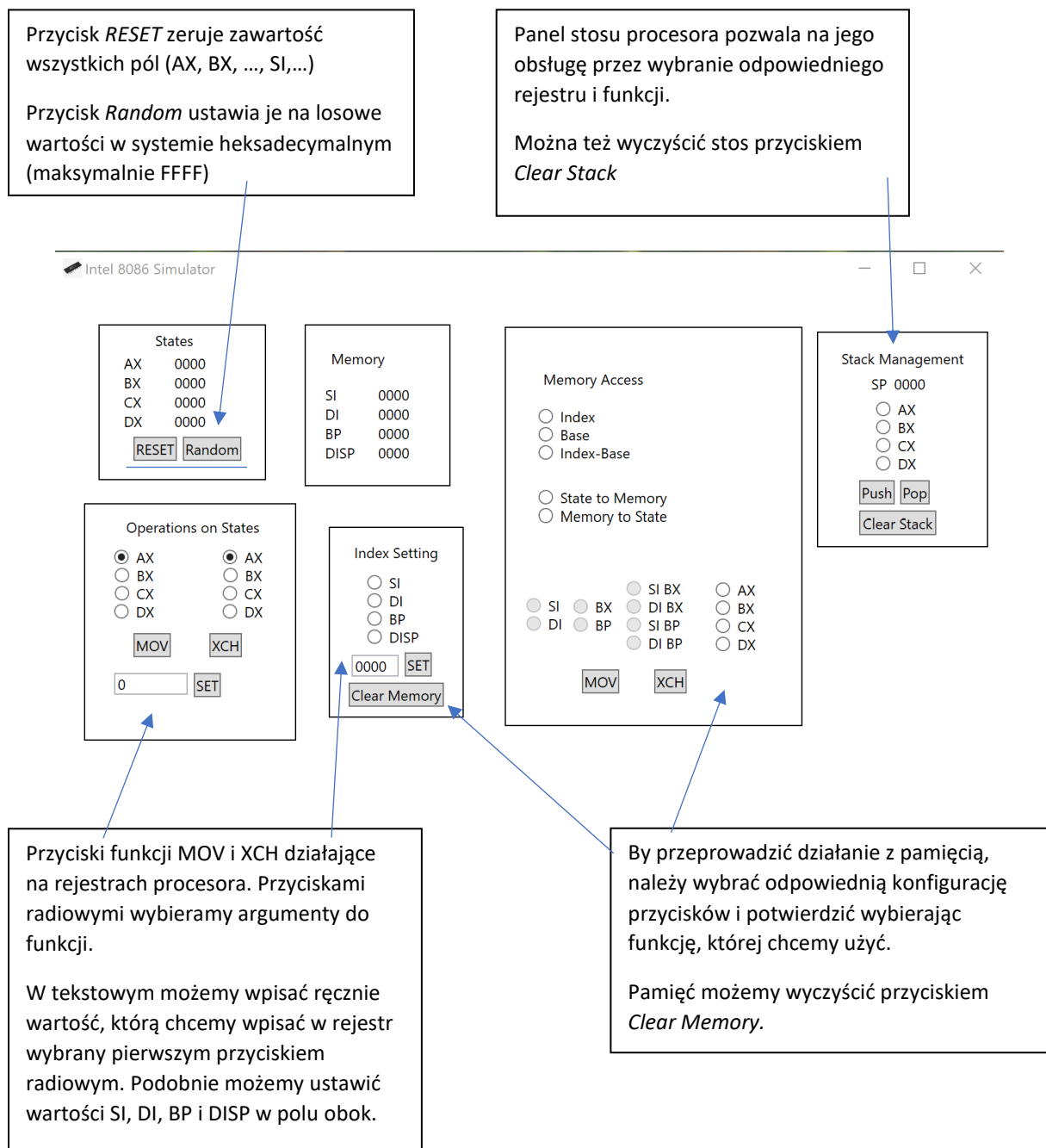
WSEI 2021

12902

1. Ogólne założenia projektu:

- Symulator posiada pełną funkcjonalność zgodną z wszelkimi wymaganiami przedstawionymi na zajęciach: modyfikowalne rejestry procesora, obsługa rozkazów MOV i XCH na rejestrach, zasymulowana pamięć 1MB, komunikacja z pamięcią, tryby indeksowania indeksowy, bazowy i indeksowo-bazowy, obsługa rozkazów MOV i XCH również na pamięci (w obie strony: procesor-pamięć, pamięć-procesor), symulacja stosu procesora, obsługa rozkazów PUSH i POP. Wszystko obudowane wygodnym GUI. (ocena: BDB)

2. Budowa symulatora:



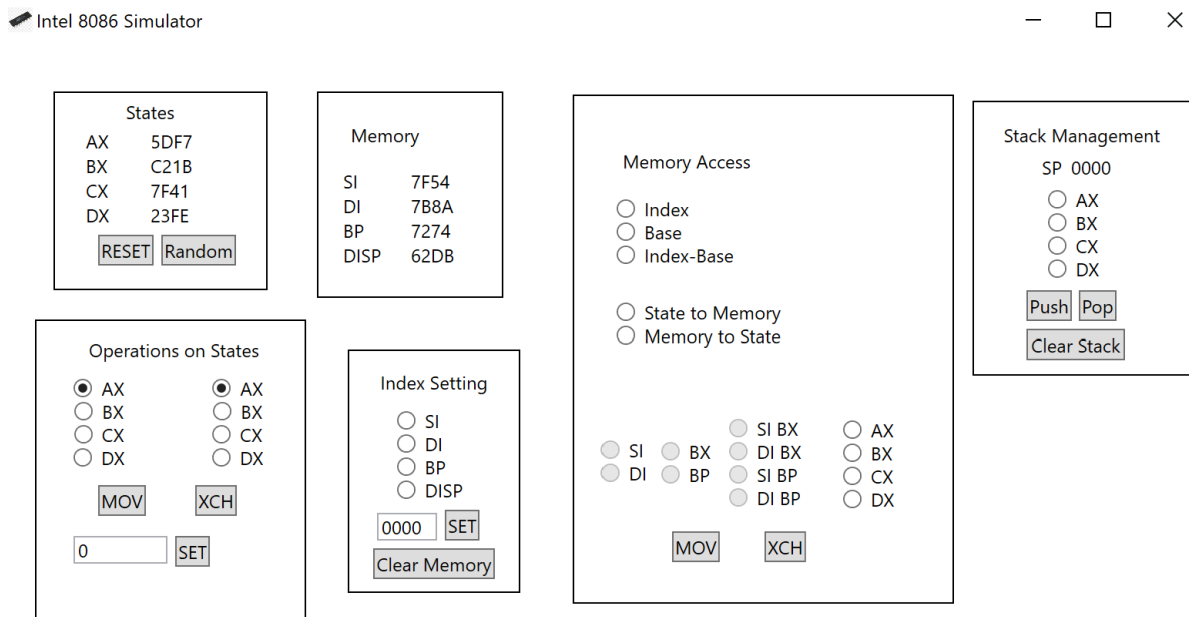
- Przy ewentualnej próbie wpisania do programu błędnych danych jak np. wybór adresu komórki z poza pamięci, czy próby wykonania funkcji POP przy pustym stosie symulator nie wyrzuci żadnych komunikatów, ale również nie wykona niepoprawnych operacji.

3. Wykorzystane technologie:

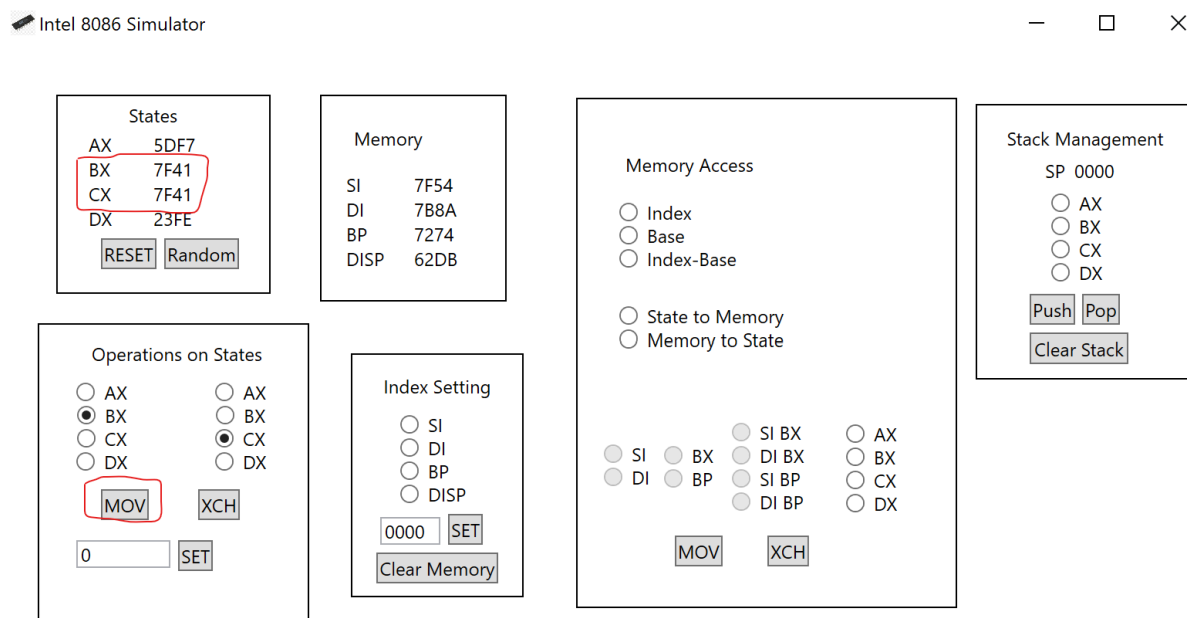
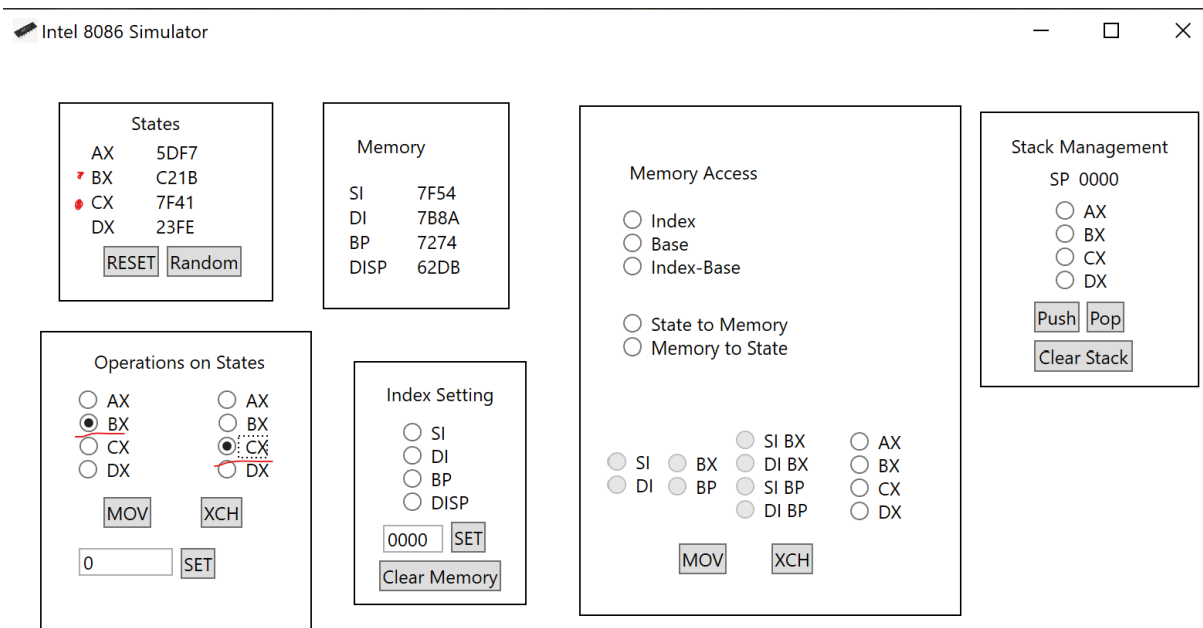
- GUI programu to aplikacja .NET Core 3.1 WPF przy użyciu języka C#

4. Demonstracja działania:

- Operacje na stanach:



Po wylosowaniu stanów możemy wykonać na nich operacje MOV i XCH. Na następnych zrzutach ekranu zostanie pokazane działanie MOV(BX, CX). Po zaznaczeniu odpowiednich przycisków, rejestrowi BX zostanie nadana wartość z rejestru CX.



W bardzo podobny sposób możemy wykonać operację XCH, jak na załączonych niżej zrzutach – znowu, po zaznaczeniu rejestrów i wciśnięciu przycisku XCH, rejestry, w tym przypadku AX i DX, zamieniają się swoją zawartością.

States	
AX	5DF7
BX	7F41
CX	7F41
DX	23FE
<input type="button" value="RESET"/> <input type="button" value="Random"/>	

Memory	
SI	7F54
DI	7B8A
BP	7274
DISP	62DB

Memory Access	
<input type="radio"/> Index	
<input type="radio"/> Base	
<input type="radio"/> Index-Base	
<input type="radio"/> State to Memory	
<input type="radio"/> Memory to State	
<input type="radio"/> SI	<input type="radio"/> BX
<input type="radio"/> DI	<input type="radio"/> BP
<input type="radio"/> SI BX	<input type="radio"/> DI BX
<input type="radio"/> SI BP	<input type="radio"/> DI BP
<input type="radio"/> AX	<input type="radio"/> BX
<input type="radio"/> CX	<input type="radio"/> DX
<input type="button" value="MOV"/> <input type="button" value="XCH"/>	

Stack Management	
SP 0000	
<input type="radio"/> AX	<input type="radio"/> BX
<input type="radio"/> CX	<input type="radio"/> DX
<input type="button" value="Push"/> <input type="button" value="Pop"/>	
<input type="button" value="Clear Stack"/>	

Operations on States	
<input type="radio"/> AX	<input checked="" type="radio"/> AX
<input type="radio"/> BX	<input type="radio"/> BX
<input type="radio"/> CX	<input type="radio"/> CX
<input checked="" type="radio"/> DX	<input type="radio"/> DX
<input type="button" value="MOV"/> <input type="button" value="XCH"/>	
0	<input type="button" value="SET"/>

Index Setting	
<input type="radio"/> SI	<input type="radio"/> DI
<input type="radio"/> BP	<input type="radio"/> DISP
0000	<input type="button" value="SET"/>
<input type="button" value="Clear Memory"/>	

States	
AX	23FE
BX	7F41
CX	7F41
DX	5DF7
<input type="button" value="RESET"/> <input type="button" value="Random"/>	

Memory	
SI	7F54
DI	7B8A
BP	7274
DISP	62DB

Memory Access	
<input type="radio"/> Index	
<input type="radio"/> Base	
<input type="radio"/> Index-Base	
<input type="radio"/> State to Memory	
<input type="radio"/> Memory to State	
<input type="radio"/> SI	<input type="radio"/> BX
<input type="radio"/> DI	<input type="radio"/> BP
<input type="radio"/> SI BX	<input type="radio"/> DI BX
<input type="radio"/> SI BP	<input type="radio"/> DI BP
<input type="radio"/> AX	<input type="radio"/> BX
<input type="radio"/> CX	<input type="radio"/> DX
<input type="button" value="MOV"/> <input type="button" value="XCH"/>	

Stack Management	
SP 0000	
<input type="radio"/> AX	<input type="radio"/> BX
<input type="radio"/> CX	<input type="radio"/> DX
<input type="button" value="Push"/> <input type="button" value="Pop"/>	
<input type="button" value="Clear Stack"/>	

Operations on States	
<input type="radio"/> AX	<input checked="" type="radio"/> AX
<input type="radio"/> BX	<input type="radio"/> BX
<input type="radio"/> CX	<input type="radio"/> CX
<input checked="" type="radio"/> DX	<input type="radio"/> DX
<input type="button" value="MOV"/> <input type="button" value="XCH"/>	
0	<input type="button" value="SET"/>

Index Setting	
<input type="radio"/> SI	<input type="radio"/> DI
<input type="radio"/> BP	<input type="radio"/> DISP
0000	<input type="button" value="SET"/>
<input type="button" value="Clear Memory"/>	

- Operacje na pamięci

Intel 8086 Simulator

States

AX	9354
BX	359E
CX	1CCE
DX	A5DC

RESET Random

Memory

SI	026A
DI	F8B4
BP	9D34
DISP	2F7B

Memory Access

☐ Index
☐ Base
☐ Index-Base

☐ State to Memory
☐ Memory to State

<input type="radio"/> SI	<input type="radio"/> BX	<input type="radio"/> SI BX	<input type="radio"/> AX
<input type="radio"/> DI	<input type="radio"/> BP	<input type="radio"/> DI BX	<input type="radio"/> BX
		<input type="radio"/> SI BP	<input type="radio"/> CX
		<input type="radio"/> DI BP	<input type="radio"/> DX

MOV XCH

Stack Management

SP 0000

☐ AX
☐ BX
☐ CX
☐ DX

Push Pop

Clear Stack

Operations on States

<input checked="" type="radio"/> AX	<input checked="" type="radio"/> AX
<input type="radio"/> BX	<input type="radio"/> BX
<input type="radio"/> CX	<input type="radio"/> CX
<input type="radio"/> DX	<input type="radio"/> DX

MOV XCH

0 SET

Index Setting

☐ SI
☒ DI
☐ BP
☐ DISP

0001 SET

Clear Memory

Na tej samej zasadzie możemy działać wykorzystując pamięć i indeksowanie jej komórek. W tym symulatorze działa to na zasadzie podobnej do wybierania stanów do wykonania działań. Wybieramy odpowiednie przyciski, najpierw tryb indeksowania, później kierunek działania, później właściwy indeks i na koniec rejestr, który będzie brał udział w działaniu.

Wartości indeksów mogą być losowane lub wpisywane ręcznie, jak przedstawiono powyżej.

Intel 8086 Simulator

States

AX	9354
BX	359E
CX	1CCE
DX	A5DC

RESET Random

Memory

SI	026A
DI	0001
BP	9D34
DISP	2F7B

Memory Access

☐ Index
☐ Base
☐ Index-Base

☐ State to Memory
☐ Memory to State

<input type="radio"/> SI	<input type="radio"/> BX	<input type="radio"/> SI BX	<input type="radio"/> AX
<input type="radio"/> DI	<input type="radio"/> BP	<input type="radio"/> DI BX	<input type="radio"/> BX
		<input type="radio"/> SI BP	<input type="radio"/> CX
		<input type="radio"/> DI BP	<input type="radio"/> DX

MOV XCH

Stack Management

SP 0000

☐ AX
☐ BX
☐ CX
☐ DX

Push Pop

Clear Stack

Operations on States

<input checked="" type="radio"/> AX	<input checked="" type="radio"/> AX
<input type="radio"/> BX	<input type="radio"/> BX
<input type="radio"/> CX	<input type="radio"/> CX
<input type="radio"/> DX	<input type="radio"/> DX

MOV XCH

0 SET

Index Setting

☐ SI
☒ DI
☐ BP
☐ DISP

0001 SET

Clear Memory

Intel 8086 Simulator

States

AX 9354
BX 359E
CX 1CCE
DX A5DC

RESET

Random

Memory

SI 026A
DI 0001
BP 9D34
DISP 2F7B

Memory Access

☒ Index
☐ Base
☐ Index-Base

☒ State to Memory
☐ Memory to State

☒ SI
☐ DI

☐ BX
☐ BP

☐ SI BX
☐ DI BX
☐ SI BP
☐ DI BP

☒ AX
☐ BX
☐ CX
☐ DX

MOV

XCH

Stack Management

SP 0000

☐ AX
☐ BX
☐ CX
☐ DX

Push

Pop

Clear Stack

Operations on States

☒ AX
☐ BX
☐ CX
☐ DX

☒ AX
☐ BX
☐ CX
☐ DX

MOV

XCH

0

SET

Index Setting

☐ SI
☒ DI
☐ BP
☐ DISP

0001

SET

Clear Memory

Intel 8086 Simulator

States

AX 9354
BX 359E
CX 1CCE
DX A5DC

RESET

Random

Memory

SI 026A
DI 0001
BP 9D34
DISP 2F7B

Memory Access

☒ Index
☐ Base
☐ Index-Base

☐ State to Memory
☒ Memory to State

☒ SI
☐ DI

☐ BX
☐ BP

☐ SI BX
☐ DI BX
☐ SI BP
☐ DI BP

☒ AX
☒ BX
☐ CX
☐ DX

MOV

XCH

Stack Management

SP 0000

☐ AX
☐ BX
☐ CX
☐ DX

Push

Pop

Clear Stack

Operations on States

☒ AX
☐ BX
☐ CX
☐ DX

☒ AX
☐ BX
☐ CX
☐ DX

MOV

XCH

0

SET

Index Setting

☐ SI
☒ DI
☐ BP
☐ DISP

0001

SET

Clear Memory

Na zrzutach przedstawione zostało działanie:

MOV(SI+DISP, AX)

Intel 8086 Simulator

States

AX	9354
BX	9354
CX	1CCE
DX	A5DC

RESET Random

Memory

SI	026A
DI	0001
BP	9D34
DISP	2F7B

Memory Access

☒ Index
☐ Base
☐ Index-Base

☐ State to Memory
☒ Memory to State

☒ SI ☐ BX ☐ SI BX ☐ AX
☐ DI ☐ BP ☐ DI BX ☐ BX
☐ DI BP ☐ SI BP ☒ CX
☐ DI BP ☐ DX

MOV XCH

Stack Management

SP 0000

☐ AX
☐ BX
☐ CX
☐ DX

Push Pop

Clear Stack

Operations on States

☒ AX ☒ AX
☐ BX ☐ BX
☐ CX ☐ CX
☐ DX ☐ DX

MOV XCH

0 SET

Index Setting

☐ SI
☒ DI
☐ BP
☐ DISP

0001 SET

Clear Memory

A także działanie: XCH(BX, SI+DISP).

Intel 8086 Simulator

States

AX	9354
BX	9354
CX	359E
DX	A5DC

RESET Random

Memory

SI	026A
DI	0001
BP	9D34
DISP	2F7B

Memory Access

☐ Index
☐ Base
☒ Index-Base

☒ State to Memory
☐ Memory to State

☐ SI ☐ BX ☐ SI BX ☐ AX
☐ DI ☐ BP ☐ DI BX ☐ BX
☐ DI BP ☒ SI BP ☐ CX
☐ DI BP ☒ DX

MOV XCH

Stack Management

SP 0000

☐ AX
☐ BX
☐ CX
☐ DX

Push Pop

Clear Stack

Operations on States

☒ AX ☒ AX
☐ BX ☐ BX
☐ CX ☐ CX
☐ DX ☐ DX

MOV XCH

0 SET

Index Setting

☐ SI
☒ DI
☐ BP
☐ DISP

0001 SET

Clear Memory

Możemy również przeprowadzić te operacje wykorzystując system indeksowo-bazowy.

States		Memory		Memory Access		Stack Management	
AX	9354	SI	026A	<input type="radio"/> Index		SP	0000
BX	9354	DI	0001	<input type="radio"/> Base		<input type="radio"/> AX	
CX	359E	BP	9D34	<input checked="" type="radio"/> Index-Base		<input type="radio"/> BX	
DX	A5DC	DISP	2F7B	<input type="radio"/> State to Memory		<input type="radio"/> CX	
<input type="button" value="RESET"/> <input type="button" value="Random"/>				<input type="radio"/> Memory to State		<input type="radio"/> DX	
				<input type="radio"/> SI BX	<input checked="" type="radio"/> AX	<input type="button" value="Push"/> <input type="button" value="Pop"/>	
				<input type="radio"/> DI BX	<input type="radio"/> BX	<input type="button" value="Clear Stack"/>	
				<input type="radio"/> SI BP	<input type="radio"/> CX		
				<input checked="" type="radio"/> DI BP	<input type="radio"/> DX		
				<input type="button" value="MOV"/> <input type="button" value="XCH"/>			

Operations on States		Index Setting	
<input checked="" type="radio"/> AX	<input checked="" type="radio"/> AX	<input type="radio"/> SI	
<input type="radio"/> BX	<input type="radio"/> BX	<input checked="" type="radio"/> DI	
<input type="radio"/> CX	<input type="radio"/> CX	<input type="radio"/> BP	
<input type="radio"/> DX	<input type="radio"/> DX	<input type="radio"/> DISP	
<input type="button" value="MOV"/> <input type="button" value="XCH"/>		<input type="text" value="0001"/> <input type="button" value="SET"/>	
<input type="text" value="0"/> <input type="button" value="SET"/>		<input type="button" value="Clear Memory"/>	

States		Memory		Memory Access		Stack Management	
AX	A5DC	SI	026A	<input type="radio"/> Index		SP	0000
BX	9354	DI	0001	<input type="radio"/> Base		<input type="radio"/> AX	
CX	359E	BP	9D34	<input checked="" type="radio"/> Index-Base		<input type="radio"/> BX	
DX	A5DC	DISP	2F7B	<input type="radio"/> State to Memory		<input type="radio"/> CX	
<input type="button" value="RESET"/> <input type="button" value="Random"/>				<input checked="" type="radio"/> Memory to State		<input type="radio"/> DX	
				<input type="radio"/> SI BX	<input checked="" type="radio"/> AX	<input type="button" value="Push"/> <input type="button" value="Pop"/>	
				<input type="radio"/> DI BX	<input type="radio"/> BX	<input type="button" value="Clear Stack"/>	
				<input type="radio"/> SI BP	<input type="radio"/> CX		
				<input checked="" type="radio"/> DI BP	<input type="radio"/> DX		
				<input type="button" value="MOV"/> <input type="button" value="XCH"/>			

Operations on States		Index Setting	
<input checked="" type="radio"/> AX	<input checked="" type="radio"/> AX	<input type="radio"/> SI	
<input type="radio"/> BX	<input type="radio"/> BX	<input checked="" type="radio"/> DI	
<input type="radio"/> CX	<input type="radio"/> CX	<input type="radio"/> BP	
<input type="radio"/> DX	<input type="radio"/> DX	<input type="radio"/> DISP	
<input type="button" value="MOV"/> <input type="button" value="XCH"/>		<input type="text" value="0001"/> <input type="button" value="SET"/>	
<input type="text" value="0"/> <input type="button" value="SET"/>		<input type="button" value="Clear Memory"/>	

Przedstawione operacje to:

MOV(DI + BP + DISP, DX) | XCH(AX, DI + BP + DISP)

- Symulacja działania stosu procesora:

Intel 8086 Simulator

States

AX 2B7B
BX 02F0
CX 5633
DX EA4E

RESET Random

Memory

SI 0E34
DI 873F
BP 3CF1
DISP E05D

Memory Access

☐ Index
☐ Base
☒ Index-Base

☐ State to Memory
☒ Memory to State

☐ SI BX
☐ DI BX
☐ SI BP
☒ DI BP

MOV XCH

Stack Management

SP 0000

☒ AX
☐ BX
☐ CX
☐ DX

Push Pop

Clear Stack

Operations on States

☒ AX ☒ AX
☐ BX ☐ BX
☐ CX ☐ CX
☐ DX ☐ DX

MOV XCH

0 SET

Index Setting

☐ SI
☒ DI
☐ BP
☐ DISP

0001 SET

Clear Memory

Na stosie możemy wykonać dwie podstawowe operacje: PUSH i POP. Na początek dodajemy kilka razy AX na stos (PUSH(AX)).

Intel 8086 Simulator

States

AX 2B7B
BX 02F0
CX 5633
DX EA4E

RESET Random

Memory

SI 0E34
DI 873F
BP 3CF1
DISP E05D

Memory Access

☐ Index
☐ Base
☒ Index-Base

☐ State to Memory
☒ Memory to State

☐ SI BX
☐ DI BX
☐ SI BP
☒ DI BP

MOV XCH

Stack Management

SP 000A

☐ AX
☒ BX
☐ CX
☐ DX

Push Pop

Clear Stack

Operations on States

☒ AX ☒ AX
☐ BX ☐ BX
☐ CX ☐ CX
☐ DX ☐ DX

MOV XCH

0 SET

Index Setting

☐ SI
☒ DI
☐ BP
☐ DISP

0001 SET

Clear Memory

Następnie możemy wykonać PUSH(BX). Każdy PUSH dodaje na stos 2B danych, przez co SP rośnie o 2.

States	Memory	Memory Access	Stack Management
AX 2B7B BX 02F0 CX 5633 DX EA4E <input type="button" value="RESET"/> <input type="button" value="Random"/>	SI 0E34 DI 873F BP 3CF1 DISP E05D	Memory Access <input type="radio"/> Index <input type="radio"/> Base <input checked="" type="radio"/> Index-Base <input type="radio"/> State to Memory <input checked="" type="radio"/> Memory to State <div> <input type="radio"/> SI <input type="radio"/> BX <input type="radio"/> SI BX <input checked="" type="radio"/> AX <input type="radio"/> DI <input type="radio"/> BP <input type="radio"/> DI BX <input type="radio"/> SI BP <input type="radio"/> BX <input type="radio"/> DI BP <input checked="" type="radio"/> DI BP <input type="radio"/> DX </div> <input type="button" value="MOV"/> <input type="button" value="XCH"/>	SP 000C <input type="radio"/> AX <input type="radio"/> BX <input type="radio"/> CX <input checked="" type="radio"/> DX <input type="button" value="Push"/> <input type="button" value="Pop"/> <input type="button" value="Clear Stack"/>

Operations on States	Index Setting
<input checked="" type="radio"/> AX <input checked="" type="radio"/> AX <input type="radio"/> BX <input type="radio"/> BX <input type="radio"/> CX <input type="radio"/> CX <input type="radio"/> DX <input type="radio"/> DX <input type="button" value="MOV"/> <input type="button" value="XCH"/> 0 <input type="button" value="SET"/>	<input type="radio"/> SI <input checked="" type="radio"/> DI <input type="radio"/> BP <input type="radio"/> DISP 0001 <input type="button" value="SET"/> <input type="button" value="Clear Memory"/>

Teraz na stosie mamy od góry wartości: BX, AX, AX, AX, AX, AX. Kiedy wykonamy operację POP, wybranemu rejestrowi zostanie przypisana wartość, którą w momencie dodawania na stos miał wpisaną rejestr BX.

States	Memory	Memory Access	Stack Management
AX 2B7B BX 02F0 CX 5633 DX 02F0 <input type="button" value="RESET"/> <input type="button" value="Random"/>	SI 0E34 DI 873F BP 3CF1 DISP E05D	Memory Access <input type="radio"/> Index <input type="radio"/> Base <input checked="" type="radio"/> Index-Base <input type="radio"/> State to Memory <input checked="" type="radio"/> Memory to State <div> <input type="radio"/> SI <input type="radio"/> BX <input checked="" type="radio"/> AX <input type="radio"/> DI <input type="radio"/> BP <input type="radio"/> DI BX <input type="radio"/> SI BP <input type="radio"/> BX <input type="radio"/> DI BP <input checked="" type="radio"/> DI BP <input type="radio"/> DX </div> <input type="button" value="MOV"/> <input type="button" value="XCH"/>	SP 000A <input type="radio"/> AX <input type="radio"/> BX <input type="radio"/> CX <input checked="" type="radio"/> DX <input type="button" value="Push"/> <input type="button" value="Pop"/> <input type="button" value="Clear Stack"/>

Operations on States	Index Setting
<input checked="" type="radio"/> AX <input checked="" type="radio"/> AX <input type="radio"/> BX <input type="radio"/> BX <input type="radio"/> CX <input type="radio"/> CX <input type="radio"/> DX <input type="radio"/> DX <input type="button" value="MOV"/> <input type="button" value="XCH"/> 0 <input type="button" value="SET"/>	<input type="radio"/> SI <input checked="" type="radio"/> DI <input type="radio"/> BP <input type="radio"/> DISP 0001 <input type="button" value="SET"/> <input type="button" value="Clear Memory"/>

Jak widzimy, operacja się powiodła zgodnie z przewidywaniami. SP zmniejszyło się, rejestr DX otrzymał poprawną wartość.

5. Kod źródłowy

- Cały kod znajduje się na repozytorium na platformie GitHub pod adresem:

<https://github.com/donFranko99/Intel8086Project>