

Projekt zaliczeniowy

Symulator rozkazów procesora INTEL 8086

Imię i nazwisko:	Paweł Trojański
Nr. Indeksu:	13335
Przedmiot:	Architektura systemów komputerowych
Grupa:	L/21 N lab8/1/IE
Język programowania:	C#
Repozytorium:	<u>GitHub</u>

Projekt ma na celu pokazania działania procesora INTEL 8086, jest to więc symulator rozkazów.

The screenshot shows the Intel 8086 simulator interface. At the top, there are four rows of registers: AX, BX, CX, and DX, each with a pink input field containing '0000'. To the right of each register name are two blue input fields for its high and low bytes: AH/AL, BH/BL, CH/CL, and DH/DL, each containing '00'. On the far right, there are four white buttons: 'MOV', 'XCHG', 'Random', and 'Zerowanie'. Below the registers, there are two white boxes labeled 'Przeznaczenie' (Destination) and 'Zawartość' (Content). Each box contains a list of register names: AX, BX, CX, DX, AH, AL, BH, BL, CH, CL, DH, and DL. The 'Przeznaczenie' box has the first four names in pink and the rest in blue. The 'Zawartość' box has the first four names in pink and the rest in blue.

Rysunek 1 - Zobrazowanie całości projektu

Całość składa się z:

- 4 komórek tekstowych przeznaczonych na 16-bitowe rejestry ogólnego przeznaczenia zapisane w systemie szesnastkowym. Pola tekstowe zabezpieczone są wzorcem opisującym łańcuch symboli (tzw. RegExp), który uniemożliwia nam wpisanie do pola wartości spoza przedziału ([A-F]; [0-9]) i o innej długości niż 4 znaki,
- 8 komórek tekstowych przeznaczonych na 8-bitowe rejestry składowe wyżej wymienionych rejestrów. One również są zapisane w systemie szesnastkowym. Możemy z nich korzystać zamiennie (tj. AX lub AH, AL; BX lub BH, BL; CX lub CH, CL; DX lub DH, DL). Pola tekstowe zabezpieczone są wzorcem opisującym łańcuch symboli, który uniemożliwia nam wpisanie do pola wartości spoza przedziału ([A-F]; [0-9]) i o innej długości niż 2 znaki,

The screenshot shows the 16-bit registers: AX, BX, CX, and DX. Each register name is in a grey box, followed by a pink input field containing '0000'.

Rysunek 2 - Rejestry powszechnego stosowania

The screenshot shows the 8-bit registers: AH, AL, BH, BL, CH, CL, DH, and DL. Each register name is in a grey box, followed by a blue input field. The values are: AH=00, AL=00, BH=00, BL=00, CH=00, CL=00, DH=00, and DL=00.

Rysunek 3 – Rejestry składowe rejestrów AX, BX, CX, DX

- 4 przyciski interaktywne do symulowania rozkazów tj.: MOV, XCHG, Random i Zerowanie (całość opisana poniżej),



Rysunek 4 - Przyciski funkcyjne do symulacji rozkazów

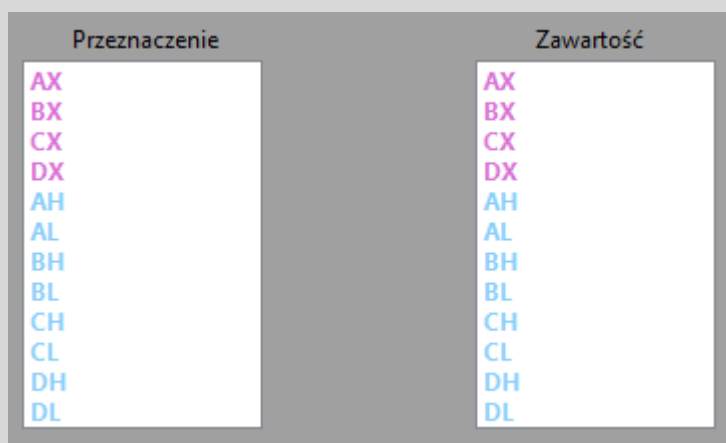
MOV – przesyła (kopiuje) zawartość komórki jednego rejestru do komórki innego na podstawie wyborów z list opisanych poniżej,

XCHG – zamienia miejscami zawartości komórek dwóch rejestrów,

Random – na podstawie wzorca opisującego łańcuch symboli generuje do każdej z komórek z rejestrami losowe dane z przedziału ([A-F]; [0-9]),

Zerowanie – zeruje wszystkie komórki z rejestrami.

- 2 list wybieranych, z których możemy wybrać komórkę, którą chcemy przenieść i do którego z rejestrów.



Rysunek 5 - Listy wybierane z rejestrami

Przykłady zastosowania symulatora i funkcjonalności

AX	GZGZ	AH	GZ	AL	GZ
BX	1F96	BH	1F	BL	96
CX	D59B	CH	D5	CL	9B
DX	2C9E	DH	2C	DL	9E

Rysunek 6 - Przykład zastosowania RegExp poprzez nieprawidłowe uzupełnienie rejestru AX tła pól tekstowych AX, AH oraz AL wyświetlane są na czerwono, a reszta poprawnych na zielono

AX	2CGZ	AH	2C	AL	GZ
BX	1F96	BH	1F	BL	96
CX	D59B	CH	D5	CL	9B
DX	2C9E	DH	2C	DL	9E

Przeznaczenie

AX
BX
CX
DX
AH
AL
BH
BL
CH
CL
DH
DL

Zawartość

AX
BX
CX
DX
AH
AL
BH
BL
CH
CL
DH
DL

MOV

XCHG

Random

Zerowanie

Rysunek 7 - Przykład przestania (skopio wania) za pomocą przycisku MOV zawartości komórki DH do komórki AH oraz wynikająca z tego automatyczna zmiana komórki rejestru AX

AX	2C9E	AH	2C	AL	9E
BX	1F96	BH	1F	BL	96
CX	D59B	CH	D5	CL	9B
DX	2CGZ	DH	2C	DL	GZ

Przeznaczenie

AX
BX
CX
DX
AH
AL
BH
BL
CH
CL
DH
DL

Zawartość

AX
BX
CX
DX
AH
AL
BH
BL
CH
CL
DH
DL

MOV

XCHG

Random

Zerowanie

Rysunek 8 - Przykład zamiany zawartości pól AX oraz DX za pomocą przycisku XCHG oraz wynikająca z tego automatyczna zmiana rejestrów składowych AH, AL oraz DH, DL

AX	0000	AH	00	AL	00	MOV XCHG Random Zerowanie
BX	0000	BH	00	BL	00	
CX	0000	CH	00	CL	00	
DX	0000	DH	00	DL	00	

Przeznaczenie	Zawartość
AX	AX
BX	BX
CX	CX
DX	DX
AH	AH
AL	AL
BH	BH
BL	BL
CH	CH
CL	CL
DH	DH
DL	DL

Rysunek 9 - Przykład wyzerowania komórek z danymi za pomocą przycisku Zerowanie

AX	4890	AH	48	AL	90	MOV XCHG Random Zerowanie
BX	C15C	BH	C1	BL	5C	
CX	9092	CH	90	CL	92	
DX	39E5	DH	39	DL	E5	

Przeznaczenie	Zawartość
AX	AX
BX	BX
CX	CX
DX	DX
AH	AH
AL	AL
BH	BH
BL	BL
CH	CH
CL	CL
DH	DH
DL	DL

Rysunek 10 - Przykład uzupełnienia komórek poprawnymi danymi w systemie szesnastkowym wygenerowanymi losowo za pomocą przycisku Random w oparciu o zastosowanie RegExp