

PROJEKT

Emulator procesora Intel 8086

Paweł Kęska nr.albumu 13075

1.Wstęp

Program emulator procesora Intel 8086 napisany w języku C#. Emulator pozwala na symulację rozkazów MOV oraz XCHG dla rejestrów oraz rejestrów i pamięci w trybach adresacji indeksowym, bazowym oraz indeksowo-bazowym. Pozwala również na operacje PUSH oraz POP na stosie.

2.Opis symulatora

Intel 8086 to 16 bitowy mikroprocesor wprowadzony na rynek 8 czerwca 1978r. Został zaprojektowany w technologii 3 μ m jako rozszerzenie 8 bitowego procesora 8080 oraz 8085. Wykonywany głównie w obudowach 40pin. Do dzisiaj jest produkowany przez niektórych dostawców.

3.Architektura użyta w projekcie

- Rejestry arytmetyczne – 16 bitowe rejestry ogólnego przeznaczenia AX,BX,CX,DX
 - AX – Akumulator
 - BX – Baza
 - CX – Licznik
 - DX – Dane
- Rejestry wskaźnikowe i indeksowe – 16 bitowe rejestry których głównym zadaniem jest wskazanie miejsca w pamięci
 - SI – Rejestr indeksowy źródła
 - BP – Wskaźnik bazy
 - DI – Rejestr indeksowy przeznaczenia
 - SP – Wskaźnik stosu
- Segment stosu
- Segment danych

4. Instrukcja obsługi

Aby uruchomić program należy dwukrotnie kliknąć na program "Intel 8086 Paweł Kęska.exe" ścieżka :
Intel 8086 Paweł Kęska\bin\Debug\net5.0-windows

Widok po uruchomieniu

Intel 8086 Paweł Kęska

	Reset	Random
AX	0000	0000
BX	0000	0000
CX	0000	0000
DX	0000	0000

☐ z rejestru do pamięci
☐ z pamięci do rejestru

☐ indeksowy
☐ bazowy
☐ indeksowo-bazowy

☐ SI ☐ DI

☐ BP ☐ BX

☐ SI BX ☐ DI BX
☐ SI BP ☐ DI BP

☐ AX ☐ BX
☐ CX ☐ DX

	Reset	Random
SI	0000	0000
DI	0000	0000
BP	0000	0000
SP	0000	0000
DISP	0000	0000

MOV AX,BX XCHG AX,BX
MOV AX,CX XCHG AX,CX
MOV AX,DX XCHG AX,DX
MOV BX,AX XCHG BX,AX
MOV BX,CX XCHG BX,CX
MOV BX,DX XCHG BX,DX
MOV CX,AX XCHG CX,AX
MOV CX,BX XCHG CX,BX
MOV CX,DX XCHG CX,DX
MOV DX,AX XCHG DX,AX
MOV DX,BX XCHG DX,BX
MOV DX,CX XCHG DX,CX

MOV XCHG
PUSH AX POP AX
PUSH BX POP BX
PUSH CX POP CX
PUSH DX POP DX

Projekt Intel 8086
Paweł Kęska
Nr Indeksu 13075

A) Rejestry

Rejestry AX, BX, CX, DX posiadają dwa pola. Pole lewe które jest tylko do odczytu i pokazuje wartość rejestru oraz prawe pole które służy **tylko** do wpisywania wartości.

Rejestry AX, BX, CX, DX można uzupełnić na dwa sposoby :

1. Wpisanie wartości w prawych okienkach oraz zatwierdzenie enterem. Okienko przyjmuje tylko liczby w systemie HEX czyli cyfry z zakresu 0-9 oraz litery od A do F.

	Reset	Random
AX	0000	0000
BX	0000	0000
CX	0000	0000
DX	0000	0000

2. Przycisk Random – który losuje cztery liczby w systemie HEX

	Reset	Random
AX	D2CE	D2CE
BX	B273	B273
CX	AC82	AC82
DX	63A2	63A2

Rejestry można również wyzerować za pomocą przycisku Reset.

	Reset	Random
AX	0000	D2CE
BX	0000	B273
CX	0000	AC82
DX	0000	63A2

Rejestry SI,DI,BP,SP,DISP posiadają dwa pola. Pole lewe które jest tylko do odczytu i pokazuje wartość rejestru oraz prawe pole które służy **tylko** do wpisywania wartości.

Rejestry SI,DI,BP,SP,DISP można uzupełnić tylko jednym sposobem:

Wpisanie wartości w prawych okienkach oraz zatwierdzenie enterem. Okienko przyjmuje tylko liczby w systemie HEX czyli cyfry z zakresu 0-9 oraz litery od A do F.

B)Rozkazy MOV oraz XCHG dla rejestrów

Emulator pozwala na symulację rozkazów MOV oraz XCHG między rejestrami. Rozkaz MOV pozwala na przeniesienie wartości z jednego rejestru do drugiego, zostawiając pierwszy rejestr z tą samą wartością. Rozkaz XCHG pozwala na zamianę wartości w dwóch rejestrach. Wartość pierwszego rejestru przechodzi do drugiego, a wartość drugiego rejestru przechodzi do pierwszego.

Aby zasymulować rozkazy MOV należy uzupełnić rejestry które chcemy wykorzystać (np. przycisk Random). Następnie należy wybrać z opcji poniżej odpowiedni rozkaz.

Przed

	Reset	Random
AX	CD4B	CD4B
BX	310F	310F
CX	410C	410C
DX	3489	3489
	MOV AX,BX	XCHG AX,BX

:

Po :

	Reset	Random
AX	CD4B	CD4B
BX	CD4B	310F
CX	410C	410C
DX	3489	3489
	MOV AX,BX	XCHG AX,BX

Analogicznie działa opcja XCHG

Przed :

	Reset	Random
AX	07AA	07AA
BX	00F7	00F7
CX	0862	0862
DX	C7E6	C7E6
	MOV AX,BX	XCHG AX,BX

Po :

	Reset	Random
AX	00F7	07AA
BX	07AA	00F7
CX	0862	0862
DX	C7E6	C7E6
	MOV AX,BX	XCHG AX,BX

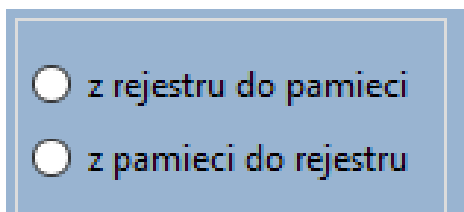
C)Rozkazy MOV oraz XCHG między rejestrami a pamięcią

Emulator pozwala również na symulacje rozkazów MOV oraz XCHG między rejestrami a pamięcią.

Wszystkie wartości z czterech rejestrów ogólnego przeznaczenia czyli AX,BX,CX,DX możemy zapisać w pamięci adresując za pomocą trzech trybów : Tryb indeksowy, Tryb Bazowy oraz Tryb Indeksowo-Bazowy.

Pamięć ma maksymalną wartość 64x1024 co daje 65536 komórek. W przypadku próby rozkazu MOV na adresie większym niż maksymalna wartość pamięci, wartość zapisze się w dwóch ostatnich komórkach pamięci.

Aby zasymulować rozkaz MOV należy najpierw wybrać czy chcemy przenieść wartość z rejestru do pamięci czy z pamięci do rejestru.



☐ z rejestru do pamieci

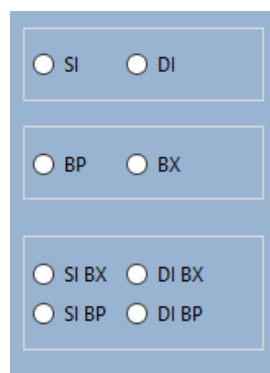
☐ z pamieci do rejestru

Następnie wybieramy tryb adresacji. W zależności od tego który tryb wybierzemy, program będzie liczył miejsce pamięci z rejestrów. Po wybraniu trybu adresacji (Addressing Modes) musimy wybrać rejestry które chcemy użyć do wyznaczenia miejsca w pamięci.

indeksowy– tryb indeksowy (Możliwe wybory SI oraz DI)

bazowy – tryb bazowy (Możliwe wybory BP oraz BX)

indeksowo – bazowy – Tryb bazowo indeksowy (Możliwe wybory DI BP,DI BX, SI BP oraz SI BX)



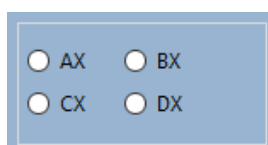
☐ SI ☐ DI

☐ BP ☐ BX

☐ SI BX ☐ DI BX

☐ SI BP ☐ DI BP

Następnie należy wybrać rejestr którego chcemy użyć:



☐ AX ☐ BX

☐ CX ☐ DX

Przykład rozkazu MOV rejestru AX do pamięci o adresie indeksowym SI 0014 oraz DISP 00AF. Po wciśnięciu przycisku MOV (Buttons for register-memory operations) wartość rejestru AX zostanie przeniesiona do komórki pamięci.

The screenshot shows the Intel 8086 Project simulator interface. On the left, there are several configuration panels with radio buttons. The first panel has 'z rejestru do pamięci' selected. The second panel has 'indeksowy' selected. The third panel has 'SI' selected. The fourth panel has 'BP' and 'BX' unselected. The fifth panel has 'SI BX', 'DI BX', 'SI BP', and 'DI BP' unselected. The sixth panel has 'AX' selected. In the center, there is a table of registers and a set of operation buttons. The 'DISP' register is set to '00AF'. The 'MOV' button is highlighted. At the bottom, the text 'Projekt Intel 8086', 'Paweł Kęska', and 'Nr Indeksu 13075' is displayed.

Register	Value
SI	0000
DI	0000
BP	0000
SP	0000
DISP	00AF

Buttons: Reset, MOV, XCHG, PUSH AX, POP AX, PUSH BX, POP BX, PUSH CX, POP CX, PUSH DX, POP DX.

Projekt Intel 8086
Paweł Kęska
Nr Indeksu 13075

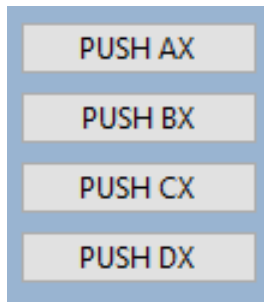
Sprawdzenie – aby sprawdzić czy program dobrze działa można zrobić operację odwrotną tj. rozkaz MOV z pamięci do rejestru. W tym celu zerujemy rejestr AX oraz zaznaczamy przycisk From memory to register.

Rozkazy XCHG dla rejestrów oraz pamięci działają analogicznie. Jedyna różnica jest taka że nie potrzebny jest wybór przycisku Operation Mode, zaznaczenie go nie wpływa na wynik rozkazu. W przypadku rozkazów na komórkach pamięci które nie zostały jeszcze zapisane wartości pamięci będą miały wartość „0000”, a w przypadku próby rozkazu na adresie większym niż liczba miejsc w pamięci rozkaz wykona się na ostatnich dwóch komórkach pamięci.

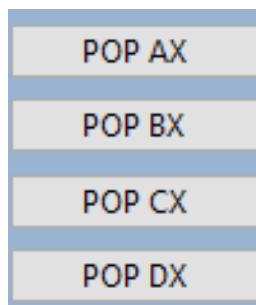
D) Stos oraz rozkazy PUSH i POP

Symulator umożliwia operacje na stosie.

Rozkaz PUSH umożliwia przeniesienie wartości wybranego rejestru do stosu. Po wykonaniu takiego rozkazu wskaźnik stosu (SP) rośnie o 2.



Rozkaz Pop umożliwia przeniesienie wartości z stosu do danego rejestru, w związku z tym iż jest to stos wartość którą można przenieść z stosu jest ostatnią wartością zapisaną w nim. Po wykonaniu takiego rozkazu wskaźnik stosu (SP) maleje o 2.



Wskaźnik stosu pokazuje obecną wartość w której jest najwyższa zapisana komórka w stosie. Wartość tą można zmienić poprzez wpisanie jakiegokolwiek w systemie HEX.



Przykład – Użycie czterech rozkazów Push AX, PUSH BX, PUSH CX oraz PUSH DX, wyzerowanie rejestrów a następnie użycie rozkazów POP w odwrotnej kolejności.

Rozkazy PUSH

	Reset	
SI	0000	0000
DI	0000	0000
BP	0000	0000
SP	0008	0008
DISP	0000	0000
	MOV	XCHG
	PUSH AX	POP AX
	PUSH BX	POP BX
	PUSH CX	POP CX
	PUSH DX	POP DX

Rozkazy POP w odwrotnej kolejności aby rejestry były ułożone tak jak początkowo.

	Reset	Random			Reset	
AX	0000	4B00	<input type="radio"/> z rejestru do pamięci <input type="radio"/> z pamięci do rejestru		SI	0000
BX	057B	057B			DI	0000
CX	5014	5014	<input type="radio"/> indeksowy <input type="radio"/> bazowy <input type="radio"/> indeksowo-bazowy		BP	0000
DX	1A11	1A11			SP	003C
	MOV AX,BX	XCHG AX,BX			DISP	0000
	MOV AX,CX	XCHG AX,CX			MOV	XCHG
	MOV AX,DX	XCHG AX,DX	<input type="radio"/> SI <input type="radio"/> DI		PUSH AX	POP AX
	MOV BX,AX	XCHG BX,AX	<input type="radio"/> BP <input type="radio"/> BX		PUSH BX	POP BX
	MOV BX,CX	XCHG BX,CX			PUSH CX	POP CX
	MOV BX,DX	XCHG BX,DX		PUSH DX	POP DX	

W przypadku próby użycia rozkazu POP na pustym stosie pokaże się komunikat.

