

Intel 8086

MICHAŁ BRÓZDA, NR ALBUMU 14223

Intel 8086 jest 16-bitowym mikroprocesorem wprowadzonym na rynek w 1978 roku (Intel, AMD, OKI, Siemens). Mikroprocesor został zaprojektowany przez firmę Intel jako następcę 8-bitowego 8080/8085.

Specyfikacja procesora:

- 16-bitowa magistrala danych
- 20-bitowa magistrala adresowa
- częstotliwość sygnału zegarowego do 10MHz
- 91 podstawowych typów rozkazów
- przestrzeń adresowa urządzeń wejścia/wyjścia – 64kB
- 7 trybów adresowania argumentów w pamięci
- dwa tryby pracy – minimalny i maksymalny
- 16-bitowa jednostka arytmetyczno logiczna (ALU)
- 16-bitowe rejestry ogólnego przeznaczenia
- 6-bajtowa kolejka rozkazów

Jego zastosowanie (w szczególności jego późniejszej odmiany z 8-bitowym interfejsem – 8088) w pierwszych ogólnodostępnych komputerach osobistych (IBM PC), doprowadziło do jego wielkiej popularyzacji i dalszego rozwoju tej rodziny procesorów (architektura x86).

W związku z historycznym znaczeniem procesora 8086 firmie Intel przydzielono identyfikator 0x8086 na liście identyfikatorów (PCI ID) dostawców urządzeń dla magistrali PCI



Rozkazy mikroprocesora 8086 są wielobajtowe. Liczba bajtów każdego rozkazu zależy od jego rodzaju i może wynosić od jednego do sześciu.

Pierwszy bajt zawiera sześciobitowy kod operacji oraz dwa bity (kierunku i szerokości). Bit D określa kierunek transmisji (0 – wynik operacji jest przesyłany z rejestru do pamięci, 1 – z pamięci do rejestru). W zależności od wartości tego bitu w rozkazie rozróżniane są operandy źródłowe i operandy przeznaczenia. Bit W określa szerokość operandu danego rozkazu (0 – operacje bajtowe, 1 – operacje na słowie 16-bitowym).

W procesorze możemy wyróżnić wiele rozkazów lecz najczęściej używanymi są:

- MOV – Przeniesienie wartości rejestru
- XCHG – Zamiana wartości rejestru
- PUSH – Zapamiętanie wartości rejestru do pamięci
- POP – Odczytanie wartości z pamięci do rejestru
- ADD - Obliczenie sumy argumentów
- SUB - Obliczenie różnicy argumentów
- DIV - Obliczenie ilorazu
- INC - Zwiększ o 1
- DEC - Zmniejsz o 1

Rejestry arytmetyczne, są to cztery 16-bitowe rejestry ogólnego przeznaczenia: AX, BX, CX, DX. Każdy z tych rejestrów może również działać jako dwa niezależne rejestry 8-bitowe:

AX	AH	AL	Akumulator
BX	BH	BL	Rejestr bazowy
CX	CH	CL	Rejestr zliczający
DX	DH	DL	Rejestr danych

Dla wielu rozkazów szczególną funkcję pełni akumulator AX (lub AL). Rozkazy wykorzystujące rejestr AX (w przeciwieństwie do pozostałych rejestrów), mają zwarty kod, a czas ich wykonywania jest najkrótszy

- **AX – Akumulator (Accumulator)**

Rejestr ten bezpośrednio współpracuje z jednostką arytmetyczno-logiczną.

- **BX – Baza (Base)**

Rejestr ten może być używany do adresowania argumentu, znajdującego się w pamięci, stanowiąc bazę do obliczania adresu.

- **CX – Licznik (Counter)**

Rejestr ten jest używany jako licznik w operacjach łańcuchowych oraz pętlach. Po każdej iteracji jego zawartość jest automatycznie dekrementowana. W rozkazach przesunięć, rejestr CL (mniej znaczący bajt rejestru CX), wykorzystywany jest jako licznik bitów.

- **DX – Dane (Data)**

Rejestr ten jest wykorzystywany w niektórych operacjach arytmetycznych do przechowywania części argumentu lub wyniku operacji (mnożenie i dzielenie 16-bitowe). Zawiera on także adres urządzenia w operacjach wejścia/wyjścia.

Rejestry segmentowe

Mamy 4 rejestry segmentowe, każdy z nich o rozmiarze 16-bitów. Zawierają one kod oraz instrukcje mikroprocesora.

CS Rejestr segmentowy kodu (Code segment register)

Wskazuje segment programu, z którego aktualnie są pobierane kolejne rozkazy do wykonania. Użytkownik nie ma dostępu do modyfikacji tego rejestru

DS Rejestry segmentowy danych (Data segment register)

Wskazuje segment, w którym zapamiętane są zmienne używane w programie.

SS Rejestr segmentowy stosu (Stack segment register)

Wskazuje segment pamięci, w którym jest zdefiniowany stos. Główne operacje to Push oraz Pop.

ES Rejestr segmentowy dodatkowy (Extra segment register)

Wskazuje dodatkowy segment danych. Jest wykorzystywany w przypadku przepełnienia rejestru DS lub w celu kopiowania danych.

Rejestry wskaźnikowe i indeksowe

Możemy rozróżnić 3 rejestry wskaźnikowe (**IP**, **BP**, **SP**) oraz 2 rejestry indeksowe (**SI**, **DI**).

Rejestry wskaźnikowe są stosowane do adresowania danych w obrębie wydzielonego obszaru pamięci zwanego segmentem stosu. Natomiast rejestry indeksowe służą do adresowania danych w obszarze pamięci zwanym segmentem danych.

Wszystkie cztery rejestry określają 16-bitowy adres względem początku danego segmentu (ang. Offset). Ponadto mogą być one wykorzystane jako argumenty większości rozkazów arytmetycznych i logicznych.

Rejestry znaczników F (ang. Flag register)

Jest to konstrukcja składająca się z szeregu jednobitowych rejestrów (wskaźników) określających stan procesora. Rejestr znaczników zawiera tzw. Słowo stanu (status word).

Poszczególne bity rejestru wskaźników dzielą się na dwie grupy:

1. Wskaźniki określające sposób działania procesora:

TF - wykorzystywany przez specjalistyczne oprogramowanie uruchomieniowe

IF - zezwala na przyjmowanie przerw zewnętrznych. Wyzerowanie tego bitu powoduje zablokowanie przyjmowania przerw zewnętrznych.

DF - określa czy zawartości rejestrów SI i DI w trakcie wykonywania rozkazów łańcuchowych mają być zwiększane (DF=0), czy zmniejszane (DF=1).

2. Wskaźniki ustawiane w wyniku wykonywania różnych rozkazów:

CF - wskaźnik przeniesienia globalnego, zmieniany rozkazami arytmetycznymi i przesunięć, zerowany rozkazami logicznymi.

OF - wskaźnik nadmiaru, zmieniany rozkazami arytmetycznymi i zerowany rozkazami logicznymi; OF jest zmieniany również w rozkazach przesunięć, których drugi argument jest równy 1.

SF - wskaźnik znaku (ujemnego wyniku), zmieniany rozkazami arytmetycznymi i logicznymi.

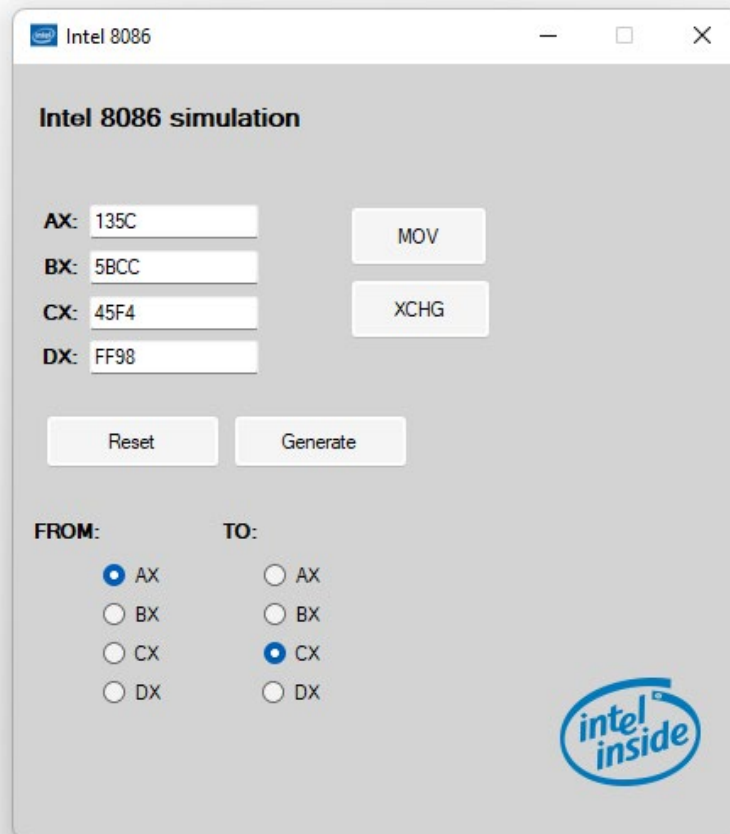
ZF - wskaźnik zera wyniku, zmieniany rozkazami arytmetycznymi i logicznymi.

PF - wskaźnik parzystej liczby jedynek w zapisie binarnym najmniej znaczącego bajtu wyniku, zmieniany rozkazami arytmetycznymi i logicznymi.

AF - wskaźnik przeniesienia pomocniczego ustawiany na potrzeby rozkazów arytmetyki dziesiętnej.

Symulator

Symulator procesora Intel 8086 zbudowany jest w oparciu o Windows FORMS. Zawiera on podstawowe rozkazy procesora takie jak: MOV i XCHG.



The screenshot shows a window titled "Intel 8086" with a standard Windows title bar. The main area is titled "Intel 8086 simulation". It contains four input fields for registers: AX (135C), BX (5BCC), CX (45F4), and DX (FF98). To the right of these fields are two buttons: "MOV" and "XCHG". Below the register fields are two buttons: "Reset" and "Generate". At the bottom, there are two groups of radio buttons labeled "FROM:" and "TO:". Under "FROM:", the "AX" radio button is selected. Under "TO:", the "CX" radio button is selected. In the bottom right corner, there is an "intel inside" logo.

Register	Value
AX	135C
BX	5BCC
CX	45F4
DX	FF98

Buttons: MOV, XCHG, Reset, Generate

FROM: ☒ AX, ☐ BX, ☐ CX, ☐ DX

TO: ☐ AX, ☐ BX, ☒ CX, ☐ DX

intel inside

Do poszczególnych elementów zostały użyte poniższe funkcje:

- Znaczniki wyboru rejestru - <RadioButton>
- Oznaczenia znaczników - <TextBlock>
- Pola wartości rejestrów - <TextBox>
- Przyciski z poleceniami - <Button>

Program zaprojektowany jest tak, aby nie pozwalał wprowadzać użytkownikowi innych wartości poza hexadecymalnymi.

Skrawek kodu zezwalający wyłącznie na wartości hex:

```
public const string HexLetters = "0123456789abcdefABCDEF\b"; // \b is the Backspace
if (!HexLetters.Contains(e.KeyChar)) e.Handled = true;
```

Kod odpowiadający za tzw. Backend symulatora został napisany w języku C#. Przy pomocy zastosowania programowania zdarzeniowego jesteśmy w stanie przypisać do poszczególnych elementów interfejsu odpowiednie funkcje, odpowiedzialne za wszelką funkcjonalność

Przycisk Generate - Zastosowanie generatora wartości w granicach liczb Hexadecymalnych, pozwala na wprowadzenie losowych wartości we wszystkich rejestrach.

Przycisk Reset - Zerowanie wartości rejestrów, rozwiązane poprzez przypisanie do pól pustych wartości typu string.

Wybieranie znaczników rejestrów (FROM, TO) - Zastosowanie zdarzenia .Checked przy pomocy przycisków typu RadioButton.

Przycisk MOVE - Na podstawie zdarzenia .Checked oraz porównania go do Prawy lub Fałszu zostaje wykonane przeniesienie wartości do wybranego przez użytkownika rejestru. Całość bazuje na strukturze if oraz else if w celu krokowego sprawdzania, które rejestry zostały zaznaczone.

Przycisk XCHG - Funkcjonalność rozwiązana w sposób bardzo zbliżony do komendy MOVE. Różnicą jest efekt końcowy, który zamienia wartości rejestrów wybranych przez użytkownika zamiast je nadpisywać.

*Aby skorzystać z przycisku MOV lub XCHG należy najpierw zaznaczyć wartości z kolumn „FROM” oraz „TO”. Bez tego akcja nie zostanie wykonana.