

# Architektura Systemów Komputerowych

## Intel 8086

Lab4/1/ISN  
Mateusz Dynur  
Nr. 13636

# Informacje o procesorze

**Procesor Intel 8086** - 16-bitowy mikroprocesor wprowadzony na rynek 8 czerwca 1978 roku. Mikroprocesor został zaprojektowany przez firmę Intel w technologii 3  $\mu\text{m}$  HMOS (ang. *High performance MOS*, później także HMOS-II, HMOS-III i CHMOS) jako rozszerzenie 8-bitowego 8080/8085. Wykonywany głównie w obudowach 40pin DIP, także jako 44-pin PLCC i 56-pin QFP (ang. *Quad Flat Package*). Wciąż jeszcze produkowany przez różnych dostawców.

## Podstawowe parametry mikroprocesora 8086:

- architektura CISC,
- przestrzeń adresowa pamięci – 1 MB w trybie rzeczywistym,
- 16-bitowa magistrala danych,
- 20-bitowa magistrala adresowa,
- częstotliwość sygnału zegarowego do 10 MHz,
- 91 podstawowych typów rozkazów,
- przestrzeń adresowa urządzeń wejścia/wyjścia – 64 kB,
- możliwość wykonywania operacji bitowych, bajtowych, o długości słowa i łańcuchowych,
- 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.

Przy projektowaniu mikroprocesora 8086, konstruktorzy firmy Intel zastosowali wiele nowych, nie występujących w mikroprocesorach 8-bitowych, rozwiązań:

- rozszerzenie możliwości adresowanie operandów
- wprowadzenie segmentacji obszaru pamięci
- mechanizmy przyspieszenia pracy
- mechanizmy dla pracy wieloprocessorowej

# Rozkazy wykonywane przez symulator

**MOV** - przeznaczony do przesyłania słów lub bajtów w operacjach typu rejestr-pamięć z (do) dowolnych rejestrów. Rozkaz ten może też przysyłać do pamięci dane określone w trybie prostym bez pośrednictwa rejestru.

```
const mov = () => {  
  setValue({  
    [radioOn]: JSON.parse(localStorage.getItem(radioWith)),  
  });  
  localStorage.setItem(  
    [radioOn],  
    JSON.stringify(JSON.parse(localStorage.getItem(radioWith)))  
  );  
  window.location.reload(false);  
};
```

Przed:

Simulacja procesora INTEL 8086

AX:	<input type="text" value="1"/>	<div>MOV</div> <div>XCHG</div> <div>RESET</div>
BX:	<input type="text" value="2"/>	
CX:	<input type="text" value="0000"/>	
DX:	<input type="text" value="0000"/>	
<div>Z    NA</div> <div>AX: <input checked="" type="radio"/> AX: <input type="radio"/></div> <div>BX: <input type="radio"/> BX: <input checked="" type="radio"/></div> <div>CX: <input type="radio"/> CX: <input type="radio"/></div> <div>DX: <input type="radio"/> DX: <input type="radio"/></div>		

© Mateusz Dymus

Po:

Simulacja procesora INTEL 8086

AX:	<input type="text" value="1"/>	<div>MOV</div> <div>XCHG</div> <div>RESET</div>
BX:	<input type="text" value="1"/>	
CX:	<input type="text" value="0000"/>	
DX:	<input type="text" value="0000"/>	
<div>Z    NA</div> <div>AX: <input type="radio"/> AX: <input type="radio"/></div> <div>BX: <input type="radio"/> BX: <input type="radio"/></div> <div>CX: <input type="radio"/> CX: <input type="radio"/></div> <div>DX: <input type="radio"/> DX: <input type="radio"/></div>		

© Mateusz Dymus

**XCHG** - służy do wymiany zawartości rejestrów i pamięci.  
Argumentem rozkazu nie mogą być rejestry segmentowe.

```
const xchg = () => {  
  let tab = [  
    JSON.parse(localStorage.getItem('radioOn')),  
    JSON.parse(localStorage.getItem('radioWith')),  
  ];  
  console.log(tab);  
  setValues({  
    [radioWith]: JSON.parse(localStorage.getItem('radioOn')),  
    [radioOn]: JSON.parse(localStorage.getItem('radioWith')),  
  });  
  localStorage.setItem([radioWith], JSON.stringify(tab[0]));  
  localStorage.setItem([radioOn], JSON.stringify(tab[1]));  
  window.location.reload(false);  
};
```

Przed:

Symulacja procesora INTEL 8086

AX:	<input type="text" value="1"/>	<div>MOV</div> <div>XCHG</div> <div>RESET</div>
BX:	<input type="text" value="2"/>	
CX:	<input type="text" value="0000"/>	
DX:	<input type="text" value="0000"/>	
<div>Z    NA</div> <div>AX: <input type="radio"/> AX: <input type="radio"/></div> <div>BX: <input type="radio"/> BX: <input type="radio"/></div> <div>CX: <input type="radio"/> CX: <input type="radio"/></div> <div>DX: <input type="radio"/> DX: <input type="radio"/></div>		

© Mateusz Dymur

Po:

Symulacja procesora INTEL 8086

AX:	<input type="text" value="2"/>	<div>MOV</div> <div>XCHG</div> <div>RESET</div>
BX:	<input type="text" value="1"/>	
CX:	<input type="text" value="0000"/>	
DX:	<input type="text" value="0000"/>	
<div>Z    NA</div> <div>AX: <input type="radio"/> AX: <input type="radio"/></div> <div>BX: <input type="radio"/> BX: <input type="radio"/></div> <div>CX: <input type="radio"/> CX: <input type="radio"/></div> <div>DX: <input type="radio"/> DX: <input type="radio"/></div>		

© Mateusz Dymur

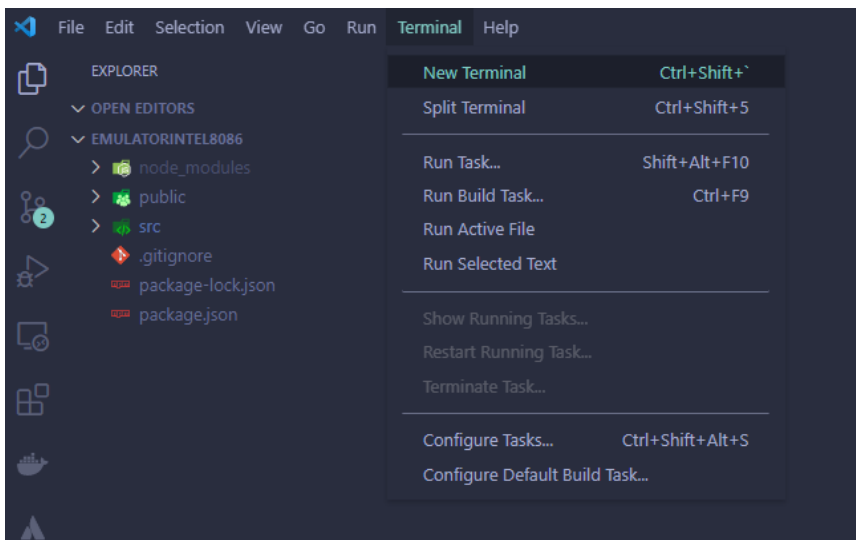
# Informacje o symulatorze

Symulator został stworzony w języku JavaScript wykorzystując bibliotekę ReactJS.

Do uruchomienia symulatora potrzebny będzie nam:

- Node16.14+
- VSCode lub inny edytor kodu

1. Po pobraniu pliku otwieramy terminal.



2. Wpisujemy „npm i” lub „npm install”, rozpocznie się wtedy instalacja biblioteki i niezbędnych narzędzi do uruchomienia aplikacji.

3. Po zakończeniu instalacji wpisujemy w terminal „npm start”, po wykonaniu polecenia, naszym oczom powinien ukazać się symulator .

