

Architektura Systemów Komputerowych

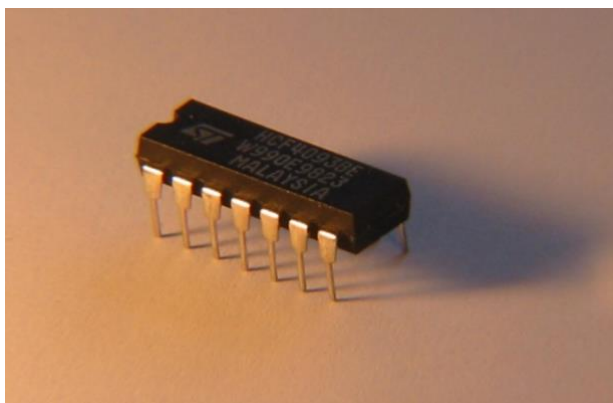
Intel 8086

Lab4/1/ISN
Dawid Foltyński
Nr. 13643

Podstawowe informacje

Procesor Intel 8086 – 16-bitowy mikroprocesor wprowadzony na rynek 8 czerwca 1978 roku. Wykonany przez Intel’a w technologii 3 nanometrowej, wykonywany głównie w obudowach 40-pin DIP (Dual in-line package), 44-pin PLCC (Plastic Leaded Chip Carrier) i 56-pin QFP (Quad Flat Package). Jego zastosowanie w pierwszych ogólnodostępnych komputerach osobistych doprowadziło do jego wielkiej popularyzacji i dalszego rozwoju tej grupy procesorów (x86). Jako ciekawostka przez jego historyczne znaczenie firmie Intel przyznano identyfikator 0x8086 na liście identyfikatorów (PCI ID) dostawców urządzeń dla magistrali PCI.

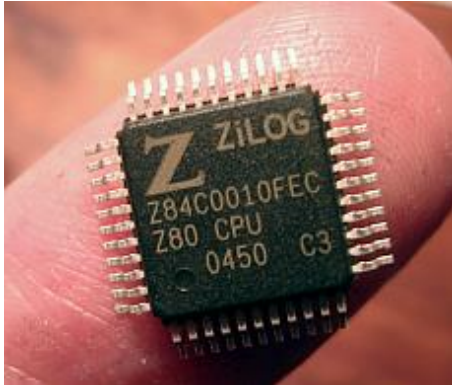
Obudowa DIP:



Obudowa PLCC:



Obudowa QFP:



Parametry mikroprocesora 8086

- Architektura CISC (Complex Instruction Set Computing) – Istotą architektury CISC jest to, iż pojedynczy rozkaz mikroprocesora wykonuje kilka operacji niskiego poziomu, np. pobranie z pamięci, operacje arytmetyczne i zapisanie do pamięci.
- Przestrzeń adresowa pamięci – 1 MB w trybie rzeczywistym. Tryb rzeczywisty nie zapewnia ochrony pamięci przed użyciem przez inny proces oraz obsługi wielozadaniowości
- 16-bitowa magistrala danych
- 20-bitowa magistrala adresowa
- Częstotliwość sygnału zegarowego do 10 MHz
- 91 podstawowych typów rozkazów
- Przestrzeń adresowa urządzeń I/O – 64 kB
- Możliwość wykonywania operacji bitowych, bajtowych, o długości słowa i łańcuchowych
- 7 trybów adresowania argumentów w pamięci
- Tryb pracy – minimalny i maksymalny
- 16-bitowa jednostka arytmetyczno-logiczna (ALU) – możliwość operacji arytmetycznych, operacji logiczne i operacje jednoargumentowe
- 16-bitowe rejestry ogólnego przeznaczenia
- 6-bajtowa (48-bitowa) kolejka rozkazów

Operacje wykonywane przez symulator

MOV – przesyła bajty lub słowa w operacjach rejestr-pamięć. Oznaczamy skąd i dokąd dana wartość ma być przesłana

Implementacja:

```
mov(src: FormControl | any, dest: FormControl | any) {  
  let source = src.value;  
  let destination = dest.value;  
  
  this.registryData  
    .get(destination)  
    ?. setValue(this.registryData.get(source)?.value);  
  
  this.registryData.get(destination)?.updateValueAndValidity();  
}
```

Przed

Emulator Intel 8086

AX	BX	CX	DX
2F2F2F	3F3F3F	FFFFFF	FFFFFF

Wybierz pola, na których chcesz wykonać operację

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

SHOW

MOV

XCHG

Po

Emulator Intel 8086

AX
2F2F2F

BX
2F2F2F

CX
FFFFFF

DX
FFFFFF

Wybierz pola, na których chcesz wykonać operację

OD
☒ AX
☐ BX
☐ CX
☐ DX

DO
☐ AX
☒ BX
☐ CX
☐ DX

SHOW

MOV

XCHG

XCHG – wymienia wartości rejestrów i pamięci.

Implementacja

```
xchg(src: FormControl | any, dest: FormControl | any) {  
  let source = src.value;  
  let destination = dest.value;  
  
  let values = [  
    this.registryData.get(source)?.value,  
    this.registryData.get(destination)?.value,  
  ];  
  
  this.registryData.get(destination)?.setValue(values[0]);  
  
  this.registryData.get(source)?.setValue(values[1]);  
  
  this.registryData.get(destination)?.updateValueAndValidity();  
  this.registryData.get(source)?.updateValueAndValidity();  
}
```

Przed

Emulator Intel 8086

AX

123321

BX

2F2F2F

CX

FFFFFF

DX

FFFFFF

Wybierz pola, na których chcesz wykonać operację

OD

☒ AX

☐ BX

☐ CX

☐ DX

DO

☐ AX

☒ BX

☐ CX

☐ DX

SHOW

MOV

XCHG

Po

Emulator Intel 8086

AX

2F2F2F

BX

123321

CX

FFFFFF

DX

FFFFFF

Wybierz pola, na których chcesz wykonać operację

OD

☒ AX

☐ BX

☐ CX

☐ DX

DO

☐ AX

☒ BX

☐ CX

☐ DX

SHOW

CLEAR MEMORY

MOV

XCHG

Jak uruchomić

Potrzebujemy node v16.10.0+ i edytor tekstu

Komenda do pobrania i uruchomienia projektu: git clone

<https://github.com/dfoltynski/architektura.git> && cd architektura && npm i && npm start