

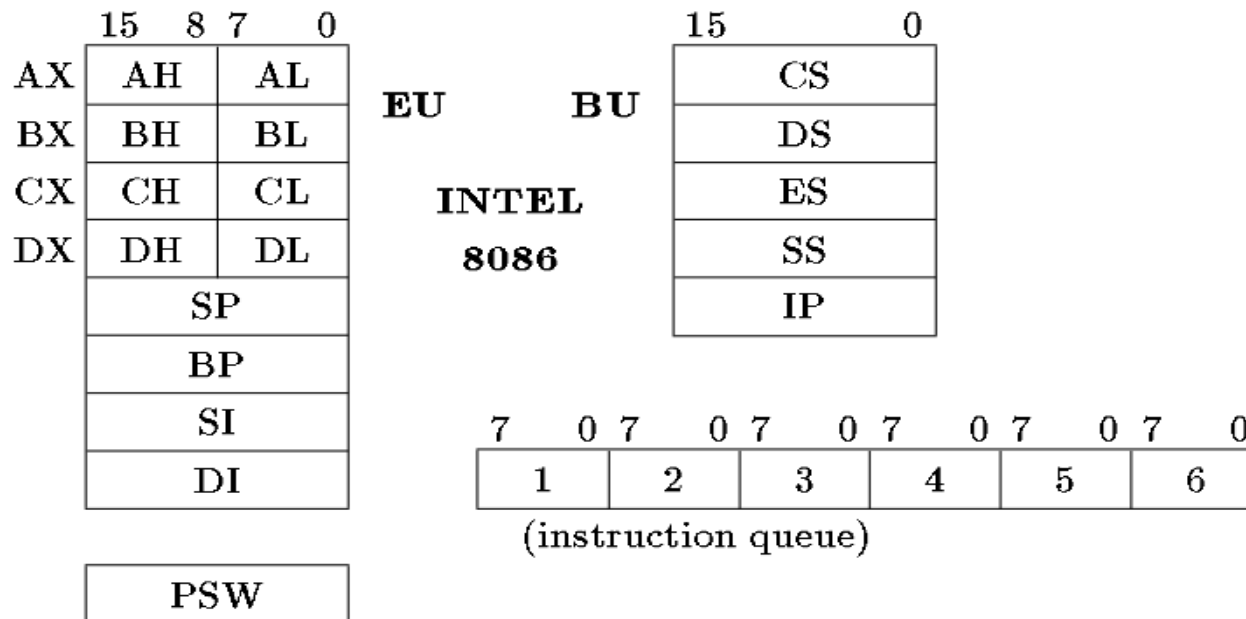
Precesor Intel 8086

Uvod

- Procesor 8086 jedan je od najvažnijih u istoriji računarstva, rodonačelnik Intelove 80x86 familije kojoj pripadaju: 8086, 80286, 80386, 80486, Pentium, Pentium II itd.
- Familija – kompatibilni procesori, mogućnost zadržavanja programa pri prelasku na novi računar

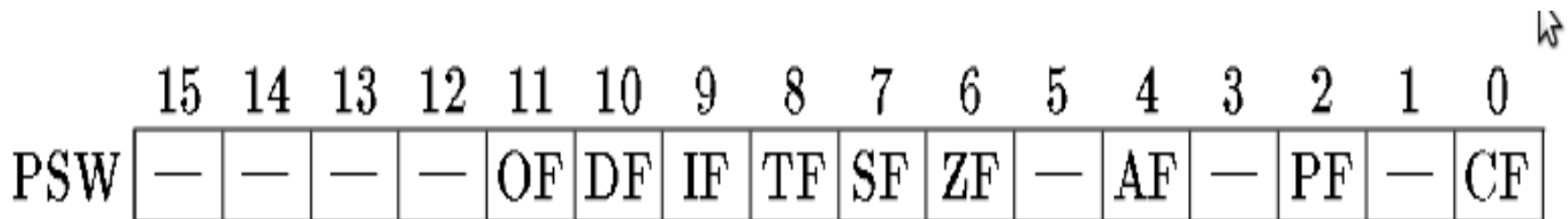
Osnovna arhitektura

- Dva glavna dijela 8086 procesora
 - Interfejs prema magistrali koji čita podatke i naredbe iz memorije
 - Izvršna jedinica koja dekodira i izvršava naredbe



Registri procesora 8086

- Oznake AX=AH:AL, slično za BX, CX, DX
- Registri opšte namjene su AX, AH, AL, BX, BH, BL, CX, CH, CL, DX, DH, DL, SP, BP, SI, DI
- Registar flagova, ProgramStatusWord



Memorija

- Magistrala ima 20 bita, može se adresirati do 2^{20} adresa koje sadrže 8 bita, što znači da je memorija kapaciteta 1MB, osnovne jedinice za rad sa memorijom su bajt i riječ (16 bita)
- Adresa (ukupno 16 bita) dijeli se na dva dijela
 - Adresa segmenta
 - Adresa bajta u segmentu (pomjeraj)
 - U heksadekadnoj notaciji adresa je SSSS:OOOO, na primjer 0040:0002 je adresa trećeg bajta u segmentu 40_{hex}
 - Adresa od 20 bita dobija se množenjem adrese segmenta sa 10_{hex} i dodavanjem pomjeraja

Registri

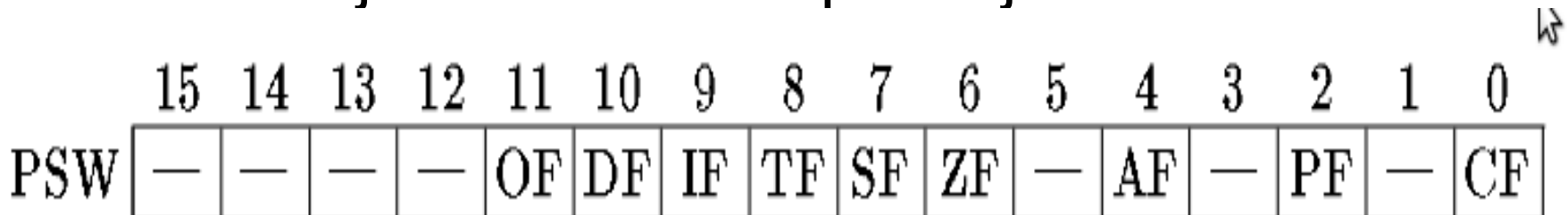
- Procesor 8086 sadrži 14 registara
 - Programski brojač, IP, pokazuje na narednu instrukciju koju procesor treba da izvrši
 - Statusni registar
 - Registri opšte namjene, 4 registra
 - Pokazivački registri, 2 registra
 - Indeksni registri, 2 registra
 - Segmentni registri, 4 registra

Statusni registar

- PSW je 16-bitni registar koji sadrži informacije o rezultatu posljednje izvršene instrukcije
- Flegovi
 - **Overflow**, prekoračenje kod aritmetičkih operacija
 - **Direction**, za kretanje unutar stringova, 0 ka višim, 1 ka nižim adresama
 - **Interrupt enable**, reagovanje na spoljašnje prekide
 - **Trap**, izvršavanje programa instrukciju po instrukciju, poslije svake instrukcije generiše prekid INT 3

Statusni registar

- Flegovi
 - **Sign**, znak rezultata prethodne instrukcije
 - **Zero**, rezultat posljednje operacije je nula
 - **Auxiliary Carry**, bit prenosa u BCD aritmetici
 - **Parity**, broj jedinica u rezultatu prethodne instrukcije je paran
 - **Carry**, bit prenosa sa najvišeg mjesta poslije izvršavanja aritmetičkih operacija



Registri opšte namjene

- Mogu se posmatrati kao 16-bitni ili kao par 8-bitnih registara, npr. $AX=AH:AL$
- AX i AL često se nazivaju akumulatorom, predstavlja podrazumijevani argument nekih instrukcija
- BX, Base Register, bazna adresa prilikom pristupa memoriji
- CX, Counter Register, čuva trenutnu vrijednost brojača, implicitno se umanjuje za rad sa petljama
- DX, implicitno se koristi za množenje i dijeljenje, kao i za adresu porta prilikom ulazno-izlaznih instrukcija

Pokazivački registri

- Pristup podacima sa stek segmenta, dodatno je dozvoljena upotreba u aritmetičko-logičkim operacijama
 - SP, Stack Pointer, pokazivač na vrh steka, prilikom formiranja apsolutne adrese uz njega se podrazumijeva SS (Stack Segment) registar
 - BP, Base Pointer, pristup podacima sa steka, ali bez izmjene steka

Indeksni registri

- Koriste se implicitno prilikom rada sa stringovima, dodatno se mogu koristiti kao registri opšte namjene u aritmetičko-logičkim operacijama
 - SI, Source Index, indeks unutar izvornog stringa, mijenja se implicitno u zavisnosti od smjera, fleg D
 - DI, Destination Index, indeks unutar odredišnog stringa, mijenja se implicitno u zavisnosti od smjera, fleg D

Segmentni registri

- Omogućavaju segmentnu organizaciju memorije, svaki registar definiše blok od 64KB memorije
 - CS (Code Segment) čuva adresu bloka memorije u kojem se nalazi izvršni kod
 - DS (Data Segment) pokazuje na blok memorije za smještanje podataka
 - SS (Stack Segment) pokazuje na segment steka
 - ES (Extra Segment) dodatni segment podataka, koristi se za rad sa stringovima

Metode adresiranja

- Načini adresiranja
 - Neposredno adresiranje
 - Registarsko adresiranje
 - Registarsko indirektno adresiranje
 - Direktno adresiranje
 - Indeksirano adresiranje
 - Bazno adresiranje
 - Bazno indeksirano adresiranje
 - Bazno indeksirano relativno adresiranje

Neposredno adresiranje

- Neposredna dodjela konstante, na primjer:
 mov ah, 34h
 mov ax, 4563h
- Za 16-bitne konstante, u memoriju se prvo smješta njen niži, a zatim viši bajt

Registarsko adresiranje

- Svodi se na preuzimanje sadržaja registra i njegovo smještanje u odredište, na primjer:

mov ax, bx

mov si, dx

mov al, dl

mov ds, ax

mov ax, cs

Registarsko indirektno adresiranje

- Sadržaj registra, navedenog između uglastih zagrada, tumači se kao adresa na kojoj se nalaze podaci, prilikom upotrebe BX, SI, DI podrazumijeva se DS, za BP koristi se SS na primjer:

```
mov al, [bx]
```

```
mov al, [bp]
```

```
mov al, cs:[bx]
```


Direktno adresiranje

- Adresa na kojoj se nalaze podaci zadaje se direktno u uglastim zagradama, adresa je 16-bitna, podrazumijeva se DS registar, na primjer:

`mov al, [0234]h` (*smisao: $al \leftarrow c(ds:234)$*)

`mov [0045h], ax`

`mov al, cs:[3452h]`

Indeksirano adresiranje

- Efektivna adresa dobija se sabiranjem registra i navedene konstante, konstanta je u principu početna adresa bloka kome se pristupa, koristi se DS, na primjer:

`mov al, 20h[si]` (*ili: `mov al, [si+20h]`*)

`mov dl, 78[di]`

`mov bh, ss:[di+120]`

Bazno adresiranje

- Bazni registar je početak bloka, a konstanta relativni položaj elementa kome pristupamo, na primjer:

`mov al, 20h[bx]` (*ili: `mov al, [bx+20h]`*)

`mov dl, 120[bp]`

`mov bh, cs:[bp+120]`

Bazno indeksirano adresiranje

- Podvarijanta indeksiranog adresiranja, ulogu konstante preuzima bazni registar, na primjer:

`mov al, ds:[bp][si]` (*ili* `mov al, ds:[bp+si]`)

`mov dl, es:[bx][di]`

Bazno indeksirano relativno dresiranje

- Pored baznog i indeksnog registra, koristi se konstanta pomjeraja, na primjer:

`mov al, ds:25h[bp][si]` (*ili* `mov al, ds:[bp+si+25h]`)

`mov dl, es:120[bx][di]`

Instrukcije

- Instrukcije za prenos podataka
- Aritmetičke instrukcije
- Logičke instrukcije
- Instrukcije pomjeranja i rutiranja
- Upravljačke instrukcije
- Instrukcije za rad sa nizovima
- Instrukcije za rad sa status registrom
- Razne instrukcije