

Registri

U centralnom procesoru postoji više različitih registara koji se mogu klasifikovati prema različitim kriterijumima: nameni, vrsti podataka koju mogu da sadrže, mogućnosti pristupa (da li su vidljivi samo programima operativnog sistema ili i korisničkim programima), itd. Najčešće korišćen kriterijum klasifikacije je *namena registara*. Registri centralnog procesora se prema nameni mogu podeliti na:

- Registre opšte namene (još se nazivaju i opšti registri)
- Registre specijalizovane namene (još se nazivaju i specijalizovani registri)

Registri opšte namene

Registri opšte namene se koriste za različite funkcije, npr za čuvanje različitih vrsta podataka, adresa i kodova generisanih pri izvršavanju programa. Korisnički program može da im pristupi, čita i menja njihov sadržaj bez ograničenja. Procesori jednostavnije konstrukcije su imali jedan opšti registar nazvan *akumulator*. Savremeni mikroprocesori sadrže veći broj registara opšte namene. U zavisnosti od dizajna procesora i odgovarajućeg mašinskog jezika, opšti registri se mogu dodatno podeliti prema nameni, npr. na:

- Akumulatore – koriste se za aritmetičko-logičke operacije i čuvanje znakovnih podataka
- Indeks registre – koriste se za indeksno adresiranje¹
- Pokazivače segmenata – sadrže početnu adresu segmenta na računarima sa segmentnim adresiranjem
- Pokazivače steka – pokazuju na početak memorije koja je određena za stek

Podela registara opšte namene nije striktna na svim procesorima. Na nekima je strogo određeno koji opšti registri mogu da sadrže podatke a koji adrese, dok na drugim programer ili operativni sistem određuju podelu.

Veličina opštih registara zavisi od arhitekture procesora. Opšti registri procesora procesora sa 64-bitnom arhitekturom imaju 64 bita. Opšti registri procesora sa 32-bitnom arhitekturom imaju

¹ U ovom načinu adresiranja referentna adresa se nalazi u adresnom delu instrukcije, a udaljenje u indeks registru. Stvarna adresa kod ovog načina adresiranja se dobija sabiranjem ovih vrednosti: $SA=A+(X)$. Naime, ovaj način adresiranja se koristi za izvođenje iteracije. Na primer, neka je u memoriju upisan niz elemenata i neka nad svakim od njih treba izvršiti neku operaciju. Na početku iteracije adresa prvog elementa u nizu se upiše u A, dok se u indeks registar upiše 0. Do ostalih elemenata niza se dolazi povećavanjem vrednosti indeks registra.

32 bita, i u slučaju potrebe, dva susedna registra se koriste kao registar od 64 bita. U opštem slučaju, registar opšte namene mora biti dovoljno veliki da može da prihvati celu reč koja se obrađuje u procesoru, odnosno dovoljno veliki da prihvati kompletnu adresu ukoliko se koristi za adresiranje.

Specijalizovani registri

U svakom procesoru postoji velik broj registara specijalizovane namene koji se koriste pri izvršavanju operacija, njihovoj kontroli, kao i za prikaz tekućeg stanja procesora i programa koji se izvršava. Većina ovih registara nije direktno dostupna korisničkim programima. Neke od njih koriste samo programi operativnog sistema, dok su drugi dostupni samo na nivou mikrokoda. Neki od specijalizovanih registara su:

- Instrukcioni registar (IR) – sadrži poslednju pročitane instrukciju
- Registar memorijskih adresa (RMA) – sadrži adresu u memoriji iz koje treba preneti sadržaj u PRM ili u koju treba upisati sadržaj iz PRM
- Prihvatni registar memorije (PRM) – sadrži reč koja je poslednja pročitana iz memorije, ili koju treba upisati u memoriju na adresu koja je sadržana u RMA
- Brojač instrukcija (PC) – sadrži adresu naredne instrukcije koju treba preneti u procesor
- Registar koji sadrži status programa koji se trenutno izvršava (PSW, eng. Program Status Word)
- Kontrolni registri – sadrže specifične informacije za svaku mašinu: tekuće stanje mašine, vektor prekida, rad u posebnom režimu izvršavanja instrukcija

Na nekim procesorima postoji specijalizovan registar koji se koristi kao pokazivač steka. Time se postiže smanjenje veličine instrukcija koje rade sa stekom jer u njima nema potrebe za eksplicitnim navođenjem ovog registra. Sa druge strane, postojanje takvih registara ograničava fleksibilnost programera.

Mikroprocesor Intel 8086

Intelov procesor 8086 spada u grupu šesnaestobitnih mikroprocesora. Nakon povezivanja sa IBM-om, ugradnjom njegovih procesora u prve PC računare Intel postaje najpopularniji proizvođač mikroprocesora na svetu i tu titulu je zadržao sve do danas. Ipak, teško je reći da su prvi Intel-ovi procesori zaslužili ovu titulu. Najveći nedostaci Intel-ovih šesnaestobitnih mikroprocesora 8086 i 80286 su adresiranje memorije po segmentima i korišćenje istih linija za adresnu magistralu i magistralu podataka. Naime, adresna magistrala je imala četiri odnosno osam dodatnih linija, tako da su ovi procesori mogli da adresiraju 1Mb odnosno 16Mb unutrašnje memorije i to sve po segmentima od po 64K. Od poboljšanja u odnosu na osmобitne procesore, šesnaestobitni donose:

- Rad sa virtuelnom memorijom²
- Multiprogramski režim rada³
- Rad tekućih linija (pipeline)⁴

Registri

Mikroprocesor i8086 ima 14 šesnaestobitnih registara, i to 12 registara za podatke i adrese, jedan pokazivač instrukcija (koji odgovara brojaču naredbi kod nekih drugih mikroprocesora), i jedan statusni registar.

8086 REGISTER ORGANIZATION																																								
<table><tr><td>ES</td><td colspan="2">Extra Segment</td></tr><tr><td>CS</td><td colspan="2">Code Segment</td></tr><tr><td>SS</td><td colspan="2">Stack Segment</td></tr><tr><td>DS</td><td colspan="2">Data Segment</td></tr><tr><td>IP</td><td colspan="2">Instruction Pointer</td></tr></table>			ES	Extra Segment		CS	Code Segment		SS	Stack Segment		DS	Data Segment		IP	Instruction Pointer		<table><tr><th>Type</th><th>Register size</th><th>Name of the Register</th></tr><tr><td rowspan="2">General purpose registers</td><td>16 bit</td><td>AX, BX, CX, DX</td></tr><tr><td>8 bit</td><td>AL, AH, BL, BH, CL, CH, DL, DH</td></tr><tr><td rowspan="2">Pointer registers</td><td>16 bit</td><td>SP, BP</td></tr><tr><td>16 bit</td><td>SI, DI</td></tr><tr><td rowspan="2">Instruction Pointer</td><td>16 bit</td><td>IP</td></tr><tr><td>16 bit</td><td>CS, DS, SS, ES</td></tr><tr><td>Flags</td><td>16 bit</td><td>Flag register</td></tr></table>		Type	Register size	Name of the Register	General purpose registers	16 bit	AX, BX, CX, DX	8 bit	AL, AH, BL, BH, CL, CH, DL, DH	Pointer registers	16 bit	SP, BP	16 bit	SI, DI	Instruction Pointer	16 bit	IP	16 bit	CS, DS, SS, ES	Flags	16 bit	Flag register
ES	Extra Segment																																							
CS	Code Segment																																							
SS	Stack Segment																																							
DS	Data Segment																																							
IP	Instruction Pointer																																							
Type	Register size	Name of the Register																																						
General purpose registers	16 bit	AX, BX, CX, DX																																						
	8 bit	AL, AH, BL, BH, CL, CH, DL, DH																																						
Pointer registers	16 bit	SP, BP																																						
	16 bit	SI, DI																																						
Instruction Pointer	16 bit	IP																																						
	16 bit	CS, DS, SS, ES																																						
Flags	16 bit	Flag register																																						
AX	AH	AL	Accumulator																																					
BX	BH	BL	Base Register																																					
CX	CH	CL	Count Register																																					
DX	DH	DL	Data Register																																					
SP			Stack Pointer																																					
BP			Base Pointer																																					
SI			Source Index																																					
DI			Destination Index																																					
FLAGS																																								

² Virtuelna memorija je način organizovanja unutrašnje memorije uz povećanje njenog kapaciteta na račun spoljašnje memorije, najčešće hard diska, čiji bajtovi su znatno jeftiniji ali i sporiji.

³ Multiprogramski režim rada se ostvaruje deljenjem vremena procesora.

⁴ Termin tekuća linija je pozajmljen iz industrijske proizvodnje gde svi radnici rade istovremeno na zajedničkom poslu u jednoj proizvodnoj liniji. Ova ideja je iskorišćena i za ubrzanje rada procesora, gde dok se jedna instrukcija donosi druga se izvršava itd

Segmentni registri

Kod ovog tipičnog predstavnika prvih Intel-ovih procesora za oformljenje memorijske adrese potrebna su dva broja: broj segmenta i dodatak. Razlog je već pomenuta podela memorije po segmentima. Jedan segment je veličine 64K. Mikroprocesor i8086 može u isto vreme da radi sa 4 segmenta od kojih svaki ima unapred propisanu ulogu. To su:

- **CS (Code Segment)** – Kodni segment u kome se zapisuju instrukcije
- **DS (Data Segment)** – Segment podataka namenjen podacima
- **SS (Stack Segment)** – Stek segment koji se koristi prilikom rada sa stekom
- **ES (Extra Segment)** – Ekstra segment ima sličnu namenu kao i segment podataka

Svakom segmentu je pridružen jedan registar, i u njemu se čuva početna adresa segmenta koji zajedno sa dodatkom, koji se nalazi u pokazivaču instrukcija (Instruction Pointer) ako se traži sledeća instrukcija, ili u nekom od pokazivačkih i indeksnih registara ako se traži podatak, daje punu memorijsku adresu. **Puna adresa** je dvadesetobitna i dobija se sabiranjem šesnaestobitnog dodatka i sadržaja segmentnog registra pomnoženog sa 16.

Primer:

Formiranje efektivne adrese iz dva registra. Ako želimo da pristupimo memoriji na fizičkoj adresi 12345h, onda setujemo DS=1230h i SI=0045h. CPU računa fizičku adresu tako što pomnoži segmentni registar sa 16 (10h hekso-dekadno), a zatim dodajući vrednost adrese registra opšte namene:

$$(1230h * 10h + 45h = 12345h):$$

$$\begin{array}{r} 12300 \\ + 0045 \\ \hline 12345 \end{array}$$

Registri podataka / registri opšte namene

Registri podataka se mogu tretirati kao 4 šesnaestobitna ili 8 osmobiitnih registara zavisno od toga da li se operiše sa rečima ili bajtovima. Šesnaestobitni su označeni sa AX, BX, CX i DX i u svakom razlikujemo viši (High) i niži (Low) bajt. Pošto se registri nalaze unutar procesora, oni predstavljaju najbržu memoriju u računaru, ali ih ima svega nekoliko.

- **AX** – akumulator se koristi prilikom množenja reči, deljenja, ulazno izlaznih operacija i nekih operacija sa azbučnim podacima. Sličnu ulogu imaju AL i AH registri kada se operiše sa bajtovima.
- **BX** – bazni registar se koristi u baznom modu adresiranja.
- **CX** – registar brojač služi kao brojač u instrukcijama za opis ciklusa, odnosno prilikom operacija sa azbučnim podacima.
- **DX** – registar podataka ima ulogu pomoćnog akumulatora prilikom množenja reči i prilikom deljenja. Često se koristi i za čuvanje adrese prihvatnog registra prilikom ulaza i izlaza podataka.

Primer:

Podela registara na viši i niži bajt. Ako je npr. AX=0011 0000 0011 1001b, onda je AH=0011 0000b, a AL=0011 1001b. Ukoliko se promeni sadržaj bilo AH, bilo AL, menja se i sadržaj AX.

Pokazivački i indeks registri

Pokazivački i indeksni registri se koriste za rad sa adresama. Da bi se pristupilo podacima u segmentu podataka, adresa segmenta se čuva u registru segmenta podataka, a dodatak u jednom od sledećih registara: BP, SI ili DI.

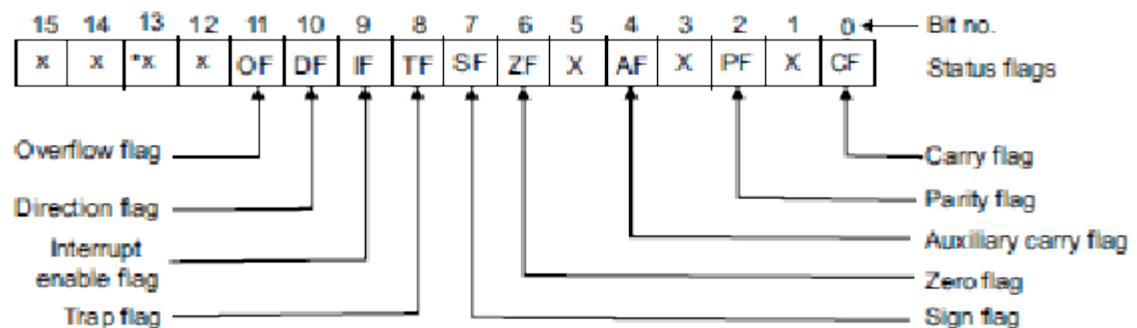
- **BP (Base Pointer)** – se koristi kod većine aritmetičkih operacija prilikom bazno indeksnog adresiranja, ali i kao dopunski pokazivač za rad sa podacima i stekovima.
- **SI (Source Index)** – je indeks registar koji se koristi u operacijama sa nizovima pri čemu ukazuje na lokacije iz koje treba uzeti izvorne podatke pa otuda i naziv izvorni indeks registar.
- **DI (Destination Index)** – ima sličnu ulogu kao i SI s tom razlikom što on ukazuje na lokacije u koje treba upisati rezultat.
- **SP (Stack Pointer)** – je klasični pokazivač steka. Prilikom rada sa stekom, adresa stek-segmenta je u stek-segment registru.

- **IP (Instruction Pointer)** – Pokazivač na sledeću instrukciju. Radi u sprezi sa CS registrom, formirajući adresu sledeće instrukcije.

Statusni registar

Statusni registar sadrži zastavice (flags) koje mogu biti podignute ili spuštene. Ulogu jedne zastavice igra jedna ćelija, odnosno jedan bit. Ako je sadržaj ćelije 1, zastavica je podignuta, u suprotnom je spuštena. Flegovi se koriste u instrukcijama grananja. Naime, često tok programa zavisi od toga da li je dobijeni rezultat nula, pozitivan, ili ima paran broj binarnih jedinica itd:

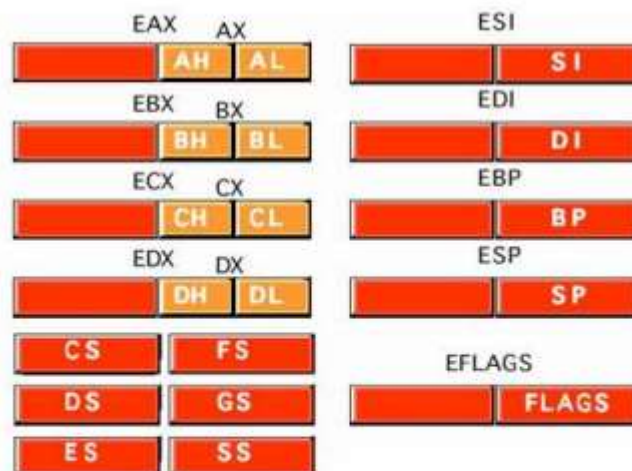
- **CF (Carry Flag)** - postavlja se na 1 ako postoji prenos kod sabiranja, odnosno pozajmica kod oduzimanja. Koristi se i kod operacije množenja, a čuva i šiftovani ili rotirani bit koji izlazi iz nekog registra kao i rezultat operacije upoređivanja.
- **PF (Parity Flag)** – fleg parnosti je 1 ako rezultat operacije sadrži paran broj jedinica.
- **AF (Auxiliary Flag)** – pomoćni fleg prenosa koristi se u radu sa pakovanim decimalnim brojevima.
- **ZF (Zero Flag)** – nula fleg dobija vrednost 1 ako je rezultat operacije nula.
- **SF (Sign Flag)** – fleg znaka je značajan kada se radi sa označenim brojevima. Ako je rezultat negativan SF se postavlja na 1. Ako se ima u vidu da su negativni brojevi zapisani u obliku potpunog komplementa, onda SF uvek sadrži bit najveće težine rezultata.
- **TF (Trap Flag)** – zamka fleg se koristi prilikom izvršavanja instrukcija korak po korak.
- **IF (Interrupt Enable Flag)** – fleg osposobljavanja prekida.
- **DF (Direction Flag)** – fleg pravca se prvenstveno koristi prilikom rada sa nizovima. Ako je DF=1 smanjuje se sadržaj indeks registra.
- **OF (Overflow register)** – fleg prekoračenja signalizira da je prilikom aritmetičke operacije došlo do prekoračenja.



Mikroprocesor Intel 80386

Tridesetdvobitni mikroprocesori su izgrađeni u minijaturnoj tehnologiji sa većim stepenom integracije⁵ tranzistora na čipu. U cilju povećanja brzine rada osnovne memorije, ugrađuju se specijalno brze memorije malog kapaciteta, tzv. keš-memorije. Ova memorija služi kao posrednik između mikroprocesora i osnovne memorije. Frekvencija rada procesora se povećava sa nekoliko MHz na 16, 20, 33 i više MHz.

Najveće ograničenje procesora i8086 bila je maksimalna dužina segmenta od 64KB, jer su svi registri bili 16-bitni, a $2^{16}=65536=64KB$, kao i relativno komplikovan način adresiranja opisan u gornjem tekstu. Intel 80386, kao prvi 32-bitni procesor uvodi novinu pod nazivom „32-bitni zaštićeni mod“, proširujući sve registre na 32 bita, što dozvoljava linearno adresiranje čak do $2^{32}=4GB$ memorije. Svi aktuelni operativni sistemi koriste ovaj mod koji takođe omogućava i rad više programa istovremeno (multitasking) u posebnim, zaštićenim memorijskim particijama. Šema registara mikroprocesora 80386 je prikazana na sledećoj slici:



32 bits: EAX EBX ECX EDX

16 bits: AX BX CX DX

8 bits: AH AL BH BL CH CL DH DL

⁵ VLSI, Very Large Scale Intergration, preko 100 000 tranzistora na čipu.

Registri procesora i80386

Registar AX je proširen na 32 bita. Novi 32-bitni registar se naziva EAX, dok zbog kompatibilnosti AX i dalje referiše na 16-bitni registar (prvih 16 bita EAX). I dalje postoje AL i AH, kao niži i viši bajt AX-a. Registar AX (ili njegova proširena verzija EAX) u opštoj terminologiji se naziva **akumulator** i ima oznaku AC. Ostali prošireni registri opšte namene su EBX, ECX, EDX, ESI i EDI. Pored ovih, i mnogi drugi registri su prošireni, BP postaje EBP, SP postaje ESP, FLAGS postaje EFLAGS i IP postaje EIP. Registar u koji se učitava kompletna instrukcija naziva se IR (Instruction Register). Na većini procesora, IR nije vidljiv programeru.

Segmentni registri su i dalje 16-bitni.

Dodata su dva nova segmentna registra, FS i GS.

Instrukcioni ciklus



Izvršavanje programa, bilo korisničkih ili sistemskih, se sastoji od uzastopnog čitanja i izvršavanja instrukcija. Instrukcioni ciklus se sastoji od faza koje se izvršavaju jedna za drugom. Na početku instrukcionog ciklusa se izvršava *faza dohvatanja*, u kojoj se vrši prenos naredne instrukcije koja se izvršava iz memorije u procesor. U narednoj fazi koja se zove *faza izvršavanja*, izvršava se preneti mašinska instrukcija. Po završetku izvršavanja instrukcije, CPU proverava da li je došlo do pojave prekida. Faza u kojoj se ova provera vrši se naziva *faza prekida*. Zatim se, u slučaju da ne postoji nikakav prekid, ide ponovo na prvu fazu ili se vrši zaustavljanje računara ako nema više instrukcija za izvršavanje. U slučaju da postoji prekid, procesor vrši sledeću akciju:

- Prekida izvršavanje tekućeg programa i čuva njegov vektor stanja. Vektor stanja uključuje adresu naredne instrukcije koja se izvršava, sadržaje registara vezanih za program i ostale relevantne informacije.
- Postavlja vrednost brojača instrukcija na početnu adresu rutine koja vrši obradu prekida (tj. Interrupt handler rutine).

Posle ovoga počinje sa izvršavanjem rutina za obradu prekida. Obično su rutine za obradu prekida deo operativnog sistema; one određuju akciju koja se preduzima u zavisnosti od vrste prekida. Po njihovom završetku se restaurira sačuvani vektor stanja i program nastavlja sa izvršavanjem.

Prekidi

Nekada normalni tok programa mora biti prekinut kako bi se izvršili događaji koji zahtevaju brz odgovor. Hardver računara obezbeđuje mehanizam koji se naziva *prekid* (eng. *Interrupts*) i koji rukuje ovakvim događajima. Na primer, u slučaju da se miš pomeri, hardver miša prekida trenutni program kako bi se obradio događaj pomeranja miša. Prekid dovodi do toga da se postavlja vrednost brojača instrukcija na početnu adresu rutine koja vrši obradu prekida (tj. Interrupt handler rutine).

Svaki Interrupt handler ima svoj broj. Na početku fizičke memorije stoji Interrupt tabela koja sadrži segmentne adrese Interrupt handler-a. Broj Interrupt-a je, u stvari, indeks zapisa u ovoj tabeli. Tabela prekida je prikazana na sledećoj slici.

INT_NUM	Short Description
0x00	Division by zero
0x01	Debugger
0x02	NMI
0x03	Breakpoint
0x04	Overflow
0x05	Bounds
0x06	Invalid Opcode
0x07	Coprocessor not available
0x08	Double fault
0x09	Coprocessor Segment Overrun (386 or earlier only)
0x0A	Invalid Task State Segment
0x0B	Segment not present
0x0C	Stack Fault
0x0D	General protection fault
0x0E	Page fault
0x0F	...

Podela prekida

- **Eksterni interapti** nastaju van procesora. Veliki broj ulazno/izlaznih uređaja podiže ovu vrstu prekida (npr. miš, tastatura, tajmer, disk kontroleri, zvučne kartice itd).
- **Interni interapti** su prouzrokovani događajem unutar procesora, ili nastankom greške ili interapt instrukcijom. Interapt uzrokovan greškom se naziva i zamka (trap). Interapti uzrokovani pozivom interapt instrukcije nazivaju se softverskim interaptima.

Povratak iz prekida. Veliki broj prekidnih rutina vraća kontrolu glavnom programu koji nastavlja kao da se ništa nije dogodilo. Zamke (*traps*) obično prekidaju glavni program.

Namena prekida. Osnovna namena prekida je da se poboljša iskorišćenost procesora.

Primeri prekida:

1. Program – prekoračenje, deljenje nulom.
2. Tajmer – generiše ga tajmer unutar CPU. OS na taj način može redovno da izvršava neke instrukcije.
3. UI – generise UI kontroler.
4. Otkaz hardvera – npr. greška pariteta memorije.

Višestruki prekidi. Prekid može nastati i tokom izvršavanja neke prekidne rutine. Ovako nastali prekidi se mogu obraditi na *sekvencijalan* ili *ugnežđen* način.

Prioritet prekida. Prekid višeg prioriteta može da izazove prekid rada prekida sa nižim prioritetom.