

Chapter 6.1.

Organizacija računara

□ OSNOVI ORGANIZACIJE RAČUNARA

- *Pipeline* mehanizam
- Direktan pristup memoriji
- Mehanizam prekida

□ ORGANIZACIJA RAČUNARA SA PROCESOROM INTEL 8086

- Unutrašnja struktura procesora
- Razmena podataka sa okruženjem
- DMA
- Mehanizam prekida

Redovni profesor dr Dušan Regodić, dipl.inž.





Osnovi organizacije računara

- ❑ Računari su digitalni sistemi koji služe za obradu podataka. Na osnovu ulaznih podataka koje zadaje korisnik, a koji predstavljaju veličine bitne za rešavanje nekog problema, računar generiše izlazne podatke koji predstavljaju rešenje tog problema.
- ❑ Podaci koje korisnik unosi, kao i rezultati koji mu se prezentuju mogu biti u različitim formatima (tekst, odmerci nekog signala itd.), ali se u računaru konvertuju u binarni oblik, jer samo tako mogu da budu obrađivani.
- ❑ Binarne reči koje predstavljaju podatke su nizovi binarnih cifara 0 i 1 sa definisanim značenjem. To znači da se za svaku binarnu reč u sistemu zna šta ona predstavlja (podatak, instrukciju, i sl.).

Redovni profesor dr Dušan Regodić, dipl.inž.

Operacije

- ❑ Obrada podataka zasniva se na izvršavanju **operacija nad binarnim rečima**.
- ❑ Vrste operacija:
 - ❖ **unarne** – izvršavaju se nad jednom binarnom reči
 - ❖ **binarne** – izvršavaju se nad dvema binarnim rečima
- ❑ Operacije se dele u četiri klase:
 - ❖ **aritmetičke** – izvršavaju se nad binarnim brojevima
 - ❖ **logičke** – izvršavaju se nad pojedinačnim razredima binarne reči
 - ❖ **operacije pomeranja** – pomeranja binarne reči za određen broj razreda uлево или удесно uz popunjavanje praznih razreda
 - ❖ **operacije prenosa** – služe za prenos binarnih reči sa jedne lokacije na drugu, pri čemu se reč koja se prenosi ne menja
- ❑ Operacije se realizuju pomoću kombinacionih i sekvenčijalnih mreža.

Redovni profesor dr Dušan Regodić, dipl.inž.

Hardver

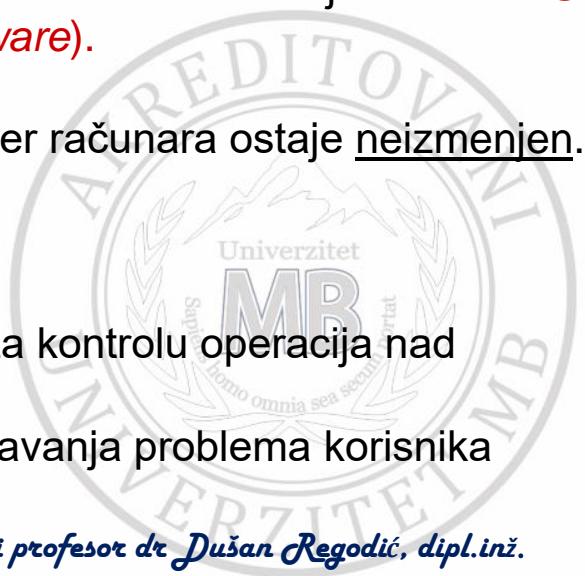
- ❑ Da bi uneo podatke u računar, korisnik mora da pristupi nekoj od jedinica za ulaz koje računar poseduje. Rezultate dobija putem izlazne jedinice računara. Ulazno-izlazne jedinice se često nazivaju **periferijama**.
- ❑ Podaci koje korisnik unese u računar smeštaju se u **operativnu memoriju**. Ona komunicira sa ostalim delovima računara, pri čemu ima najživlju komunikaciju sa procesorom računara.
- ❑ **Procesor** obavlja potrebne operacije nad podacima koji se nalaze u operativnoj memoriji.
- ❑ S obzirom da je operativna memorija brza i zbog toga skupa i ograničenog kapaciteta, svi podaci sa kojima računar trenutno ne radi smeštaju se u **eksternu**, tj. **spoljašnju memoriju** koja je znatno sporija, ali ima mnogo veći kapacitet od operativne memorije.
- ❑ **Ulazno-izlazne jedinice, operativna memorija, procesor i eksterne memorije** čine glavne fizičke delove računara koji se jednim imenom nazivaju **HARDVER** (*hardware*).



Redovni profesor dr Dušan Regodić, dipl.inž.

Softver

- Sam hardver nije dovoljan za rad računara. Da bi računar mogao da izvrši neku obradu podataka, neophodno je definisati program rada računara u kome će precizno biti zadate sve aktivnosti u okviru obrade, kako po pitanju sadržaja, tako i po pitanju redosleda njihovog izvođenja.
- Program predstavlja niz pojedinačnih instrukcija ili naredbi koji se čuva u memoriji računara. **Svi** programi u računaru se jednim imenom nazivaju **SOFTVER (software)**.
- Izborom programa menja se način obrade podataka, dok hardver računara ostaje neizmenjen.
- Programi se mogu svrstati u dve kategorije:
 - **sistemski softver** – programi na niskom nivou koji služe za kontrolu operacija nad hardverom
 - **aplikativni softver** - programi koje se pišu za potrebe rešavanja problema korisnika



Redovni profesor dr Dušan Regodić, dipl.inž.



Izvršavanje instrukcije (1)

□ Procesor izvršava program koji je prethodno smešten u operativnu memoriju instrukciju po instrukciju i to onim redosledom kojim su instrukcije navedene u programu, sve dok ne dođe do kraja programa. Ovaj postupak se može predstaviti ciklusima koji sadrže sledeća četiri koraka:

1. dohvatanje instrukcije iz memorije
2. dohvatanje operanda, ako se to zahteva u instrukciji
3. izvršavanje instrukcije
4. ako se u instrukciji zahteva, upis rezultata u memoriju ili slanje na odgovarajuću perifernu jedinicu

Napomena: Operandi su binarni brojevi nad kojima se vrše željene aritmetičke ili logičke operacije. Nalaze se u memoriji zajedno sa instrukcijama, ili se dobijaju od neke periferije.

Redovni profesor dr Dušan Regodić, dipl.inž.

Izvršavanje instrukcije (2)

- Format instrukcije se sastoji iz dva dela:
 - operacionog koda, koji definiše operaciju
 - adresnog dela, koji specificira podatke, tj. operative
- Da bi procesor izvršio operaciju, najpre pomoću adresnog dekodera dekoduje kod operacije, a zatim na osnovu adresnog dela instrukcije doprema operande.
- Adresni deo instrukcije može da sadrži jednu ili više adresa (ili podataka) u zavisnosti od broja operanada.

Jednoadresna instrukcija

OP	#A1
----	-----

Dvoadresna instrukcija

OP	#A1	#A2
----	-----	-----

Troadresna instrukcija

OP	#A1	#A2	#A3
----	-----	-----	-----

- Pošto se operandi dopreme u registre za operative, aritmetičko-logička jedinica izvršava operaciju definisanu kodom operacije.

Redovni profesor dr Dušan Regodić, dipl.inž.



Prenos instrukcija i podataka

- ❑ Prenos podataka između procesora i memorije obavlja se preko bidirekcionih (dvosmernih) veza koje se nazivaju **magistralom podataka**.
- ❑ Izbor memorijske lokacije iz koje treba uzeti podatak ili u koju treba upisati podatak obavlja se pomoću **adrese** koju određuje procesor.
- ❑ Adresa se šalje memoriji preko unidirekcionih (jednosmernih) veza koje se nazivaju **adresnom magistralom**.
- ❑ Prenos po magistralama je paralelan, što znači da se svi biti memorijske reči prenose istovremeno. Širina adresne magistrale, u većini slučajeva, određuje **kapacitet operativne memorije**. Tako, ako adresa ima n bita, pomoću nje se može adresirati $C = 2^n$ različitih memorijskih lokacija, pa C predstavlja maksimalni kapacitet memorije.

Redovni profesor dr Dušan Regodić, dipl.inž.

Upis podatka

Upis binarnog podatka u memoriju

- procesor adresira željenu memorijsku lokaciju postavljanjem njene adrese na adresnu magistralu
- procesor postavlja podatak koji treba da se upiše na magistralu podataka
- procesor šalje komandu memoriji da obavi upis podatka



Redovni profesor dr Dušan Regodić, dipl.inž.



Čitanje podatka

Čitanje binarnog podatka iz memorije

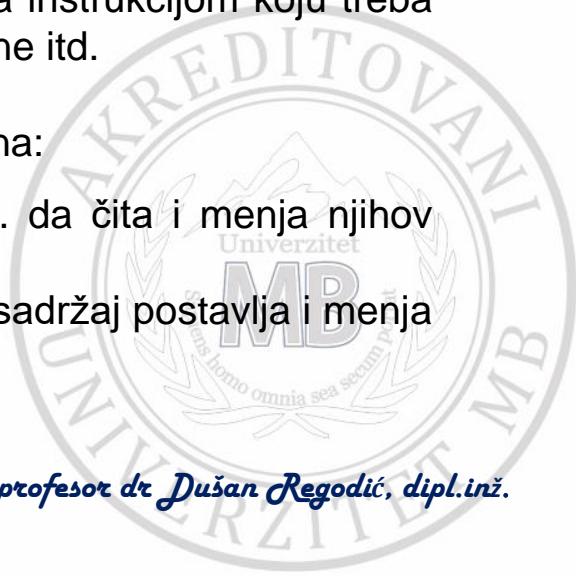
- procesor adresira memorijsku lokaciju iz koje želi da pročita podatak postavljanjem njene adrese na adresnu magistralu
- procesor šalje komandu memoriji da na magistralu podataka postavi adresirani podatak
- procesor preuzima podatak sa magistrale podataka

Redovni profesor dr Dušan Regodić, dipl.inž.



Registri procesora

- Pošto je u organizaciji računara procesor odgovoran na izvršavanje instrukcija, on mora da ima mogućnost lokalnog čuvanja izvesne količine podataka. Za to se koriste brzi **procesorski registri**.
- Registri mogu da imaju različite namene. Na primer, postoji registar u kome se čuva adresa naredne instrukcije koju treba dohvatiti, zatim registar sa instrukcijom koju treba izvršiti, registri u koje se smeštaju operandi, registri opšte namene itd.
- U zavisnosti od načina pristupa, registri se mogu klasifikovati na:
 - **adresibilne registre** - programer može da im pristupa, tj. da čita i menja njihov sadržaj
 - **interne registre** - nisu dostupni programeru, već se njihov sadržaj postavlja i menja automatski tokom rada procesora



Redovni profesor dr Dušan Regodić, dipl.inž.

Pipeline mehanizam (1)

Pipeline predstavlja koncept koji je uveden sa ciljem da se ubrza proces izvršavanja programa. Koncept se zasniva na korišćenju konkurentnosti prilikom izvršavanja instrukcija.

□ Pretpostavimo da se proces izvršavanja instrukcije odvija u četiri faze:

IF (*Instruction Fetch*): dohvatanje instrukcije iz memorije i njeno smeštanje u registar za instrukcije

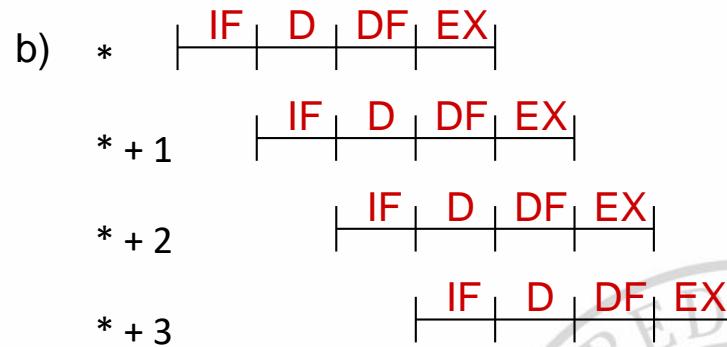
D (*Decode*): dekodiranje operacionog koda instrukcije

DF (*Data Fetch*): dohvatanje podataka (operanada) bilo iz memorije ili iz odgovarajućih registara

EX (*Execute*): izvršavanje instrukcije

Redovni profesor dr Dušan Regodić, dipl.inž.

Pipeline mehanizam (2)



Izvršavanje instrukcija (a) obično (b) u *pipeline-u*

- Ubrzanje koje se može ostvariti zavisi od broja stepena (stages) ili dubine *pipeline-a*, tj. faza u izvršavanju instrukcije. Cena ubrzanja je u složenijem i skupljem procesoru. Ipak, skoro svi današnji procesori podržavaju *pipeline* koncept.

Redovni profesor dr Dušan Regodić, dipl.inž.

Direktan pristup memoriji (1)

Direktan pristup memoriji - DMA (*Direct Memory Access*) omogućava direktan prenos podataka između periferije i memorije, bez posredovanja procesora.

- ❑ U sistemu bez DMA, procesor mora da kopira deo po deo podataka sa izvora i da ih prosleđuje do odredišta, a pri tome nije **raspoloživ** ni za kakvu drugu aktivnost.
Dodatno vreme se gubi zato što su periferije mnogo sporije od operativne memorije.

- ❑ Uvođenjem DMA koncepta, procesor je osloboden poslova oko prenosa podataka i za to vreme može da obavlja druge aktivnosti. Ipak, treba imati u vidu da u tim aktivnostima ne može da koristi magistralu.

Redovni profesor dr Dušan Regodić, dipl.inž.



Direktan pristup memoriji (2)

Princip rada

DMA prenos podatka obavlja se tako što, slanjem odgovarajuće komande DMA kontroleru, procesor inicira prenos, a zatim potpunu odgovornost preuzima DMA kontroler (generisanje adresa, slanje podataka). Po završetku prenosa, DMA kontroler obaveštava procesor da je prenos završen i da je magistrala slobodna.

DMA prenos se primjenjuje:

- kada je potrebno preneti veliku količinu podatka, pa bi transfer preko procesora znatno usporio prenos
- kada je periferija relativno brza (primer takve periferije je hard disk)

Redovni profesor dr Dušan Regodić, dipl.inž.



Mehanizam prekida (1)

Mehanizam prekida je uveden kako bi se izbeglo da procesor troši vreme čekajući na spoljašnje događaje.

Problem:

Većina periferija je znatno sporija od procesora. Ukoliko procesor treba da pošalje podatke nekoj periferiji, na primer štampaču da ih odštampa, procesor mora da sačeka da periferija završi svoj posao, tj. štampanje prispelih podataka, kako bi poslao nove podatke za štampanje. Čekanje na periferiju (dok ona obavi svoj posao) predstavlja izgubljeno vreme za procesor, jer je on tada besposlen.

Rešenje:

Navedeni problem se rešava uvođenjem mehanizma prekida koji omogućava efikasniji rad računara sa periferijama. Korišćenje prekida omogućava procesoru da izvršava druge instrukcije za vreme dok periferija obavlja svoj posao.

Redovni profesor dr Dušan Regodić, dipl.inž.

Mehanizam prekida (2)

- Zahvaljujući mehanizmu prekida, procesor ne čeka na oslobođanje periferije, već opslužuje periferiju na njen zahtev.

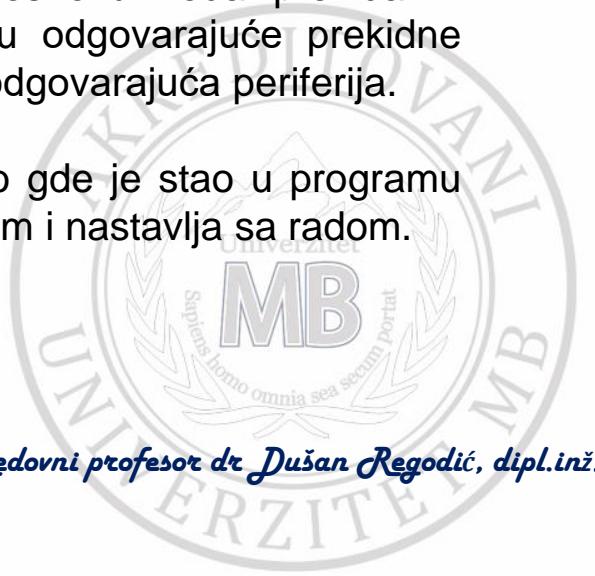
Postupak opsluživanja prekida odvija se na sledeći način:

1. Kada periferija postane spremna za prijem podataka od procesora, ona to signalizira procesoru slanjem **zahteva za prekidom**. Zahtev sadrži **kod prekida** koji odgovara periferiji koja ga je poslala.
2. Po priјemu zahteva, procesor završava izvršavanje tekuće instrukcije i na kratko prekida izvršavanje tekućeg programa kako bi opslužio prispeli zahtev.

Redovni profesor dr Dušan Regodić, dipl.inž.

Mehanizam prekida (3)

3. Svaki procesor ima skup prekida koje je u stanju da opsluži. Za svaki prekid unapred je definisana tzv. **prekidna rutina** koja predstavlja podprogram koji treba da se izvrši u slučaju prispeća zahteva za tim prekidom. Prekidne rutine su smeštene u memoriji na određenim adresama. Sve adrese prekidnih rutina smeštene su u **tabelu prekida** (*Interrupt Pointer Table*) koja se, takođe, nalazi u memoriji. Procesor opslužuje prekid tako što, na osnovu koda prekida iz prispelog zahteva, pronalazi u tabeli prekida adresu odgovarajuće prekidne rutine i izvršava rutinu. U prekidnoj rutini se opslužuje odgovarajuća periferija.
4. Po završetku prekidne rutine, procesor se vraća tamo gde je stao u programu koji je izvršavao u trenutku nailaska zahteva za prekidom i nastavlja sa radom.



Redovni profesor dr Dušan Regodić, dipl.inž.

Mehanizam prekida (4)

- ❑ Da bi mehanizam prekida mogao uspešno da funkcioniše, neophodno je obezbediti da se pri povratku iz prekidne rutine procesor nađe i potpuno istim uslovima u kojima je bio kada je počeo opsluživanje prekida. Ti uslovi se nazivaju **kontekstom procesora**.
- ❑ Kad procesoru stigne zahtev za prekidom i on ga prihvati, pre nego što pređe na izvršavanje odgovarajuće prekidne rutine, procesor mora da sačuva kontekst u kome je izvršavao tekući program. Kontekst obično čine trenutni sadržaji pojedinih registara procesora.
- ❑ Kontekst je neophodno sačuvati kako bi, nakon završetka prekidne rutine, procesor znao odakle treba da nastavi izvršavanje glavnog programa i kakav je bio status nakon poslednje operacije koju je ALU izvršila pre prekida. Naime, prekidna rutina koristi iste procesorske registre kao i glavni program, pa obično menja njihov sadržaj.

Redovni profesor dr Dušan Regodić, dipl.inž.



Mehanizam prekida (5)

Postoje dve vrste prekida:

- spoljašnji ili eksterni prekidi** - prekidi koji dolaze od periferija
- unutrašnji prekidi** - prekidi koji su posledica izvršavanja instrukcije prekida, ili posledica neke neregularnosti u izvršavanju tekuće instrukcije
- Prekidima se pridružuju **prioriteti** koji ukazuju na njihovu važnost. Prednost u opsluživanju imaju prekidi višeg prioriteta.
- Interni prekidi su višeg prioriteta od eksternih.

Redovni profesor dr Dušan Regodić, dipl.inž.

Okruženje sa procesorom Intel 8086

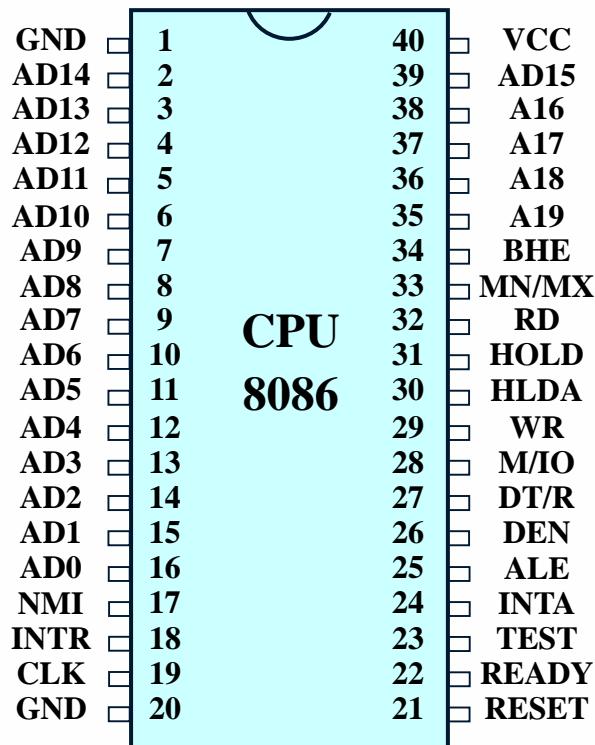
Karakteristike procesora Intel 8086

- radni takt procesora je, u zavisnosti od verzije, 2, 5, 8, ili 10MHz
- 16-bitna magistrala podataka
- 20-bitna adresna magistrala (moguće je adresirati 2^{20} bajtova, tj. 1MB memorije)
- magistrala podataka i adresna magistrala su multipleksirane, tj. koriste iste pinove na procesorskom čipu
- postoje dve linije za spoljašnje prekide
- postoji poseban I/O (ulazno/izlazni) adresni prostor veličine 64K za adresiranje perifernih jedinica
- postoji podrška za DMA



Redovni profesor dr Dušan Regodić, dipl.inž.

Raspored pinova na čipu procesora 8086 (1)



Redovni profesor dr Dušan Regodić, dipl.inž.



Raspored pinova na čipu procesora 8086 (2)

Pin	Opis
VCC	pozitivan kraj napajanja procesora (+5V)
GND	negativan kraj napajanja procesora (masa)
AD0÷AD15	multipleksirane magistrale podataka i adresa
A16÷A19	dodatne 4 linije za adresnu magistralu
NMI [Not Masked Interrupt]	ulaz na koji se dovode nemaskirani prekidi
INTR [Interrupt Request]	ulaz na koji se sa kontrolera prekida dovodi signal o pristiglo zahtevu za spoljašnjim prekidom
INTA [Interrupt Acknowledge]	izlaz kojim procesor signalizira kontroleru prekida da je spreman da prihvati zahtev za prekidom
CLK [Clock]	ulaz za takt na kome radi procesor

Redovni profesor dr Dušan Regodić, dipl.inž.

Raspored pinova na čipu procesora 8086 (3)

Pin	Opis
BHE [Bus High Enable]	izlaz kojim procesor signalizira da je u 16-bitnom modu rada
MN/MX [MiNimal/MaXimal]	ulaz na osnovu koga procesor zna da li je u minimalnom ili maksimalnom modu rada, tj. da li je jedini procesor u sistemu ili ne
RD [Read]	signal kojim procesor zahteva podatak iz adresirane memorijске ćelije ili adresirane periferije
WR [Write]	signal kojim procesor upisuje podatak u adresiranu memorisku ćeliju ili adresiranu periferiju
HOLD	signal kojim DMA kontroler traži od procesora da mu prepusti magistralu
HLDA [Hold Acknowledge]	signal kojim procesor signalizira DMA kontroleru da mu je preustio magistralu

Redovni profesor dr Dušan Regodić, dipl.inž.

Raspored pinova na čipu procesora 8086 (4)

Pin	Opis
M/IO	signal kojim procesor govori ostatku sistema da li je adresa koju je postavio na adresnu magistralu adresa memorije ili adresa periferije
ALE [Address Latch Enable]	signal kojim procesor signalizira da je na magistralu postavio adresu
DEN [Data Enable]	signal kojim procesor signalizira da je na magistralu postavio podatak
READY	ulazni signal kojim memorija kaže procesoru da je spremna za sledeći ciklus čitanja ili upisa podataka
RESET	ulazni signal kojim se resetuje procesor
TEST	ulaz koji procesoru kaže da je sistem u test modu i da instrukcije izvršava jednu po jednu (<i>step mode</i>)

Redovni profesor dr Dušan Regodić, dipl.inž.



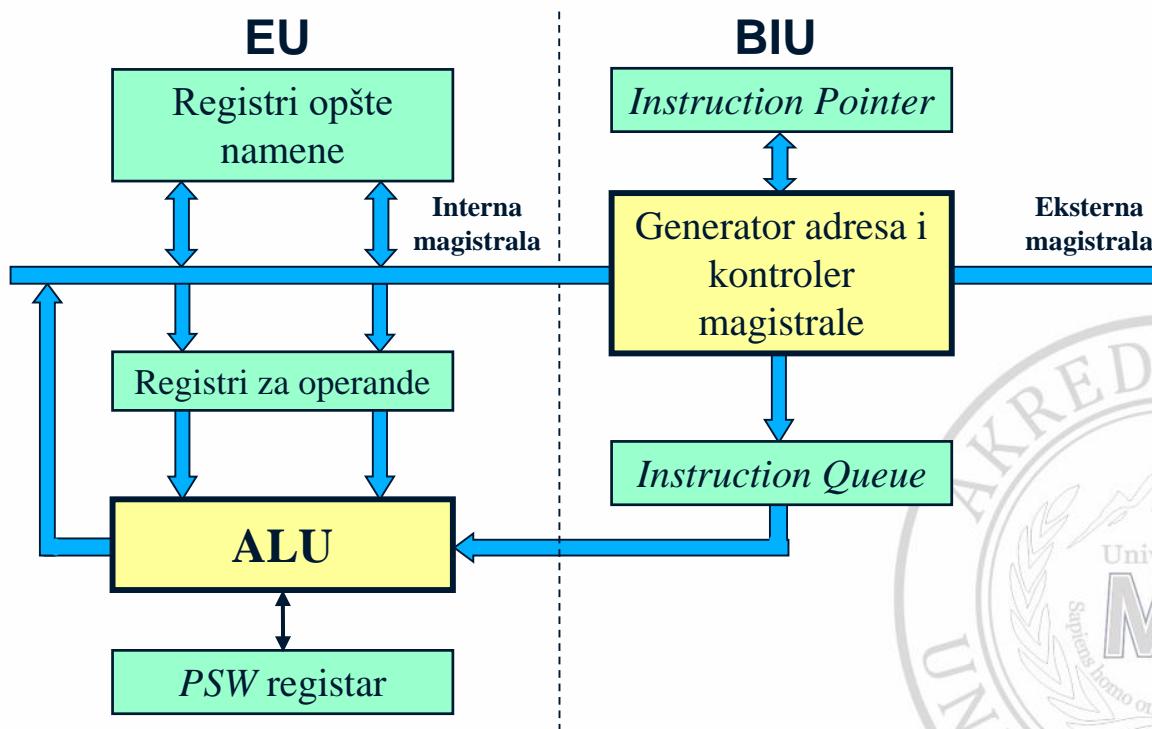
Unutrašnja struktura procesora Intel 8086 (1)

Procesor Intel 8086 sastoji se od dve osnovne jedinice:

1. EU (*Execution Unit*) – izvršna jedinica koja izvršava instrukcije
 2. BIU (*Bus Interface Unit*) – jedinica za spregu sa magistralom koja dohvata instrukcije iz memorije, čita operande i šalje rezultat u memoriju ili ka odgovarajućoj periferiji
-
- Izvršna jedinica i jedinica za spregu sa magistralom rade nezavisno jedna od druge i najčešće rade istovremeno.

Redovni profesor dr Dušan Regodić, dipl.inž.

Unutrašnja struktura procesora Intel 8086 (2)



Redovni profesor dr Dušan Regodić, dipl.inž.

Execution Unit (1)

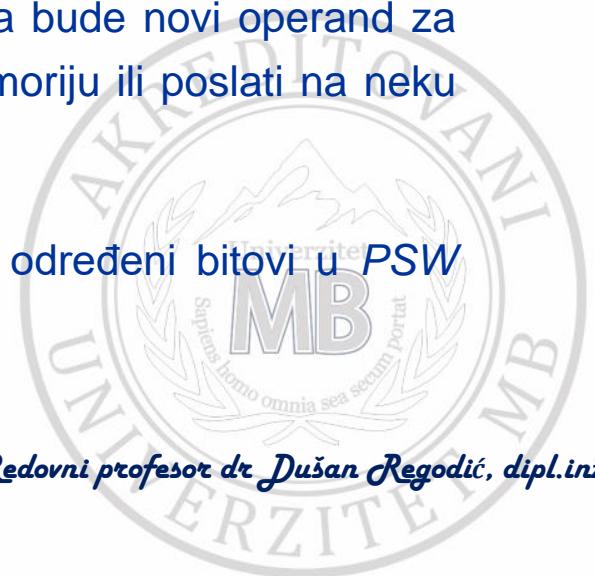
- ❑ Izvršna jedinica nije direktno povezana sa sistemskom magistralom.
- ❑ Centralni deo izvršne jedinice je ALU koja, u zavisnosti od instrukcije, obavlja aritmetičke ili logičke operacije nad operandima.
- ❑ Instrukciju koju teba da izvrši, ALU dobija iz Instruction Queue koji se nalazi unutar BIU.
- ❑ Dobijenu instrukciju ALU izvršava nad operandima koji se nalaze u dva 16-bitna registra čiji su izlazi vezani na ulaze ALU.
- ❑ Operandi u registre dolaze iz memorije, sa periferije, ili iz registara opšte namene ukoliko su rezultat prethodne operacije koju je ALU izvršila.



Redovni profesor dr Dušan Regodić, dipl.inž.

Execution Unit (2)

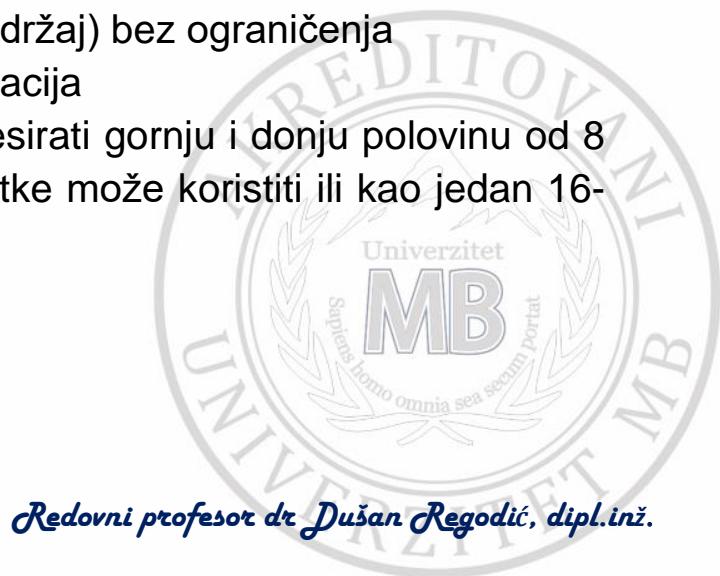
- Ukoliko neka instrukcija zahteva podatak iz memorije ili sa neke od periferija koji nije raspoloživ u registrima za operande, izvršna jedinica prosleđuje taj zahtev BIU, koja zatim obezbeđuje traženi podatak.
- Rezultat izvršavanja neke instrukcije u ALU može da bude novi operand za sledeću instrukciju, ili podatak koji treba smestiti u memoriju ili poslati na neku od periferija.
- Kao rezultat izvršene operacije mogu se postaviti i određeni bitovi u PSW registru.



Redovni profesor dr Dušan Regodić, dipl.inž.

Registri opšte namene

- Postoji 8 16-bitnih registara opšte namene koji su svrstani u dve grupe od po 4 registra.
- Prvu grupu čine **registri za podatke**.
 - služe za čuvanje različitih vrsta podataka generisanih tokom izvršavanja programa
 - korisnik može da im pristupa (čita i menja sadržaj) bez ograničenja
 - koriste se u većini aritmetičkih i logičkih operacija
 - specifični su jer im je moguće posebno adresirati gornju i donju polovinu od 8 bitova, što znači da se svaki registar za podatke može koristiti ili kao jedan 16-bitni, ili kao dva 8-bitna registra
- Drugu grupu čine **indeksni i pointer registri**.



Redovni profesor dr Dušan Regodić, dipl.inž.

PSW registar (1)

- ❑ **PSW** (*Processor Status Word*) registar služi za pamćenje nekih bitnih osobina poslednjeg rezultata dobijenog iz ALU, kao i određenih parametara procesora.
- ❑ Svaka informacija je predstavljena jednim bitom regista koji se zove **fleg** (*flag* – zastavica).
- ❑ Procesor Intel 8086 ima 16-bitni PSW koji sadrži:
 - ❖ 6 statusnih flegova
 - ❖ 3 kontrolna flega
 - ❖ 7 bitova koji se ne koriste

Redovni profesor dr Dušan Regodić, dipl.inž.



PSW registar (2)

Statusni flegovi

- predstavljaju informacije o rezultatu operacije izvršene u ALU
- može ih menjati jedino EU
- korisnik može samo da čita njihove vrednosti, ali ne i da ih menja

Kontrolni flegovi

- predstavljaju parametre bitne za rad procesora
- njihovu vrednost postavlja korisnik

Redovni profesor dr Dušan Regodić, dipl.inž.



Statusni flegovi

Statusni flegovi	Opis
CF [Carry flag]	Setovan na 1 ako je bilo prenosa ili pozajmice pri računanju najvišeg bita rezultata. Dobija novu vrednost nakon svake instrukcije sabiranja, oduzimanja ili rotiranja.
PF [Parity flag]	Setovan na 1 ako rezultat ima paran broj jedinica. Koristi se za proveru ispravnosti prenosa podataka do periferija.
AF [Auxiliary Carry flag]	Setovan na 1 ako je pri računanju rezultata bilo prenosa ili pozajmice između nižeg i višeg bajta.
ZF [Zero flag]	Setovan na 1 ako je rezultat operacije 0.
SF [Sign flag]	Setovan na 1 ako je rezultat negativan broj, tj. počinje sa 1 (negativni brojevi se predstavljaju u komplementu dvojke).
OF [Overflow flag]	Setovan na 1 ako dođe do prekoračenja, tj., zbog veličine, rezultat ne može da bude upisan na predviđenu lokaciju.

Redovni profesor dr Dušan Regodić, dipl.inž.



Kontrolni flegovi

- Kontrolni flegovi
 - DF [Direction flag]
 - IF [Interrupt flag]
 - TF [Trap flag]

□ IF zabranjuje ili dozvoljava spoljašnje prekide koji dolaze po INTR liniji. Spoljašnji prekidi koji stižu po NMI liniji ili unutrašnji prekidi se ne mogu zabraniti ovim flegom.

Redovni profesor dr Dušan Regodić, dipl.inž.



Bus Interface Unit

❑ BIU je jedinica za spregu sa magistralom koja vrši sve operacije sa memorijom i periferijama koje zahteva EU. Preko nje se, na zahtev EU, prenose podaci između procesora i memorije ili periferija.

❑ BIU sadrži:

- generator adresa i kontroler magistrale
- *Instruction Pointer* registar
- *Instruction Queue*



Redovni profesor dr Dušan Regodić, dipl.inž.

Instruction Pointer registar

- ❑ *Instruction Pointer* (IP) je registar koji sadrži adresu sledeće instrukcije koju BIU treba da prenese iz memorije u procesor.
- ❑ IP je u suštini brojač koji inkrementira sadržaj nakon preuzimanja nove instrukcije iz memorije.
- ❑ U slučaju da u programu treba napraviti skok, neophodno je promeniti vrednost IP-a. Nakon toga, prva sledeća instrukcija koju treba izvršiti je ona koja se nalazi na adresi upisanoj u IP.



Redovni profesor dr Dušan Regodić, dipl.inž.

Instruction Queue

- ❑ Tokom izvršavanja instrukcije u EU, BIU dohvata naredne instrukcije iz memorije i smešta ih u internu memoriju, tj. *Instruction Queue*.
- ❑ Korišćenje *Instruction Queue-a* obezbeđuje da EU bude uvek snabdevena novim instrukcijama.
- ❑ U *Instruction Queue* može da se smesti do 6 instrukcija koje čekaju da se izvrše.
- ❑ U slučaju da EU izvrši instrukciju koja prebacuje kontrolu na drugu lokaciju u memoriji (kao na primer instrukcije skoka GOTO, CALL, JUMP,...), jedinica za spregu sa magistralom:
 - resetuje *Instruction Queue*
 - dohvata instrukciju sa nove adrese i direktno je prosleđuje EU
 - nastavlja da puni *Instruction Queue* narednim instrukcijama



Redovni profesor dr Dušan Regodić, dipl.inž.

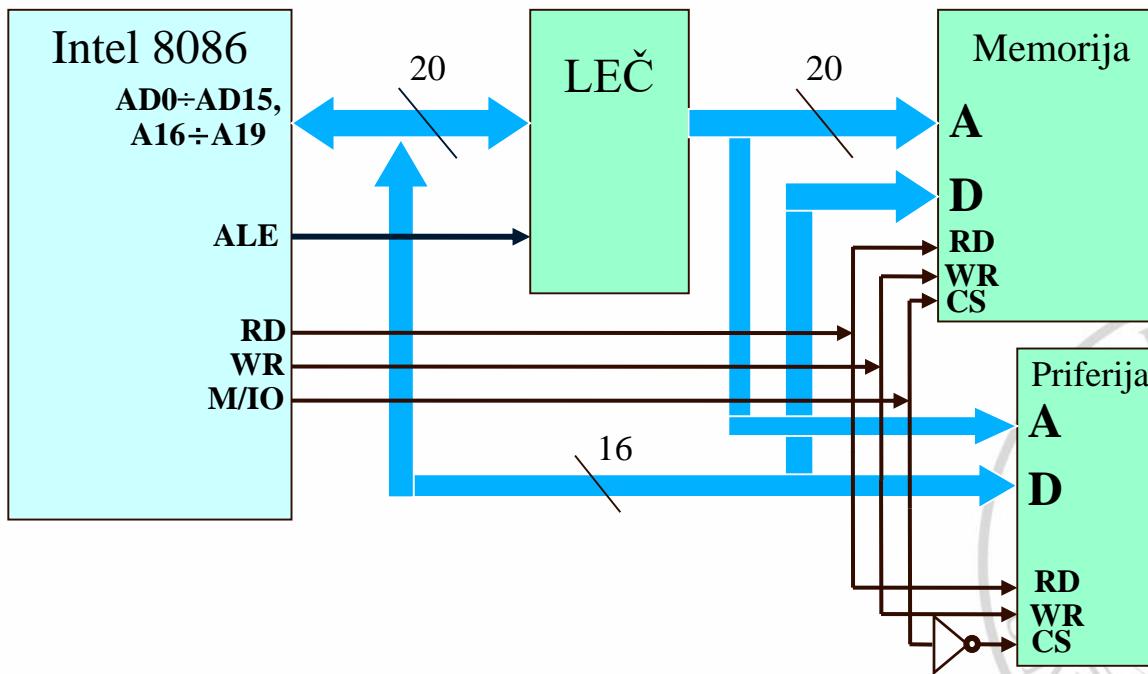
Razmena podataka sa okruženjem

- ❑ Potreba za razmenom binarnih informacija između procesora i okruženja (memorija, periferije) je veoma česta.
- ❑ Komunikacija sa memorijom je potrebna zbog:
 - čitanja instrukcija iz memorije
 - čitanja podataka iz memorije
 - upisa rezultata u memoriju
- ❑ Procesor komunicira sa periferijama (šalje im i prima podatke od njih) na sličan način kao i sa memorijom.

Redovni profesor dr Dušan Regodić, dipl.inž.



Povezivanje sa okruženjem



Redovni profesor dr Dušan Regodić, dipl.inž.



Razmena podataka sa memorijom (1)

□ Upis binarnog podatka u memoriju:

- procesor adresira željenu memorijsku lokaciju postavljanjem njene adrese na adresnu magistralu
- na magistralu podataka procesor postavlja podatak koji želi da upiše
- setovanjem signala **WR** (*write* – pin29) procesor zadaje komandu memoriji da obavi upis podatka

□ Čitanje binarnog podatka iz memorije:

- procesor adresira memorijsku lokaciju iz koje želi da pročita podatak postavljanjem njene adrese na adresnu magistralu
- setovanjem signala **RD** (*read* – pin32) procesor zadaje komandu memoriji da na magistralu podataka postavi adresirani podatak
- procesor preuzima podatak sa magistrale podataka

Redovni profesor dr Dušan Regodić, dipl.inž.

Razmena podataka sa memorijom (2)

- **Problem:** adresna magistrala i magistrala podataka su multipleksirane, pa nije moguće da se na pinovima procesora istovremeno pojave i adrese i podaci, što je neophodno prilikom upisa i čitanja sadržaja.
- Navedeni problem se rešava upotrebom leča. Leč je sekvensijalno kolo slično registru, s tom razlikom što se prilikom upisa podatka u leč, podatak automatski pojavljuje na njegovom izlazu. Uloga leča je da privremeno zapamti adresu.
- **Postupak postavljanja adrese i podatka:**
 - procesor najpre postavlja adresu na multipleksiranu magistralu
 - zatim aktivira **ALE** signal kojim se adresa upisuje u leč i prosleđuje na njegov izlaz
 - pošto se izlaz leča vodi na adresni ulaz memorije, adresa je prosleđena do memorije
 - sada procesor može na multipleksiranu magistralu da postavi podatak za upis ili da preko magistrale pročita podatak iz memorije

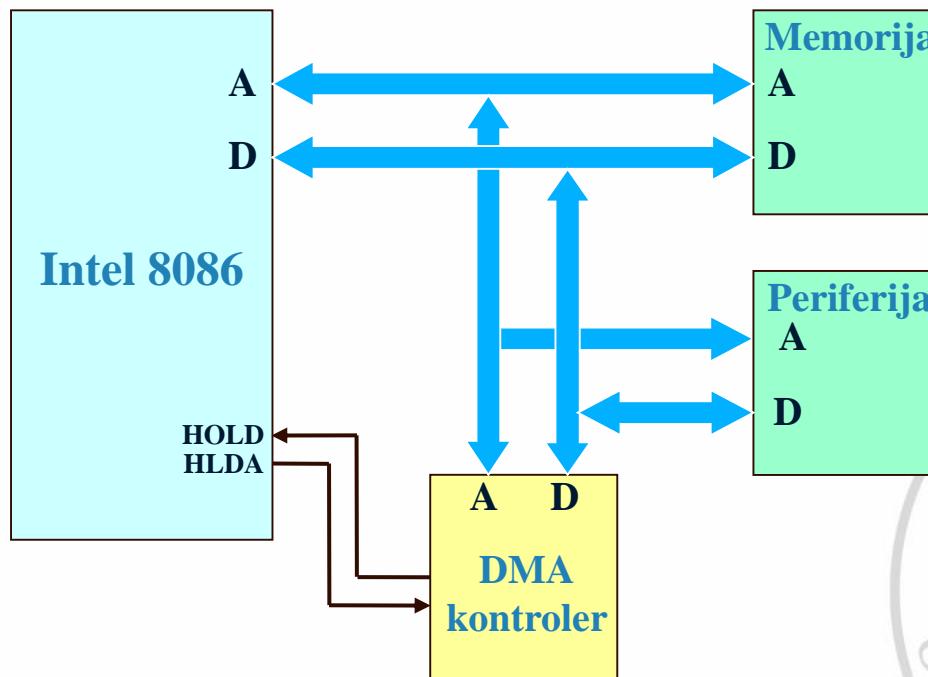
Redovni profesor dr Dušan Regodić, dipl.inž.

Razmena podataka sa periferijom

- Postoje dve razlike u komunikaciji sa periferijama u odnosu na komunikaciju sa memorijom:
 - Prva razlika je u tome što se sada ne adresira memorijska lokacija već periferija. Naime, svaka periferija ima svoju adresu. Kada procesor na adresnu magistralu postavi adresu periferije, odgovoriće mu samo periferija čiju je adresu postavio.
 - Druga razlika je u tome što periferije i memorija moraju da znaju čiju adresu je procesor postavio na magistralu. Adrese memorijskih lokacija i adrese periferija se poklapaju. Kada ne bi postojala nikakva dodatna informacija o tome kome procesor želi da se obrati, desilo bi se da mu istovremeno odgovore i memorija i periferija, što bi dovelo do greške u sistemu zbog sudara podataka koji dolaze sa različitih strana na istu magistralu podataka.
- Problem se rešava **M/IO** signalom koji procesor postavlja na 1 kada želi da se obrati memoriji, odnosno na 0 kada želi da se obrati periferiji. Ovaj signal se vodi na **CS** (*Chip Select*) ulaze i memorije i periferije.

Redovni profesor dr Dušan Regodić, dipl.inž.

Povezivanje procesora sa DMA kontrolerom



Na slici nije prikazan adresni leč, ali se podrazumeva da on postoji unutar procesora.

Redovni profesor dr Dušan Regodić, dipl.inž.

DMA ciklus

DMA prenos se obavlja na sledeći način:

- ❑ DMA kontroler zahteva od procesora da mu omogući pristup magistrali tako što aktivira signal **HOLD** (pin31).
- ❑ Kada procesor primi signal **HOLD**, završava tekući ciklus na magistrali i aktivira signal **HLDA** (*Hold Acknowledge* - pin30), čime predaje magistralu DMA kontroleru.
- ❑ Dok DMA kontroler koristi magistralu, procesor ne sme da joj pristupa (ne može da pristupa memoriji u cilju upisa ili čitanja), ali može da nastavi sa obradom podataka koji se trenutno u njemu nalaze.
- ❑ Kada završi DMA ciklus, DMA kontroler deaktivira **HOLD** signal, čime oslobađa magistralu i predaje je procesoru.

Redovni profesor dr Dušan Regodić, dipl.inž.

Mehanizam prekida

- ❑ Procesor Intel 8086 ima mogućnost opsluživanja kako internih, tako i eksternih prekida.
- ❑ Unutrašnji prekidi nastaju prilikom izvršavanja instrukcije INT. Na osnovu njenog koda prekida, procesor aktivira odgovarajuću prekidnu rutinu i opslužuje prekid.
- ❑ Za prijem zahteva za spoljašnjim prekidima, predviđena su dva pina procesora:
 - **INTR** (*Interrupt Request* – pin18)
 - **NMI** (*Not Masked Interrupt* – pin17)



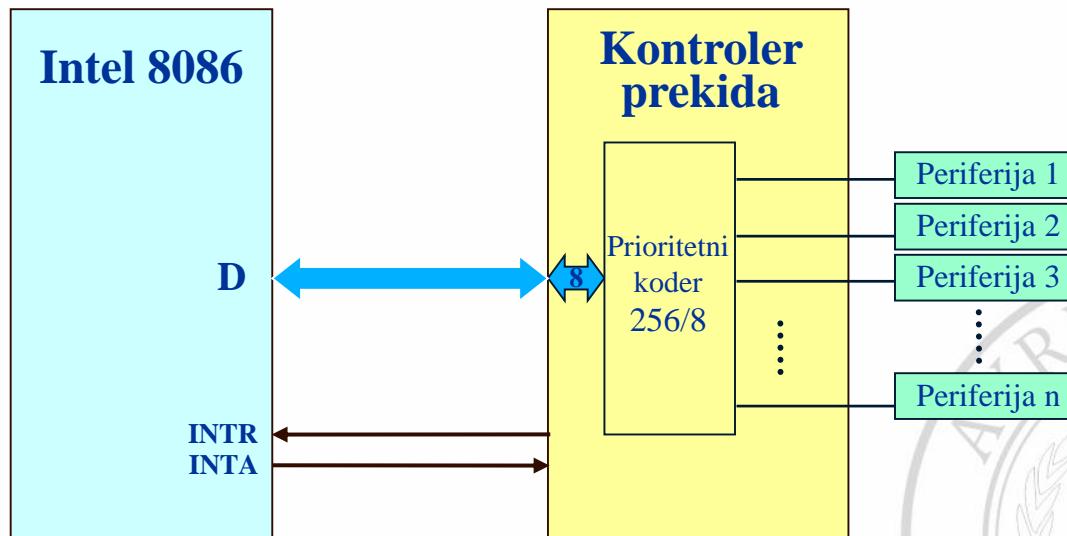
Redovni profesor dr Dušan Regodić, dipl.inž.

INTR (1)

- Linija **INTR** je obično povezana sa programabilnim **kontrolerom prekida** (Intel 8259).
- Uloga kontrolera prekida je da prihvata zahteve za prekidima upućene od strane periferija koje su sa njim povezane.
- Nakon dobijanja zahteva, kontroler određuje koji od pristiglih zahteva ima najviši prioritet i ukoliko je pristigli zahtev višeg prioriteta od prioriteta zahteva koji se trenutno obradjuje, aktivira liniju **INTR**.
- Za upoređivanje prioriteta prispelih zahteva koristi se **prioritetni koder**. To je vrsta kodera koja dopušta dovođenje signala na više ulaza, pri čemu se na izlazu pojavljuje kodirana informacija koja odgovara aktivnom ulazu sa najvišim prioritetom.

Redovni profesor dr Dušan Regodić, dipl.inž.

INTR (2)



Redovni profesor dr Dušan Regodić, dipl.inž.

INTR (3)

- ❑ Nakon aktiviranja linije **INTR**, procesor prihvata ili ne prihvata zahtev za prekidom u zavisnosti od vrednosti IF flega u PSW registru. Ako je navedeni fleg postavljen na 1, zahtev se prihvata, a ako je 0, zahtev se ignoriše.
- ❑ Ukoliko procesor prihvati zahtev, najpre završava izvršavanje tekuće instrukcije. Stoga vreme potrebno da procesor prihvati prekid (vreme kašnjenja prekida) zavisi od dužine instrukcije koja se trenutno izvršava (najveće je kašnjenje za instrukcije množenja i deljenja).
- ❑ Nakon toga, procesor odgovara na zahtev aktiviranjem linije INTA (*Interrupt Acknowledge – pin24*), čime se kontroler prekida obaveštava da na magistralu podataka treba da postavi 8-bitni podatak koji predstavlja **kod prekida** i nosi informaciju o tome koja periferija je zahtevala prekid.
- ❑ Na osnovu koda prekida dobijenog sa magistrale podataka, procesor određuje koju prekidnu rutinu treba da pozove.

Redovni profesor dr Dušan Regodić, dipl.inž.

Tabela prekida (1)

- Tabela prekida – *Interrupt Pointer Table* je tabela koja uspostavlja vezu između koda prekida i prekidne rutine (programa) koju treba izvršiti da bi se opslužio taj prekid.
- Na osnovu koda prekida, iz tabele prekida se može dobiti adresa u memoriji na kojoj se nalazi prekidna rutina koju treba izvršiti.
- Tabela prekida ima 256 ulaza, za svaki tip prekida koji procesor može da opsluži, po jedan ulaz.
- Za pamćenje adrese prekidne rutine koriste se 4 bajta, pa tabela uvek zauzima prvi kilobajt memorije.

Redovni profesor dr Dušan Regodić, dipl.inž.

Tabela prekida (2)

Adresa memorijske lokacije	4 bajta
0	Adresa prekidne rutine čiji je kod 0
4	Adresa prekidne rutine čiji je kod 1
8	Adresa prekidne rutine čiji je kod 2
12	Adresa prekidne rutine čiji je kod 3
.	.
.	.
1020	Adresa prekidne rutine čiji je kod 255
1024	ostatak memorije
.	.

Redovni profesor dr Dušan Regodić, dipl.inž.



Čuvanje konteksta procesora (1)

- Kad procesoru stigne zahtev za prekidom i on ga prihvati, pre nego što pređe na izvršavanje odgovarajuće prekidne rutine, procesor mora da sačuva **kontekst** u kome je izvršavao tekući program.
- Kontekst obuhvata:
 - sadržaj PSW (*Processor Status Word*) registra
 - trenutnu vrednost IP (*Instruction Pointer*) registra

Redovni profesor dr Dušan Regodić, dipl.inž.





Čuvanje konteksta procesora (2)

- ❑ Kontekst procesora se čuva u posebnom delu memorije koji se naziva **stek** (*stack*).
- ❑ Neposredno pre izvršavanja prekidne rutine, sadržaji PSW i IP registara se automatski šalju na stek. Po završetku prekidne rutine se, takođe automatski, reinicijalizuju.
- ❑ U praksi se na steku čuvaju i sadržaji pojedinih registara opšte namene, ali to mora da naloži korisnik (programer) softverskim putem.
- ❑ U zavisnosti od tipa prekida, programer utvrđuje koji registri mogu promeniti stanje tokom izvršavanja prekidne rutine i njih eksplicitno smešta na stek. Takođe, programer mora sam da reinicijalizuje ove registre softverskim putem po povratku iz prekidne rutine.

Redovni profesor dr Dušan Regodić, dipl.inž.

Stek

- Stek podržava LIFO (last in, first out) disciplinu pristupa, tj. sa steka se najpre uzima podatak koji je poslednji stavljen na stek.

vrh,dno steka →



Upis podatka u stek

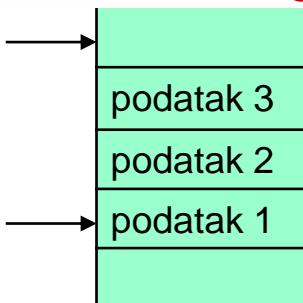
vrh steka →

dno steka → podatak 1



vrh steka →

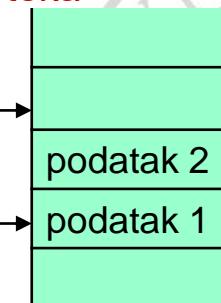
dno steka →



Čitanje podatka sa steka

vrh steka →

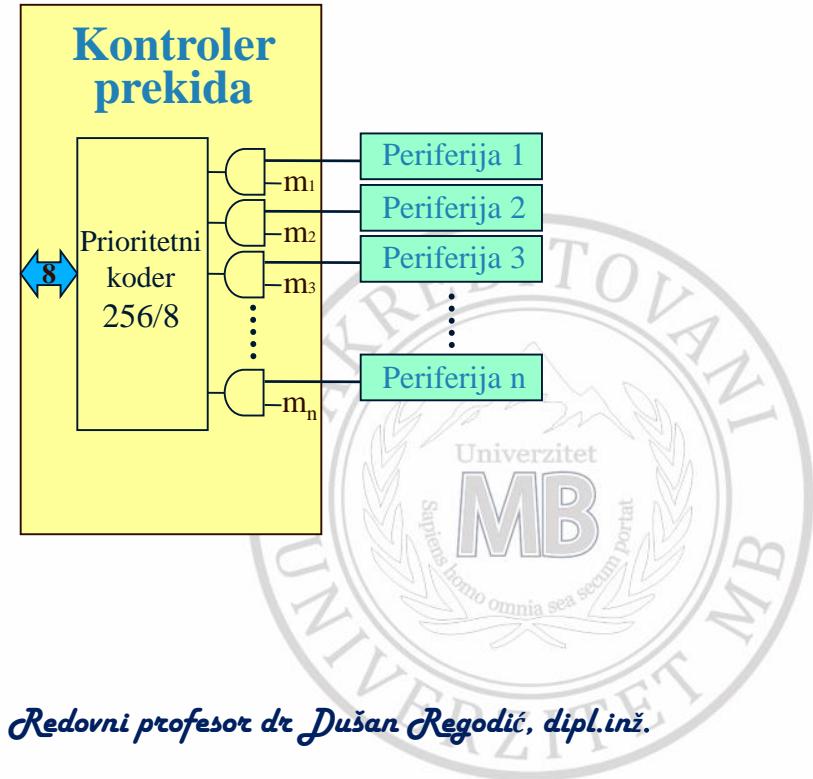
dno steka →



Redovni profesor dr Dušan Regodić, dipl.inž.

Maskiranje prekida

- ❑ Procesor Intel 8086 ima mogućnost da maskira pojedine prekide tako što kontroleru prekida šalje odgovarajuću komandu.
- ❑ Maskirani prekidi se ignorisu od strane kontrolera prekida i informacija o njihovom pojavljivanju se ne prosleđuje procesoru.
- ❑ Prekid se maskira bitom maske m_i na ulazu logičkih I kola tako što se taj bit postavi na vrednost 0.



Redovni profesor dr Dušan Regodić, dipl.inž.



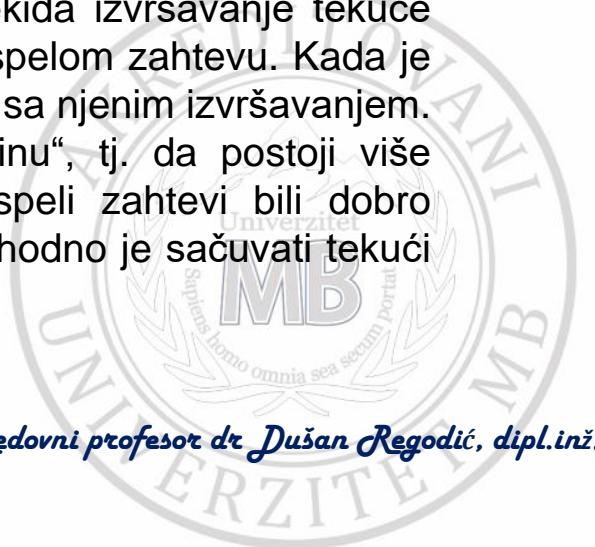
NMI

- ❑ Eksterni prekidi do procesora mogu stići i po **NMI** liniji. U ovom slučaju reč je o prekidima koji mogu da imaju katastrofalne posledice po procesor, na primer informacija o gubitku napajanja, greška prilikom transfera na magistrali, greška u radu memorije, itd.
- ❑ Prekidi koji stižu po **NMI** liniji su nemaskirajući, tj. ne postoji mogućnost da budu maskirani. Oni se ne mogu zabraniti ni IF flegom u PSW registru.
- ❑ Prekidi koji stižu po **NMI** liniji su višeg prioriteta od bilo kog prekida koji stiže po **INTR** liniji, što znači da u slučaju istovremenog pristizanja prekida sa obe linije, najpre se obrađuje prekid koji je stigao po **NMI** liniji.

Redovni profesor dr Dušan Regodić, dipl.inž.

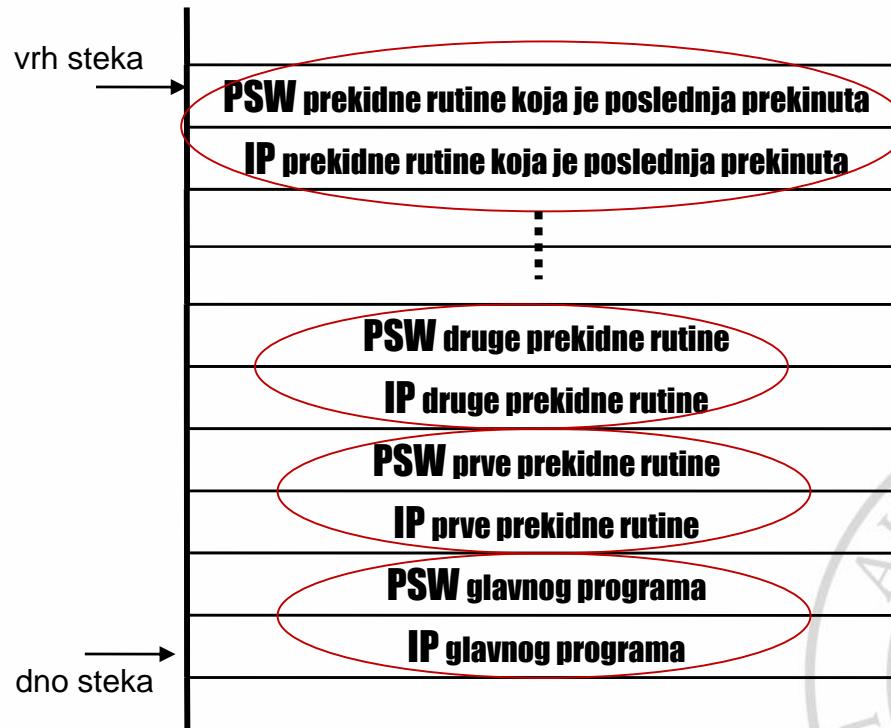
Opsluživanje prekida (1)

- ❑ Zbog velike raznovrsnosti mogućih zahteva za prekidom, uvedena je stroga disciplina u njihovom opsluživanju koja se zasniva na **prioritetima**.
- ❑ Najviši prioritet imaju interni prekidi, zatim spoljašnji koji dolaze po NMI liniji i na kraju spoljašnji koji stižu po INTR liniji.
- ❑ Hijerarhija u prioritetu je bitna ukoliko tokom izvršavanja jedne prekidne rutine dođe novi zahtev za prekidom. Ako je prispeli zahtev nižeg prioriteta, mora na sačaka da bi bio opslužen. Međutim, ako je višeg prioriteta, procesor prekida izvršavanje tekuće prekidne rutine i prelazi na prekidnu rutinu koja odgovara prispelom zahtevu. Kada je završi, procesor se vraća na rutinu koju je prekinuo i nastavlja sa njenim izvršavanjem. Ovakav način opsluživanja prekida može da ide i u „dubinu“, tj. da postoji više prekidnih rutina čije je izvršavanje u toku. Da bi svi prispeli zahtevi bili dobro opsluženi, pri svakom prelasku na novu prekidnu rutinu neophodno je sačuvati tekući kontekst procesora.



Redovni profesor dr Dušan Regodić, dipl.inž.

Opsluživanje prekida (2)



Redovni profesor dr Dušan Regodić, dipl.inž.

Istorija razvoja Intel familije procesora (1)

- Sedamdesetih godina prošlog veka, kada je firma **IBM** projektovala svoj prvi PC računar, na raspolaganju su bila dva proizvodača mikroprocesora: **Intel** i **Motorola**. Iako su Motorolini procesori po mišljenju IBM-ovih inženjera bili prihvativiji jer su imali bolje karakteristike, izbor je pao na Intel.
- Pri izboru su prevagnule sledeće činjenice:
 - Intel-ov procesor je bio jeftiniji
 - Intel-ov procesor je bio kompatibilan sa svim ranijim verzijama, a garantovana je i kompatibilnost sa svim budućim generacijama procesora
 - IBM je dobio prava za proizvodnju Intel-ovog procesora
 - za Intel-ov procesor je postojao gotov operativni sistem **CP/M** firme *Digital Research*, koji je u to vreme bio industrijski standard za mikroračunare



Redovni profesor dr Dušan Regodić, dipl.inž.

Istorija razvoja Intel familije procesora (2)

Intel 4004

- ❑ prvi Intel-ov procesor nastao u novembru 1971.godine
- ❑ upakovan u samo jedan čip, za čiju proizvodnju se koristila tada dostupna 10µm tehnologija
- ❑ veoma dobre karakteristike za to vreme
 - radni takt od 0.4 do 0.8 MHz
 - 4-bitna magistrala podataka
 - maksimum od 640 bajta eksterne memorije



Redovni profesor dr Dušan Regodić, dipl.inž.



Istorija razvoja Intel familije procesora (3)

Intel 8008

- ❑ procesor druge generacije nastao u aprilu 1972.god.
- ❑ glavna poboljšanja u odnosu na Intel 4004
 - 8-bitna magistrala podataka
 - maksimum od 16KB eksterne memorije

Intel 8080

- ❑ centralni deo prvog komercijalno raspoloživog računara – Altair
- ❑ u toku nekoliko meseci prodato ih je na hiljade komada, pa je vrlo brzo dospeo na prvo mesto liste prodaje
- ❑ osnovne karakteristike
 - radni takt od 2MHz
 - maksimum od 64KB memorije



Redovni profesor dr Dušan Regodić, dipl.inž.

Istorijski razvoj Intel familije procesora (4)

Intel 8086

- prvi 16-bitni procesor nastao sredinom 1978.godine, koga je IBM ugradio u svoj prvi personalni računar PC/XT
- proizvodio se u, za to vreme, veoma naprednoj 3µm tehnologiji
- osnovne karakteristike
 - radni takt u verzijama 2, 5, 8 i 10MHz
 - 20-bitna adresna magistrala omogućavala je pristup RAM-u od 1MB (tada se nije uočavala kasnija potreba za većom memorijom)



Redovni profesor dr Dušan Regodić, dipl.inž.

Istorija razvoja Intel familije procesora (5)

Intel 80286

- 16-bitni procesor nastao 1982.godine, kada ga je IBM ugradio u svoj *Advanced technology personalni računar PC/AT*
- za 6 godina proizvodnje instalirano je 15 miliona ovih PC računara
- bio je prvi “pravi” procesor; uveo je koncept zaštićenog moda (*protected mode*) koji omogućava da više programa rade istovremeno, nezavisno jedan od drugog (*multitasking*); ova mogućnost je kasnije našla primenu u operativnim sistemima, na pr. u Windows OS
- osnovne karakteristike
 - radni takt u verzijama 6, 10, 12, a kasnije i 20MHz
 - adresirao je 16MB RAM-a
 - imao je 134000 tranzistora



Redovni profesor dr Dušan Regodić, dipl.inž.



Istorija razvoja Intel familije procesora (6)

Intel 80386

- ❑ prvi 32-bitni procesor nastao krajem 1985.godine, koji predstavlja veliki tehnološki napredak u Intel-u
- ❑ do 1989.godine razvijena je cela familija ovih procesora; poslednja verzija mogla je da adresira 4GB RAM-a, a radni takt je bio 33MHz
- ❑ u ovom procesoru se prvi put koristio **pipeline** princip za izvršavanje instrukcija
- ❑ svi čipovi u 386 familiji bili su *pin-to-pin* kompatibilni
- ❑ u 386 familiji je postojala softverska kompatibilnost sa familijom 286
- ❑ familija 386 je ostvarila veliki napredak prvenstveno u razvoju čipova i uvela je standarde koje su mnogi kasniji čipovi pratili

Redovni profesor dr Dušan Regodić, dipl.inž.



Istorija razvoja Intel familije procesora (7)

Intel 80486

- 32-bitni procesor nastao početkom 1989.godine
- iako su prve verzije u 486 familiji radile na istom taktu kao poslednje verzije u familiji 386, 80486 je bio duplo brži zbog poboljšane arhitekture
- radni takt u verzijama od 25 do 100MHz (1994.god.)
- prvi procesor sa integriranom keš memorijom veličine 8KB u koju se korišćenjem pipeline-a stavljali sledeća instrukcija ili podatak, tako da procesor nije morao da pristupa spoljašnjoj memoriji; ovo je bio veliki pomak u brzini obrade podataka
- veliko poboljšanje je bilo i uvođenje integrisanog matematičkog koprocesora koji je procesoru omogućavao da aritmetičke operacije nad brojevima u pokretnom zarezu izvršava u jednom taktu

Redovni profesor dr Dušan Regodić, dipl.inž.

Istorija razvoja Intel familije procesora (8)

Pentium I mart 1993.g.	Promena naziva (nije 586) zbog pravnih problema. Radni takt do 233MHz. Dva nezavisna keša (code i data). Omogućeno izvršavanje dve instrukcije za vreme jednog takta. Od takta 75MHz, dva procesora mogla da rade zajedno u sistemu. Najpoznatiji Pentium Pro i Pentium MMX. 32-bitna unutrašnja magistrala podataka, 64-bitna spoljašnja magistrala (duplo brže nego kod 486).
Pentium II 1998.g.	Neuspeli pokušaj objedinjavanja najboljih osobina prethodnika. Radni takt do 450MHz. 32KB L1 keša (po 16KB za podatke i instrukcije) i do 512KB L2 keša (duplo sporiji od L1, ali mnogo brži od sistemске memorije).
Pentium III feb. 1999.g.	SSE set instrukcija (MMX-ov set proširen sa 70 novih instrukcija) kojim je poboljšano procesiranje 3D animacija. Radni takt do 1GHz. Da bi povećao sigurnost <i>online</i> transakcija, Intel je odlučio da na svaki procesor ugradi serijski broj čipa (<i>processor serial number - PSN</i>) koji je mogao da se pročita preko mreže ili <i>Internet-a</i> . Korisnici su smatrali da je to napad na njihovu privatnost i Intel je bio prinuđen da dopusti isključenje PSN-a u BIOS-u.
Pentium IV jun 2002.g.	Nova, NetBurst arhitektura CPU dizajnirana tako da u budućnosti omogući povećanje brzine . Net Burst se sastoji od 4 nove tehnologije : Hyper Pipelined Technology (brzina procesora se povećava proširenjem <i>pipeline-a</i>), Rapid Execution Engine (avedene dve ALU koje rade duplo brže), Execution Trace Cache (bile su neophodne prepravke na kešu) i 400MHz sistemska magistrala . Radni takt do 3.6GHz.

Redovni profesor dr Dušan Regodić, dipl.inž.



Važno!

Ova prezentacija je nekomercijalna. Slajdovi mogu da sadrže materijale preuzete sa Interneta, stručne i naučne građe, koji su zaštićeni Zakonom o autorskim i srodnim pravima. Ova prezentacija se može koristiti samo privremeno tokom usmenog izlaganja nastavnika u cilju informisanja i upućivanja studenata na dalji stručni, istraživački i naučni rad i u druge svrhe se ne sme koristiti.

Član 44 - Dozvoljeno je bez dozvole autora i bez plaćanja autorske naknade za nekomercijalne svrhe nastave: (1) javno izvođenje ili predstavljanje objavljenih dela u obliku neposrednog poučavanja na nastavi; - ZAKON O AUTORSKOM I SRODΝIM PRAVIMA ("Sl. glasnik RS", br. 104/2009 i 99/2011)

Dušan Regodić
dusanregodic5@gmail.com





**HVALA VAM
NA PAŽNJI**



Redovni profesor dr Dušan Regodić, dipl.inž.