

VII - Ulazno-izlazni podsistem

SADRŽAJ

7.1 Uloga ulazno-izlaznog podsistema

7.2 Klasifikacija U/I uređaja

7.3 Hardver značajan za U/I podsistem

7.4 Uniformni interfejs ka aplikacijama

7.5 Usluge koje obezbeđuje U/I podsistem

7.6 Performanse U/I sistema

7.1 - Uloga U/I podsistema

- Upravlja, kontroliše i koordinira rad U/I uređaja i operacija.
- Zadužen za komunikaciju procesa sa spoljašnjim svetom
- Predstavlja interfejs računara prema spoljašnjem svetu.
- Ostvaruje vezu između registara CPU i/ili operativne memorije, s jedne strane i perifernog uređaja, s druge strane
- Obezbeđuje što jednostavniji interfejs prema korisniku
- Dekodiranje adrese uređaja (i/ili izbor uređaja)
- Dekodiranje komandi (vrsta U/I komande)
- Privremeno čuvanje podataka (*buffering*)
- Upravljanje prenosom podataka i vremensko usklađivanje
- Izveštavanje centralnog procesora o stanju U/I modula
- Otkrivanje grešaka i izveštavanje procesora o njima

7.1 - Funkcije U/I podsistema

U/I podsistem treba da ostvari **četiri osnovne funkcije**:

1. **Vođenje evidencije** o statusu svih uređaja, što zahteva posebne mehanizme; jedan od najčešćih je ostvaren korišćenjem datoteke (kontrolnog bloka) koja je pridružena svakom od uređaja.
2. **Davanje dozvole za korišćenje uređaja**, odlučivanje o tome koliko dugo i kada će se U/I uređaj koristiti.

Postoji veliki broj tehnika kojima se ovo postiže:

- a) **dodeljivanjem** (*dedicated*) - uređaj se dodeljuje jednom procesu
 - b) **deljenjem** (*shared*) - jedan uređaj zajednički koriste više procesa
 - c) **virtuelizovanjem** - vrši se simuliranje fizičkog uređaja
3. **Alokacija** - fizičko dodeljivanje uređaja procesima sa kontrolnim jedinicama i kanalima preko kojih se uređaj povezuje u sistem
 4. **Dealokacija** (ukidanje) dozvole i tehnika. Dealokacija se može izvršiti na nivou **procesa** ili **posla** (*job*). Na nivou poslova, uređaj se dodeljuje jednom poslu sve dok se on ne završi, dok se na nivou procesa uređaj pridružuje samo u trenutku kada taj proces to zahteva.

7.1 - Problemi koji se rešavaju kod U/I

Upravljanje U/I predstavlja složen proces kontrole povezivanja i prenosa podataka između CPU i memorije sa jedne strane i perifernih uređaja (diskova, štampača, tastatura, modema i sl.) sa druge strane.

1. Razlike se odnose u **prikazu podataka** na pojedinoj jedinici, odnosno **kodovima** koji se upotrebljavaju. Periferije su složeni elektronski ili elektromehanički uređaji koji najčešće rade sa analognim signalima, dok su CPU i memorije digitalni uređaji koji koriste digitalne signale.
2. Razlika u **brzinama** kojim periferni uređaji izvode operacije. Brzina prenosa podataka (protok bitova) na U/I uređajima je mnogo manja od brzine prenosa u računaru.
3. Razlike u **načinu prenosa** podataka ka i od U/I uređaja.
4. Razlike u **operacijama** koje U/I uređaji mogu izvoditi. Razlike nisu samo u ulaznim, odnosno izlaznim operacijama nego i među jedinicama istog tipa postoje razlike, tako na primer magnetnu traku treba i možemo premotavati, dok kod diska ne postoji takva operacija

7.1 - Osnovni ciljevi U/I podsistema

- **Nezavisnost uređaja** – različiti modeli uređaja istog tipa, bez obzira na njihovog proizvođača, moraju se jednako posmatrati. Treba obezbediti i programsku nezavisnost od konkretne vrste U/I uređaja (podaci sa FD, HD, CD/DVD ili flash ROM-a se tretiraju isto)
- **Nezavisnost znakovnog koda** – programska podrška ne sme da bude zavisna od znakovnog koda (*fireware*) U/I uređaja. Korisniku se podaci predstavljaju u standardnom obliku putem translacione tabele.
- **Efikasnost** – U/I uređaju su znatno sporiji od ostalih računarskih resursa pa je poželjno da U/I podsistem obezbedi konkurentnost u radu
- **Uniformni interfejs ka aplikacijama i korisniku** – potrebno je omogućiti jednostavan, po mogućstvu uniformni rad sa U/I uređajima kako bi korisnicima pojednostavili rad.

7.2 - Klasifikacija U/I uređaja

- **Namena uređaja** - za dugotrajno **skladištenje podataka** (HD, DVD), za **prenos podataka** (modem, mrežna kartica) i U/I uređaji za **spregu sa korisnikom** (tastatura, miš, monitor)
- **Smer prenosa** - ulazni, izlazni i ulazno/izlazni
- **Način prenosa** - serijski i paralelni.
- **Jedinici prenosa** - podaci se mogu prenositi **bajt po bajt** (*stream of bytes*) kod terminala ili kao veliki **blokovi** kod diskova. Dakle, **prema jedinici prenosa** U/I uređaje možemo podeliti na **karakter uređaje** (*Character devices*) i **blok uređaje** (*Block devices*).
- **Metoda pristupa** - sekvencijalni pristup (magnetne trake) i direktni ili slučajni pristup (HD, CD, DVD)
- **Mogućnost deljenja** - nedeljivi (monopol.), deljivi i virtualni U/I uređaji
- **Brzina prenosa** (*data rate*) – brzi i spori U/I periferni uređaji
- **Mogućnost upisa** - samo čitanje (CD, DVD) samo pisanje (monitori, štampači) i čitanje i pisanje (HD)

7.2 - Klasifikacija U/I uređaja

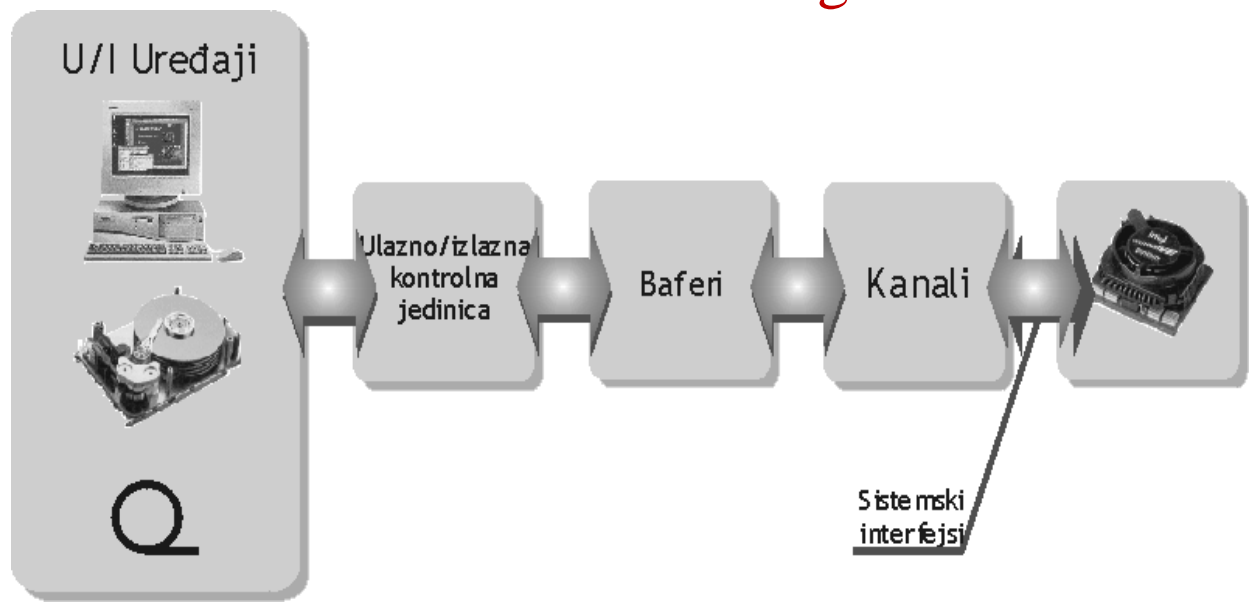
- **Prema stanju** – u zavisnosti od stanja u kome se nalazi proces kada koristi U/I uređaj može se izvršiti podela na **blokirajuće** i **neblokirajuće**.
- **Način rukovanja greškama** (*Error Handling*) - OS može da vrati podatke koji su čitani sa HD kada uređaj nije raspoloživ zbog privremene nemogućnosti upisa. Većina sistema vraća broj greške ili kod kada U/I zahtev ne može da se izvrši. Izveštaj se čuva u datoteci *System error log*.
- **Časovnik i tajmer** - predstavlja posebnu klasu U/I uređaja koji treba da obavi sledeće funkcije:
 1. Prikazivanje tekućeg vremena
 2. Prikazivanje proteklog vremena
 3. Tajmerski okidač za neku operaciju **X** u nekom trenutku *t*

Koristi se kod **realizacije tehnike predpražnjenja** (*preemption*) kod raspoređivanja procesa

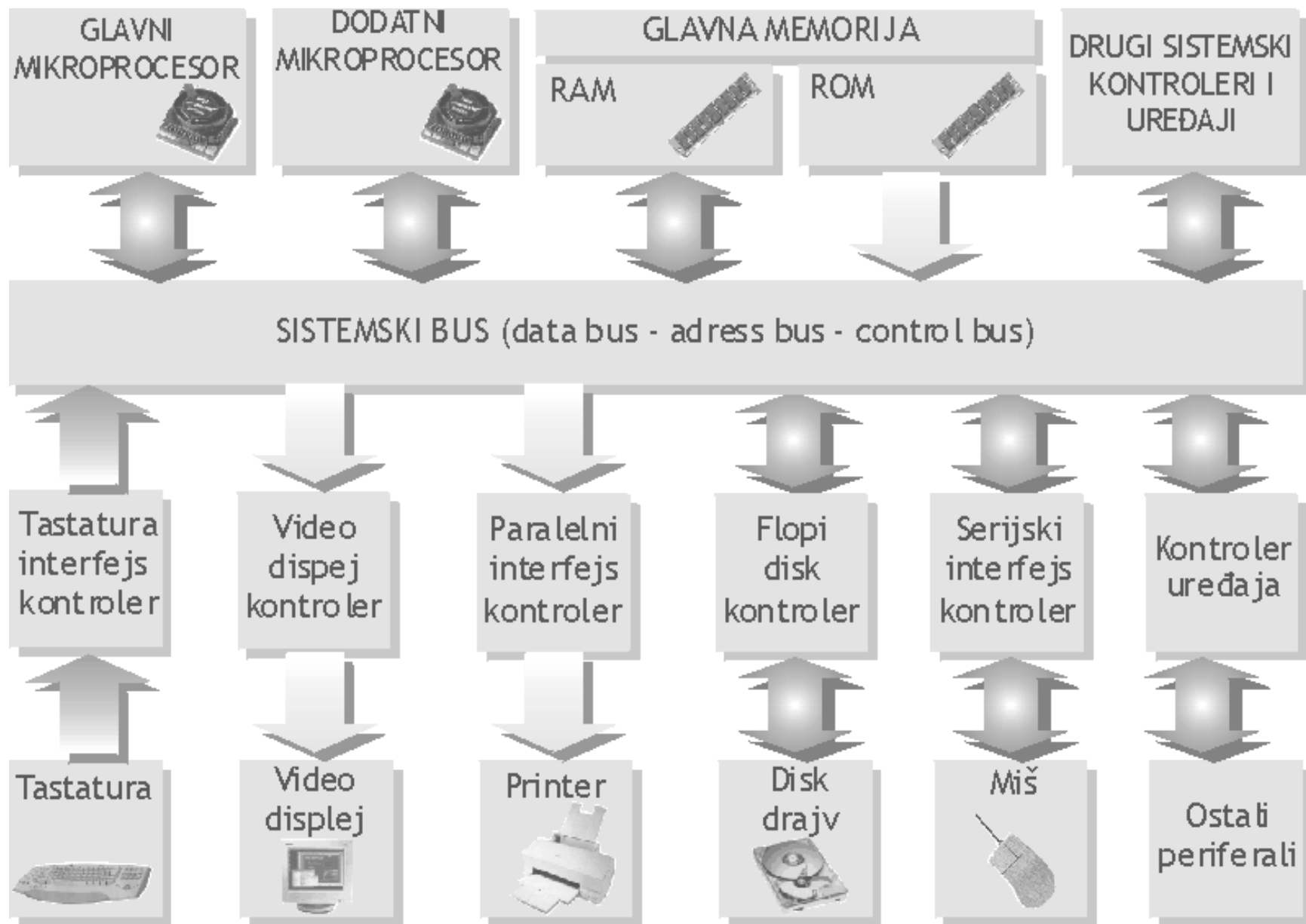
7.3-Hardver značajan za U/I podsystem

- Sve jedinice računarskog sistema povezane su međusobno unificiranim i standardizovanim sistemom veza – **interfejsima**.
- **Interfejs** je univerzalni naziv za **skup linija za prenos informacija i elektronskih kola** koji upravljaju prenosom signala po linijama.
- U računarskom sistemu postoje sledeći interfejsi: **operativna memorija - procesor, procesor - kanal, U/I interfejs (kanal - kontroler periferne jedinice)** i **interfejsi perifernih jedinica**.
- Deo systemske magistrale (bus-a) koji povezuje periferne uređaje sa centralnim delom računara zove se **ulazno/izlazna magistrala**.

- Ona je povezana sa svim periferijama preko njihovih **U/I kontrolnih jedinica** (kontrolera).



7.3-Hardver značajan za U/I podsystem



7.3-Hardver značajan za U/I podsistem

- **Kontroler** svakog perifernog uređaja ima svoj dekođer - sklop koji prepoznaje kod svog perifernog bloka.
- Kada registruje "svoj kod" kontroler **obezbeđuje puteve između U/I magistrale i svog perifernog uređaja.**
- Računar inače generiše kod koji se sastoji iz tri grupe podataka: **operacije, adrese uređaja i funkcijskog koda** (komande za kontroler).
- Načini rada kontrolera: **priključak** (*port*), **magistrala** (*bus*) i **uređaj**
- Svaki kontroler ima **jedan ili više registara** preko kojih komunicira
- Registrima se može pristupiti putem **U/I i memorijskim naredbama**
- Tipičan kontroler sadrži sledeće registre:
 - 1. Kontrolni registar** - postavlja se režim rada, CPU upisuje podatke
 - 2. Statusni registar** - opisuje trenutno stanje, CPU samo čita podatke
 - 3. Ulazni registar** - služi da se smeste podaci koji su očitani
 - 4. Izlazni registar** - služi da se smeste podaci koji su namenjeni nekom U/I uređaju

7.3-Hardver značajan za U/I podsistem

- Rad centralnog procesora i operativne jedinice, sa jedne i perifernih jedinica sa druge strane, usklađuje se **korišćenjem posrednika - kanala**.
- **Kanali** predstavljaju mikroprocesore specijalne namene, čiji je osnovni zadatak da, u skladu sa primljenim naredbama CPU, **izvrše zahtevane ulazno/izlazne operacije** i izvrše **prenos podataka ka ili od U/I uređaja** i to bez daljeg angažovanja CPU.
- **Rad kanala kontroliše CPU** tako što samo inicijalizuje operaciju kojom nadalje upravljaju kontrolne jedinice i upravljački uređaji kanala
- **Time se procesor oslobađa** i njegov se rad ne prekida.
- Tek po **završetku rada** perifernih jedinica, **šalje se informacija** o tome procesoru. Ovo je inače, karakteristika multiprogramskog rada.
- **Kanal može istovremeno da opslužuje jednu ili više U/I jedinica.**
- Ako radi sa više perifernih jedinica kaže se da radi u **multipleksnom režimu rada**.
- Multipleksiranje se vrši **u vremenu**, tako što se jedinični intervali dodeljuju pojedinim perifernim jedinicama.

7.3 - Vrste prenosa podataka kod I/O

➤ Da bi prenos podataka između I/O uređaja i CPU ili memorije računara bio omogućen, **U/I kontroler mora da obavi sledeće funkcije:**

- ✓ dekodira adrese
- ✓ prepozna upravljačke signale
- ✓ prihvatiti podatke (npr. iz ulaznog uređaja) i pripremiti ih za dalji prenos (npr. prema memoriji)
- ✓ otkriva greške u prenosu
- ✓ omogući **vremensku sinhronizaciju** svih navedenih aktivnosti

➤ **Načini adresiranja** perifernih uređaja su:

- a) **Memor.preslikavanje registara** U/I kontrolera (*Memory mapped I/O*)
- b) **Posebno adresiranje** U/I kontrolera (*Isolated I/O*).

➤ **Prenos podataka između računara i I/O periferija** delimo na:

1. **Programirani prenos podataka**
2. **Prekidni prenos podataka**
3. **Direktni pristup memoriji (DMA)**

7.3 - Programirani prenos podataka

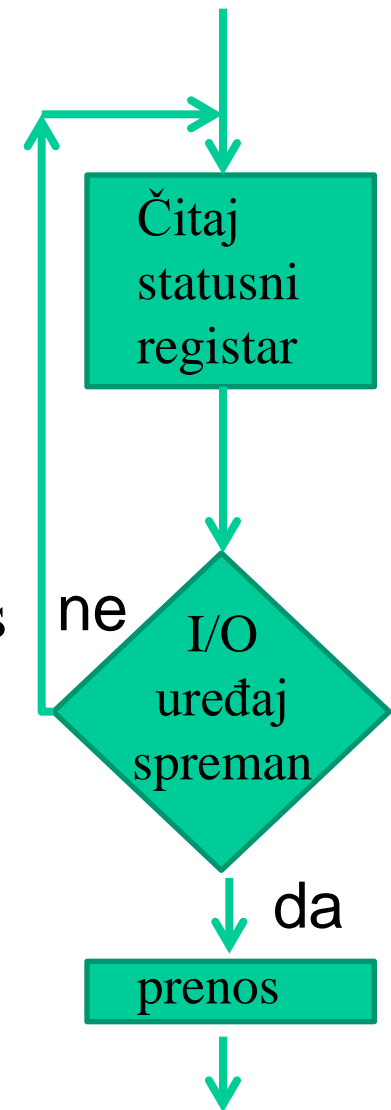
➤ U glavni program uključeni su i **prekidi za vreme kojih CPU obavlja komunikaciju sa perifernim jedinicama** (očitanje nekog stanja, davanje nekih novih podataka, korekcija periferne jedinice...)

➤ Osnovne vrste programiranog U/I prenosa:

1. bezuslovni U/I prenos - najjednostavniji, ali i najnesigurnija metoda prenosa. Prenos podataka između računara i I/O uređaja izvodi se a da se pre toga ne proverava da li je I/O uređaj spreman za prenos

2. uslovni U/I prenos – prvo se proverava da li je I/O uređaj spreman za komunikaciju. Ako nije, CPU u petlji čeka sve dok se ne stvore uslovi za prenos podataka. U/I kontroler mora imati dodatni registar - statusni registar koji otkriva stanje I/O uređaja.

3. prozivka (polling) - vrši se ciklično prozivanje I/O uređaja koji su spojeni na jedan CPU



7.3 - Programirani prenos podataka

Prednosti:

- postoji relativno **jednostavan program**
- sigurnost prenosa koji se odvija **najvećom mogućom brzinom** tj. čim je I/O uređaj spreman
- **ne trebaju dodatna pristupna kola** (čipovi).

Nedostaci:

- **Vreme čekanja** nekih uređaja je kraće od vremena odziva i tada treba primeniti neki drugi način prenosa
- **CPU nepotrebno troši vreme na prozivanje I/O uređaja** koji nisu tražili prenos podataka

Primena:

1. Koristi se za prenos **male količine podataka** bajta ili reči.
2. Koristi se u slučaju kad je **I/O uređaj uvek spreman za prenos podataka** a njegovo **vreme odziva poznato i stalno**.
3. Primeri su **analogno-digitalni pretvarači**.

7.3 - Tehnika prekida

- U ovom načinu rada I/O šalje zahtev za prekidom na poseban - prekidni ulaz CPU (može da bude jedan ili više prekidnih ulaza)
- Nakon primljenog zahteva CPU završava tekuću instrukciju (zaustavlja glavni program), smešta stanja internih registara u magacin, zabranjuje novi prekid (prekid prekida) i prelazi na program za obradu prekida
- Nedostatak ovog načina je gubitak vremena dok CPU ustanovi sa koje jedinice je došao zahtev za prekid.
- To se rešava vektorskim prekidom.
- U ovom načinu rada periferna jedinica uz zahtev za prekidom šalje adresu potprograma sa koje glavni program treba da nastavi da radi (potreban je poseban kontroler tkz. *interrupt controler*).
- Još neki nedostaci ovog načina su:
 1. poskupljenje uređaja radi posebnih kola koja šalju prekidnu adresu
 2. asinhroni rad, što znači da zahtev za prekidom može doći bilo kada pa obrada prekida mora biti programski dobro napravljena da ne bi došlo do gubitka podataka bitnih za glavni program.

7.3 - Tehnika prekida

S obzirom na izvore prekida, razlikujemo:

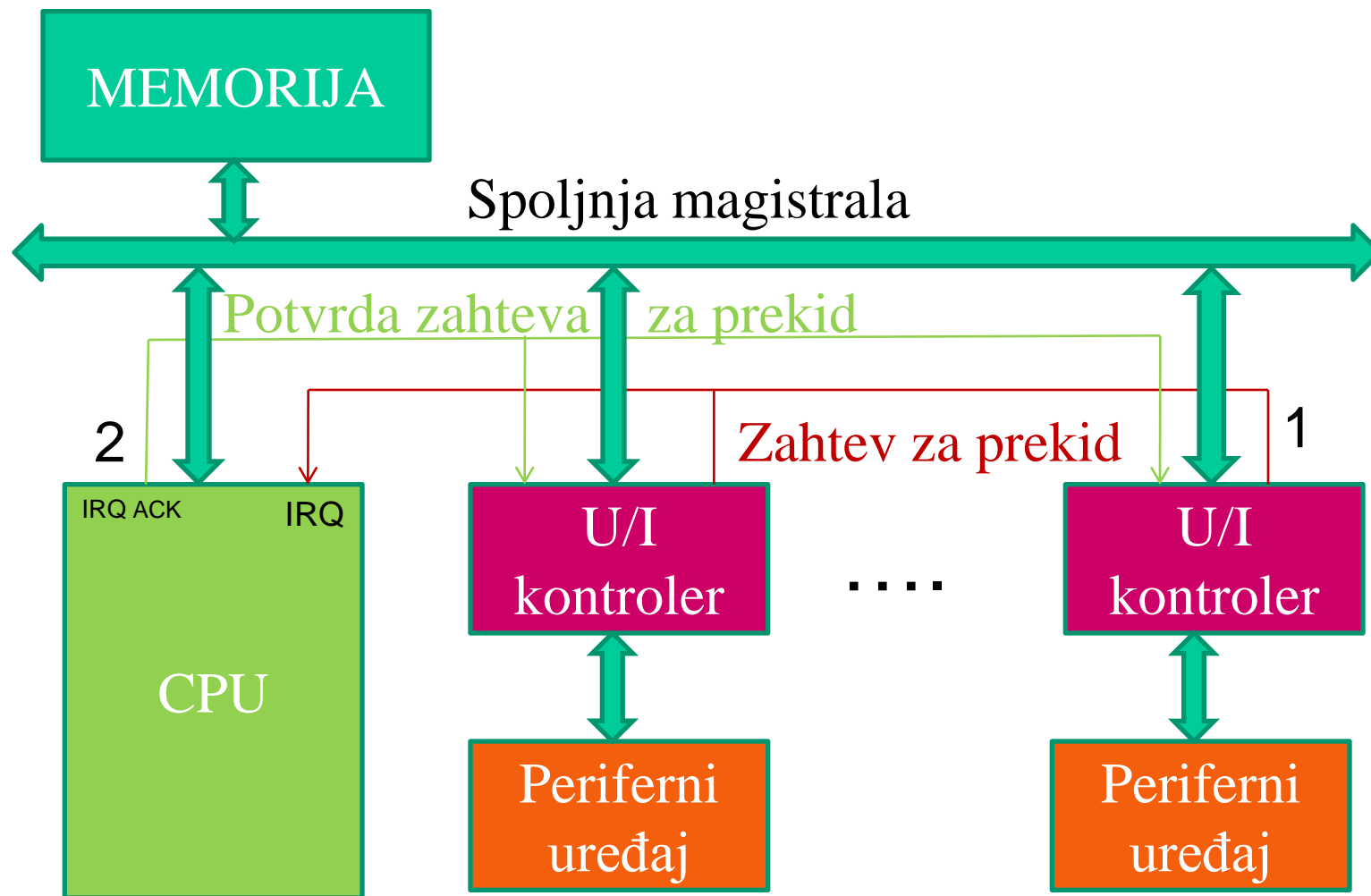
1. **U/I prekide** (*hardware interrupt*) – pripadaju grupi **maskirajućih prekida** i predstavljaju zahteve za U/I operacijom
2. **Vremenske prekide** (*timer interrupts*) – osiguravaju da se neke operacije ne izvršavaju previše dugo – nemaskirajući prekidi
3. **Prekide zbog grešaka** – imaju najviši prioritet jer greške mogu dovesti do isključivanja računara – nemaskirajući prekidi
4. **Programske prekide** (*software interrupts*) – generišu ih programi i takođe spadaju u nemaskirajuće prekide
5. **Sistemske pozivi SVC** (*supervisory call, system call*) – generišu ih programi kada traže neku pomoć OS tj. njegovog jezgra prilikom prelaska iz korisničkog u supervizorski režim rada
6. **Izuzetci** (*Exception*)

Prekidnim U/I prenosom otklanjaju se nedostaci programir. prenosa. Nije potrebno poznavati vreme odziva I/O uređaja, a CPU ne mora gubiti vreme čekajući ili prozivajući I/O uređaje.

7.3- Tehnika prekida

1. U/I uređaj **generiše zahtev za prekid IRQ**, i preko U/I kontrolera šalje ga CPU
 2. CPU nakon završetka tekuće instrukcije **utvrđuje hoće li prihvatiti ili ne**. Ako prihvata zahtev, CPU odgovara signalom potvrde prekida **IACK** (*Interrupt acknowledge*). Od vrste prekida zavisi da li će CPU postaviti prekidnu masku kojom se onemogućavaju drugi prekidi.
 3. U magacin se **smeštaju sadržaji programskog brojača i sadržaji registra**
 4. U prog.brojač se **upisuje adresa prekidnog programa** (vektor prekida)
 5. **Izvodi se prekidni program**
 6. Nakon izvršenja prekidnog programa **izvodi se naredba za povratak iz prekida - RTI** (*return from interrupt*), koja ima zadatak da obnovi sadržaje program.brojača i registara na stare/upamćene vrednosti, kako bi se omogućilo da ranije prekinuti program nesmetano nastavi da radi.
- Kod **više istovremenih zahteva za prekid** oni će se obrađivati od najvažnijeg-najviši prioritet, prema najmanje važnom-najniži prioritet
 - Prekidanje prekidne rutine naziva se **gnježđenje prekida** (*nesting*).

7.3- Tehnika prekida



Prekidna struktura s jednom linijom zahteva za prekid

7.3 - DMA (Direct Memory Access)

- DMA rad je analogan izvršenju "*Multiple Data Move*" instrukcije.
- DMA se izvršava od strane porta hosta a ne od strane CPU-a.
- Operandi koji su neophodni za DMA se zovu DMA parametri i oni su:
 1. **memorijska adresa** - adresa glavne memorije ka ili iz koje se podaci mogu preneti (upisivati/čitati).
 2. **dužina bloka** - broj koji predstavlja veličinu oblasti glavne memorije u/iz koje treba preneti (upisivati/čitati) podatak.
 3. **adresa uređaja** - adresa U/I uređaja za memorisanje.
- Postoje 2 tipa DMA u zavisnosti koliki je broj prenetih podataka:
 1. **DMA sa jednostrukim ciklusom** - obavlja se samo jedan prenos u toku jedne DMA operacije (korak 3). Prednost - kratko zauzeće magistrale
 2. **Paketni (burst) DMA** - u toku jedne DMA operacije vrši se prenos većeg broja podataka. Kontroler uređaja mora biti u stanju da **baferuje** (privremeno čuva) podatke. Prednost - **veća brzina i manja cena** logike, a mana što se za duži period **zauzima sistemska magistrala**.

7.3 - DMA (Direct Memory Access)

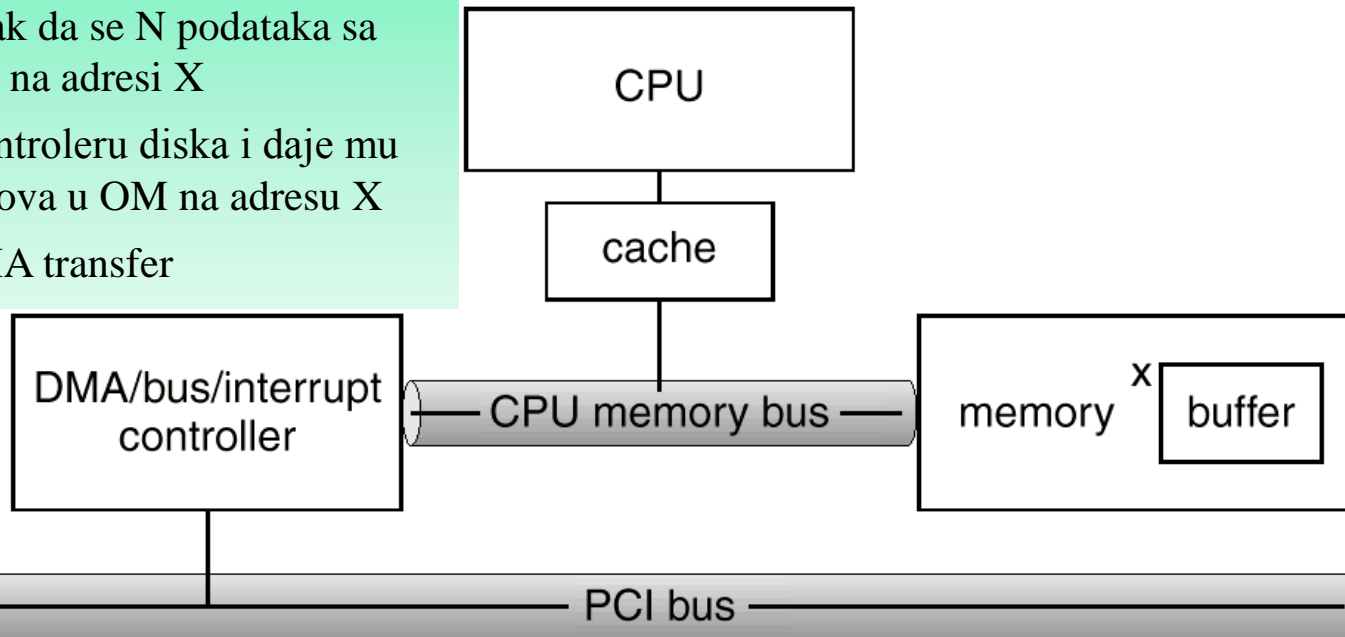
Primer: DMA operacija kod štampanja odvija se na sledeći način:

1. vrši se prenos DMA parametara ka portu hosta i startuje se štampač.
2. kako je registar podataka štampača prazan, port uređaja zahteva da se u registar podataka upiše jedan znak, čime su ostvareni uslovi da štampanje može da počne; ovo se izvodi preko DMA kanala.
3. DMA kanal u portu hosta zahteva od CPU upravljanje nad sistemskom magistralom, kada se magistrala dodeli DMA prenos počinje.
4. DMA operacija kojom se vrši punjenje registra za podatke obuhvata:
 - a) postavljanje memorijske adrese na sistemsku magistralu,
 - b) obavlja se operacija čitanje memorije,
 - c) upisuje se pročitani podatak u registar podataka,
 - d) oslobađa se sistemski magistrala,
 - e) povećava se memorijska adresa za naredni podatak u nizu,
 - f) smanjuje se vrednost brojača prenetih podataka.
5. Naredne akcije koje se preduzimaju zavise od vrednosti brojača i to ako je Brojač $\neq 0$, sledi korak 3 a ako je Brojač=0, zaustavlja se štampač i generiše se prekid koji signalizira da je operacija štampanja završena.

7.3 - DMA prenos

DMA prenos podataka sa diska u operativnu memoriju

1. Disk drajver dobija zadatak da se N podataka sa diska prenese u memoriju na adresi X
2. Disk drajver se obraća kontroleru diska i daje mu zadatak da prenese N bajtova u OM na adresi X
3. Disk kontroler inicira DMA transfer
4. Nakon uspostave veze disk kontroler šalje svaki bajt DMA kontroleru



5. DMA kontroler prenosi N bajtova na adresi X u OM, povećava memorijske adrese ($X+1$) i smanjuje broj prenetih bajtova ($N-1$)
6. Kada $N=0$, DMA putem prekidnog signala (*interrupt*) obaveštava CPU da je transfer završen

7.4-Uniformni interfejs ka aplikacijama

- **Drajver** (*driver*) je softver koji upravlja uređajem.
- Jedan drajver može da opsluži više primeraka uređaja iste klase.
- Drajveri uređaja koji nisu ni blokovski ni znakovni se **razlikuju od drajvera klasifikovanih uređaja**.
- Van drajvera su vidljive samo operacije, kao što su, **operacije ulaza ili izlaza**, koje omogućavaju jednoobrazno korišćenje U/I uređaja.

Tipične operacije drajvera **blokovskih uređaja** su:

1. operacija **inicijalizacije**
2. operacije **ulaza i izlaza blokova** - koristi sloj za upravljanje datotekama

Tipične operacije drajvera **znakovnih uređaja** su:

1. **operacija inicijalizacije**,
2. **operacije ulaza i izlaza znakova**
3. **upravljačka operacija** - omogućuju dinamičko podešavanje funkcion. karakteristika znakovnih uređaja (brzine prenosa znakova).

- Za adresu svake od ovih operacija predviđeno je posebno polje u elementu **tabele drajvera** koja definiše svaki drajver.

7.5-Usluge koje obezbeđuje U/I sistem

- **Raspoređivanje U/I operacija** – podrazumeva određivanje dobrog rasporeda kod izvršavanja U/I operacija (primer čitanja HD)
- **Baferovanje** – predstavlja deo memorije koji čuva trenutne podatke.
 - ✓ Usklađivanje različitih brzina
 - ✓ Prilagođavanje različitih veličina transfera podataka
 - ✓ Održavanje semantike kopiranja – obezbeđuje da korisnički podaci ne mogu da se promene prilikom kopiranja
- **Keširanje** – oblast brze systemske memorije koja čuva kopije podataka i time poboljšava performanse pristupa tj. R/W prema I/O uređaju
- **Spuler** – bafer koji **privremeno čuva podatke koji su namenjeni nekom nedeljivom uređaju** (šampač). Koristi se tehnika *spuling* (*Simultaneous Peripheral Operations On-Line*) kojom se na disku formira vituelni uređaj u koji se upisuju podaci u formatu za taj uređaj
- **Upravljanje greškama** – U/I uređaji mogu otkazati rad iz mnogih razloga. Neke od njih OS može sam da otkloni (**privremene greške**), a neke nije u stanju (**trajna greška**) pa samo obaveštava korisnika o njoj.

7.6 - Performanse U/I podsistema

1. **Nivo aplikativnog softvera** – programski kod je fleksibilan i najčešće bezopasan po sistem. Na ovom nivou **nije potrebno ponovno punjenje drajvera ni resetovanje sistema** ali se zato i ne mogu koristiti osnovne funkcionalne strukture jezgra kao što su niti, poruke i td.
2. **Nivo drajvera** – ovde je moguće koristiti sve strukture podataka jezgra a samim tim i bolje performanse. **Problem je što bilo koja greška može dovesti do pada sistema**
3. **Nivo hardvera** – nivo koji omogućava da se **najviše poboljšaju performanse U/I podsistema** (RAID strukture). Problem je što se ovde programira na **najnižem nivou** (assembler) tako da je veoma teško to uraditi sa nekim višim programskim jezicima. Samim tim **razvoj traje veoma dugo** i zahteva se specijalna oprema za testiranje.

Hvala na pažnji !!!



Pitanja

? ? ?