



CS120 - ORGANIZACIJA RAČUNARA

Struktura I/O sistema

Lekcija 08

PRIRUČNIK ZA STUDENTE

CS120 - ORGANIZACIJA RAČUNARA

Lekcija 08

STRUKTURA I/O SISTEMA

- ✓ Struktura I/O sistema
- ✓ Poglavlje 1: Struktura I/O sistema
- ✓ Poglavlje 2: Karakteristike uređaja za skladištenje podataka
- ✓ Poglavlje 3: Karakteristike kontrolera hard diskova
- ✓ Poglavlje 4: I/O magistrale za hard diskove
- ✓ Poglavlje 5: RAID kontroleri
- ✓ Poglavlje 6: Pokazne vežbe
- ✓ Poglavlje 7: Zadaci sa samostalnim radom
- ✓ Poglavlje 8: Domaći zadatak
- ✓ Ulaz/Izlaz- Opšta razmatranja

Copyright © 2017 – UNIVERZITET METROPOLITAN, Beograd. Sva prava zadržana. Bez prethodne pismene dozvole od strane Univerziteta METROPOLITAN zabranjena je reprodukcija, transfer, distribucija ili memorisanje nekog dela ili čitavih sadržaja ovog dokumenta., kopiranjem, snimanjem, elektronskim putem, skeniranjem ili na bilo koji drugi način.

Copyright © 2017 BELGRADE METROPOLITAN UNIVERSITY. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, without the prior written permission of Belgrade Metropolitan University.

▼ Uvod

UVOD

Postoje dva osnovna načina za realizaciju ulaza/izlaza kod mikroračunara. Kod prvog od njih, izolovanog ulaza/izlaza, čitanje U/I portova i upis u njih vrši se posebnim operacijama.

Komunikacija između računara i njegovog okruženja ostvaruje se preko U/I uređaja. U/I uređaji su delovi mašine namenjeni za čuvanje informacija, obavljanje konverzije podataka koji se prihvataju iz okruženja u oblik pogodan za čitanje od strane računara, i konverziju podataka na svojim izlazima u formi pogodnoj i razumljivoj za okruženje.

U/I uređaje koji se povezuju na CPU zovemo zajedničkim imenom *U/I sistem*. Jedan kontroler uređaja (**device controller**) i prateće uređaje koji se povezuju na kontroler zovemo **U/I podsistem** (**I/O subsystem**).

Kontroler uređaja može da predstavlja jednostavan interfejs i nalazi se između CPU-a i U/I uređaja. U nešto složenijem izvođenju može da upravlja radom nekoliko uređaja koji mogu biti nezavisni od rada CPU-a. U svim slučajevima kontroleri uređaja omogućavaju realizaciju funkcija upravljanja i baferovanja koje su neophodne za korektan rad povezanih U/I uređaja.

Kontroler uređaja može biti fizički u U/I uređaju, CPU-u, ili pakovan kao posebna jedinica. Neki od kontrolera uređaja mogu upravljati radom samo jednog uređaja, jer, ili je obično samo jedan uređaj takvog tipa potreban za rad sistema, ili zbog performansnih razloga (npr. veća brzina u radu kada se samo po jedan uređaj vezuje na jedan kontroler). Često, ipak, kontroler može biti zajednički za veći broj uređaja istog tipa, jer najveći broj uređaja nije istovremeno aktivan, tako da kontroler može da radi u multiprogramskom režimu rada.

Postoje dva osnovna načina za realizaciju ulaza/izlaza kod mikroračunara. Kod prvog od njih, izolovanog ulaza/izlaza, čitanje U/I portova i upis u njih vrši se posebnim ulaznim, odnosno izlaznim operacijama. Kod drugog načina, memorijski preslikanog ulaza/izlaza, U/I portovi se smatraju lokacijama u memoriji, tako da se čitanje i upis vrši istim naredbama kao i čitanje i upis u glavnu memoriju.

▼ Poglavlje 1

Struktura I/O sistema

STRUKTURA I/O SISTEMA I VEZA SA OSTALIM DELOVIMA RAČUNARA

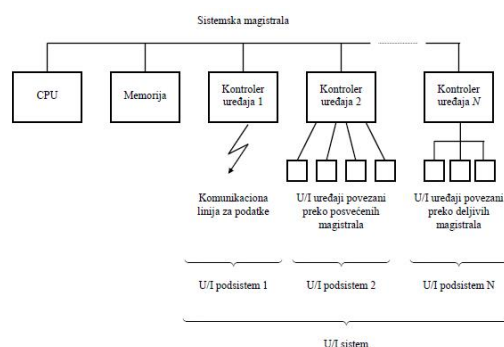
Spoljni uređaji često koriste različite formate podataka i dužine reči nego što je to slučaj sa računarom na koji su priključeni.

Pored procesora i memorije, treći ključni element računarskog sistema jeste U/I podsistem koga čini skup U/I modula. Svaki U/I modul predstavlja interfejs između systemske magistrale i izvesnog broja U/I uređaja. U/I modul nije samo skup mehaničkih veza već poseduje izvesni "stepen inteligencije", odnosno logiku koja vrši komunikaciju između systemske magistrale i perifernih uređaja.

Razlozi za postojanje U/I modula su: Postojanje mnogo vrsta spoljnih uređaja sa različitim principima rada, Brzina prenosa podataka spoljnih uređaja je obično mnogo manja nego kod procesora ili memorije, Spoljni uređaji su mnogo brži od memorije ili procesora. Spoljni uređaji često koriste različite formate podataka i dužine reči nego što je to slučaj sa računarom na koji su priključeni.

I/O modul ima dve osnovne funkcije:

Sprega procesora i memorije preko systemske magistrale. - Sprega sa jednim ili više perifernih uređaja (Slika 1).

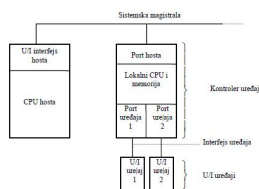


Slika 1.1 Struktura U/I sistema i veza sa ostalim delovima računara. [Izvor: Autor]

Kontroler uređaja može da predstavlja jednostavan interfejs i nalazi se između CPU-a i U/I uređaja. U nešto složenijem izvođenju može da upravlja radom nekoliko uređaja koji mogu biti nezavisni od rada CPU-a. U svim slučajevima kontroleri uređaja omogućavaju realizaciju funkcija upravljanja i baferovanja koje su neophodne za korektan rad povezanih U/I uređaja. Kontroler uređaja može biti fizički u U/I uređaju, CPU-u, ili pakovan kao posebna jedinica. Neki od kontrolera uređaja mogu upravljati radom samo jednog uređaja, jer, ili je obično samo jedan uređaj takvog tipa potreban za rad sistema, ili zbog performansnih razloga (npr. veća brzina u radu kada se samo po jedan uređaj vezuje na jedan kontroler). Često, ipak, kontroler

može biti zajednički za veći broj uređaja istog tipa, jer najveći broj uređaja nije istovremeno aktivan, tako da kontroler može da radi u multiprogramskom režimu rada.

Opšta struktura U/I podsistema prikazana je na slici 2.



Slika 1.2 Opšta struktura U/I podsistema. [Izvor: Autor]

KLASIFIKACIJA I/O UREĐAJA

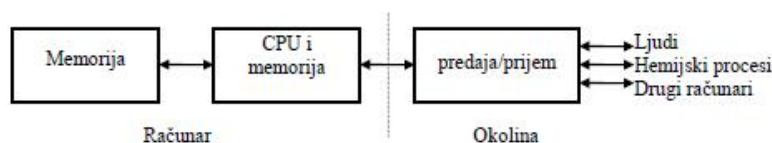
Skoro svaki računar opšte namene se može konfigurisati za specifičnu aplikaciju dodavanjem U/I uređaja koji su prilagođeni za tu aplikaciju.

U/I uređaji

U/I uređaji poseduju veliki broj osobina koje ih čine različitim od CPU-a ili memorije. Ova razlika je pre svega rezultat mehaničke prirode U/I uređaja (što čini da oni budu manje pouzdani i znatno sporiji u odnosu na elektronske uređaje) i raznovrsnosti u načinima njihove realizacije.

Klasifikacija

Skoro svaki računar opšte namene se može konfigurisati za specifičnu aplikaciju dodavanjem U/I uređaja koji su prilagođeni za tu aplikaciju. Ove U/I uređaje, koje često zovemo uređaji, možemo klasifikovati na sledeći način: **uređaji za memorisanje** - često se koriste od strane sistema za memorisanje podataka i njihovo povezivanje (čitanje), i **predajno/prijemni** (source/sink - pravi prevod je izvor/ponor) - koji se prvenstveno koriste od strane sistema za komuniciranje sa njegovim okruženjem (slika 3).



Slika 1.3 Veza računara i okruženja . [Izvor: Autor]

Uređaji za memorisanje

Ovi uređaji se mogu posmatrati kao memorijsko proširenje. Imajući u vidu da je memorija sistema realizovana elektronskim komponentama relativno skupa, i da je njen sadržaj nestalan nakon nestanka napajanja (misli se na RAM-ove), skoro kod svih sistema se danas ugrađuje mehanički uređaj za memorisanje. Svaki uređaj za memorisanje se karakteriše kapacitetom memorisanja.

Primopredajni uređaji

Svrha primopredajnih uređaja je da obezbede komuniciranje između računara i njegovog okruženja. Prijem se ostvaruje preko uređaja tipa tastatura, analizator govora, digitajzer, miš i dr., ili preko uređaja tipa modem, senzor ili prekidač. Predajni uređaji, a to mogu biti štampači, grafički displeji, modemi, aktuatori i dr., generišu izlaznu informaciju. Komunikacione linije koje se koriste za spregu sa drugim računarima ili računarskim mrežama mogu se takođe smatrati primopredajnim uređajima.

OSOBINE I/O UREĐAJA

Osobine uređaja koje imaju direktan uticaj na rad računarskog sistema su vreme pristupa, broj podataka koji se prenosi u jedinici vremena (propusnost) i procenat grešaka koje se javljaju.

Osobine U/I uređaja

Osobine uređaja koje imaju direktan uticaj na rad računarskog sistema su vreme pristupa, broj podataka koji se prenosi u jedinici vremena (propusnost) i procenat grešaka koje se javljaju u toku rada.

Vreme pristupa

Medijumi za memorisanje karakterišu se sekvencijalnim a ne proizvoljnim pristupom. Podatak kome pristupamo ne može se trenutno pročitati (ili se trenutno upisati) nego je najpre potrebno neko vreme da se obavi određeno mehaničko kretanje. Ovo vreme se zove *vreme pristupa*. Na primer, kod pomeranja glave diska, vreme pristupa čini vreme potrebno da se pozicionira glava (*seek time*) i vreme dok željeni sektor ne prođe izpod glave (*rotational delay*). Vreme pristupa U/I uređajima je značajno duže od vremena pristupa glavnoj memoriji (oko 25ms za pomeranje glave na disku prema 0,5μs za pristup glavnoj memoriji, a to znači da je taj odnos reda $5 \cdot 10^4$). Sa druge strane, vreme pristupa može takođe da varira značajno, ne samo zbog različitih uređaja (pristup informaciji na disku je brži od onog na magnetnoj traci), nego i zbog različitih tipova podataka koji se čuvaju na jednom istom uređaju.

Pronalaženje proizvoljne reči na magnetnoj traci može da traje dugo, ali nalaženje sledeće reči može biti znatno kraće. Zbog toga, prirodno je da uređaji za memorisanje čitaju i upisuju podatke u blokovima, tako da pristup jednom podatku povlači sa sobom i pristup većem broju podataka (celom bloku podataka).

Vreme za prenos podataka

Vreme za prenos podataka U/I uređaja zavisi od iznosa podataka koji se prenose (obim bloka), i brzine prenosa podataka (vreme između sukcesivnih elemenata podataka). Obim bloka je određen tipom uređaja. Znakovno orijentisani uređaji (terminali) predaju svoje podatke bajt po bajt, dok blok orijentisani uređaji (diskovi ili trake) vrše prenos blokova podataka. Obim bloka je određen faktorima kao što su: vreme pristupa, vreme prenosa, fragmentacije, i zahtevom za baferovanje i dr.

Brzina prenosa podataka je često mala u odnosu na prenos tipa memorija-memorija. Na primer, brzina prenosa ka/iz diska može biti 1MB/s. Ako usvojimo da je sektor na disku obima 1kB, za prenos jednog sektora biće potrebno 1ms.

VEROVATNOĆA GREŠKE

Komunikacioni uređaji su često podložni greškama, zbog dugih neoklopljenih linija, koje su podložne radijacijama i drugim tipovima smetnji.

Verovatnoća greške

Zbog mehaničke prirode najvećeg broja uređaja za memorisanje, javlja se veći procenat greške u toku prenosa podataka u odnosu na onaj koji važi za prenos CPU↔memorija. Zbog fizičkog kretanja medijuma na kome je upisana informacija, velike gustine pakovanja (bitovi/inču), čestica u vazduhu, amortizacije i tolerancije javljaju se greške. Komunikacioni uređaji su često podložni greškama, zbog dugih neoklopljenih linija, koje su podložne radijacijama i drugim tipovima smetnji.

Detekcija i korekcija ovih grešaka zahteva neki oblik hardverske redundancije kao i obezbeđenje neke mogućnosti da se pomoću softverskih sredstava ove greške isprave. Ovo se izvodi korišćenjem dodatnih bitova za proveru (korišćenjem kodova za korekciju greške), povratnom predajom primljene poruke (echo), ili ponovnim prenosom poruke.

OBRADA I/O PREKIDA I UPRAVLJAČKI PROGRAMI

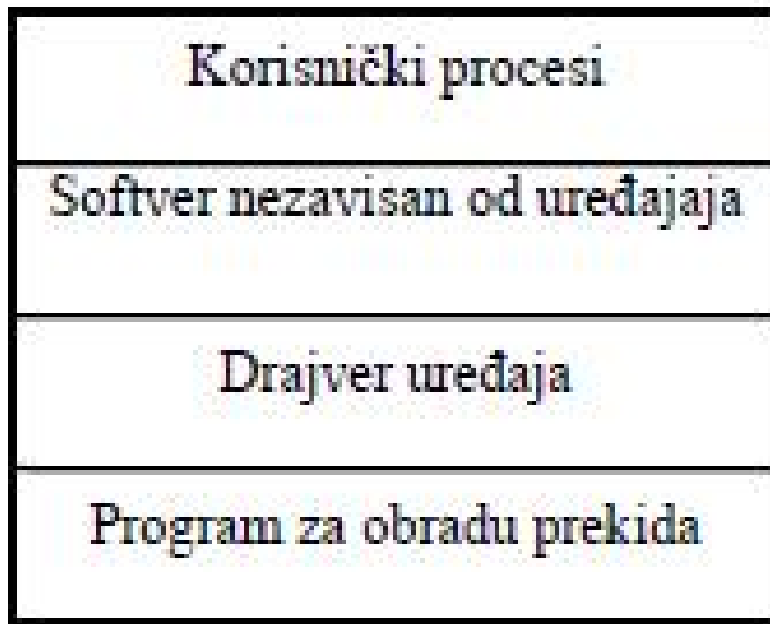
Glavna uloga operativnih sistema je da upravlja svim U/I uređajima.

I/O i operativni sistem računara

Glavna uloga operativnih sistema je da upravlja svim U/I uređajima. Upravljanje se sastoji u korišćenju U/I komandi i obradi prekida i grešaka. Pored toga, operativni sistem treba da obezbedi visok nivo intrefejsa prema uređajima.

Interfejs treba da bude konzistentan za sve tipove uređaja kao i da bude jednostavan za korišćenje. Da bi ispunio ovo, U/I softver se organizuje na način kao što je prikazano na slici 4, tj. organizuje se u nivoe.

Na najnižem nivou se nalazi program za obradu prekida (**interrupt handler**). Program za pokretanje uređaja (**device driver**) sadrži sve kodove koji su karakteristični za taj tip uređaja. U suštini, to je jedini deo operativnog sistema koji prepoznaje prikačeni tip uređaja, kao što je tip korišćene U/I komande, fizička organizacija uređaja, mehaničke osobine uređaja i način koji ukazuje koje akcije treba preuzeti kada se jave greške u uređaju.



Slika 1.4 Organizacija I/O softvera . [Izvor: Autor]

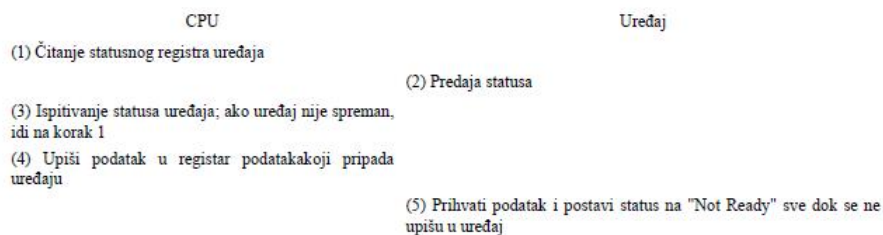
DIREKTNI (PROGRAMIRANI) I/O

Direktni (programirani) I/O je najprostiji oblik U/I-a, gde CPU i kontroler uređaja formiraju jedinstveni proces, čime se omogućava bilo kakav oblik konkurentnog rada.

Direktni (programirani) I/O

To je najprostiji oblik U/I-a, gde CPU i kontroler uređaja formiraju jedinstveni proces, čime se omogućava bilo kakav oblik konkurentnog rada. Sledeći program prikazuje sekvencu koraka koja je potrebna da se ostvari U/I operacija tipa upis.

Koraci 1-3 formiraju petlju u kojoj CPU ispituje U/I uređaj. Ovo je neophodno zbog sinhronizacije CPU-ovog rada sa radom kontrolera uređaja koji se odnosi na taj proces je prikazan na slici 6.



Slika 1.5 Sekvenca koraka koja je potrebna da se ostvari U/I operacija tipa upis . [Izvor: Autor]

▼ Poglavlje 2

Karakteristike uređaja za skladištenje podataka

ELEKTROMEHANIČKE KOMPONENTE DISKOVA

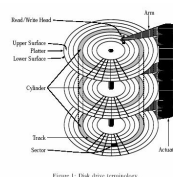
Unutrašnje staze, staze bliže centru imaju manju površinu pa samim tim i manje magneta.

Elektromehaničke komponente

Disk uređaj se sastoji od rotacionih kružnih ploča koje rotiraju oko zajedničke ose. Obe površine ploče su presvučene magnetnim materijalom. Svaka površina ima pridruženu upisno/čitajuću glavu i za većinu disk uređaja samo jedna glava može biti aktivna u jednom trenutku vremena. Glave se linearno pokreću pomoću sopstvenog servo-sistema, kao na slici 1.

Oblik podataka na disku (Disk layout)

Minimalna količina informacija na disku je blok ili sektor koji po pravilu obuhvata 512 bajtova. Svaka kružna površina se deli na koncentrične krugove koji se zovu staze (tracks), a staze se dalje dele na sektore. Skup staza na podjednakom rastojanju od centralne ose, predstavlja strukturu koja se naziva cilindar. Osnovi parametar koji opisuje stazu je gustina staze (track density), odnosno broj sektora po stazi. Magnetna tehnologija sve više napreduje, pa je gustina staze sve veća, što iziskuje potrebu efikasnijeg korišćenja magneta.



Slika 2.1 Interna struktura diska. [Izvor: A. S. Tanenbaum, Structured computer organization, 6th edition, Pearson Prentice Hall, 2012.]

Unutrašnje staze, staze bliže centru imaju manju površinu pa samim tim i manje magneta. U tom kontekstu, za razliku od prvih generacija diskova, uvodi se tehnika zona na disku, gde se cilindri grupišu u zone iste gustine (BZR = bit zone recording). Na taj način se povećava i kapacitet diska kao i brzina čitanja sa medijama, ali kontrolna elektronika diska ima mnogo složeniji zadatak da upravlja promenljivom gustinom staza, odnosno kompleksnijim oblikom podataka na disku.

Logičko-fizičko mapiranje LBN->CHS

Tradicionalno mapiranje jednodimenzionalne adrese LBN (Logical Block Number) u trodimenzionalnu fizičku adresu adresu CHS (cylinder/head/sector) se obavlja počevši od LBN=0, koja se pridružuje prvom bloku na početnom cilindru (outermost) i onda se LBN blokovi dodaju sukcesivno, sve do punog kapaciteta diska. Međutim brojni faktori utiču na dosta izmena i osetno povećanje složenosti u mapiranju. Ti faktori su:

Rezervisani prostor na disku (reserved space for firmware)

Većina savremenih disk uređaja rezerviše veoma mali deo diska za sopstveni firmver i to je obično ili na samom početku ili na samom kraju diska

STAZNO I CILINDERSKO POMERANJE (TRACK SKEW & CYLINDER SKEW)

Logički pomeraj se takođe dešava između poslednje staze svakog cilindra i prve staze narednog cilindra (cylinder skew). Pomeraji se definišu za svaku zonu posebno.

BZR (Bit Zone Recording)

Kao što je već opisano BZR omogućava povećanje performansi i kapaciteta savremenih diskova, ali je zato firmveru koji upravlja diskom prilično otežan posao. Za svaki LBN se najpre mora odrediti zona na disku kojoj pripada, a onda se u datoj zoni saglasno njenoj gustini određuje CHS (cylinder/head/sector).

Stazno i cilindarsko pomeranje (track skew & cylinder skew)

Kako za aktiviranje glave na susednu stazu potrebno izvesno vreme (head switch time), prvi logički blokovi na susednim stazama se pomeraju (track skew) u odnosu na prve fizičke blokove, inače može doći do gubitka skoro cele rotacije pri sekvencijanom pristupu disku. Logički pomeraj se takođe dešava između poslednje staze svakog cilindra i prve staze narednog cilindra (cylinder skew). Pomeraji se definišu za svaku zonu posebno.

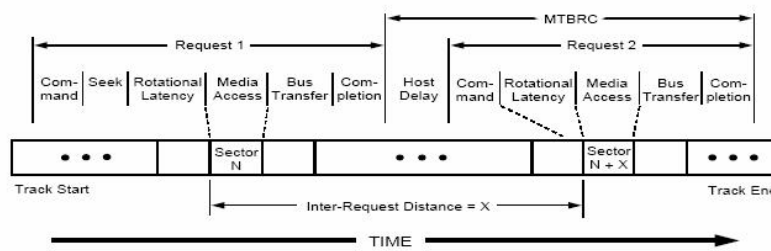
Upravljanje defektima (defect management)

Savremeni disk uređaji omogućavaju 100% naznačenog kapaciteta i pored značajnog broja defekata na mediumu koji se pojavljuju bilo u fabrikaciji uređaja bilo naknadno u toku rada disk uređaja.

Da bi se omogućilo 100% iskorišćenje kapaciteta, savremeni diskovi imaju rezervne regione na mediumu (spare regions). Taj prostor se rezerviše na kraju staza, cilindra i zona kako bi se defektivnim blokovima našla zamena. Upravljanje defektima se odvija na 2 načina, tehnikom klizanja (slipping) i tehnikom ponovne dodele LBN adrese (relocation). Tehnika klizanja se obavlja na 2 nivoa, klizanje na nivou bloka (sector slipping) i klizanje na nivou staze (track slipping) i ta tehnika se obavlja jedino na nivou fizičkog formatiranja celog disk uređaja (low-level formatting).

Komponente disk transfera

Na slici 2 su prikazane komponente disk transfera:



Slika 2.2 Komponente disk transfera. [Izvor: A. S. Tanenbaum, Structured computer organization, 6th edition, Pearson Prentice Hall, 2012.]

SEEK FAZA (POZICINIRANJE UPISNO/ČITAJUĆIH GLAVA)

Vreme poziciniranja je složena funkcija koja je za bliže distance može aproksimirati funkcijom kvadratnog korena, dok se za veće distance može aproksimirati linearnom funkcijom.

Komanda faza

Ova faza obuhvata pakovanje komande registre diska, odnosno u task-file-register disk kontrolera. Kontroler-inicijator kreira više bajtnu komandu, koju šalje u određenom poretku u task-file-register diska.

Seek faza (Poziciniranje upisno/čitajućih glava)

Da bi pristupio zahtevanom sektoru, servo mehanizam mora da najpre obavi pozicioniranje sa tekuće pozicije na zahtevani cilindar (**seek time**), potom treba aktivirati odgovarajuću glavu (**head switch**), a onda treba sačekati rotaciju (**rotational latency**), odnosno da se zahtevani sektor postavi na poziciju glave. Vreme poziciniranja je složena funkcija koja je za bliže distance može aproksimirati funkcijom kvadratnog korena, dok se za veće distance može aproksimirati linearnom funkcijom, slika 3.

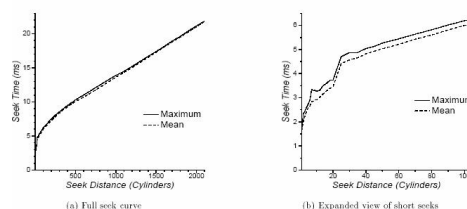


Figure 3: Seagate ST41601N: Extracted seek curves.

Slika 2.3 Tipične seek krive. [Izvor: A. S. Tanenbaum, Structured computer organization, 6th edition, Pearson Prentice Hall, 2012.]

Proizvođači diskova deklarišu tri vremenska parametra vezana za pozicioniranje:

Vreme punog pozicioniranja (**Full stroke seek**)

Deklariše se kao vreme potrebno da glava diska obavi maksimalno pozicioniranje sa nultog na poslednji cilindar.

Vreme srednjeg pozicioniranja (average seek or random seek)

To je vreme potrebno da se servo mehanizam pozicionira sa trenutne pozicije na bilo koju poziciju i to vreme je reda veličine 10 msec kod savremenih diskova. Ovo vreme se navodi samo u karakteristikama diska kao globalno, inače se ne može koristiti sa simulaciju. Umesto konstatnog vremena, uvodi se kriva pozicioniranja kao najbolji reprezent funkcije pozicioniranja. Kriva je složena i nepravilna, relativno se može dobro aproksimirati na 2 načina:

a) korenskom (sqrt) funkcijom za bliske distance

b) linearnom funkcijom za dalje distance

VREME ROTACIONOG KAŠNENJA (ROTATIONAL LATENCIES)

Vreme linearnog pozicioniranja je vreme potrebno da se servo mehanizam pozicionira, sa trenutne pozicije na susedni cilindar i iznosi nekoliko msec (1-5 msec).

Vreme linearnog pozicioniranja (linear seek or track-to-track seek)

To je vreme potrebno da se servo mehanizam pozicionira, sa trenutne pozicije na susedni cilindar i iznosi nekoliko msec (1-5 msec). Ovo vreme je bitno, kada disk zahtev uključuje više sukcesivnih cilindra istovremeno.

Vreme rotacionog kašnjenja (rotational latencies)

To je vreme potrebno da se servo mehanizam pozicionira na početni blok uključen u zahtev. Uzima se prosečno rotaciono kašnjenje, koje je jednako vremenu polovine jedne rotacije diska. Na primer, u slučaju kada brzina okretanja iznosi 3600 rev/min, vreme rotacionog kašnjenja iznosi 8msec.

Trotate_min=0

Trotate_max=one revolution

Trotate_avg=1/2 (one revolution)

Vreme aktiviranja glave (head switch time)

To je vreme potrebno da se nova glava aktivira i pojava se uvek dešava kad se promeni površina magnetnog mediuma.

Media transfer faza

Sa površine diska se može preneti najviše podataka koliko iznosi kapacitet jedne staze za jednu revoluciju, tako da brzina transfera zavisi od kapaciteta cilindra i od brzine okretanja diska. Maksimalna brzina transfera u ovom slučaju jednaka:

$$V_{\text{disk_to_disk_buffer}} = \text{Capacity_of_Track} / \text{Time_of_one_revolution}$$

Iz izraza je jasno da gušće popunjene zone imaju veću brzinu transfera. Tipične veličine su od 20MB/s do 100MB/s.

Bus transfer faza

To je praktično maksimalna brzina interfejesa, koji ne zavisi od mehaničkih parametara diska. Brzina transfera je reda veličine 1-320 MB/sec, za paralelne interfejse, a 3Gb/sec za serijske disk interfejse.

Statusna faza

To je faza u kojoj disk priprema i emituje inicijatoru statusnu informaciju. To vreme kao i vreme trajanja komande faze je relativno malo i dosta ga je teško izmeriti precizno.

UPRAVLJANJE SPOLJNIM INTERFEJSOM (EXTERNAL INTERFACE MANAGEMENT)

Svaki disk pripada određenom interfejsu, odnosno perifernjaskoj magistrali (SCSI, IDE, IPI). Svaki interejs ima svoj skup signala, komandi, poruka i razne vrste precizno definisanih ciklusa.

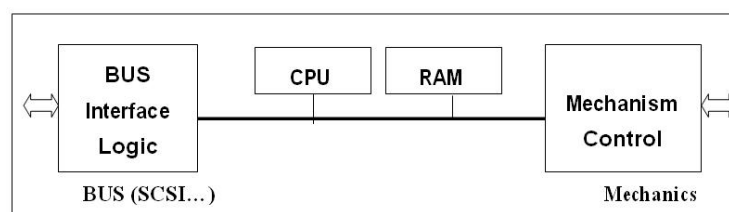
Kontrolna elektronika diska (On-Board Control Logic)

Kontrolna elektronika diska je prikazana na slici 4.

Bazirana na sopstvenom procesoru, koji izvršava firmver zadužen za kompletno upravljanje svim delovima disk uređaja. Za savremene diskove, ugrađeni procesori su tipične jačine 32 bitnog CPU. Pored sopstvenog procesora, sve je veća količina RAM memorije prisutna na kontrolnoj ploči diska, koja ima više namensku ulogu. Najvažnije funkcije firmvera odnosno kontrolne ploče su:

Upravljanje spoljnim interfejsom (External Interface Management)

Svaki disk pripada određenom interfejsu, odnosno perifernjaskoj magistrali (SCSI, IDE, IPI). Svaki interejs ima svoj skup signala, komandi, poruka i razne vrste precizno definisanih ciklusa na magistrali.



Slika 2.4 Kontrolna elektronika diska. [Izvor: A. S. Tanenbaum, Structured computer organization, 6th edition, Pearson Prentice Hall, 2012.]

Postoje 4 glavne interakcije između diska i periferijske magistrale: startovanje komande (**request arrival**), završetak komande (**request completion**), prenos podataka (data transfer) i isključenje/priključenje na magistralu (disconnect/reconnect), koje omogućava aktiviranje više diskova u vremenu i svaki uređaj se priključuje na magistralu samo kada treba da prenosi podatke, dok u ostalim slučajevima, kada nema transfera, oslobađa magistralu za druge uređaje.

Obrada komande (Request Processing)

Kada kontrolna elektronika dobije komandu od nadređenog kontrolera, komande se mora inicijalno obraditi odnosno dekodirati, što zahteva izvesno vreme (**command overhead time**). Vreme dekodiranja zavisi od više faktora kao što su vrsta ciklusa (**read/write**), od veličine transfera, od stanja u kešu na samom disku (**on-board cache hit/miss**).

Keširanje na disk uređaju (On-Board Cache)

U savremenim disk uređajima prisutna je relativno velika količina RAM memorije reda veličine više MB, koja obavlja dve glavne funkcije. Prva i osnovna funkcija je usklađivanje brzine transfera disk mediuma i disk interfejsa (**speed-matching buffer**), a tu funkciju imaju praktično svi disk uređaji i stari i novi.

▼ Poglavlje 3

Karakteristike kontrolera hard diskova

KOMPAKTNI KONTROLERI

Bazirani su na savremenom stepenu razvoja integriranih kola kada imamo pojavu jako složenih i kvalitetnih periferijskih kontrolera u vidu jednog integrisanog kola (on-chip design).

Analizirajući savremene disk kontrolere, namenjene za PC arhitekturu, po pitanju složenosti, funkcionalnosti i performansi deli se na tri funkcionalne grupe. To su: **1. Kompaktni kontroleri** (Compact controllers) **2. Baferski kontroleri** (Tracking-Buffer controllers) **3. Potpuni keš kontroleri** (Full-Caching controllers)

Kompaktni kontroleri

Bazirani su na savremenom stepenu razvoja integriranih kola kada imamo pojavu jako složenih i kvalitetnih periferijskih kontrolera u vidu jednog integrisanog kola (on-chip design). Njihova velika prednost je jednostavnost, pouzdanost i niska cena. Kvalitetna periferijska IC kola omogućavaju jednostavnu, kompaktnu realizaciju disk kontrolera, koji su relativno jeftini i pouzdani. Pri tome, implementirajući firmversko-softverska prilagođenja i ubrzanja (firmware driver + device driver), mogu se za partikularno hardversko-sistemske okruženje postići relativno pristojne performanse. Kompaktni disk kontroleri su pogodni za jednog korisničke operativne sisteme dok kod višekorisničkih sistema ne mogu dati neke bolje rezultate.

Kompaktni disk kontroleri zahtevaju kvalitetna periferijska IC kola koja poseduju dovoljnu funkcionalnost, da samostalno upravljaju perifericom (one chip design). Takva IC kola su dostupna i realna i sa takvim izborom periferijskog IC kola, realizacija kompaktnog kontrolera zahteva još elementarnu dekodersku logiku za selekciju kao i bafersku logiku za spregu periferijskog IC kola sa sistemskom magistralom. Pri tome se može upotrebiti baferska memorija za usklađivanje između sistemske magistrale i periferala. Usklađivanje se odnosi na nivou brzine i na nivou veličine podataka.

Glavne karakteristike kompaktnih kontrolera su da nemaju poseban lokalni procesor za upravljanje kontrolerom i ne poseduju veliku bafersku memoriju na kontrolerskoj kartici.

Baferski kontroleri (FIFO architecture)

Baferski kontroleri su optimalna ili kompromisna klasa za većinu operativnih sistema i većinu aplikativnih primena, kada je u pitanju kompromis između performansi koje postižu i cene koštanja. Baferski kontroleri su bolji od kompaktnih kontrolera mada su i skuplji. Sadrže relativno veliku bafersku memoriju koja je organizovana na FIFO principu.

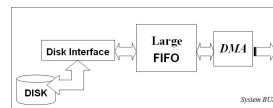
BAFERSKI KONTROLERI (FIFO ARCHITECTURE)

Karakteristično za ovu klasu kontrolera je implementacija lokalnog procesora koji obavlja upravljanje kontrolerom i upotreba veće količine memorije koja je realizovana na FIFO principu

Upotreba velike baferske memorije, omogućava izuzetno dobro usklađivanje brzina diska i sistemske magistrale, kao jednu redukovanu vrstu keširanja diska, koja se obezbeđuje tehnikom čitanja unapred (**read-ahead**). FIFO arhitektura uključuje u sebi jednostavnije lokalne procesore za upravljanje FIFO baferom i radom kontrolera. Razlog za to je što se kod ovog tipa kontrolera ne implementiraju složeniji algoritmi upravljanja, jer ne poseduje keš memoriju, bar ne pravu keš memoriju, mada i FIFO bafer funkcioniše kao neka vrsta keš memorije.

FIFO kontroler ima malo vreme za procesiranje komande (**controller command overhead**), jer ne primenjuje složene keš algoritme, koji intenzivno pretražuju gde su podaci, da li su validni itd. U odnosu na keš kontrolere (**full caching controller**), imaju daleko manje memorije na kartici, tipično od 32K ili 64K.

Konstruktori ovakvog tipa kontrolera, ističu superiornost baferskih kontrolera nad konkurencijom kada su u pitanju sekvencijani pristupi disku. Cena im je daleko niža od potpunih keš kontrolera (**full caching**) kontrolera. Na slici 1 je prikazana je blok šema FIFO arhitekture.



Slika 3.1 FIFO arhitektura. [Izvor: A. S. Tanenbaum, Structured computer organization, 6th edition, Pearson Prentice Hall, 2012.]

Karakteristično za ovu klasu kontrolera je implementacija lokalnog procesora koji obavlja upravljanje kontrolerom i upotreba veće količine memorije koja je realizovana na FIFO principu. Upotreba velike FIFO memorije, omogućava rasterećenje sistemske magistrale, jer se kompletan prenos podataka obavlja sa diska u FIFO memoriju, bez obzira na dužinu transfera, a na drugoj strani iz FIFO memorije, podaci se prebacuju u operativnu memoriju sa daleko većom brzinom prenosa. Pri tome se sistemska magistrala ne opterećuje direktno sa disk transferom, koji je spor, a istovremeno se FIFO (**dualport FIFO**) prazni daleko većom brzinom preko DMA kontrolera, ili to obavlja glavni CPU.

Rastererećenje procesora je značajno, jer kompletno upravljanje kontrolerom obavlja lokalni procesor. Kod baferskih kontrolera upotrebljava se relativno velika baferska memorija, reda veličine od 16KB ili 32KB do 128KB.

Pored funkcije da prihvataju podatke sa diska i istovremeno oslobađuju podatke na drugoj strani FIFO bafera, usklađujući pri tome brzine periferala i sistemske magistrale, velika FIFO memorija na kontroleru se koristi za keširanje diska. U suštini to nije pravo keširanje, jer je FIFO bafer nezgodan, u smislu što ne može lako i brzo da se pristupi bilo kom delu podataka koji su u njemu.

KEŠ KONTROLERI (FULL CACHING ARCHITECTURE)

Keš kontroleri predstavljaju najsloženije i najkvalitetnije kontrolerske kartice.

Keširanje diska se obavlja tehnikom čitanja unapred (**read lookahead**), a sam proces obavlja lokalni procesor. Disk kontroler, kada dobije zahtev za čitanje određenog broja sektora, uvek pročita više nego što se zahteva (obično celu stazu ili cilindar) u nadi da će sledeći zahtev biti sekvencijalan. Ukoliko sledeći zahtev bude sekvencijalan, tada se kontroler ne obraća disku, već podatke prosleđuje iz FIFO bafera (**memory to memory cycle**). Takođe kontroler može u neaktivnom stanju da preko svog lokalnog procesora obavi predikciju sledećeg disk zahteva i pročita sektore u FIFO bafer.

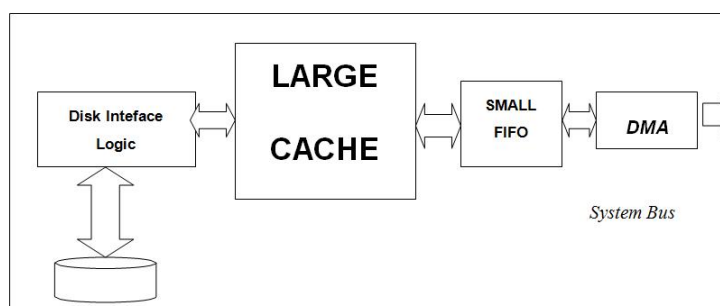
Keširanje na ovakav način nije pravo, ili totalno, keširanje jer se podaci primaju ili prosleđuju na FIFO principu ili na modifikovanim FIFO principima što u svakom slučaju otežava pretraživanja ili pristup podacima unutar FIFO bafera. Ovakva koncepcija keširanja daje izuzetno dobre rezultate kod sekvencijalnih pristupa disku.

Baferski kontroleri (**tracking-buffer controllers**) nemaju složene keš algoritme i njihove rutine su jednostavnije u odnosu na keš (**full caching**) kontrolere. EPROM memorija dovoljnog kapaciteta sadrži kompletnu rutinu koja upravlja baferskom (**tracking buffer**) karticom. Po pravilu se koristi tehnika čitanja unapred (**read-ahead**) na nivou cele staze ili multipla staza.

Keš kontroleri (**Full caching architecture**)

Predstavljaju najsloženije i najkvalitetnije kontrolerske kartice. Ispoljavaju kvalitetne osobine po pitanju performansi i to zahvaljujući boljim lokalnim procesorima kao i složenom algoritmu koji koriste za keširanje diska. Cena im je relativno visoka.

Na slici 2 je prikazana je blok šema keš arhitekture. Keš arhitektura uključuje kvalitetnije lokalne procesore (CPU), 16 bitne ili 32 bitne, veliku keš memoriju, mali FIFO bafer za spregu sa sistemskom magistralom. Keš arhitektura dosta komplikuje i poskupljuje dizajn ali poboljšava performanse sistema. Algoritam za upravljanje keš memorijom je relativno složen. Veličina keš memorije može biti relativno velika i reda je više megabajta.



Slika 3.2 Keš arhitektura. [Izvor: A. S. Tanenbaum, Structured computer organization, 6th edition, Pearson Prentice Hall, 2012.]

▼ Poglavlje 4

I/O magistrale za hard diskove

DVE VRSTE DISKOVA: IDE I SCSI

Prioritet svakog uređaja određen je njegovim identifikacionim brojem koji se postavlja preko preklopnika na uređaju.

Dve vrste diskova predstavljaju de-facto standarde na PC računarima: IDE i SCSI. IDE uređaji su dobili naziv po elektronici integrisanoj na samom uređaju (**Integrated Drive Electronics**). Ovoj klasi uređaja pripadaju relativno jeftini diskovi solidnih performansi: kapaciteta do 200GB, sa brzinama okretanja ploča od 5400-7200 obrtaja u minuti. Kontroleri za IDE uređaje su ugrađeni na matičnim pločama računara (**on-board controller**) i pružaju interfejs ka računaru pri brzinama od 33-133 Mbit/s. Realna brzina čitanja i pisanja na same magnetne površine je znatno manja, tako da na kontroleru postoji bafer u koji se podaci smeštaju pre upisa na sam disk - na ovaj način se sprečava da performanse sistema značajno padnu prilikom rada sa diskovima. Svaki IDE kontroler ima dva kanala - primarni (**primary**) i sekundarni (**secondary**), a na svaki kanal se mogu vezati najviše dva uređaja u odnosu master-slave. Uređaji vezani na različite kanale mogu da primaju ili šalju podatke računaru istovremeno. Na jednom kanalu može biti samo jedan uređaj aktivan u jednom trenutku vremena. Svaki IDE uređaj ima preklopnike (**jumpers**) koje treba podesiti u željeni režim rada - master ili slave pre vezivanja na kontroler. Administrator sistema određuje način na koji će uređaji biti vezani na IDE kontroler - vezivanje dva brza diska sa kojih se često čita i na koje se često piše na jedan kanal, a dva spora CD-ROM uređaja, koji se povremeno koriste, na drugi kanal nema smisla. Uređaje treba vezati tako da se performanse sistema održe na najvišem mogućem nivou.

SCSI uređaji predstavljaju profesionalnu klasu širokog spektra uređaja - diskova, CD-ROM uređaja, traka, skenera itd. Kontroler za SCSI diskove nije integrisan na matičnim pločama i kupuje se odvojeno. Elektronika na SCSI kontroleru je komplikovanija, interfejs ka računaru je brži (do 320MB/s), a na kontroler je moguće vezati od 7 do 15 uređaja, u zavisnosti od kontrolera. SCSI uređaji se ne nalaze u master-slave odnosu, već se na kontroler vezuju prema prioritetima. Prioritet svakog uređaja određen je njegovim identifikacionim brojem koji se postavlja preko preklopnika na uređaju. Princip je sledeći: prioritet uređaja sa ID=0 je najviši i treba ga dodeliti sistemskom disku, a prioritet uređaja sa ID=15 je najniži. Identifikacioni broj ID=7 rezervisan je za SCSI kontroler.

ATA/ATAPI/IDE

ATA interfejs je jedan od najpopularanijih i najrasprostranjenih interfejsa. Uzrok popularnosti niska cena a visoke performanse. Poreklo naziva ATA je Advanced Technology Attachment.

PARALELNI ATA (PATA)

Zbog svoje jednostavnosti i niske cene (u poređenju sa SCSI interfejsom), ATA interfejs je najpopularniji popularnosti za desktop i prenosive PC računare.

Alternativni naziv je IDE (**I**ntegrated **D**rive **E**lectronics) zato što je skoro kompletna elektronika kontrolera integrisana na samom disk uređaju. ATA je dugo vremena forsiran kao paralelni 16 bitni interfejs PATA (Parallel ATA). U novije vreme imamo pojavu serijskog ATA interfejsa, SATA (Serial ATA).

Paralelni ATA (PATA)

Zbog svoje jednostavnosti i niske cene (u poređenju sa SCSI interfejsom), ATA interfejs je najpopularniji popularnosti za desktop i prenosive PC računare (Slika 1).

ATA magistrala je nezavisna od mikroprocesorske i sistemske magistrale. Samo dva uređaja mogu biti priključeni na ATA magistralu, koja se još i naziva ATA kanal. Na računarskim sistemima, po pravilu se nalaze 2 ATA kanala, koju su obično integrisana na samoj procesorskoj ploči.

U standardnim realizacijama, jedan ili 2 ATA uređaja mogu biti priključeni na ATA kanal.



Slika 4.1 ATA konfiguracija po jednom kanalu. [Izvor: <https://en.wikipedia.org/wiki/Parallel ATA>]

Ako su oba uređaja priključena na ATA kanal, jedan od njih se obeležava kao gospodar (*master* ili *device #0*), a drugi se obeležava kao sluga (*slave*, *device #1*). Naziv master/slave je relativno nekorektan.

Za većinu data transakcija oba uređaja su jednaka i kanal je single-task organizovan, u jednom trenutku samo jedan uređaja može biti aktivan i nema arbitracije. Master ima dodatnu funkciju za upravljanje dijagnostikom, reset stanjem i izvesnim kontrolnim signalima. Slika prikazuje maksimalnu ATA konfiguraciju po jednom kanalu koja se sastoji od ATA kontrolera i 2 ATA uređaja, povezanih u master/slave konfiguraciji. Setovanje uređaja kao master ili slave, obavlja se preko kratkospojnika (**jumper**) na samom disku ili preko kabla (**Cable Select**).

Za paralelni ATA važi da je 16 bitni interfejs, a brzina transfera je različita zavisno od moda koji se primenjuje. Postoje 3 režima rada za ATA kanal sa raznim podvarijantama:

- PIO modovi (**P**rogrammed **I**nterface **O**utput)
- DMA modovi (**D**irect **M**emory **A**ccess)
- UDMA modovi (**U**ltra **D**MA **M**odes).

EVOLUCIJA ATA INTERFEJSA (EVOLUTION OF THE ATA INTERFACE)

ATA standard je uveo IBM, 1980 godine. Potom se ATA intefejs permanentno razvijao, povećavajući performanse i dodajući nove funkcije i osobine.

ATA interfejs uključuje relativno malo klasa uređaja;

- magnetne diskove; • izmenjive diskove (ZIP) ;
- CD-R-RW uređaje; • DVD uređaje ;

Evolucija ATA Interfejsa (Evolution of the ATA Interface)

ATA standard je uveo IBM, 1980 godine. Potom se ATA intefejs permanentno razvijao, povećavajući performanse i dodajući nove funkcije i osobine. Prva realizacija je bila takva, da je kontrolerska logika bila ugrađena na disk uređaj, čime je praktično sve sa disk kontrolera potisnuto na sam disk uređaj, IDE.

Potom se razvija poboljšanje IDE u EIDE (**Enhanced IDE**), što je omogućilo brže disk transfere i podržavalo logičko blok adresiranje (LBA), za pristup diskovima većim od 524MB.

ATAPI (**Advanced Technology Attachment Packet Interface**) je proširenje koje je omogućilo da druge vrste periferala dele ATA magistralu. ATAPI definiše protokole za kontrolisanje uređaja kao što su CD-ROM uređaji i trake.

ATA modovi za prenos podataka (**ATA Transfer Modes**)

Visoke brzine uvek su bili imperativ za ATA interfejs. ATA maskimalna brzina transfera je redovno povećavana počevši od 1994 godine, kada je ATA interfejs postao deo ANSI standarda. ATA podržava 3 primarna data transfer režima-moda:

- **PIO Mode** — Prenos podataka između diska i memorije zahteva CPU intervenciju, zapravo prenos podataka realizuje sam procesor.
- **DMA Mode** — Prenos podataka između diska i memorije ne zahteva CPU intervenciju, zapravo prenos podataka realizuje klasični DMA kontroler, kakvih ima 8 na CPU ploči.
- **Ultra DMA (UDMA) Mode** — Ovaj mod ima istu funkcionalnost kao i klasičan DMA, ali dodaje prenos podataka na obe ivice takta (adds double transition clocking) čime postiže duplu brzinu data transfera, a takođe dodaje CRC kod za detekciju i korekciju greške čime se postiže povećani integritet podataka.

LVD SIGNALI

Serijska tehnologija upotrebljava LVD šemu koja se koristi 2 para data linija, koje prenose i primaju, signale niskog napona.

ATA-7, takođe se naziva Ultra ATA-133, dodaje podršku za Ultra DMA mod 6 (133 MB/ sec). Ovaj mod je poslednji mod iz PATA serije, iza toga imamo nagli zaokret ka serijskoj arhitekturi, tj uvodi se Serial ATA.

Serial ATA (SATA)

Da bi se prevazišle limitacije paralelne I/O arhitekture, serverska industrija migrira ka serijskim I/O tehnologijama. Serijska I/O tehnologija ima mogućnost da smanjuje dimezije kablova, ima manju potrošnju i veće performanse data transfera. Serial I/O tehnologije imaju sledeće osobine:

- LVD (Low-voltage differential LVD signaling)
- Point-to-point konekcije
- 8b/10b kodovanje

Glavni problem sa paralelnim arhitekturama je, što sa povećanjem brzina transfera, javlja se izobličenje signala (**signal skew**) i ISI efekat (**Inter-symbol Interference**), osobito za veće dužine kablova. Serijska arhitektura prenosi podatke bit po bit, tako da nema ISI efekata, a i greška se lako detektuje i ispravlja na nivou jednog bita.

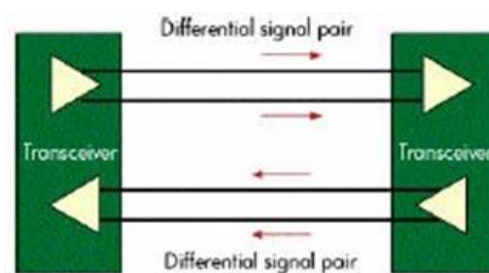
Serijska arhitektura radi sa sledećom filozofijom:

- 1 bit podataka
- vrlo visoka vrednost za takt (~GHz)

LVD signali

Serijska tehnologija upotrebljava LVD šemu koja se koristi 2 para data linija, koje prenose i primaju, signale niskog napona. Podatak se predstavlja kao naponska razlika između 2 žice u paru, kao na slici 7.

Zato što koristi male napone, LVD može da radi na mnogo većim brzinama nego na paralelnoj arhitekturi. Mali naponi smanjuju efekte kapacitivnosti, induktivnosti i šuma. Izvor šuma teži da doda istu količinu napona u obe žice, tako da naponska razlika ostaje ista.



Slika 4.2 LVD šema. [Izvor: https://en.wikipedia.org/wiki/Low-voltage_differential_signaling]

POINT-TO-POINT KONEKCIJE

Point-to-Point arhitektura može podržati više simultanih konekcija između kontrolera I I/O uređaja, pri čemu svi uređaji rade u svojoj punoj brzini, nema master/slave odnosa.

Point-to-point konekcije

Serijska arhitektura uvodi dizajn point-to-point linkova između kontrolera i više I/O uređaja. **Point-to-Point** arhitektura može podržati više simultanih konekcija između kontrolera I I/O uređaja, pri čemu svi uređaji rade u svojoj punoj brzini, nema master/slave odnosa, nema čekanja. Kontroler i sistemska magistrala su dužni da obezbede svim serijskim uređajima nezavisne kanale, koji rade u punoj brzini, dakle preklapljeno, čime se postižu visoke performanse.

8b/10b kodovanje

U paralelnim bas arhitekturama, podaci i takt se prenose zajedno paralelnim žicama, između kotrolera i diskova. Povećavati taktnu frekvenciju, odnosno brzinu prenosa podataka, je sve teže, zato što dolazi do izobličenja signala, a i šum može dovesti do ozbiljnih grešaka, kada se podaci i takt prenose paralelno. Serijska arhitektura koduje signale takta u niz podataka, tako da se šalje samo jedan signal, koji sadrži i takt i podatke, a to sve po jednoj liniji. Zahvaljujući tome, podaci mogu da se šalju na jako visokim frekvencijama, u odnosu na paralelne magistrale.

Serijska komunikacija zahteva od uređaja da konvertuje paralelni podatak-bajt u serijsku niz i obrnuto.

Svaki serijski uređaj, mora da sadrži kombinovani uređaj SerDes (**serializer/deserializer**), koji se sastoji od sledećih komponenti, slika 8:

- paralelni interfejs
- TX FIFO: FIFO keš (**First-In-First-Out**) za izlaz
- RX FIFO i za ulaz
- 8 bit/10 bit (8b/10b) enkoder i dekodeer
- serijalizator i deserijalizator

8b/10b enkoder konvertuje svaki 8 bitni data bajt u 10 bitni karakter za trasmisiju, čime se omogućava da se takt informacija koduje u serijski niz podataka. To će uvesti 20% overhead-a u niz podataka, ali se ovim omogućava pouzdan prenos podataka i detekcija greške.



Slika 4.3 Tehnika 8b/10b. [Izvor: https://www.wikiwand.com/en/8b/10b_encoding]

SERIAL ATA TEHNOLOGIJA

Serial ATA je uvedena sa maksimalnom brzinom od 1.5 Gb/s, ili 150 MB/s sa 20% encoding overhead ("8b/10b encoding").

Serial ATA tehnologija

SATA nastavlja tamo gde je PATA 133 stao u razvoju a to su brzine interfejsa iznad 133MB/s. Pored brzine SATA nudi masu prednosti.

SATA specifikacije su se pojavile u tri dela ili faze:

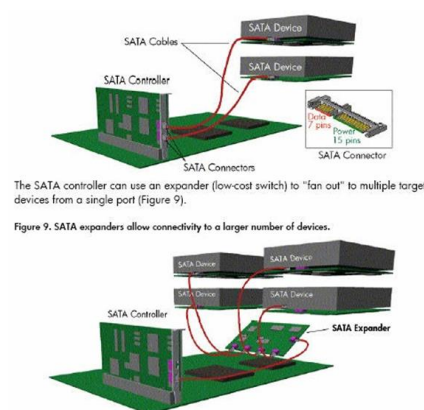
- SATA 1.0 (tranzicija paralelne u serijsku arhitekturu sa brziom od 150MB/s)
- SATA II Phase 1 (poboljšanje SATA 1.0)
- SATA II Phase 2 (povećanje brzine na 300MB/s)

SATA 1.0

SATA 1.0 specifikacija je objavljena u Augustu 2001 i prihvaćena u standardu ANSI T13. Specifikacija se fokusira na povećanje brzine i migraciju ka serijskoj arhitekturi. Serial ATA je uvedena sa maksimalnom brzinom od 1.5 Gb/s, ili 150 MB/s sa 20% encoding overhead ("8b/10b encoding").

SATA 1.0 je projektovan da zameni parallel ATA, što znači da ispuni zahteve desktop računara (**non-hot plug**). SATA nije hardware kompatibilan sa PATA; međutim SATA je potpuno kompatibilan sa PATA protokolom i zato je softverski kompatibilan sa postojećim ATA driverima

Diskutovaćemo glavne hardverske komponente za SATA. SATA kontroler se integriše u matičnu ploču ili je to PCI kompatibilna kartica. Svaki SATA kontroler ima određeni broj SATA portova u koje mogu da se konekuju SATA uređaji, interno ili eksterno, slika 10.



Slika 4.4 SATA konfiguracija. [Izvor: <https://wiki.kobol.io/helios64/sata/>]

SATA II

SATA Phase II povećava data transfer na 3 Gb/s, ili 300 MB/s koja je faktorisana sa 20 % 8b/10b encoding overhead-a.

SATA zamenjuje paralelni 40 pinski PATA konektor sa 7-pinskim SATA konektorom (**four signal lines and three ground lines**) i malim kablom dužine do 1 metar. Tanji kablovi omogućavaju bolju integraciju i bolje rashlađivanje (**airflow**), slika 11.

SATA odbacuje master/slave koncept i dozvoljava samo jedan uređaj po kablju, koga sistem vidi kao master SATA uređaj. Ova point-to-point konekcija omogućava svakom disku da komunicira sa kontrolerom u punoj brzini, bez ometanja i čekanja drugih diskova. SATA uvodi i specijalni konektor, 15-pinski, koji omogućava hot-plug mogućnost, neophodnu za RAID sisteme i servere.

SATA II

SATA II je nastavak SATA 1.0 specifikacije. SATA II se razvijala u 2 faze. SATA II-Faza-I specifikacija, pojavila se u Oktobru 2002, pobojšala je osobine SATA uređaja da mogu da zadovolje serverske i mrežne aplikacije. SATA II -Faza-II specifikacija, se pojavila 2003, povećava brzinu na 3Gb/s



Slika 4.5 SATA kablovi. [Izvor: <https://support.hp.com>]

SATA II unosi sledeće osobine koje povećavaju performanse:

Obrada komandi u redu čekanja (**Native command queuing**): omogućava da disk primi više komandi od CPU i da ih procesira u poretku da maksimizira protok podataka. Izvršavanje bez strogog redosleda (**Out-of-order execution and delivery**) drži uređaje u maksimalno zaposljenom stanju. Kombinuje se sa prethodnom metodom, tako što se zadate komande izvršavaju različito od svog vremena nailaska. Razbacivanje podataka (**data scatter/gather**): scatter/gather lista je struktura podataka, koja pomaže DMA u lociranju memorijskih regiona u koje se upisuju podaci iz bafera diska. Veoma je korisna zbog paging tehnike, koja jedan bafer sa diska može razbacati po raznim delovima fizičke memorije.

▼ Poglavlje 5

RAID kontroleri

RAID STRUKTURE

Definicija RAID je: RAID je struktura, koja organizuje nezavisne diskove u jedan veliki, visoko performansni logički disk.

RAID: ime, definicija, motivacija

Ime RAID ima sledeće značenje:

prvo ime : **Redundant Arrays of Inexpensive Disks**

moderno ime: **Redundant Arrays of Independent Disks**

Prvo ime RAID (**Redundant Arrays of Inexpensive Disks**) predstavlja redundantno polje jeftinih diskova, a odnosilo se na prvobitnu zamisao RAID koncepta, da više malih jeftinih diskova budu konkurencija velikom skupom disku. Međutim to ne odgovara stvarnosti, jer nikada nisu postojali skupi diskovi, a sve su veći i veći, a sve jeftiniji.

Drugi pokušaj je da se RAID proglasi kao **Redundant Arrays of Independent Disks**, redundantno polje nezavisnih diskova, takođe ne odgovara baš najbolje, diskovi u RAID strukturi nisu baš nezavisni, nego zajedno čine jedan logički disk.

Verovatno najbolji naziv je polje diskova (**disk array**).

Definicija RAID je: RAID je struktura, koja organizuje nezavisne diskove u jedan veliki, visoko performansni logički disk.

RAID KONCEPT

Povećanje performanse se realizuju na 2 načina: paralelnim operacijama sa više diskova istovremeno i konkurentnim disk operacijama takođe sa više diskova istovremeno.

RAID-početni koncept

U početku su mislili da umesto velikog i skupog diska, naprave isti kapacitet sa malim diskovima. Ali tako da RAID struktura bude jeftinija, brža i pouzdanija od jednog velikog diska. Međutim nikada nije postajao veliki, a skup disk.

No RAID, kao koncept je potpuno uspeo. Danas je RAID sinonim za:

- ogromni kapacitet

- jako visoke performanse
- jako visoku pouzdanost

RAID unosi dve fundamentalne osobine:

1. povećanje performansi preko paralelizma operacija

Kada imamo više diskova u jednoj logičkoj celini kakva je RAID, povećanje performanse se realizuju na 2 načina: paralelnim operacijama sa više diskova istovremeno i konkurentnim disk operacijama takođe sa više diskova istovremeno.

OSNOVNE RAID OSOBINE

Minimalno deljenje je takozvani bit-level stripping, gde svaki bit ide na poseban disk.

Za paralelizam operacija koristi se tehnika deljenja podataka između različitih diskova, pri čemu se uvodi minimalna jedinica podataka, koja se može nalaziti na jednom disku kontinualno (**stripe unit**). U slučaju malih stripe jedinica, jedan veći logički blok fizički se smešta na više fizičkih diskova, dok u slučaju velikih stripe jedinica, mali zahtevi se opslužuju sa jednog fizičkog diska, što može povećati konkurentnost više nezavisnih disk operacija u istom trenutku.

Minimalno deljenje je takozvani bit-level stripping, gde svaki bit ide na poseban disk.

Sledeći nivo podele je na nivou bajta (**byte-level stripping**), gde svaki bajt ide na poseban disk. Karakteristike bit i bajt nivoa deljenja da u svakom disk pristupu učestvuju svi diskovi u paraleli, čime se postiže maksimalna brzina transfera, ali konkurentnost celog RAID sistema je jednaka 1, samo jedan zahtev u jednom trenutku.

Sledeći slučaj je deljenje na nivou bloka (**block-level striping**), blokovi se dele između različitih fizičkih diskova na sledeći način:

blok i -> pripada disku po formuli $(i \bmod n) + 1$, gde je n broj diskova u RAID strukturu. Veličina bloka za deljenje je stepen broja 2, može biti manja od veličine bloka na disku, može biti jednaka 512 bajtova ili umnožak 512 bajtova. Kada imamo veće blokove, imamo povoljnu situaciju u oba slučaja, paralelizam za velike transfere, konkurentnost za male zahteve.

2. povećanje pouzdanosti preko redundanse

Svaki disk ima svoju pouzdanost, koja se meri srednjim vremenom do otkaza, MTTF (obično 100.000 časova). Ako bi imali RAID strukturu od N diskova, tada bi se verovatnoća otkaza jednog takvog sistema povećala N puta. Rešenje za problem pouzdanosti je uvođenje redundanse, koja predstavlja čuvanje ekstra informacija, koje normalno ne bi bile potrebne, ali se čuvaju da u slučaju otkaza diskova ne dođe do gubitka informacija. Najprostija tehnika koja se može koristiti je duplirati svaki disk, da ima svoju identičnu kopiju a naziva tehnika ogledala (**mirroring or shadowing**).

Svaki logički disk se sastoji od 2 fizička diska, pa u slučaju otkaza jednog od diskova, podaci se dobijaju sa onog drugog diska. Podaci će nestati samo ako strada i drugi disk, dok se prvi ne zameni sa zdravim. Srednje vreme otkaza jednog ogledala diskova zavisi od dva faktora: srednjim vremenom do otkaza pojedinačnih diskova kao i od srednjeg vremena za reparaciju.

PERFORMANSE KOD RAID SISTEMA

Ključni faktor RAID performansi je paralelizam. Treba naglasiti da se kod RAID sistema, osetno razlikuju read i write performanse.

U principu umesto otkazalog diska stavlja se novi disk, kojim se ogledalo reparira, odnosno kopiraju se svi podaci sa zdravog diska, a ogledalo će da pukne ako u toj reparaciji strada zdravi disk. Srednje vreme za koje par u ogledalu gubi podatke (mean time to data loss) jednako je:

$$\text{mean time to data loss} = (\text{mean time to failure})^2 / (2 \times \text{mean time to repair})$$

Osnovni nedostatak tehnika ogledala je što se za jedan logički disk koriste 2 diska, a to je 50% iskorišćenja kapaciteta.

Druga tehnika koja se koristi za povećanje pouzdanosti je tehnika parnosti (parity). Za grupu diskova uvodi se jedan disk parnosti, tako u slučaju otkaza bilo kog diska iz grupe, podaci mogu da se rekonstruišu.

Performanse kod RAID sistema

Ključni faktor RAID performansi je paralelizam. Treba naglasiti da se kod RAID sistema, osetno razlikuju read i write performanse. Prilikom čitanja, čitaju se samo podaci. Prilikom upisa, upisuju se podaci, ali i redundantne informacije, kao što su kopija u mirroring sistemima ili parnost u parity sistemima.

Zato su *read performanse*, uvek, osetno bolje nego *write performanse*.

Definicija performansi

Definišaćemo četiri klase performansi karakterističnih za disk RAID sisteme:

Random Read Performanse:

- obuhvataju veliki broj, malih read zahteva
- ove performanse opisuju ponašanje RAID sistema, na veliki broj malih slučajnih čitanja, obično u malim u datotekama RAID sistema
- Tipično, ove performanse su najznačajnije za baze podataka u kojima ima veliki broj transakcija sa malim datotekama i to tako da dominiraju read transakcije u odnosu na write transakcije

Random Write Performanse:

- obuhvataju veliki broj, malih write zahteva

- ove performanse opisuju ponašanje RAID sistema, na veliki broj malih slučajnih upisa, obično u malim datotekama RAID sistema

PERFORMANSE KOD RAID SISTEMA: READ/WRITE PERFORMANSE, I RAID KLASE

Treba naglasiti da su write performanse po pravilu osetno gore u odnosu na read performanse, za većinu RAID konfiguracija.

- Tipično, ove performanse su najznačajnije za baze podataka u kojim ima veliki broj transakcija sa malim datotekama, ali tako da dominiraju write transakcije u odnosu na read transakcije. Treba naglasiti da su write performanse po pravilu osetno gore u odnosu na read performanse, za većinu RAID konfiguracija.

Sequential Read Performanse:

- obuhvataju mali ili srednji broj, velikih disk zahteva za čitanje
- ove performanse opisuju ponašanje RAID sistema, kada se obavlja čitanje velikih datoteka u velikim sekvencijalnim zahtevima
- Tipično, ove performanse su glavne za aplikacije u kojima dominira veliki read u odnosu na write, na primer server sa velikim grafičkim datotekama u kome dominira čitanje.

Sequential Write Performanse:

- obuhvataju mali ili srednji broj, velikih disk zahteva za upis
- ove performanse opisuju ponašanje RAID sistema, kada se obavlja upis velikih datoteka u velikim sekvencijalnim zahtevima
- Tipično, ove performanse su glavne za aplikacije u kojima dominira veliki upis u odnosu na čitanje, na primer server sa velikim grafičkim datotekama u kome dominira upis, kao što je video ili audio editovanje.

RAID klase

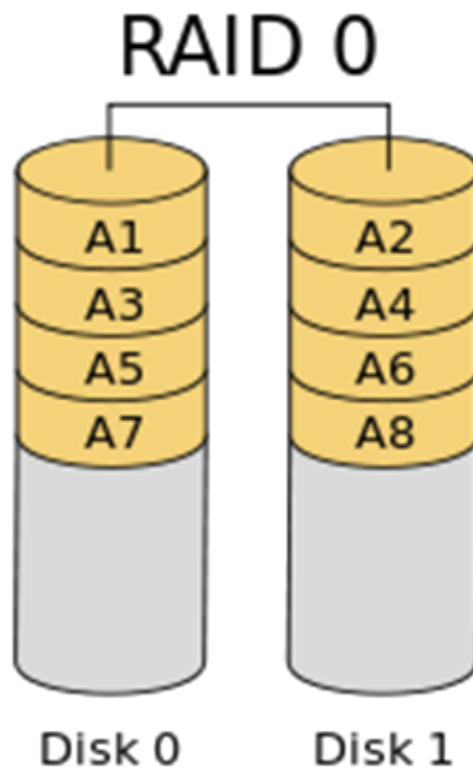
Na osnovu deljenja podataka i redundantne informacije, RAID sisteme ćemo podeliti u četiri klase:

- Mirroring
- Striping Without Parity
- Striping With Parity
- Combined systems (Nested systems)

RAID0

Kod RAID0, podaci se podjednako dele na oba diska. RAID0 ne poseduje redundanciju, niti je otporan na greške.

Kod RAID0, podaci se podjednako dele na oba diska.
 RAID0 ne poseduje redundanciju, niti je otporan na greške.
 Glavni razlog primene ove tehnike jeste brzina.
 Ukoliko se RAID0 pravi od n HDD-a različitih veličina,
 Rezultirajući HDD će biti n X (HDD najmanje veličine)
 Primer: Disk0 = 120 GB, Disk1 = 320 GB
 RAID0 konfiguracija 240GB
 Ukoliko jedan HDD otkáže sa radom, otkazuje ceo RAID0 sistem.
 Primarno se koristi u naučne svrhe.

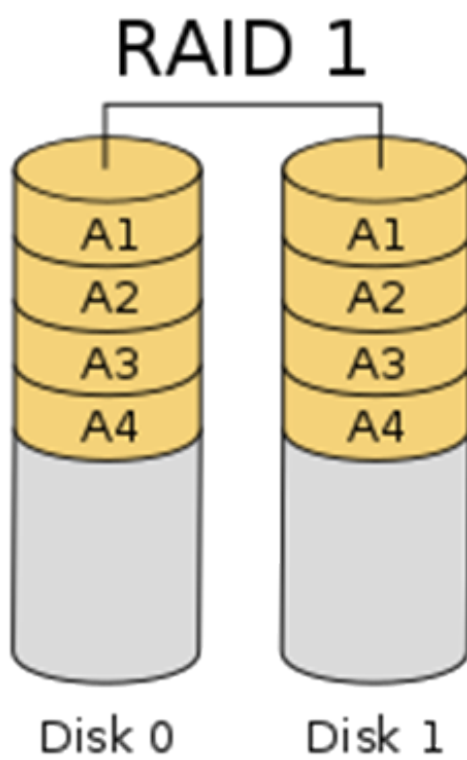


Slika 5.1 RAID0 Mehanizam [Izvor: Autor]

RAID1

Kod RAID1, podaci se kopiraju dele na oba diska. RAID1 poseduje redundanciju, i jeste otporan na greške.

Kod RAID1, podaci se kopiraju dele na oba diska.
 RAID1 poseduje redundanciju, i jeste otporan na greške.
 Glavni razlog primene ove tehnike jeste pouzdanost.
 Ukoliko se RAID1 pravi od n HDD-a različitih brzina,
 Rezultirajući HDD će biti kao kod HDD najmanje brzine
 Primer: Disk0 = 5400 rpm, Disk1 = 7200 rpm
 RAID1 konfiguracija 5400 rpm
 RAID1 radi sve dokle bar jedan HDD radi.



Slika 5.2 Slika-1 RAID1 Mehanizam [Izvor: Autor]

▼ Poglavlje 6

Pokazne vežbe

ZADATAK 1

Zadatak 1: Srednje vreme rotacionog kašnjenja (30 minuta)

Zadatak #1 (10 minuta)

Za diskove čiji su rotacione brzine obrtanja jednake 3600rpm, 5400rpm, 7200rpm, 10K, 15K, odrediti srednje vreme rotacionog kašnjenja.

Rešenje:

$$T_{\text{rotate_min}} = 0$$

$$T_{\text{rotate_max}} = T_{\text{one_revolution}}$$

$$T_{\text{rotate_average}} = (T_{\text{rotate_min}} + T_{\text{rotate_max}}) / 2 = T_{\text{one_revolution}} / 2$$

$$T_{\text{rotate_average}} = \frac{1}{V_{\text{rotate_sec}}} = \frac{60}{V_{\text{rpm}}} = 30 / V_{\text{rpm}}$$

$$V_{\text{rpm}} = 3600 \text{ rpm} \Rightarrow T_{\text{rotate_average}} = 30 / 3600 = 8.33 \text{ ms}$$

$$V_{\text{rpm}} = 5400 \text{ rpm} \Rightarrow T_{\text{rotate_average}} = 30 / 5400 = 5.55 \text{ ms}$$

$$V_{\text{rpm}} = 7200 \text{ rpm} \Rightarrow T_{\text{rotate_average}} = 30 / 7200 = 4.17 \text{ ms}$$

$$V_{\text{rpm}} = 10000 \text{ rpm} \Rightarrow T_{\text{rotate_average}} = 30 / 10000 = 3 \text{ ms}$$

$$V_{\text{rpm}} = 15000 \text{ rpm} \Rightarrow T_{\text{rotate_average}} = 30 / 15000 = 2 \text{ ms}$$

ZADATAK 2

Zadatak 2 (15 minuta)

Zadatak # (5 minuta)

Izračunati kapacitet hard diska koji poseduje 1046 cilindara, 16 glava i 63 sektora veličine 512 B.

Rešenje:

$$C = 1046 * 16 * 63 * 512 = 527184 \text{ kB} = 514 \text{ MB}$$

ZADATAK 3

Zadatak 3 (45 minuta)

Zadatak #3 (30 minuta)

Hard disk ima 24247 cilindara, 24 staze/cilindru, 793 sektora/stazi, a veličina sektora je 512 B. Broj obrtaja magnetnog diska je 7200 obrtaja/min, pri čemu je minimalno vreme pozicioniranja 0,8 ms, a srednje vreme pozicioniranja 7,4 ms. Odrediti:

- Kapacitet diska u GB;
- Vreme potrebno za iščitavanje 1 MB podataka koji su sekvencijalno raspoređeni na disku;
- Vreme potrebno za iščitavanje 1 MB podataka koji su slučajno raspoređeni na disku.

Rešenje:

a) $C = 24247 \cdot 24 \cdot 793 \cdot 512 = 214,88 \text{ GB}$

b) $N_{\text{sek}} = \frac{1 \text{ MB}}{512 \text{ B}} = 2048 \rightarrow 1 \text{ MB podataka se nalazi na 2048 sektora na hard disku}$

$N_{\text{st}} = \frac{2048 \text{ sektora}}{793 \text{ sektora/stazi}} = 2,58 \rightarrow 1 \text{ MB podataka se nalazi na 3 staze na disku}$

Vreme pristupa jednoj stazi se računa prema sledećem obrascu:

$$T_{pr} = T_{poz_{st}} + \frac{1}{2r} + \frac{b}{rN}$$

$$r = 7200 \text{ ob/min} = 120 \text{ ob/s}$$

$$T_{poz_{st}} = 7,4 \text{ ms}$$

$$N = 793 \cdot 512 = 396,5 \text{ kB} \rightarrow \text{kapacitet staze}$$

$$b = b_1 + b_2 + b_3 \rightarrow \text{pročitajte se dve cele staze i 58\% treće staze}$$

$$b_1 = b_2 = N = 396,5 \text{ kB}$$

$$b_3 = 0,58 \cdot N = 230 \text{ kB}$$

Prema tome, ukupno vreme potrebno za iščitavanje 1 MB podataka je:

$$T_{pr} = T_{poz_{st}} + \frac{1}{2r} + \frac{b_1}{rN} + T_{poz_{min}} + \frac{1}{2r} + \frac{b_2}{rN} + T_{poz_{min}} + \frac{1}{2r} + \frac{b_3}{rN}$$

traženje
prve
staze
 rotaciono
kašnjenje
 čitavanje
prve
staze
 prelazak
na
sledeću
stazu
 rotaciono
kašnjenje
 čitavanje
druge
staze
 prelazak
na
sledeću
stazu
 rotaciono
kašnjenje
 čitavanje
treće
staze

$$T_{poz_{min}} = 0,8 \text{ ms}$$

$$T_{pr} = 43,02 \text{ ms}$$

Slika 6.1 Rešenje pod b). [Izvor: Autor]

c) $N_{\text{sek}} = \frac{1 \text{ MB}}{512 \text{ B}} = 2048 \rightarrow 1 \text{ MB podataka se nalazi na 2048 sektora na hard disku}$

$$b = 512 \text{ B}$$

$$N = 396,5 \text{ kB}$$

Pošto se podaci nalaze nasumično raspoređeni na disku, to znači da u najgorem slučaju moramo pročitati svih 2048 sektora na kojima se može naći po 512 B našeg podatka od 1 MB, tako da je srednje vreme pristupa:

$$T_{pr} = 2048 \cdot \left(T_{poz_{st}} + \frac{1}{2r} + \frac{b}{rN} \right) = 2048 \cdot (7,4 + 4,17 + 0,01) = 23,715 \text{ s}$$

Slika 6.2 Rešenje pod 3). [Izvor: Autor]

✓ Poglavlje 7

Zadaci sa samostalni rad

ZADACI ZA SAMOSTALNI RAD ZA LEKCIJU #8

Zadaci za samostalni rad za lekciju #8 rade se okvirno 45 min.

Zadaci za samostalni rad:

Hard disk poseduje:
broj_indeksa/2 cilindara,
prve_dve_cifre glava,
druge_dve_cifre sektora, veličine 1024 B.

Izračunati kapacitet u kB/MB/GB.

▼ Poglavlje 8

Domaći zadatak

DOMAĆI ZADATAK #8

Domaći zadatak #8 okvirno se radi 45 minuta

Domaći zadatak #8

Hard disk ima X cilindara, 24 staze/cilindru, Y sektora/stazi, a veličina sektora je 512B.

Broj obrtaja diska je 10800 obrtaja/min, pri čemu je minimalno vreme pozicioniranja 0,6 ms, a srednje vreme pozicioniranja 5,4 ms.

X = broj_indeksa * 6,

Y = poslednje_dve_cifre_broja_indeksa (za 00 koristiti 99).

Odrediti:

- a) Kapacitet diska u GB (1kB = 1024B; 1MB = 1024kB; 1GB = 1024MB);
- b) Vreme potrebno za iščitavanje 5 MB podataka koji su sekvencijalno raspoređeni na disku;
- c) Vreme potrebno za iščitavanje 5 MB podataka koji su slučajno raspoređeni na disku.

Predaja domaćeg zadatka:

Domaći zadatak slati odgovarajućem predmetnom asistentu, sa predmetnim profesorom u CC.

Predati domaći zadatak koristeći .doc/docx uputstvo dato u prvoj lekciji.

▼ Ulaz/Izlaz- Opšta razmatranja

ZAKLJUČAK

Recime lekcije #8

Pored procesora i memorije, treći ključni element računarskog sistema jeste U/I podsistem koga čini skup U/I modula.

Svaki U/I modul predstavlja interfejs između systemske magistrale i izvesnog broja U/I uređaja.

U/I modul nije samo skup mehaničkih veza već poseduje izvesni “stepen inteligencije”, odnosno logiku koja vrši komunikaciju između systemske magistrale i perifernih uređaja.

Razlozi za postojanje U/I modula su:

- Postoji mnogo vrsta spoljnih uređaja sa različitim principima rada,
- Brzina prenosa podataka spoljnih uređaja je obično mnogo manja nego kod procesora ili memorije.
- U određenim slučajevima spoljni uređaji su mnogo brži od memorije ili procesora.

Spoljni uređaji često koriste različite formate podataka i dužine reči nego što je to slučaj sa računarom na koji su priključeni.

U/I modul ima dve osnovne funkcije:

Sprega procesora i memorije preko systemske magistrale.

Niska cena mikroprocesora otvorila je put ka povećanju inteligencije (funkcionalnosti) U/I podsistema. Umesto da se koriste jednostavne U/I komande (odgovaraju fizičkom nivou) koriste se logičke i/ili simboličke U/I komande visokog nivoa. veliki broj funkcija dražvera uređaja (obrađivanje grešaka i optimizacija vremena pristupa) može da izvrši mikroprocesor U/I podsistema, nazvan U/I procesor (IOP).

Literatura:

A. Tanenbaum, Structured Computer Organization,
Chapter 04, pp. 100 - 116,
Chapter 05, pp. 117 - 130.