

-Organizacija računara-

-Skripta – II test-

Kontrolna (upravljачka) jedinica

-Ona upravlja pojedinim koracima u obradi podataka i to na osnovu informacija sadržanih u instrukciji koju zahvata iz memorije. Sinhronizuje U/I jedinice, memoriju i aritmetičko-logičku jedinicu.

-U odnosu na implementaciju kontrole jedinice, postoje dva pristupa:

- direktan (hardverski)
- mikroprogramski

-U odnosu na ove implementacije, procesori su se delili na RISC i CISC procesore, s tim što danas ta granica nije jasno povučena, već jedni ka drugima konvergiraju.

-Uslovno rečeno, upravljačka jedinica je jedna „crna kutija“ koja ima određenu količinu ulaza, određenu količinu izlaza i unutrašnju memoriju kako bi znala gde je trenutno u svom radu i šta sledeće treba da radi.

-To je sekvencijalno kolo – kao što se sećamo, u tom slučaju izlaz ne zavisi samo od ulaza, već i od internog memorijskog stanja kola.

-**Ulazi** mogu biti recimo: takt, flegovi, različiti upravljački signali sa systemske magistrale, instrukcijski registar (tu se smešta adresa instrukcije), itd.

-**Izlazi** mogu biti recimo: različiti upravljački signali ka sistemskoj magistrali (upravlja radom računara), upravljački signali koji ne izlaze izvan procesa i ostaju u CPU i upravljaju njegovim radom, itd.

***Upravljački signali** su signali koji upravljaju radom procesora, ali i čitavog računara.

Direktna (hardverska) realizacija CPU

-Što se tiče **direktne (hardverske)** implementacije CPU, ona je karakteristična za **RISC** procesore. Ona generiše fiksne sekvence kontrolnih signala gde stanje signala zavisi od ulaza i od stanja u kom se procesor nalazi u toku izvršavanja određene funkcije.

-Karakteristike RISC procesora:

- relativno mali instrukcijski set procesora
- jedna instrukcija po mašinskom ciklusu (*mašinski ciklus* nije takt, već vreme za dobavljanje operanada iz dva registra, računanje i smeštanje rezultata u registar) i sve instrukcije podjednako dugo traju
- sve instrukcije su bile vrlo jednostavne, imale su istu dužinu, fiksni format i brzo su se dekodirale, takođe su imale jednostavne adresne modove
- imali su veliki broj registara
- većina operacija je bila iz registra u registar

*Kod RISC procesora je sve hardkodovano, nema ispravke.

-Prednosti direktne realizacije:

- veća brzina izvršavanja operacija zato što ima jednostavan format instrukcija
- manji broj komponenti i trošilo se znatno manje struje za rad

*Prvi ovakav procesor je bio ARM procesor sredinom 80-tih godina.

-Mane direktne realizacije:

- jako je teško realizovati izmene: popraviti greške, dodavati nove instrukcije
- zbog prethodno navedenog je bila komplikovana sama realizacija a i testiranje ovakvih procesora

Mikroprogramska realizacija CPU

-Stanja svih upravljačkih signala fizički dolaze iz memorije (memorija ima određeni broj lokacija) i u svakoj lokaciji bitovi koji su smešteni predstavljaju stanje upravljačkih signala za neki konkretni trenutak u vremenu. Zatim, clock ulazi u brojač i pokreće ga, broji i brojač izbacuje binarni broj koji služi da se nakon dekodiranja u jedan od 2^n ulaza pozove jedna memorijska ćelija. To što se pročita iz te memorijske ćelije su kontrolni signali koji upravljaju svim komponentama.

-Sadržaj te pročitane memorijske ćelije se naziva **mikroinstrukcija(mikrokod)**, dok je sekvenca mikroinstrukcija **mikroprogram(firmware)**.

-Karakteristike CISC procesora:

- jedna instrukcija se izvršava u velikom broju ciklusa takta
- komplikovani adresni modovi
- komplikovani format instrukcija jer je on varijabilne dužine, pa je samim tim i dekodiranje komplikovano

-Sadržaj jedne lokacije mikroprogramske memorije je višebitna reč, gde je svaki bit reči jedan upravljački signal.

-Svaka mašinska instrukcija procesora je reprezentovana skupom reči u mikroprogramskoj memoriji gde svaka reč predstavlja skup upravljačkih signala u jednom momentu.

-Prednosti CISC-a u odnosu na RISC:

- manje programskih instrukcija u mašinskom kodu (.exe fajlovi su kraći)
- moćnije su instrukcije nego kod RISC-a
- mikroprogramiranje je jednostavnije od dizajniranja hardverski implementiranje kontrolne jedinice
- mogućnost ispravke mikrokoda
- uslovno rečeno, manja cena i manje grešaka, mada se ispotavilo da to nije baš istina

-Mane CISC-a:

- sporije od hardverske implementacije u nekim slučajevima, recimo dekodiranje kompleksnih instrukcija zbog promenljive dužine i formata instrukcija
- zbog malog broja registara, parametri i lokalne promenljive se smeštaju na stek, pa je onda sporije izvršavanje funkcija i procedura, dok RISC procesori stavljaju sve u registre jer ih imaju baš dosta

-Zaključak: obe tehnologije konvergiraju jedna ka drugoj

- ❖ CISC procesori imaju sve više i više registara
- ❖ RISC procesori postaju sve kompleksniji jer imaju veći broj instrukcija i one su kompleksnije nego što su bile ranije

Instrukcijski ciklus

-Ovaj ciklus realizuje procesor.

-Tok ciklusa:

- 1) dobavljanje naredbe iz memorije (fetch instruction)
- 2) interpretiranje naredbe – dekodiranje da bi se utvrdilo koju akciju treba da izvrši
- 3) dobavljanje podataka (fetch data) – dobavljanje podataka iz memorije ili U/I uređaja ukoliko to naredba zahteva (*opciono*)
- 4) procesiranje podataka – izvršenje nekih aritmetičkih ili logičkih operacija nad podacima ukoliko to naredba zahteva (*opciono*)
- 5) upisivanje podataka – upisivanje rezultata u memoriju ili U/I jedinicu ako to naredba zahteva (*opciono*)

-Svaka instrukcija mašinskog jezika se sastoji iz niza koraka (fetch, decode, itd.), a svaki korak se sastoji iz niza mikrooperacija. Svaka mikrooperacija predstavlja osnovnu funkcionalnu operaciju u procesoru.

1) Dobavljanje instrukcije (fetch instruction)

- ❖ PC (Program Counter) sadrži adresu sledeće instrukcije
- ❖ Adresa iz PC se izbacuje na adresnu magistralu kao binarni broj (PC → address bus)
- ❖ Kontrolna jedinica usput izbacuje i par specijalnih upravljačkih signala koji će reći da se čita iz memorije (sve to izađe na kontrolnu i adresnu magistralu)
- ❖ Memorija shvati da se procesor obraća njoj, pročita koja je adresa na adresnoj magistrali, dobavi traženi sadržaj i izbacuje na data magistralu (sabirnica podataka)
- ❖ Procesor zatim pokupi izbačen rezultat sa sabirnice podataka i smesti u instrukcijski registar (memory → IR)
- ❖ Dok se to dešava, paralelno se i PC uvećava za 1 kako bi onda pokazivao na sledeću instrukciju ili operand
- ❖ Na kraju, na osnovu instrukcijskog registra, dobijena instrukcija definiše koji deo mikrokoda će se aktivirati (IR → microcode_addr)

-Pravilan redosled mikroinstrukcija se mora poštovati (prvo ide $\text{address_bus} \leq \text{PC}$, pa onda $\text{PC} \leq \text{PC} + 1$).

-Konflikti se moraju izbegavati (istovremena čitanja i pisanja i sl.). U našem procesoru može istovremeno da radi:

$\text{address_bus} \leq \text{PC}, \text{data_bus} \leq \text{memory}, A \leq \text{data_bus}, \text{PC} \leq \text{PC} + 1$

2) Dobavljanje operandata

- ❖ Ako instrukcija zahteva dobavljanje operandata, onda se oni dobave pre izvršenja instrukcije

Primer neposrednog adresiranja:

LD A, broj (LD = load)

t1: $\text{address_bus} \leq \text{PC}, \text{data_bus} \leq \text{memory}, A \leq \text{data_bus}, \text{PC} \leq \text{PC} + 1$

t2: idi na fetch

3) Izvršenje instrukcije

- ❖ Može biti različitog oblika i to zavisi od instrukcije koja se izvršava
- ❖ Može da sadrži: čitanje/pisanje po memoriji, ulazno/izlazne operacije, rad sa registrima, aritmetičko/logičke operacije

4) Provera prekida

- ❖ Kada procesor treba da proveriti da li je došlo do prekida i ukoliko se prekid dogodio, procesor treba da sačuva u memoriji vrednost PC da bi znao dokle je stigao sa izvršenjem i da bi po završetku prekida mogao nastaviti šta je radio i onda će se na stack staviti i flegovi
- ❖ 3 koraka:
 - Stack Pointer se smešta na adresnu magistralu $\text{address_bus} \leq \text{SP}$
 - PC se snima u memoriju $\text{memory} \leq \text{PC}$, $\text{SP} \leq \text{SP} + 1$
 - u PC se smešta adresa prekidne rutine ($\text{PC} \leq \text{interrupt_handler}$)
 - kada se vraća iz interrupt handler-a, to je isto kao i instrukcija RET.

-Kao što je prethodno opisano se radilo dugo vremena, dok se nije shvatilo da tok se izvršava neka instrukcija (možda ona dugo traje), paralelno se može raditi i nešto drugo npr, dobavljati sledeća instrukcija. Ova pojava se naziva **prefetch**.

-Međutim, prefetch neće raditi kada imamo skokove jer svaki skok poništava prefetch. Ono što imamo u današnjim modernism procesorima je čitav deo hardvera koji se bavi predviđanjem skoka (*branch prediction*).

-Pipelining je preklapanje svih stepena izvršenja instrukcije:

- Fetch instruction (FI)
- Decode instruction (DI)
- Calculate operands (CO)
- Fetch operands (FO)
- Execute instructions (EI)
- Write result (WO)

*Grafik na slajdu 34 nam pokazuje kako se preklapaju instrukcije. Na primeru 9. instrukcije, ona se izvrši u okviru 14 taktova, a da nema preklapanja bilo bi 54 takta (za svaku instrukciju po 6 taktova).

*U praksi ovo nikada nije ovako idealno, procesor nekad uspe nešto da uradi, a nekada i ne. Ubrzanje nije 5, 6 puta, ali je sigurno duplo brže sa pipeliningom nego bez njega.

Memorija

-Podsećanje:

- Služi za prihvatanje, čuvanje (pamćenje, memorisanje) i predaju podataka i programa
- Element koji pamti elementarnu informaciju (1 bit) je memorijski element
- Memorijski elementi se udružuju u memorijsku lokaciju (memorijsku ćeliju).
- Skup memorijskih ćelija je memorijski modul (blok)
- Memorijski modul je najčešće adresibilan
- Adresa je jedinstveni prirodan broj dodeljen svakoj ćeliji.

-Sa aspekta pristupa:

- ❖ sa sekvencijalnim (serijskim) pristupom (magnetna traka, bušena traka, ...)
- ❖ sa cikličkim (periodičnim) pristupom (hard disk, floppy disk, CDROM)
- ❖ sa proizvoljnim (slučajnim) pristupom (RAM – Random Access Memory)

-Sa aspekta mogućnosti promene:

- ❖ promenljiva memorija (RAM memorija)
 - statička
 - dinamička
- ❖ polupromenljiva memorija
 - PROM – Programmable Read Only Memory
 - EPROM – Erasable PROM
 - EEPROM – Electrically EPROM
 - fleš memorija
- ❖ stalna memorija - (ROM – Read Only Memory)

RAM

-Uglavnom je adresabilna što znači da memorija počinje od nekog broja. Kada procesor želi da se obrati nekoj memoriji, on u principu izbacuje na adresnu magistralu taj broj koji predstavlja adresu memorije kojoj procesor želi da pristupi.

-Omogućava čitanje i pisanje podataka (bitova).

-Kada nema napajanja, gubi se sadržaj i samim tim ova memorija ne može služiti za trajno smeštanje podataka.

-Dinamička memorija, za razliku od statičke, mora da se osvežava jer je „zaboravna”, dok statička memorija dokle god ima napajanja čuva sadržaj.

Dinamički RAM

- ❖ zaboravna je zato što memorijska ćelija koja čuva podatak to radi u kondenzatoru koji se prazni tokom vremena, pa se onda mora stalno dopunjavati (osvežavati)
- ❖ kondenzator gubi tj. iz njega curi elektricitet, pa se zato stalno dopunjava svakih nekoliko milisekundi; ovo su mali kondenzatori i to znači da može da stane malo napajanja i na sve to što u njih stane imaju tu osobinu da elektricitet neminovno otiče iz njih
- ❖ u trenucima osvežavanje je nedostupna za rad, ali to svakako vrlo kratko traje (to je dobra realizacija)
- ❖ ima manju cenu od statičke memorije, ali je sporija
- ❖ koristi se za operativnu memoriju računara
- ❖ ovaj RAM je realizovan sa dva tranzistora gde jedan imitira kondenzator, a drugi imitira prekidač

-Funkcionisanje DRAM-a:

- kada se na logičkoj jedinici čita ili piše adresna linija, transistor koji se ponaša kao prekidač je uključen i provodi bit line do kondenzatora
- ako smo čitali naelektrisanje (ako je 1 bilo zapamćeno), naelektrisanje će se izliti na drugu liniju koja odlazi na pojačivače koji pojačavaju taj signal pa se to zapamti i prosledi do procesora, a ukoliko smo pročitali nulti napon (0 je bilo zapamćeno), nikakvo naelektrisanje se ne pojavljuje
- ukoliko treba da se upiše neki podatak, jedinica se upisuje tako što ovaj kondenzator pogodimo adresnom linijom, logičkom jedinicom nateramo da transistor (prekidač) proradi i napuni kondenzator naelektrisanjem; ako se upisuje nula, u slučaju da je i pre toga isto bila nula – ništa se ne dešava, ali u slučaju da je bila 1, ona će se isprazniti kroz logičku nulu koju pokušavamo da upišemo (praznimo kondenzator)

Statički RAM

- ❖ napravljen je od D flip flopova – potrebno nam je 6 tranzistora da bi realizovali flip flop (ne samo 2, kao što je to u osnovnom slučaju)
- ❖ nema curenja električnog naboja (ne zahteva osvežavanje)
- ❖ kompleksnije je zbog posedovanja 6 tranzistora, većih je dimenzija
- ❖ brža je, ali je zato i skuplja
- ❖ koristi se u keš memoriji i za register

-Funkcionisanje SRAM-a: (slika na slajdu 9)

- Međusobno povezani tranzistori omogućavaju dva stabilna stanja
 - Stanje 1 – C1 visok napon, C2 nizak – T1 T4 isključeni, T2 T3 uključeni
 - Stanje 0 – C2 visok napon, C1 nizak – T2 T3 isključeni, T1 T4 uključeni
- Tranzistori T5 i T6 se aktiviraju adresnom linijom – propuštaju stanje flip-flopa na bit linije B i B'
- Pisanje – napon se dovodi na B, a njegov komplement na B', a odatle u flip-flop
- Čitanje – stanje flip-flopa se dovodi na bit line

Poređenje DRAM-a i SRAM-a

- ❖ Zajedničko im je to što obe memorije gube podatke kada izgube napajanje, što znači da im je obema tokom rada potrebno napajanje
- ❖ **DRAM** - jednostavnije, manje, veća gustina pakovanja, manja cena, zahteva osvežavanje sadržaja, sporija, moduli velikog kapaciteta
- ❖ **SRAM** – brža, skuplja, zauzima više mesta, koristi se u keš memoriji

ROM

-Služi za trajan smeštaj podataka i nije potrebno napajanje da bi se sadržaj pamtio.

-Koristi se za programiranje mikrokoda u procesoru i smeštaj sistemskih programa (npr BIOS)

*U BIOSu se čuva minimalan kod koji je potreban za pristup hard disku i učitavanje operativnog sistema u RAM, kao i neke zajedničke funkcije koje će biti korišćenje od strane operativnog sistema (low level funkcije za obraćanje hardveru). Moderni BIOS-i nisu napravljeni od ROM memorije, već od EEPROM.

-Vrste ROM-a:

- 1) Standardni ROM – upis na njega tokom proizvodnje, zato su jako skupe greške, ne može se menjati, skupo je za male serije
- 2) PROM (Programmable (once) ROM) – samo jednom se može upisati sadržaj i za to je potrebna specijalna oprema za programiranje
- 3) Izmenljivi ROM
 - a. EPROM – briše se ultraljubičastim svetlom (UV lampe)
 - b. EEPROM – memorijska ćelija se briše električnim signalom i ovde je sporiji upis od čitanja
 - c. Flash memory – blokovi memorije se brišu električnim signalom (flash-om) i velika je gustina pakovanja

Osvežavanje DRAM

-Kolo za osvežavanje se nalazi u memorijskom integralnom kolu. Memorijsko kolo se privremeno isključuje dok se osvežava.

-Osvežavanje:

- prolazi se kroz sve vrste
- pročita se i zatim upiše nazad pročitana vrednost

-Troši vreme i obaraju se performanse memorije

Organizacija memorije u integrisanim kolima

-Primer

256 KB se organizuje u 8 puta po 256 Kb memorije:
svako integrisano kolo ima 256 Kb, odn. 5122 1-bitnih ćelija (512 vrsta i 512 kolona)
za 256 Kb potrebno 18 adresnih linija ($2^{18} = 256 \times 1024$)

Detekcija/korekcija grešaka

-Greške mogu biti:

- trajne (hard failure)
 - naprezanje/opterećenje, greška u proizvodnji, istrošenost
 - memorijska ćelija – trajno prikazuje 0 ili 1 ili ponekad vrati 0, ponekad 1
- privremene (soft error) - problemi u napajanju, smetnje, alfa čestice

-Detekcija greške Hemingovim kodom (i mogućnost korekcije):

- ❖ ideja: na 4 bita informacije idu 3 bita Hemingovog koda koji imaju vrednost takvu da je broj jedinica u svakom krugu paran (pogledati sliku na slajdu 23)
- ❖ povećanjem broja bitova informacije se ne povećava broj bitova Hemingovog koda linearno, već je pravilo da ako je M bitova reči a K dodatnih bitova, imamo:
$$2^k \geq M + K + 1$$
- ❖ Provera ispravnosti: sračuna se ponovo K dodatnih bitova, istim algoritmom, uradi se XOR između originalnih i sračunatih K bitova; ako se dobiju sve nule, reč je ispravna, u suprotnom rezultujući bitovi predstavljaju redni broj bita koji je neispravan
- ❖ Dodatni bitovi (check bit) se nalaze na pozicijama u reči, gde imamo samo jedan bit u binarnom zapisu (1, 2, 4, 8, 16, 32, itd.)
- ❖ Dodatni bitovi se računaju tako što se uradi XOR onih bitova, čija pozicija izražena binarno, sadrži jedinicu gde je i jedinica check bita

*problem se ovde javlja kod detekcije višebitnih grešaka, ovo dobro radi za jednobitne greške

Hemingov SEC-DED kod

-SEC (Single Error Correcting)

-SEC-DED (Single Error Correcting, Double Error Detecting)

-Ideja: dodatni bit koji čini da ukupan broj jedinica bude paran

- detektuje i ispravlja 1-bitne greške
- detektuje 2-bitne greške

-Savremene DRAM memorije koriste 9 dodatnih bitova na svakih 128 bitova, što čini 7% dodatnog memorijskog prostora

Vrste DRAM memorije u računarima

-SDRAM (Synchronous DRAM)

- podaci se prenose sinhrono sa taktom magistrale
- burst režim omogućuje da se RAS jednom postavi, a da se kolone čitaju u paketu
- odličan učinak prilikom transfera u/iz bloka memorije

-RAMBUS DRAM (RDRAM)

- nakon 480 ns, brzina prenosa ide do 1.6 Gbps
- specijalan memorijski kontroler i magistrala

-DDR SDRAM (Double Data Rate SDRAM)

- podaci se šalju dva puta u jednom taktu magistrale (na rastućoj i opadajućoj ivici takta)
- DDR2 – dva puta više podataka od DDR u jedinici vremena
- DDR3 – dva puta više podataka od DDR2 u jedinici vremena

Keš memorija

-Memorija se nalazi u procesoru (registri i keš napravljeni od D flip flopova), imamo operativnu (radnu, glavnu) memoriju (napravljena od dinamičkog RAM-a) i masovnu memoriju (hard disk, flopi disk, CD, DVD,...).

-Parametri memorije sa stanovišta performansi:

- Vreme za pristup (access time) – vreme između pojave adrese na adresnoj magistrali i razmene podataka
- Memorijski ciklus (Memory Cycle) – vreme koje protiče između dva pristupa memoriji
- Brzina prenosa (Transfer Rate) – brzina kojom se podaci prenose

-U idealnom slučaju želimo da nam vreme pristupa i memorijski ciklus budu što manji (kraći), a brzina prenosa što veća.

-Fizička realizacija memorije:

- ❖ Poluprovodnička memorija – RAM
 - statička RAM memorija
 - dinamička RAM memorija
 - SSD
- ❖ Magnetska – Diskovi (HDD, FDD), traka
- ❖ Optički – CD, DVD, Blu Ray
- ❖ Ostali tipovi – Hologram

-Hijerarhijska organizacija memorije (piramida – slajd 5) – što idemo ka vrhu piramide, tu je brzina memorije veća, a kapacitet memorije se smanjuje.

-Hijerarhijska lista po brzini:

- Registri (statička memorija)
- L1 keš (statička memorija)
- L2 keš (statička memorija)
- Operativna memorija (dinamička memorija)
- Keš za disk (u operativnoj memoriji, na uređaju)
- Hard disk
- Optički uređaji
- Traka

-Uvek težimo tome da dobijemo što veću brzinu prenosa sa što većim kapacitetom, međutim, što više poboljšavamo performanse, cena je sve veća ili je možda tehnološki i tehnički potpuno neizvodljivo realizovati bolje ili je isuviše komplikovano (što je takođe jako skupo).

-Keš memorija se nalazi između operativne memorije i procesora. Koristi se statička RAM memorija; deo koda i podataka se pribavi u keš memoriju i procesor radi sa keš memorijom.

-Računar bez keš memorije – moguće, ali veoma skupo rešenje; za operativnu memoriju bi se koristila statička RAM memorija, ali i dalje ostaje problem dužine vodova, odnosno kašnjenja (operativna memorija je daleko od procesora). Zaključak je da ne vredi praviti takav računar.

-Keš kontroler podatke i deo koda iz operativne memorije u keš i to nudi procesoru. Procesor toga nije svestan tj. kojoj memoriji pristupa, već mi kao korisnici primetimo da je brže kada radimo sa keš memorijom u odnosu sa operativnom memorijom.

-Iako keš memorije imamo za red veličine manje od operativne memorije, to ne umanjuje nešto bitno performanse našeg programa, u svakom slučaju ih poboljšava.

-Postoji tendencija u programu da se reference na memorijske lokacije grupišu (Locality of Reference) recimo grupisanje koda (petlje) ili grupisanje podataka (nizovi, matrice). Ova tendencija opravdava postojanje keš memorije.

-Bez obzira koja organizacija keša je u pitanju (ima ih 3), svaka od njih se bazira na ideji da uz neki blok ili keš line postoji neki tag i to je sastavni deo svake keš memorije.

*Cache line (blok) – u njemu se nalaze originalni bajtovi koji su dobavljeni iz RAM-a.

*Tag – služi za jedinstveno identifikovanje bloka iz operativne memorije da bi znao koji se blok trenutno nalazi u kom slotu (keš liniji ili bloku).

-Kada se jedan blok dovuče iz memorije, on napuni ceo slot. U zavisnosti od toga koliko je veliki keš line, koliko je veliki tag i koji je način mapiranja adrese iz operativne memorije na slot u keš memoriji, postoje tri realizacije keš memorije.

-Proces kako radi keš memorija:

- Procesor zahteva sadržaj memorijske lokacije
- Proverava se keš (da li ima sadržaj ove lokacije)
- Ako je u kešu, dobavlja se (veoma brzo)
- Ako nije u kešu (cache miss), učitava se čitav blok iz memorije u keš i zatim se sadržaj prosledi procesoru
- Keš sadrži tagove koji identifikuju koji blok iz operativne memorije se nalazi u kom slotu i obično je to deo adrese

-Dizajn keš memorije:

1. Veličina – ograničavajući factor je samo veličina čipa
2. Način mapiranja (Mapping Function): direktni, asocijativni, set-asocijativni i oni predstavljaju internu arhitekturu keš memorije
3. Algoritam zamene (Replacement Algorithm) – kada je keš pun, koji slot izbaciti iz keša?
 - a. najranije korišćen, a bez pogodaka (Least Recently Used – LRU)
 - b. izbaciti blok koji je najduže u kešu (First In First Out – FIFO)
 - c. najređe korišćen (Least Frequently Used - LFU)
 - d. izbaciti blok koji ima najmanje pogodaka
4. Način upisa (Write Policy) – write through – write back
5. Veličina bloka
6. Broj keš memorija – više slojeva (L1, L2, L3) – razdvojiti instrukcije od podataka ili ne?

1. Veličina

- cena je u srazmeri sa veličinom – veći keš, veća je i cena
- brzina je u srazmeri sa veličinom – više keša, veća brzina (ali samo do neke tačke)

2. Način mapiranja

- Kako se adresa iz operativne memorije mapira na cache line/slot/block u kešu

-Direktno mapiranje-

-Predefinisani keš line za svaki block, samim tim mi nemamo dovoljno tih blokova. Ukoliko je blok u kešu, nalazi se na određenoj lokaciji.

-Adresa se sastoji iz dva dela:

- *Least Significant* w bitova identifikuju bajt unutar reči
- *Most Significant* s bitova identifikuju memorijski blok i sastoji se iz r bitova koji identifikuju keš slot i $s-r$ bitova koji reprezentuju tag.

-Dobre strane su što je jednostavno i nije skupo, dok su loše strane to što je fiksna lokacija u kešu za svaki blok pa se dešava da ukoliko program radi sa dva bloka koji završavaju u istom slotu, tada su promašaji keša veliki (cache miss)

-Asocijativno mapiranje-

-Ovde se blok iz operativne memorije može učitati u bilo koji slot čime se komplikuje realizacija keša, ali je fleksibilnije

-Ukoliko bi probali da pretražujemo serijski, to bi bilo užasno sporo

-Adresa iz memorije se sastoji iz taga i identifikatora reči (word)

-Imamo s bitova za tag

-Uz svaki tag imamo komparator koji poredi tag iz memorijske adrese sa svim pojedinačnim tagovima keša što znači da se svaki tag mora proveriti kada se traži u kešu

-Pretraga tagova je kompleksna jer recimo za 16384 linija imamo 16384 komparatora

-Dobre strane su što je fleksibilno, dok su loše strane to što je kompleksnija realizacija, skuplje je i potrebna je velika količina memorije za asocijativni nivo (za 16K slotova (64 KB keša), potrebno je $16384 \times 22 = 360448$ bitova asocijativne memorije (44 KB asocijativne memorije) za tagove, dok je za direktno mapiranje je bilo potrebno **16K** dodatne memorije za tagove).

-Set-asocijativno mapiranje-

-Ovo je kombinacija prethodno dva navedena mapiranja koja koristi dobre strane od oba.

-Keš se sastoji iz skupova (setova) i svaki set se sastoji iz zadatog broja slotova.

-Direktno mapiranje paralelno 2, 4 ili 8 direktnih keš slotova gde se interno na asocijativan način shvata koji od ova 3 paralelna slotova zapravo sadrži ono što nama treba, šta tražimo. Spolja izgleda kao direktno mapiranje: središnji sloj služi da se pogodi blok od 4 bajta, a donja dva bita nam govore koja od 4 bita trebaju procesoru.

-Blok iz operativne memorije se mapira na liniju u zadatom setu – npr. blok B može da bude u bilo kojem slotu (liniji) seta i tipično su 2 linije (slotova) po setu – 2 way associative mapping gde se zadati blok može naći u jednom od dva slotova u setu i 2 komparatora tagova

-Ako imamo 4-way set-associative: – 4 slotova po setu i 4 komparatora tagova

3. Algoritmi zamene

-Kada je keš pun, a potrebno je da se sadržaj iz operativne memorije donese u keš... – koji slot isprazniti?

-Ako imamo direktno mapiranje, nema posebnog algoritma, već se prazni onaj slot koji je dodeljen bloku i tu dolazi blok sa istim brojem slotova

-Kod dva druga mapiranja (asocijativno i set-asocijativno), koristi se hardverski implementiran algoritam:

- najranije korišćen, a bez pogodaka (Least Recently Used – LRU)
- izbaciti blok koji je najduže u kešu (First In First Out – FIFO)
- najređe korišćen, tj. ima najmanje pogodaka (Least Frequently Used - LFU)

4. Način upisa

-Kada procesor nešto snima u memoriju, on nije svestan da nešto od toga završi u kešu, samo je pitanje da li to odmah treba da završi i u memoriji ili ne

-Problemi su kada je više procesora u računaru, gde svaki može da ima svoj keš i postojanje U/I uređaja koji direktno upisuju/čitaju memoriju.

-Ovi navedeni problem se rešavaju putem 2 pristupa: write through i write back.

a. Write through

- sva pisanja idu i u memoriju i u keš i uvek će imati sačuvane podatke
- više procesora može da prati saobraćaj ka operativnoj memoriji i da ažurira keš
- mane su što svaki upis u memoriju mora da zauzme magistralu da bi keš kontroler snimio to i u operativnu memoriju, a to je veliki obim saobraćaja i time se usporava upis u memoriju

b. Write back

- sva pisanja se inicijalno rade samo u keš
- svakim upisom odvojimo jedan dodatni bit sa strane – UPDATE bit i postavimo mu vrednost na 1
- kada se izbacuje blok iz keša, pogleda se ovaj UPDATE bit i ako je vrednost 1, snimimo u memoriju sadržaj, a ako je nula, samo pregazimo novim sadržajem
- zbog ove tehnike keš memorije ostalih procesora ispadaju iz sinhronizacije
- U/I uređaji moraju da pristupaju memoriji kroz keš ili se namerno može izbegavati memorija koja je keširana (sistemski poziv za alokaciju memorije koja nije keširana)

5. Veličina bloka

-Što je blok (linija, slot) veći, veći je hit ratio.

-Rast bloka nije neograničen – posle određene veličine, hit ratio opada sa daljim povećavanjem bloka; povećanje veličine bloka smanjuje broj keš linija (kapacitet je konstantan); povećanje veličine bloka povećava verovatnoću da će neki podatak biti nepotreban (lokalitet koda i podataka).

-U modernim procesorima veličina bloka je od 8 do 32 bajta.

6. Broj keš memorija

-Inicijalno je postojao jedan keš

-Dva nivoa keša:

- L1 na procesoru (Pentium 16 KB)
- L2 na ploči, izveden kao SRAM memorija: brz koliko i procesor, ali sporiji od L1, veći kapacitet od L1 (sadrži više podataka od L1), 256/512 KB, kod nekih procesora, L2 je na procesorskom čipu, ali ne i u procesoru!

-Tri nivoa keša: L1, L2 i L3 – Pentium 4 (L1 – 16 KB, L2 – 512 KB, L3 – 1 MB)

-**Objedinjen (unified) keš**: i instrukcije i podaci u jednoj keš memoriji. Ako se više koriste instrukcije od podataka, keš se više puni instrukcijama (load balancing).

-**Razdvojen keš**: poseban keš za instrukcije i za podatke

- dva posebna keša
- posebno dobar u sistemima sa paralelnim izvršavanjem naredbi i pipelining-om
- kod jednog keša: ako se radi pre-fetch, a tekuća instrukcija zahteva pristup kešu (čitanje/pisanje), jedna od ove dve operacije mora da čeka, što nije slučaj sa razdvojenim keš memorijam

Masovna memorija

-Čuva podatke i programe kada se računar ugasi (tada RAM ne funkcioniše) i kada veličina podataka prevazilazi kapacitet RAM memorije.

-To su: **magnetni diskovi** (hard disk, flopi disk), **SSD diskovi**, **optički uređaji** (CD-ROM, CD-Recordable (CD-R), CD-R/W, DVD, Blu Ray) i **magnetna traka**.

Magnetni diskovi

-Imaju podlogu u obliku diska presvučenu magnetnim materijalom. Magnetni material može da se namagnetise i razmagnetise. Ranije je podloga bila od aluminijuma, a sada je od stakla koje garantuje uniformniju površinu što povećava čitljivost podataka, smanjeni broj oštećenja površine (smanjen broj grešaka), manja je udaljenost glave od površine, relativno je čvršća i ima veću otpornost na udarce.

-Ideja diskova se zasnima upravo na tom namagnetisanju feromagnetskog materijala kojim je presvučen disk.

-Čitanje i pisanje preko navoja žice zvanog glava. Za vreme čitanja i pisanja, glava je fiksirana, a ploča se rotira ispod nje. Čitanje je uznapredovalo, pisanje je isto kao i nekad.

-**Čitanje (nekada)** – magnetno polje koje se pomera ispod glave izaziva indukciju struje navoju glave i bio je isti navoj i za čitanje i za pisanje.

-**Čitanje (sada)** – odvojene glave za čitanje i pisanje, jedna uz drugu, delimično oklopljeni senzor koji menja otpornost u zavisnosti od magnetnog polja (magnetoresistive sensor – MR sensor): električna otpornost senzora zavisi od smera magnetnog polja i ovo omogućuje rad na većim frekvencijama (veća gustina zapisa i brzina).

-**Pisanje** – struja kroz navoj izaziva pojavu magnetnog polja. Impulsi se šalju u navoj i magnetno polje izaziva namagnetisanje materijala ispod glave. Različit smer struje izaziva različit smer namagnetisanja.

-Podaci na disku su organizovani tako da u zavisnosti od pozicije glave na disku, možemo videti da ona opisuje imaginarne koncentrične krugove koji se nazivaju **staze**. Postoji izvesni razmak između staza, a smanjenjem razmaka se povećava kapacitet.

-Staze se dele na sektore koji predstavljaju najmanji blok podataka. On je obično veličine 512 bitova.

-Konstantna je ugaona brzina (brzina rotacije diska) i to znači da je isti broj bitova, samo što je na većim poluprečnicima diska veći razmak između bitova (ali ih je i dalje 512).

-Sektori po stazama imaju oblik luka i glava se pomera do zadate staze i čeka da sektor naiđe ispod. Ovde se gubi proctor na spoljnim stazama jer je manja gustina podataka,

-Iz prethodno navedenog razloga je smišljen koncept **Multiple zoned recording** – upotreba zona povećava kapacitet ,gde svaka zona ima fiksni broj bitova po traci s tim što je stazama koje su dalje od centra povećan broj sektora (i bitova) i to je programibilno i brzina obrtaja se povećava kako se udaljavamo od centra.

-Pronalaženje sektora:

- ❖ Glava mora da pronađe stazu i početak sektora
- ❖ Da bi disk znao na kojoj traci i koji je sektor unutra, disk mora da se formatira tako što se zapisuju dodatne informacije (ovo je samo low level formatiranje) koje nisu dostupne korisniku i sadrže oznake staze i sektora.

*Seek – kada se glava radijalno pomera

-Low level formatiranje – svaki sektor se zna gde počinje i čita se redom: bajt za sinhronizaciju, id trake, id glave, id sektora, CRC (kontrolni kod) – da se podaci ne bi međusobno poklapali, i na kraju podaci.

-Winchester disk format:

- svaka staza ima 30 sektora fiksne dužine od po 600 bajtova svaki
- svaki sektor ima 512 bajtova podataka i 88 bajtova dodatnih podataka

-Fizičke karakteristike magnetnih diskova:

1. Fiksna ili pomerajuća glava

- a. Fiksna glava – jedna glava po stazi i glave su montirane na fiksnom postolju, svaka iznad svoje staze
- b. Pomerajuća glava – jedna glava po strani i glava je montirana na postolju koje se može pomerati iznad staza
*sada su pomerajuće glave jer fiksna glava nije moguća zbog gustog pakovanja staza u diskovima.

2. Promenljivi ili fiksni medijum(disk)

- a. Promenljivi (removable) disk – može da se ukloni iz uređaja i da se zameni drugim diskom, “neograničen kapacitet” i jednostavan prenos podataka
- b. Fiksni disk – trajno montiran u uređaju
*moderni hard diskovi su trajno montirani u uređaju, mada se mogu odšrafiti, nije samo material removable, već je ceo hard disk removable.

3. Jednostrani magnetni materijal ili dvostrani

- ❖ Magnetni sloj može biti na obe ili samo na jednoj strani podloge
- ❖ Imamo dve glave kako ne bi obrtali disk, već se čita ili piše sa obe strane i obe glave se istovremeno pomeraju (nema odvojen mehanizam u odnosu na stranu)

4. Jedna ili više ploča

- ❖ Više ploča: jedna glava po strani, glave su poravnate i spojene na istu podlogu, staze istog prečnika formiraju cilindre
- ❖ Podaci su razmešteni po cilindru gde je cilindar je skup staza sa istim prečnikom; smanjuje pomeranje glave i povećava brzinu prenosa (transfer rate)
*moderni imaju više dvostranih ploča

5. Mehanizam glave

- ❖ Procep i udaljenost glave od površine diska bi trebalo da budu što manji jer bi onda obezbedili veću gustinu zapisa podatak, ali je onda veća verovatnoća greške zbog nečistoća i nasavršenosti podloge
- ❖ Tri vrste mehanizama glave: kontakti (floppy), fiksni razmak, plutajući (leteći) – Winchester i njega imaju moderni hard diskovi
- ❖ Flopi je removable medijum (mali kapacitet i spor je) gde je glava odmah na ploči tako da je udaljenost minimalna, tako da praktično struže disk što dovodi do oštećenja

-Floppy disk je raznih veličina (u inčima), malog je kapaciteta, spor je, jeftin, ali nepouzdan jer se radom struže po njemu zbog mehanizma glave.

-Winchester hard disk:

- ❖ Hermetički zatvoren disk, ali unutar njega mora biti vazduh zbog plutajuće glave
- ❖ Jedna ili više ploča
- ❖ Lagane glave lebde na sloju vazduha koji kruži usled okretanja diska
- ❖ Veoma mali razmak između glave i ploče

-**Performanse diskova** – gleda se nekoliko stvari:

- ❖ Vreme za pozicioniranje na stazu (seek time) – vreme potrebno da se glava pomeri na zadatu stazu
- ❖ Rotaciono kašnjenje (rotational latency) – vreme potrebno da zadati sektor dođe ispod glave
- ❖ Access time = Seek + Latency – vreme pristupa
- ❖ Vreme potrebno za prenos (transfer time):

$$T = b/(rN)$$

b – broj bajtova za prenos

r – brzina okretanja diska (obrtaja u sekundi)

N – broj bajtova po traci

SSD (Solid State Disk)

-Alternativa hard diskovima i za razliku od hard diskova koji imaju magnetni medijum, SSD ima memorijska integrisana kola.

-Realizovan je električno izbrisivom memorijom – EEPROM (flash memorija) koja čuva sadržaj i posle nestanka napajanja.

-Problem je što je ograničen broj upisa u memorijske ćelije.

-Podaci se snimaju u stranice (4KB), organizovane u blokove (512KB)

- najmanja količina podataka je jedna stranica
- može se pisati u stranicu, samo ako je prethodno obrisana
- brišu se čitavi blokovi, a ne stranice

-Zbog prethodno navedenog postoji **garbage collection** – čitav blok se prekopira u memoriju, stranice koje su obrisane se ne kopiraju, blok se obriše, pa se snime samo neobrisane stranice (na drugo mesto). Ovo za posledicu ima: write amplification, odn. zbog nekoliko obrisanih stranica, čitav blok se ponovo upisuje (obično na drugu lokaciju da ne "haba" isti blok).

-**TRIM** – kada OS briše datoteku, on zapravo ne briše njen sadržaj, već je samo označi kao slobodnu, da bi kasnije prebrisao njen sadržaj (sektori koje zauzima ostaju netaknuti, a samo se datoteka označi kao obrisana). SSD to ne zna, pa i dalje čuva sadržaj obrisanih stranica prilikom garbage collecting-a. OS šalje TRIM komandu prilikom brisanja datoteke, sa spiskom stranica koje je zauzimala da ih SSD uređaj ne čuva prilikom garbage collecting-a.

-**Prednosti** u odnosu na magnetne diskove: brže se startuje (nema mehaničkih delova), manje kašnjenje (latency), pošto nema glavu koja čeka da podatak naiđe, ne pati od problema fragmentacije diska, jednaka brzina transfera bez obzira na lokaciju podatka, manja buka i radna temperatura, nije osetljiv na vibracije i magnetno polje, lakši i manje troše struje (sem ultra-brzih SSD diskova).

-**Mane:** ograničen broj upisa u ćeliju, garbage collection, koji se mora odraditi kod obrisanih stranica i skuplji 5 do 10 puta od magnetnog disk.

RAID (Redundant Array of Independent/Inexpensive Disks)

-Ukupno je 6 nivoa u upotrebi, a nama su najbitnija dva.

-Razlog za ovu ideju je što se skup fizičkih diskova vide kao jedan od strane operativnog sistema, druga stvar je što gomila tih diskova se vidi kao jedan, ali određenog kapaciteta:

- 1) totalni zbir svih kapaciteta diskova (stripe)
- 2) dva diska čuvaju iste podatke iz razloga dobrog back up-a ako jedan otkaže (mirror)

RAID 0

-Nema redundantnosti, princip stripe i podaci su distribuirani po diskovima.

-Ideja je da se dobije na brzini

-Uzastopni zahtevi verovatno neće završiti na istom disku. Velika verovatnoća da će podaci biti na više diskova. Umesto da se pomeri na sledeću traku u jednom disku, čita se sledeća traka drugog disk.

-Ovde je mana što su podaci razdvojeni između ta dva diska.

RAID 1

-Podaci su duplirani (mirror disks), raspodeljeni su po diskovima, sve kopije svake trake na odvojenim diskovima i onda se čita sa bilo kog diska, a piše se po svim diskovima.

-Spasavanje podataka je jednostavno: zameni se неисправan disk i prekopiraju se podaci; nema gubitka vremena (podaci se uvek mogu pročitati sa ispravnog diska).

-Mana je što je ovo dosta skupo.

RAID 2

- Diskovi su sinhronizovani.
- Veoma male trake – često na nivou bajta/reči.
- Dodatni bitovi za detekciju/korekciju greške se upisuju u dodatne hard diskove. Koristi se Hemingov kod (Hamming code).
- Puno redundanse – skupo i retko se koristi.

RAID 3

- Poput RAID 2, samo jedan dodatni disk, bez obzira na ukupan broj diskova.
- Po jedan bit parnosti za sve odgovarajuće bitove u diskovima se upisuje u dodatni disk.
- Podaci se rekonstruišu na osnovu ispravnih diskova i parnosti sa dodatnog diska – jednostruke greške odnosno samo ako nije otkazalo više od jednog diska.

Optički uređaji

1) CD-ROM

- Polikarbonat presvučen refleksnom materijom, obično aluminijumom. Podaci su smešteni kao rupe.
- Podaci se čitaju laserom.
- Konstantna gustina podataka – konstantna linearna brzina (promenljiva ugaona brzina) i zavisnosti da li je glava blizu ili daleko od centra, uređaj menja brzinu.
- Ima spiralnu stazu čime se još više koplikuje čitanje, mora da prati spiralu. Random access pristup se ovde teško postiže.
- Standard je da je kapacitet 700B.
- Format:
 - SYNC – identifikacija početka bloka
 - ID – zaglavlje sa pozicijom bloka i modom (0 – prazan blok, 1 – podaci sa korekcijom greške, 2 – podaci bez korekcije greške)
 - Data – podaci
 - ECC – dodatni podaci za korekciju greške (u modu 1), još podataka u modu 2 (do ukupno 2336 bajtova)
- Dobre strane: velik kapacitet, jednostavna tehnologija za masovnu proizvodnju, promenljiv medijum (removable), robusno
- Loše strane: skupo za male serije, sporo, može samo da se čita (read only)

- 2) **CD-Recordable (CD-R)** - fabrički se naprave "plikovi" jakim laserom i u uređaju se zapisivanje svodi na "pucanje plikova" slabim laserom.

- 3) **CD-RW** - izbrisivi medijum; Promena faze (phase change), materijal ima različitu refleksivnost u dva stanja (čitanje i pisanje) i stanje se menja laserom, mada je ograničen broj upisa laserom.
- 4) **DVD** - Višeslojni medijum (multi-layer). Veoma velik kapacitet (4.7GB po sloju) – maksimalan kapacitet oko 17GB (dva sloja, dve strane).
- 5) **Izmenjivi DVD:**
- Izmenjivi samo jednom: – DVD -R i DVD +R
 - DVD-RW – izbrisivi DVD
 - DVD – RAM – izbrisivi DVD koji nema spiralnu stazu, već sektore i staze poput hard diska.
- 6) **Blue ray tehnologija**
- Nastavak razvoja DVD tehnologije – drugačiji laser (plavi), koji ima manju talasnu dužinu svetla, pa može da se fokusira na manju površinu, što omogućuje veći kapacitet
 - Još veći kapacitet – 25 GB po sloju: 2 sloja (50 GB), tri sloja (100 GB) i četiri sloja (128 GB)

Magnetne trake - Serijski pristup podacima, sporo, veoma jeftino i koristi se za arhiviranje.

Sabirnica

-Povezuje ključne elemente hardvera: procesor, memoriju i ulazno-izlazni podsistem. Drugačije se naziva magistrala ili bus.

-Različiti su elementi povezivanja za različit element hardvera.

-Memorija prima i šalje podatke, prima adresu(lokaciju) za čitanje ili pisanje i takođe prima kontrolne signale za; čitanje(OE), pisanje (WE) i tajming(takt – clock).

-Ulazno-izlazni podsistem je sličan memoriji i njemu se procesor obraća. Za izlaz, on prima podatke od procesora/memorije i šalje podatke perifernoj jedinici. Za ulaz, on prima podatke od perifernoj jedinici i šalje podatke procesoru/memoriji.

-Ulazno-izlazni podsistem prima kontrolne signale od procesora i šalje kontrolne signale ka perifernim jedinicama. Takođe, prima adresu od procesora (npr. Adresu(port) perifernoj jedinici) i šalje signale procesoru za prekid.

-Procesor želi da se obraća memoriji ili kontrolerima i on čita instrukcije i podatke, upisuje podatke (nakon obrade), šalje kontrolne signale ostalim elementima, prima zahteve za prekide i reaguje na njih.

-Sabirnica je komunikacioni put (mnoštvo paralelnih žica) koji spaja dve ili više jedinica na računaru. Pošto je deljeni medijum, ne mogu svi istovremeno komunicirati jer bi se tada pomešali signali, tako da je pristup magistrali ekskluzivan.

-Može da se prenosi više bitova istovremeno (8-bitna magistrala, 16-bitna, 32-bitna). Svaka sabirnica sadrži tri podskupa žica:

- 1) Žice za podatke (**data bus**) – nebitno da li su to podaci ili instrukcije, ove žice služe za prenos istih. Širina podatka: 8, 16, 32, 64 i više bita.
- 2) Žice za adrese (**address bus**) – služi za identifikaciju izvora/odredišta podataka. Širina adresne magistrale definiše maksimalnu količinu memorije u sistemu. Npr 16-bitna adresna magistrala omogućava adresiranje 64KB memorije.
- 3) Žice kontrolni signala (**control bus**) – kontrola i tajming. Signali za čitanje/pisanje u/iz memorije, zahtevi za prekidom, takt(clock) signali.

-Fizička realizacija sabirnice je zapravo mnoštvo paralelnih linija(žica) na matičnoj ploči. Takođe sadrži konektore na matičnoj ploči, a nekada su kablovi trakasti(ribbon cables).

Broj sabirnica u računaru

-Jedna sabirnica:

- ❖ Veliki broj uređaja na jednoj sabirnici vodi propagaciji kašnjenja jer veliki broj uređaja zahteva veću dužinu žica na sabirnici, što dovodi do kašnjenja signala i takođe dolazi do zagušenja jer ukupan kapacitet može biti prevaziđen ukoliko ima puno uređaja.
- ❖ Većina sistema koristi više sabirnica da prevaziđe ove probleme.

-Više sabirnica:

- ❖ Obično su hijerarhijski organizovani
- ❖ Procesor i keš komuniciraju posebnom sabirnicom, memorija i procesor su povezani sistemskom sabirnicom, dok su svi ostali uređaji spojeni na dodatnu sabirnicu (*expansion bus*). Ovo međutim posustaje kod brzih uređaja.
- ❖ Sabirnica visokih performansi (*high performance bus*) - brzi uređaji se spajaju na sabirnicu velike brzine (*high speed bus*), dok se sporiji uređaji spajaju na dodatnu sabirnicu (*expansion bus*)

Karakteristike sabirnica

1. Po upotrebi linija:

- Posećene(dedicated) - odvojene linije za adresu i podatke
 - Multiplekirane – neke žice dele funkciju. Kada je adresa na sabirnici, specijalna kontrolna linija se aktivira(*address valid* i *data valid* kontrolna linija). Prednost je što je manji broj žica, ali mane su što je kompleksnija kontrola i degradacija performansi.
- *posvećene su bolje sa stanovništva upravljanja (jednostavnije je).

2. Arbitraža na sabirnici – javlja se kada je potrebno da jedna od jedinica preuzme sabirnicu

*samo jedan modul (element) može da kontroliše sabirnicu u jednom momentu.

***Način arbitraže:**

- Centralizovan – jedna specijalna jedinica kontroliše pristup sabirnici, a to je *Bus Controller* ili arbitar (*Arbiter*)
- Distribuirana – svaki modul ima svoju kontrolnu logiku i može zatražiti kontrolu nad sabirnicom. Svi su ravnopravni.

3. Tajming – koordinacija događaja na sabirnici

- Sinhrona – događaji diktirani taktom i postoji posebna linija za takt (clock line). Sve jedinice su spojene na liniju za takt i menjaju svoje stanje u skladu sa taktom.
- Asinhrono – po postavljanju odgovarajućih signala sa sabirnicu, procesor izbacuje komandu (čitanje/pisanje). Uređaj po prijemu svih signala pripremi podatke za upis/čitanje i potvrđuje njihovo prisustvo na sabirnici signalom Acknowledge (ACK)

PCI (Peripheral Component Interconnection) bus linije

-Obavezne:

- ❖ Sistemske linije (takt i reset linije)
- ❖ Adresne linije i linije za podatke
- ❖ Interface control (kontrolne linije)
- ❖ Arbitraža – nisu deljene linije i direktno su vezane za PCI bus arbitra
- ❖ Linije za kontrolu grešaka

-Opcione:

- ❖ Linije za prekid – nisu deljene linije i svaki uređaj je direktno vezan za kontroler prekida
- ❖ Podrška za keš
- ❖ Proširenje za 64-bitnu sabirnicu
- ❖ JTAG/Boundary Scan – za testiranje

Ulazno-izlazni podsistem

-Omogućuje razmenu podataka između procesora i perifernih jedinica i nalazi se između procesora i perifernih uređaja.

-Ovaj podsistem postoji jer ne možemo spojiti periferijske uređaje direktno na sistemsku magistralu jer imam problem sa razmenom različitih vrsta podataka, njihovim veličinama, brzinama, oblika i u tom slučaju bi procesor morao da radi sve.

-Povezan sa jedne strane sa procesorom i memorijom preko sistemske magistrale. Sa druge strane povezan sa jednom ili više perifernih jedinica.

-Perifernije jedinice mogu biti:

- ❖ Čitljive od strane čoveka (*human readable*) – monitor, štampač, tastatura
- ❖ Mašinski čitljive (*machine readable*) – uređaji za nadgledanje i kontrolu
- ❖ Komunikacije – modem, mrežna kartica

*za sve navedeno mora postojati odvojeni kontroler (u/i podsistem)

-Funkcije U/I podsistema:

- ❖ Kontrola i tajming
- ❖ Komunikacija sa procesorom
- ❖ Komunikacija sa perifernom jedinicom
- ❖ Baferizovanje podataka
- ❖ Detekcija greške

-Koraci u komunikaciji sa perifernom jedinicom:

- ❖ Procesor proverava status periferije preko UI podsistema – da li je jedinica available
- ❖ UI podsistem vraća status
- ❖ Ako je periferija spremna, procesor zahteva transfer podataka
- ❖ UI podsistem pribavlja podatke od uređaja
- ❖ UI podsistem šalje podatke procesoru

-Načini realizacije U/I podsistema:

- 1) Programiran
- 2) Sistem prekida
- 3) Direct memory access (DMA)

1) Programiran

- Procesor ima direktnu kontrolu nad ulazno-izlaznim uređajem: proverava status, zadaje čitanje/pisanje uređaju i prenosi podatke
- Detalji: Procesor zahteva UI operaciju, UI modul izvršava operaciju, UI modul postavlja statusne bitove, procesor proverava statusne bitove periodično, UI modul ne obaveštava direktno procesor, UI modul ne prekida procesor, procesor može da čeka ili da proveri status kasnije
- Sama komunikacija UI-a i procesora se odvija preko UI komandi
- UI komande - procesor zadaje adresu UI modula (identifikuje modul). Procesor zadaje komandu: *kontrola* (govori modulu šta da radi), *test* (provera statusnih bitova), *čitanje/pisanje* (modul šalje/prima podatke ka/iz perifernog uređaja)
- Adresiranje perifernog uređaja je slično pristupu memoriji, svaki uređaj ima jedinstveni identifikator i komande sadrže identifikatore (adrese)
- **Mapiranje UI uređaja:**
 - Memorijski mapirani UI (ARM procesori) – uređaji dele isti adresni prostor, nema specijalni UI komandi
 - Odvojeni UI (Intel procesori) – odvojeni adresni prostor, specijalne linije za pristup UI i za pristup memoriji i posebne komande za pristup UI-u, adresa uređaja je *port*
- Ovo se ne praktikuje jer procesor stalno proverava da li se nešto desilo, pritisnulo i slično i time se bespotrebno troši procesorsko vreme

2) Sistem prekida

- Prevazilazi problem kada procesor čeka na izvršenje UI komande
- Procesor ne proverava status uređaja
- UI modul prekida procesor kada obavi posao
- Procesor zadaje UI komandu, UI modul dobija podatke od perifernog uređaja, dok procesor radi nešto drugo, UI modul prekida procesor, procesor traži podatke od UI modula i UI modul šalje podatke procesoru
- Procesor zadaje UI komandu, radi nešto drugo i na kraju svakog instrukcijskog ciklusa proverava da li se desio prekid; ukoliko jeste: procesor zapamti gde je stao sa radom tako što back up-uje program counter, register i flagove i zatim procesira prekid
- Identifikacija modula koji je izazvao prekid može se realizovati: posebnom prekidnom linijom (ne sme se preterivati sa brojem linija), software poll (sporo), hardware poll (svi dele istu liniju za prekid) i Bus Master (prvo se zauzima sabirnica od strane UI-a da bi se izazvao prekid pa procesor potvrđuje prekid)
- Višestruki prekidi: svaki UI modul ima određen prioritet. Prekidi višeg prioriteta mogu da prekinu one nižeg nivoa i kod istovremenog zahteva, prekid višeg nivoa ima prednost nad nižim.

3) Direct memory access (DMA)

- Dodatan hardver se spaja na sabirnicu (DMA kontroler)
- DMA kontroler preuzima posao prenosa podataka
- Po završetku transfera, izaziva prekid da bi signalizirao procesoru završetak rada
- Procesor zadaje DMA kontroleru sledeće: tip operacije (čitanje/pisanje), adresu uređaja (id uređaja), početnu adresu memorijskog bloka odvojenog za transfer, količinu podataka koja će biti prenet
- Procesor nastavlja sa radom
- DMA kontroler obavlja kompletan transfer i DMA kontroler izaziva prekid po završetku
- **Krada ciklusa (Cycle Stealing)** - tokom prenosa podataka, DMA kontroler preuzima sabirnicu (procesor je tada suspendovan, samo stoji – nije prekid (interrupt). DMA kontroler prenosi jednu reč i procesor se suspenduje pre nego što pokuša da preuzme sabirnicu – pre prihvata operanada ili zapisa rezultata. Ovo usporava procesor, ali ne toliko koliko dva prethodna načina transfera
- Mesta gde možemo prekinuti procesor u radu: tokom dobavljanja i dekodiranja instrukcije od strane procesora i tokom izvršenja operacije. Ovakva realizacija je bolja jer procesor i DMA kontroler rade uporedo, nisu zaglavljani, iako je sporije, kada se sveukupno posmatra je ovako brže
- **3 vrste DMA konfiguracije:**
 1. Jedna zajednička sabirnica i svaki transfer zauzima sabirnicu dva puta: (UI uređaj => DMA i DMA => memorija) i procesor je suspendovan dva puta
 2. Sistemska sabirnica i DMA kontroler sa posebnom vezom ka perifernom uređaju. Kontroler može da podrži više od jednog perifernog uređaja. Svaki transfer zauzima sabirnicu samo jednom – DMA => memorija. Procesor je suspendovan samo jednom (hard disk kontroler)
 3. Posebno sistemska, a posebno UI sabirnica. Svaki transfer zauzima sistemska sabirnicu samo jednom – DMA => memorija. Procesor suspendovan samo jednom. Još efikasnije, pošto DMA kontroler ima samo jedan UI interfejs ka UI modulima (unificiran) (USB kontroler)

4) UI kanali

- UI podsistem postaje računarski sistem sa posebnim posvećenim procesorom
- Povećana brzina rada, smanjeno opterećenje procesora
- procesor zadaje UI podsistemu UI instrukcije i UI modul (kontroler) izvršava te instrukcije
- UI procesor se još zove i I/O Channel
- Produžetak koncepta DMA - ne bavi se samo transferom podataka od/ka periferiji i UI procesor izvršava UI instrukcije

Interfejsi

-Služe za spajanje sa periferijskim uređajem.

-**Serijski:** RS-232 (eksterni modemi), USB (kamere, tastatura, miš), Firewire (digitalne kamere, rekorderi), SATA (hard disk, CD-ROM).

-**Paralelni:** paralelni port (štampanici), SCSI (hard disk, CD-ROM), PATA (hard disk, CD).

USB

-Universal Serial Bus

-Sastoji se iz kontrolera i USB uređaja

-Uređaji povezani po principu daisy chain – ne više od 127 uređaja

-Moguće proširiti sistem USB hubovima (USB hub) – ne više od 5 nivoa

-Nema terminatora

-Omogućuje *plug-and-play* (standardizovana identifikacija uređaja) i *hotswap* (nije potreban restart po spajanju i moguće spajanje i odspajanje bez gašenja računara)

-Zadaci na testu-

1) Računski metod računanja Hemingovog koda

- Zadato 8 bitova podataka
- 4 bita za Hemingov kod
- Potrebna tabela za rešavanje

2) Načini mapiranja keš memorije: jedna grupa direktno, druga set-asocijativno

- Dobićemo 32-bitnu adresu, treba odrediti koliko bitova je za word (reč), koliko za identifikaciju cache line-a, a koliko ostaje za tag

Ivana Milutinović