

## Premoščevalnik - most (bridge)

Premoščevalnik (bridge) je novejšega izvora kot ponavljalnik. Je zahtevnejša naprava, saj podatke (pakete) ne le ojači in pošlje na ustrezen vmesnik, temveč jih najprej shrani v medpomnilnik. Podatke (pakete) v medpomnilniku preveri in v primeru, da premoščevalnik ne ugotovi napake, paket pošlje naprej na ustrezen vmesnik (enega ali več) premoščevalnika, glede na ciljni naslov. Ker se paket najprej shrani, in ker je potreben določen čas, da premoščevalnik ugotovi, na kateri vmesnik poslati paket, so premoščevalniki počasnejši od ponavljalnikov. Najbolj sta se uveljavili dve vrsti premoščevalnikov:

- transparentni premoščevalniki (transparent bridges), asami določajo pot po kateri bo potoval paket. Takšno ime imajo zato, ker so končnim postajam "nevidni" oziroma transparentni. Transparentni premoščevalniki uporabljajo STA (*Spanning Tree Algorithm*) algoritem za iskanje optimalnih poti pri prenosu paketov. Sporočila, ki si jih med seboj pošiljajo premoščevalniki, se imenujejo BPDU (*Bridge Protocol Data Units*). Priključimo jih med enake lokalne računalniške mreže, ne da bi pri tem spreminjali obstoječe stanje. STA algoritem se uporablja za povezovanje omrežij Ethernet.
- Premoščevalniki SRB (Source Routing Bridges) pa delujejo nekoliko drugače. Ze iz imena je razvidno, da mora končna postaja, ki zeli poslati paket, sama ugotoviti optimaino pot. Informacijo o optimaini poti mora vključiti v poslani paket. Naloga premoščevalnikov je, da na podlagi informacije, ki je vsebovana v paketu samem, paket "dostavijo" na segment na katerem se nahaja naslovljena končna postaja. SRB algoritem se uporablja za povezovanje omrežij Token Ring.

Naslov MAC (Media Access Control) je enoumno določen naslov računalnika, definiran na nivoju strojne opreme. V omrežjih Ethernet je to njegov naslov v krajevni mreži. Ko je računalnik povezan na internet, se naslovu MAC priredi naslov IP. Naslovi MAC so 48 bitni, v šestnajstiškem številskem sistemu pa so predstavljeni z 12 mestnim številom.

Med postopkom inicializacije pregleduje premoščevalnik vse MAC naslove paketov. Vsak novi izvorni naslov shrani v interno tabelo, razdeljeno na posamezne segmente. Če je povezanih zelo veliko število delovnih postaj, postanejo tabele obsežne in delovanje premoščevalnika se upočasni.

Paketi potujejo med premoščevalniki po različnih poteh. Da ne bi prišlo do kroženja paketa zaradi vzporednih poti se uporablja STA (*Spanning Tree Alghoritem*) algoritem, ki ima nalogo da:

- ustvari topologijo brez zank (med dvema točkama je možna le ena pot, ostale povezave niso aktivne)
- v primeru izpada aktivne povezave aktivira neaktivno povezavo tako, da je možna povezava med poljubnima premoščevalnikoma, vendar samo po eni poti.

Katere postaje bodo aktivne in katere neaktivne, izračuna STA algoritem na podlagi uteži, ki jih določi administrator omrežja. Ta jih izbere tako, da so aktivne tiste povezave, ki jih želi. Informacijo o stanju povezav in izmenjavo tabel med premoščevalniki zagotavljajo sporočila BPDU, ki se periodično izmenjujejo med premoščevalniki.

Ker premoščevalniki operirajo z MAC naslovi, ki so definirani na 2. (logični) plasti referencnega modela OSI, pravimo, da so premoščevalniki naprave, ki delujejo na 2. plasti modela OSI.

Premoščevalnik ima naslednje lastnosti:

- število premoščevalnikov v omrežju ni omejeno,
- fizična razsežnost omrežja, ki ga povezujejo premoščevalniki je takorekoč neomejena,
- s premoščevalniki lahko deino povečamo prepustnost omrežja,
- omogočajo povezavo različnih LAN omrežij med seboj: Ethernet, Token Ring, FDDI....
- Med seboj jih ločimo po:

o številu segmentov, ki jih povezujejo,

o LAN tehnologijah, ki jih podpirajo: Ethernet, Token Ring, FDDI....

o algoritmih za določanje poti po kateri bo potoval paket.

Poznamo:

- lokalne premoščevalnike (angleško *local bridge*), ki med seboj povezujejo segmente lokalnega omrežja in
- oddaljene (angleško *remote bridge*), ki med seboj povezujejo oddaljena lokalna omrežja.

Premoščevalniki so počasne naprave, zato jih v lokalnih omrežjih izpodrivajo preklopna stikala, v prostranih omrežjih (povezovanje oddaljenih lokalnih omrežij) pa usmerjevalniki.

## Koncentrator – HUB

Danes je najbolj pogosto uporabljan ponavljalnik prav gotovo 10BaseT koncentrator (hub), ki je v bistvu ponavljalnik z več vmesniki. Navadno imajo koncentratorji 8,12,16 ali 24 vmesnikov. Delovne postaje, ki so povezane med seboj s prenosnim medijem in na tej povezavi ne vsebujejo nobene naprave za medmrežno povezovanje tvorijo svoj segment omrežja. Če postaje ločuje koncentrator, pravimo, da so postaje na istem logičnem segmentu, saj potujejo vsi podatki do vseh postaj.

## Diski

Diskovni mehanizem je zapečaten enota, ki jo uporablja za trajno skladiščenje podatkov in programov. Vsebuje čvrste plošče v obliki diskov iz aluminija ali stekla. Površinska gostota pomnjenja se meri vzdolž steze na disku v enotah Mbit/inč<sup>2</sup> ali Gbit/inč<sup>2</sup>. Današnje gostote znašajo 6 do 20 Gbit/inč<sup>2</sup>

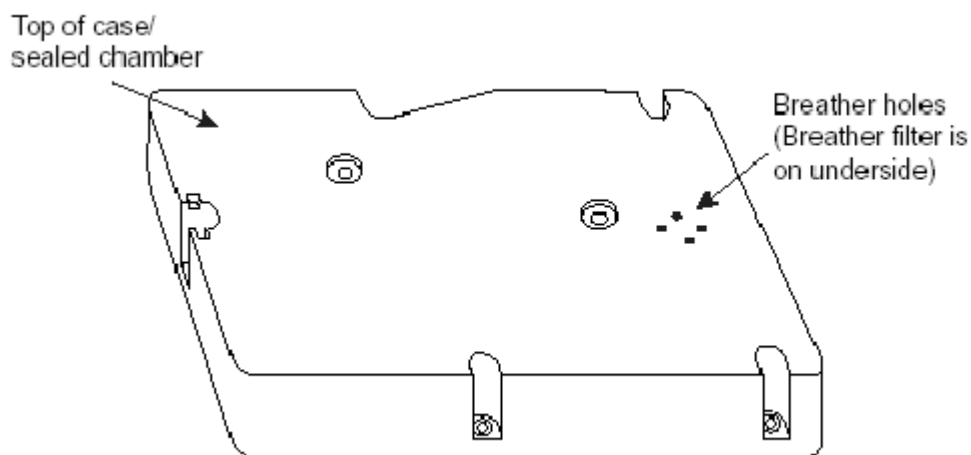
Osnovni deli diska do:

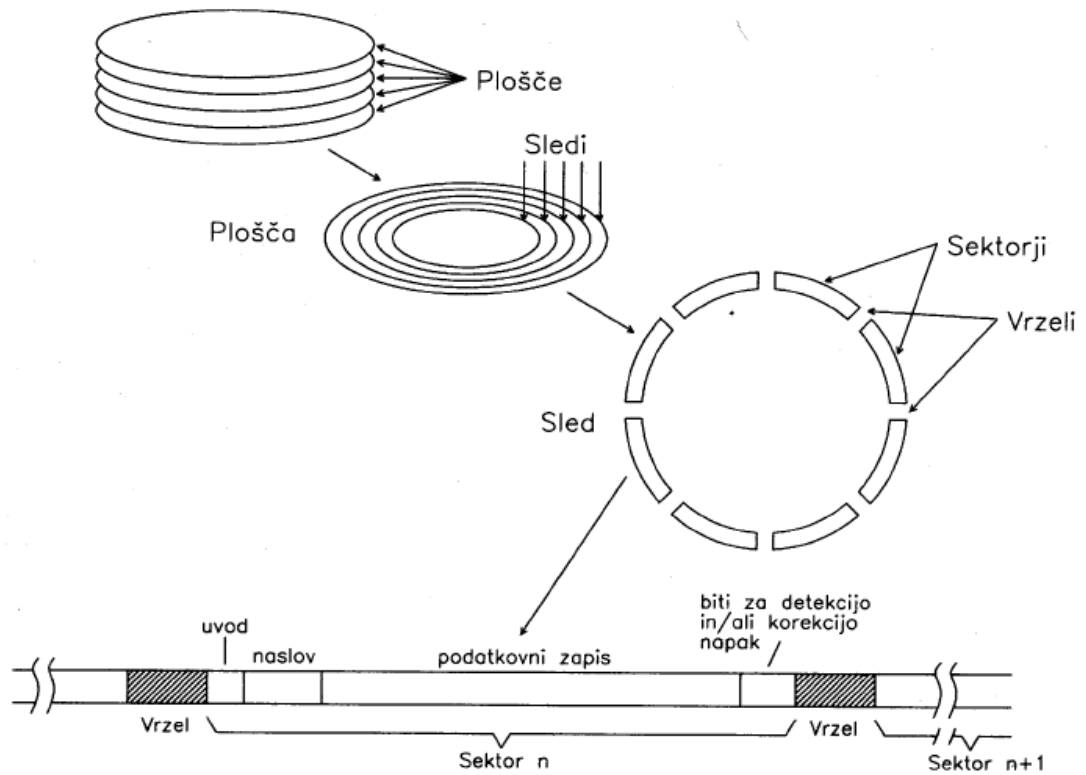
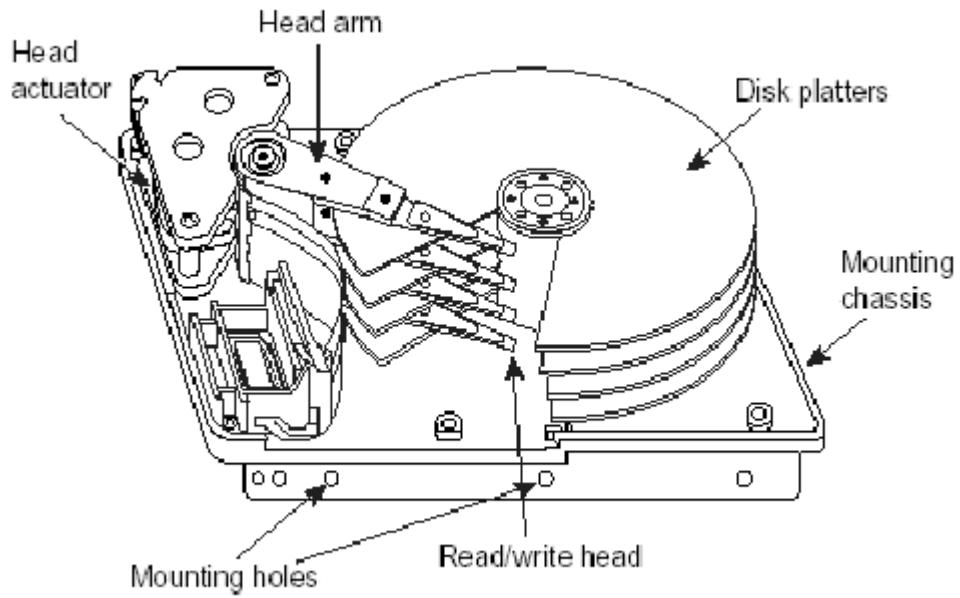
- ~ Plošče
- ~ Logične plošče
- ~ Glave za branje/pisanje
- ~ Kabli in priključki
- ~ Pristopni mehanizem (activator glav)
- ~ Deli za naravnanje (kratkospojilniki, stikala)
- ~ Krmilnik mehanizma
- ~ Motor z osjo

Glave berejo in vpisujejo podatke v koncentričnih krogih – stezah, kli se delijo na segmente – sektorje. Vsak sektor lahko običajno pomni 512 bajtov. Steze se nahajajo na več ploščah, ki so postavljene ena nad drugo. Mnogi diski imajo prek 20000 stez po inču. Steze vseh plošč, ki so na enaki oddaljenosti od centra diska, sestavljajo cilindar. Vsaka površina plošle ima svojo glavo, te pa so pritrjene na skupni nosilec. Prvobitni diski so se vrteli s hitrostjo 3600 obratov v minuti, sedaj pa dosegajo vrednosti do 10000.

Glave se med delovanjem ne dotikajo površine plošče, pač pa zaradi aerodinamične oblike in velike hitrosti lebdiijo nad površino. Pri izključitvi napajanja se spustijo na površino za pristajanje. Lebdenje jim omogoča zelo tanka zračna blazina. Če se ta zaradi prahu ali vibracij stanjša, laho glava pade in udari v površino. Padec poškoduje disk, s tem pa se izgubi nekoliko bitov ali pa se disk popolnoma uniči. Večina mehanizmov uporablja posebna maziva za plošče, da bi vzdržale vsakodnevne vzlete in pristajanja. HDA je sklop glav in diskov, ki se zapečatijo v prostorih brezhibne čistoče.

Steza je en krog podatkov na eni strani plošče. Prevelika je za učinkovito shranjevanje podatkov (preko 100kB) in malih datotek. Zato se steze razdelijo na zabeležene oddelke – sektorje. Ko se disk formatira se ustvarjajo področja za identifikacijo, ki jih krmilnik izkorišča za obeleževanje sektorjev in prepoznavanje začetka in konca vsakega sektorja. Zaradi teh delov prihaja do razlike med neformatirano in formatirano kapaciteto pomnjenja diska. Vsak sektor ima uvod, naslov za označitev sektorja, del za shranjevanje podatkov in kontrolni niz bitov. Med sektorji se nahajajo vrzeli.





## Bralni Pomnilnik

Za branje pomnilnike veljata dve značilnosti. Prva je ta, da lahko njihovo vsebino le beremo, druga pa, da se vpisana informacija zadrži v pomnilniku tudi po izklopu napajanja. Prva lastnost nima bistvenega pomena, saj nam omogočajo tudi bralno pisalni pomnilniki. Predvsem je pomembna lastnost, da lahko zadržijo vpisano informacijo tudi po izklopu napajanja, kar pa se vedno dogaja, ko končamo delo z računalnikom. Čemu pa potrebujemo trajno vpisano informacijo, ki je sami ne moremo spreminjati? Za »obuvanje« ali zagon računalnika in nalaganje celotnega operacijskega sistema je potreben neki program, ki ostane v delovnem pomnilniku tudi, ko ni več napajalne napetosti. Procesor takoj po resetiranju s prav določene lokacije v delovnem pomnilniku pobere naslov in ga zapiše v programski števec. Ta naslov kaže na začetne postopke, ki se pričnejo izvajati ob zagonu računalnika. Za vpis teh programskih procedur in morebitnih pripadajočih podatkov pa potrebuje bralni pomnilnik.

-Organizacijsko so ti pomnilniki besedno orientirani v 8-bitne besede. Besedna orientacija ima svoj smisel v tem, da je ves program vpisan v enem čipu. Pri bitni orientaciji bi bilo to rešljivo tako, da bi bil vsak bit neke instrukcije v svojem čipu in šele osem pravilno sestavljenih čupob bi dalo ustrezen program. Vsekakor zelo zapletena rešitev, ki bi v praksi povzročala veliko težav. Principialno organizacijo bralnega pomnilnika prikazuje slika 5.3.

Izhodi iz naslovnega dekodirnika so posamezne besedne linije, ki povezujejo bitne celice iste besede; na prikazani sliki je beseda dolga le tri bite, praktično pa je vedno daljša. Posamezne bitne celice so sestavljene iz programirljivih povezav in veznih elementov, kot so diode ali tranzistorji. Izhodi iz naslovnega dekodirnika so vhodi v pomnilniško matriko, izhodi iz matrike pa so podatkovne linije za posamezne podatkovne bite D. Za določeno vhodni naslov se vzbudi le ena besedna linija, podatki, ki so zapisani v njenih bitnih celicah, pa se preberejo s podatkovnih linij na izhodnih bralnih ojačevalnikov. Glede na izvedbo bitne celice in njeno povezave v pomnilniško bralnih ojačevalnikov. Glede na izvedbo bitne celice in njeno povezave v pomnilniško matriko, delimo bralne pomnilnike v naslednje skupine:

- pomnilniki, ki so programirani v masko (ROM-pomnilniki)
- električno programirljive celice (PROM-pomnilniki)
- reprogramirljive celice (EPROM, EEPROM-pomnilniki).

ROM-pomnilniki imajo zelo preprosto izvedeno bitno celico, ki je lahko kar polprevodniška dioda ali tranzistor s programirljivo povezavo; pri tem predstavlja neprekinjena povezava (dioda ali tranzistor sta vključena v matriko) eno logično stanje, prekinjena povezava, ko dioda ali tranzistor nista vključena v matriko, pa drugo logično stanje. Neprekinjena vez predstavlja logično stanje 1 na vhodu v bralni ojačevalnik, ker pa ta deluje kot invertirani ojačevalnik, dobimo na podatkovnem izhodu stanje 0. prekinitev povezave pa predstavlja na vhodu v ojačevalnik logično stanje 0, na podatkovnem izhodu pa dobimo logično stanje 1. Kapacitete ROM-pomnilnikov so ponavadi od  $32\text{ K} * 8$ ,  $64\text{ K} * 8$  pa do  $1\text{ M} * 8$  z dostopnimi časi od 100 do 300 nanosekund. Pomnilnike teh izvedb imenujemo tudi mask pomnilniki ROM.

PROM-pomnilniki imajo prav tako za bitno celico tranzistor ali polprevodniško diodo, le da je ta element povezan v matriko prek elektriško oslABLJENE povezave, ki jo imenujemo varovalka (fusible link). Pri neprogramiranem pomnilniku so vse te povezave prevodne in s tem tudi vsi tranzistorji ali diode vključene, vsi vhodi v bralne ojačevalnike so zato v stanju 1 in vsi podatkovni izhodi so v stanju 0, če ojačevalniki invertirajo logični stanje. Takšna izvedba pomnilnika omogoča uporabniku, da si pomnilnik sam sprogramira, kar stori s programatorjem, ki s povišano napetostjo prežge varovalke tistih celic, kjer se v skladu s programom zahtevajo prekinitve povezav. Brisanje vsebine kakor tudi ponovno programiranje nista mogoča, zato se tudi ti pomnilniki vse redkeje uporabljajo.

EPROM-pomnilniki imajo za uporabnika zelo ugodno lastnost, ki se kaže v tem, da lahko njihovo vsebino brišemo in nato ponovno programiramo. Brisanje vsebine opravimo z ultravijolično svetlobo, ki vstopa v mikrovezje skozi majhno okence na čipu. Po tem okencu se tudi prepoznajo EPROM.-pomnilniki. Pomnilno celico sestavlja MOSFET-tranzistor s plavajočimi vrati (FAMOS- Floating Gate Avalanche Inj MOS).

Potem pa poznamo še EEPROM-pomnilnik ki pa je po izvedbi skoraj povsem enak EPROM-pomnilnikom, le da imajo trikrat tanjšo izolacijsko plast okoli plavajočih vrat.

## BCD-kodiranje

Množico B naj sestavljajo cifre desetiškega številskega sestava takole:

$$B = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}.$$

Množico B želimo kodirati z elementi binarne množice A

$$A = \{0, 1\}$$

tako, da bo koda enakomerna in da bo zadovoljen pogoj  $n \leq m^q$

n	q
1	0
2	1
4	2
8	3
16	4
32	5
64	6
128	7

Določiti moramo, kolika naj bo dolžina besede, to je vrednost q, da bo koda enakomerna. V tabeli je dana odvisnost n od q in vidimo, da za kodiranje 8 elementov potrebujemo trimestno besedo, za kodiranje 16 elementov pa štirimestno besedo, ki sicer omogoča kodiranje 16 različnih elementov, množica A pa jih vsebuje le 10. Šest besed je preveč, imenovali jih bomo redundantne kombinacije. Opisani način kodiranja imenujemo binarno kodiranje dekade ali BCD-kodiranje. Izmed 16 razpoložljivih kombinacij lahko za kodiranje 10 znakov množice desetiških cifer praviloma izberemo 10 kateri kolih kombinacij. Tako dobimo različne vrste BCD-kod, ki so prikazane v tabeli spodaj.

Desetiška cifra	Vrsta BCD - kode					
	8421	2421	5421	735(-6)	Excess-3	ciklična
0	0000	0000	0000	0000	0011	0001
1	0001	0001	0001	1001	0100	0101
2	0010	0010	0011	0111	0101	0111
3	0011	0011	0011	0010	0110	1111
4	0100	0100	0100	1011	0111	1110
5	0101	1011	1000	0100	1000	1100
6	0110	1100	1001	1101	1001	1000
7	0111	1101	1010	1000	1010	1001
8	1000	1110	1011	0110	1011	1011
9	1001	1111	1100	1111	1100	001

V zvezi z navedeno tabelo lahko povzamemo nekaj značilnosti:

- ~ Prvi štirje načini kodiranja kodirajo cifro 0 s štirimi ničlami, kar lahko privede do dvoumnih podatkov, saj izpad napajanja da enak rezultat kot takrat, ko je podatek enak nič;
- ~ Pomanjkljivosti prvih štirih načinov odpravljata zadnji dve kodi, pri katerih je podatek ničla kodiran tako, da so v kodi navzoče enice;
- ~ Ciklična koda ima poleg tega še eno značilnost, in sicer to, da je sprememba iz ene pozicije v drugo vedno storjena le na enem mestu, zato takšno kodo imenujemo enokoračna koda.