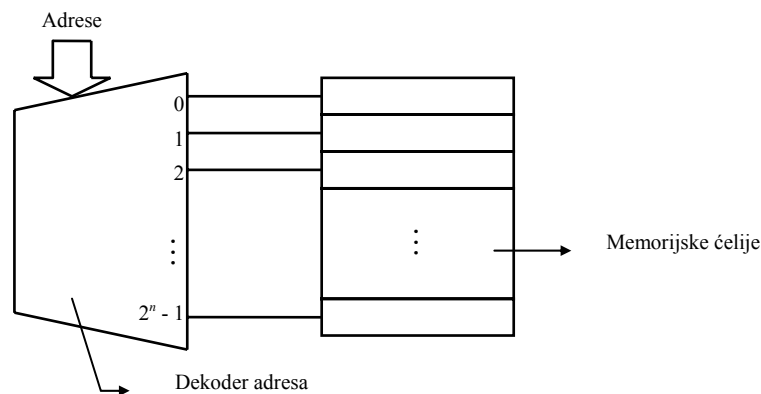


3. GLAVNA MEMORIJA

3.1. Interna organizacija glavne memorije

Glavna memorija je deo računarskog sistema u kome se čuvaju programi i podaci i organizovana je u lokacije. Svaka lokacija se sastoji od istog broja ćelija. U memorijskoj ćeliji može da se čuva samo jedna od dve različite vrednosti. Ove vrednosti su binarne cifre 0 i 1 koje su poznate kao *bitovi*, tj. osnovna jedinica memorije je binarna cifra nazvana bit. Veličina koja je smeštena u memoriji, nezavisno od toga da li ona predstavlja deo instrukcije ili podataka, je u binarnoj formi. Isti zaključak važi i za deo informacije koji se prenosi preko magistrala ili se smešta u druge delove računara.

Svaka lokacija ima svoj broj koji se zove *adresa*. Program pristupa specificiranoj lokaciji na osnovu adrese. Ako memorija ima n lokacija, tada adrese primaju vrednosti od 0 do $n-1$. Susedne lokacije imaju uzastopne adrese. Ako je adresa m -tobitna, tada maksimalan broj lokacija koje se direktno adresiraju iznosi 2^m . Na slici 3.1 prikazana je simbolička prezentacija memorije. Pravougaono polje predstavlja skup lokacija u kojima je moguće vršiti upisivanje ili čitanje. Trapezno polje se odnosi na adresni dekodler pomoću koga se bira (selektuje) jedna od memorijskih lokacija čija je adresa prisutna na ulazu.

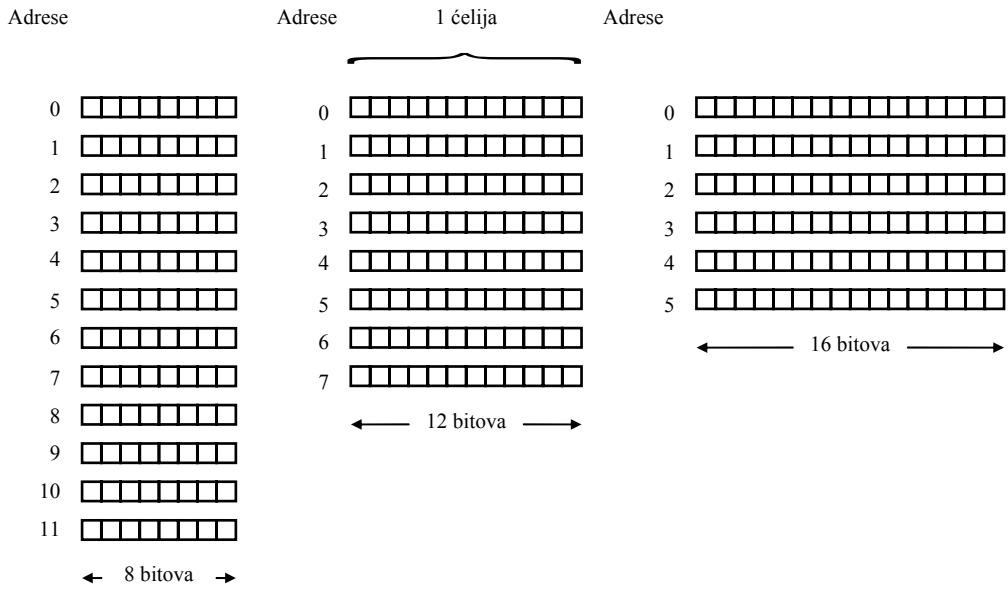


Sl. 3.1. Simbolička prezentacija memorije.

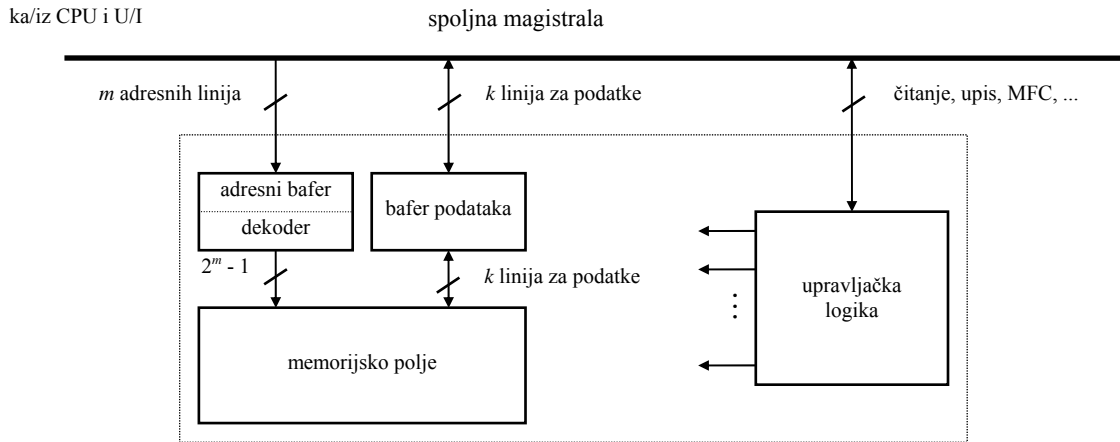
Obično se u memorijskim lokacijama smešta jedan bajt, a to je grupa od 8 bitova. Reči su veće grupe bitova a čine ih obično dva ili više bajtova. Računar kod koga je reč 16-bitna ima dva bajta po reči, dok računar kod koga je reč 32-bitna ima četiri bajta po reči. Nibla (*nibble*) je skup od četiri bita.

Na slici 3.2 prikazana su tri načina organizacije 96-bitne memorije. Uočimo da su kod 8-bitne organizacije za potpuno kodiranje potrebna četiri bita a kod 12-bitne i 16-bitne po tri.

Na slici 3.3 prikazana je interna organizacija memorije. Kao i CPU, memorija poseduje adresni bafer i bafer za podatke. Upravljačko kolo prima signale čitanje/upis (read/write) koji se predaju od strane CPU-a ili U/I podsistema preko spoljne magistrale. Na osnovu ovih ulaza upravljačko kolo uz pomoć adresnog dekodera generiše signal pomoću koga se obavlja interna selekcija memorijske lokacije. Upravljačko kolo generiše i druge signale pomoću kojih se upravlja internim radom memorije.



Sl. 3.2. Tri načina organizacije 96-bitne memorije.

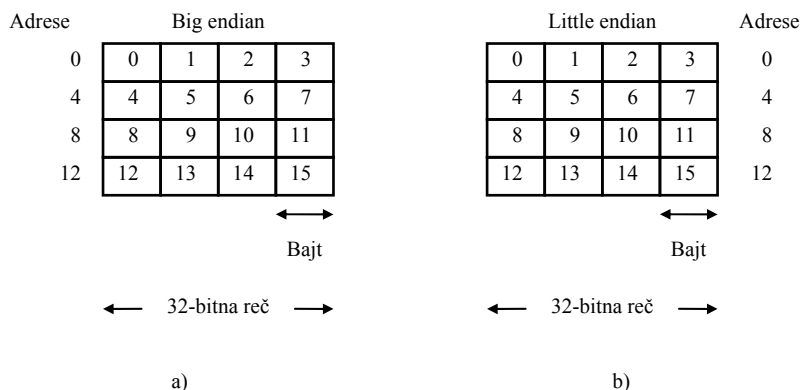


Sl. 3.3. Interna organizacija memorije.

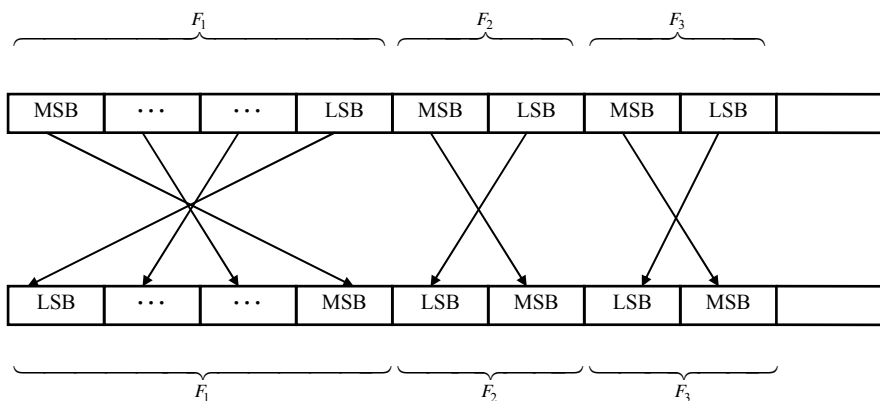
3.2. Urednost bajtova

Bajtovi u reči se mogu numerisati sa leva-na-desno, ili sa desna-na-levo. Na slici 3.4 prikazan je deo memorije 32-bitnog računara. Notacija prikazana na slici 3.4a odgovara proizvodima firme Motorola, i zove se "big endian", a notacija sa slici 3.4b odgovara proizvodima firme Intel, i zove se "little endian".

Važno je shvatiti da će kod oba slučaja, "little endian" i "big endian", 32-bitni broj čija je numerička vrednost, recimo 6, biti predstavljen pomoću tri 3 bita oblika 110 na desnoj strani (LS bitovi) i 29 bitova 00...0 na levoj strani. Kod "big endian" šeme (shodno slici 3.4) to su bitovi bajta 3 (ili 7, ili 11, ...), dok su kod "little endian" šeme to bitovi bajta 0 (ili 4, ili 8, ...). Kod oba slučaja reč koja sadrži ovu celobrojnu vrednost ima adresu 0 (binarna prezentacija).



Sl. 3.4. (a) Notacija Big endian. (b) Notacija Little endian.



Sl. 3.5. Konverzija zapisa od "big endian" na "little endian".

Sa programske tačke gledišta irelevantno je koji se tip adresiranja koristi, mada postoje i situacije kada moramo biti svesni "endian" načina adresiranja. Ako se binarni podaci prenose između dve mašine, ili, konkretnije, vrši se prenos binarnih datoteka (fajlova) iz IBM PC (zasnovan na CPU Intel 80386) ka radnoj stanici SUN-3 (zasnovana na CPU Motorola 68030) javiče se problemi. Format podataka oba procesora, što znači, način kako se vrši predstavljanje celobrojnih vrednosti (*integer*) i znakova (*character*) su identični sa izuzetkom razlike koja se odnosi na način rada "endian". Analizirajmo slučaj zapisa koji sadrži četvorobajtno polje F1, iza koga slede dva dvobajtna polja F2 i F3 (slika 3.5). Analizirajući sliku 3.5 vidimo da način kako će se izvršiti premeštanje bajtova (konverzija iz jednog formata u drugi) zavisi od detaljnog poznavanja izgleda uredenosti podataka (*data layout*).

3.3. Karakteristike glavne memorije

Bazična tehnologija za implementaciju glavne memorije koristi poluprovodnička integrisana kola. U suštini, kao i kod CPU to su LSI ili VLSI čipovi velike gustine pakovanja i jako izražene kompleksnosti. Sledeće osobine sa projektantske tačke gledišta su ključne za izbor memorije:

- **Brzina** - vreme koje protekne između iniciranja operacije i završetka te operacije (na primer između Read i MFC). Brzina se izražava u zavisnosti od sledeća dva termina:
 - a) vreme pristupa (*access time*) - definiše se kao vreme koje protekne od trenutka kada je adresa stabilna do trenutka kada se memorija odazove sa stabilnim podacima.
 - b) vreme ciklusa (*cycle time*) - minimalno vremensko kašnjenje koje je potrebno da prođe između dve sukcesivne memorijske operacije (na primer, vreme između dve sukcesivne operacije čitanja). Vreme

ciklusa određuje koliko se brzo može pristupati memoriji (na kontinualnoj osnovi). Vreme ciklusa je obično duže od vremena pristupa.

- **Gustina** - koliko se bitova može čuvati (smestiti) po memorijskom čipu. U opštem slučaju ovo određuje koliko je čipova potrebno za implementaciju glavne memorije računara.
- **Disipacija snage** - kod memorijskih čipova disipacija snage specificira se u zavisnosti od snage koja se troši kada se vrši obraćanje nekoj lokaciji koja pripada tom čipu (*operating power*), i snage koja se troši kada se ne vrši obraćanje (*standby power*).
- **Cena komponenata** - obično cena ukupnog memorijskog podsistema (primarna i sekundarna memorija) ima veliki uticaj na ukupnu cenu računarskog sistema.
- **Drugi faktori** - u ovu grupu spadaju pouzdanost, kompatibilnost sa drugim memorijskim čipovima, organizacija memorije u zavisnosti od broja bitova po lokaciji i dr.

3.4. Hijerarhija memorije

Projektantska ograničenja kod računarskih memorija mogu se iskazati kroz tri pitanja: Koliko je memorije potrebno? Koliko je memorija brza? Koliko je memorija skupa?

Odgovor na pitanje koliko memorije treba ugraditi će uvek ostati otvoren.

Na drugo pitanje relativno je lakše odgovoriti. Da bi se postigle dobre performanse memoriju je bolje ugraditi u CPU. Vreme pristupa unutrašnjoj memoriji je kraće od vremena pristupa spoljnoj memoriji.

Odgovor na treće pitanje bi glasilo: Cena memorije mora biti u skladu sa ostalim komponentama sistema.

U ovom trenutku se koriste različite tehnologije za implemetaciju memorijskog sistema. Imajući u vidu različit spektar korišćenih tehnologija, moguće je izvesti sledeće zaključke:

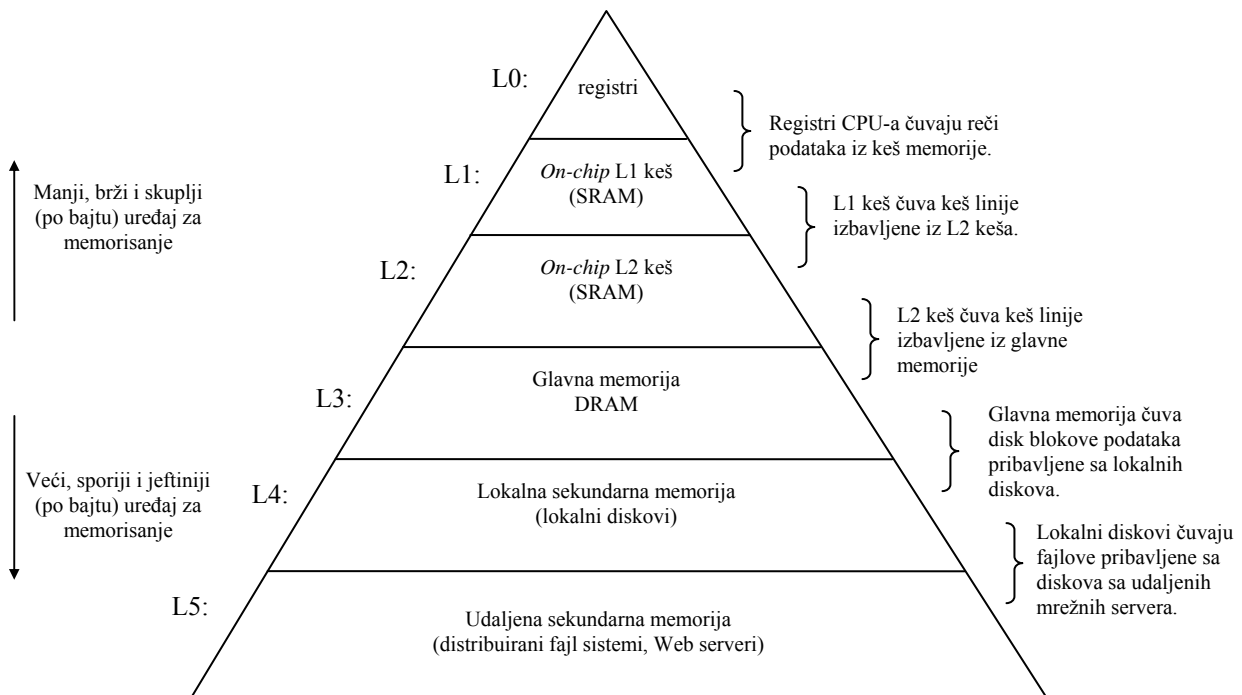
- kraće vreme pristupa - veća cena po bitu,
- veći kapacitet - manja cena po bitu,
- veći kapacitet - duže vreme pristupa.

Dilema koja se projektantu postavlja je sledeća: Projektant želi da ugradi u svoj sistem memoriju većeg kapaciteta iz dva razloga:

- veći kapacitet je uvek potreban,
- cena po bitu je tada niža.

Na žalost, da bi ispunio zahteve vezane za performanse u pogledu brzine rada, projektant mora da koristi:

- skupu memoriju,
- memoriju malog kapaciteta sa veoma kratkim vremenom pristupa.



Sl. 3.6. Hijerarhija memorije.

Rešenje za ovu dilemu se ne ogleda u ugradnji jedinstvenog tipa memorijskih komponenti ili tehnologija, nego korišćenja memorijske hijerarhije. Tipična hijerarhija je prikazana na slici 3.6. U tabeli 3.1 su prikazane performanse i kapacitet različitih tipova memorije.

Tab. 3.1. Performanse memorije i kapacitet memorisanja.

Memorijski medijum	Tipično prosečno vreme pristupa	Propusna moć	Kapacitet medijuma	Obim bloka prebačenog sa višeg na niži nivo	Ko upravlja prenosom podataka	Tehnologija izrade
Registri CPU-a	200ps-1ns	0.5-60GB/s	256B-1kB	Reč obima 2B ili 4B	Upravljačka jedinica CPU-a	CMOS SRAM
L1 keš memorija	5-10ns	0.8-1GB/s	16-64kB	Linija 4-32B	Primarni keš kontroler	CMOS SRAM
L2 keš memorija	15-40ns	0.1-0.3GB/s	128kB-1GB	Linije 4-128B	Sekundarni keš kontroler	CMOS SRAM
Glavna memorija	50-100ns	20-80MB/s	256MB-1GB	Stranice 4kB	MMU (jedinica za upravljanje kodom memorije)	CMOS DRAM
Slotovi proširenja glavne memorije	75-500ns	800kB-30MB/s	1-10GB	Stranice 4kB	MMU	CMOS DRAM
Disk keš	60-500ns	900kB-30MB/s	1-10MB	Blokovi 4kB	Kontroler uređaja	CMOS DRAM
Kruti disk	5-50ms	1200-6000kB/s	100-500GB	Fajlovi obima MB	Kontroler uređaja	Magnetni medijum
Flopi disk	95ms	100-200kB/s	1.44MB	Fajlovi obima MB	Kontroler uređaja	Magnetni medijum
CD-ROM	100-500ms	500-4000kB/s	600MB-20GB	Fajlovi obima MB	Kontroler uređaja	Optički zapis
Trake (cartridge)	0.5s pa više	2000kB/s	1-10TB	Fajlovi obima MB	Kontroler uređaja	Magnetni medijum

3.5. Tipovi poluprovodničkih memorija

Postoje dve familije memorijskih čipova:

- RAM (*Random Access Memory*) - bilo kojoj lokaciji se može pristupiti radi čitanja ili upisa u nekom fiksnom vremenskom trenutku koji je nezavisan od adrese lokacije.
- ROM (*Read Only Memory*) - memorijskim lokacijama se može pristupati samo radi čitanja.

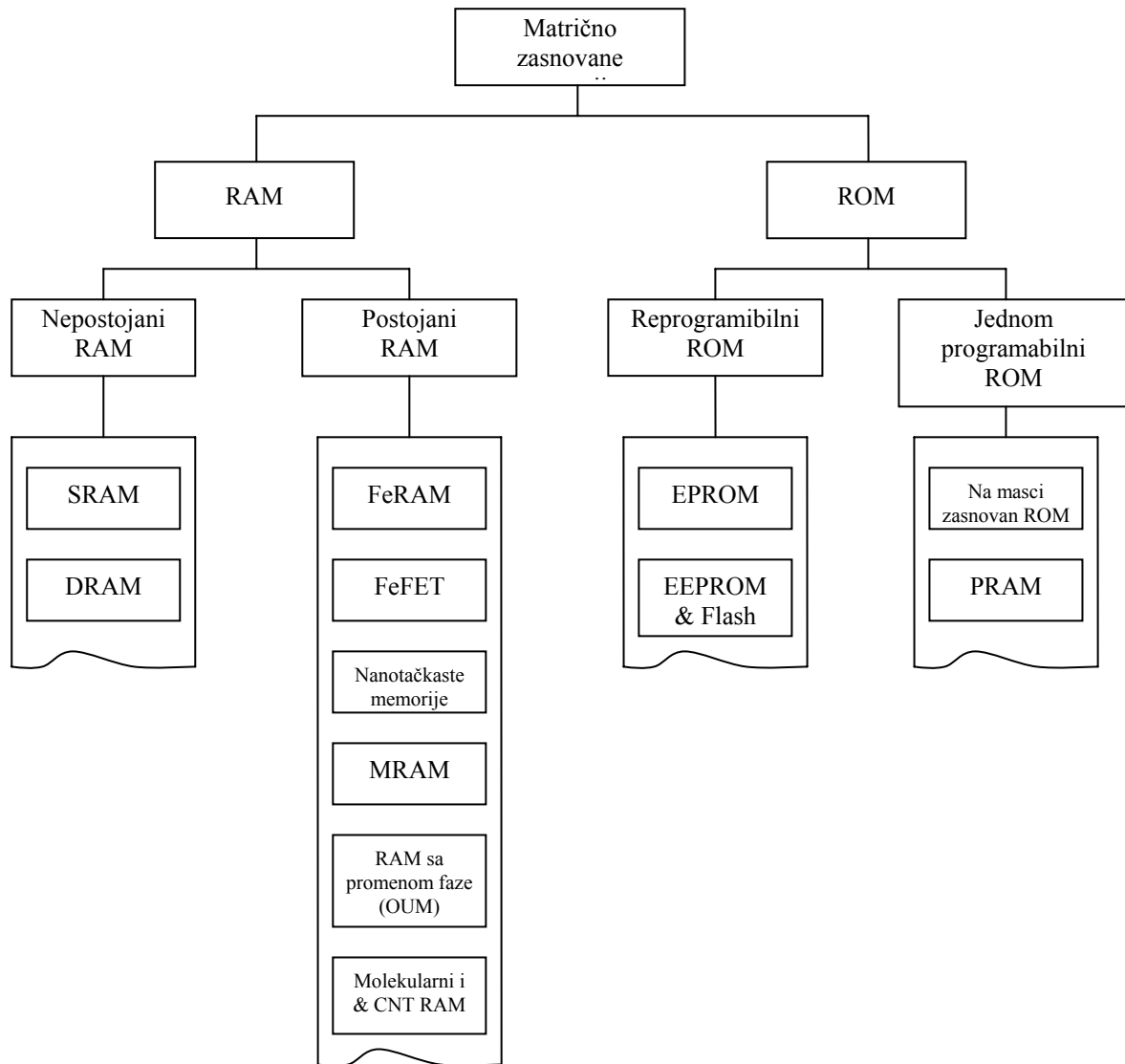
Na slici 3.7 prikazane su kategorije najbitnijih tipova memorijskih čipova zasnovanih na matricama.

3.5.1. ROM

Jednom računarskom sistemu je memorija potrebna za smeštaj naredbi programa, kao i promenljivih i konstanti. Naredbe se često smeštaju u ROM a podaci u RAM.

Informacija koja je smeštana u ROM je nepromenljiva. Ona se zadržava čak kada se i napajanje sistema isključi. Sa druge strane, RAM čuva informaciju samo dok je povezan na izvor za napajanje.

Sadržaj ROM-a se može definisati u toku fabričke izrade memorijskog čipa, pa u tom slučaju kažemo da se radi o "maskiranom" ROM-u. Korisnik je taj koji proizvođaču memorijskih čipova specificira sadržaj memorije, a proizvođač projektuje fotolitografsku masku sa korektnim bit oblicima. Proces maskiranja podataka u ROM je skup i dugotrajan, ali kada se završi, sami memorijski čipovi su relativno jeftini. Maskirani ROM se zbog toga koristi kod visokoserijskih aplikacija, tj. tamo gde program ne treba menjati.



Sl. 3.7. Kategorije matrično zasnovanih memorija.

Drugi tip ROM-a je PROM (*Programmable Read Only Memory*). Cena PROM-a je u odnosu na maskirani ROM viša kada su u pitanju veće količine, ali je prednost ta što proizvođač računarskog sistema može po potrebi programirati ovaj čip, a time i prilagođavati program konkretnoj aplikaciji. Proces programiranja kojim se u stvari puni ROM podacima ili naredbama obavlja se posebnim uređajem koji se zove PROM programer. PROM-ovi se isporučuju u sledećim varijantama:

- PROM-ovi tipa "topljivi osigurač". Fabrički su u svim memorijskim lokacijama upisane "0" ili "1". U toku programiranja, u zavisnosti od dovedene informacije, metalne veze (topljivi osigurači) pregorevaju ili ne, i na taj način se pamti informacija. Pregorevanje je nereverzibilan proces. Ako je potrebno promeniti podatke (sadržaj neke lokacije) tada je neophodno programirati novi čip, jer stari više nije upotrebljiv. PROM-ovi tipa "topljivi osigurač" kao memorijski čipovi nemaju veliki kapacitet i koriste se uglavnom kod logičkih aplikacija (tabela za preslikavanje ili dekodiranje), a ne kao medijum za smeštaj programa ili podataka.
- Drugi tip PROM-ova koji se danas veoma mnogo koristi se UV-EPROM-ovi. UV-EPROM se briše kada su duže vreme (10 minuta) izloženi zračenju ultraljubičaste svetlosti. Na gornjoj strani čipa postoji transparentni prozor preko koga se omogućava prodor svetlosti za brisanje ka stvarnoj površini memorijskog čipa. EPROM se može brisati/programirati više puta, a programiranje se izvodi posebnim uređajem koji se zove EPROM programator. Kod većih količina cena po komadu EPROM-a je veća nego kod maskiranog ROM-a. Osnovna prednost je ta što se može programirati i brisati veći broj puta. Ova činjenica ukazuje da je EPROM veoma

pogodan za izradu prototipova u toku eksperimentalne faze (dok se konačni program ne razvije) i za realizaciju sistema koji se proizvode u malim serijama.

- c) Treći tip PROM-a je EEPROM (*Electrical Erasable Programmable Read Only Memory*). EEPROM je sličan UV-EPROM-u sa izuzetkom što se on može programirati i brisati električnim putem. Drugi naziv za EEPROM je EAPROM (*Electrical Alterable Read Only Memory*).

3.5.2. Statički i dinamički RAM

U RAM memorijskom prostoru računarskog sistema se čuva informacija koja se menja u toku normalnog rada sistema. Postoje dva tipa RAM-ova: statički i dinamički. U oba slučaja se gubi smeštena informacija kada se napajanje isključi. Razlika između statičkih i dinamičkih RAM-ova je u načinu pamćenja podataka. Kod statičkih RAM-ova postoji flip-flop za svaku lokaciju u koju se smešta podatak, a informacija ostaje zapamćena sve dok se ne promeni stanje flip-flopa (tj. dok se ne upiše nova informacija) ili ne isključi napajanje čipa. Kod dinamičkih RAM-ova se mehanizam pamćenja zasniva na jednom tranzistoru i kondenzatoru za svaki bit. Informacija se upisuje kao nula ili pražnjenjem kondenzatora između *gejta* i *sorsa* tranzistora. Dinamički RAM jeftiniji od statičkog ali zbog pražnjenja kondenzatora mora da se vrši *osvežavanje* sadržaja (dopunjavanje kondenzatora). U tom cilju je potrebno ugraditi dodatnu logiku koja se zove *logika za osvežavanje memorije*. Zadatak ove logike je da u pravom trenutku vrši dopunjavanje kondenzatora. U suštini se ovaj postupak izvodi automatski (obično nezavisno od rada CPU-a), ali je za realizaciju ove logike potrebno predvideti prostor na memorijskim pločama.

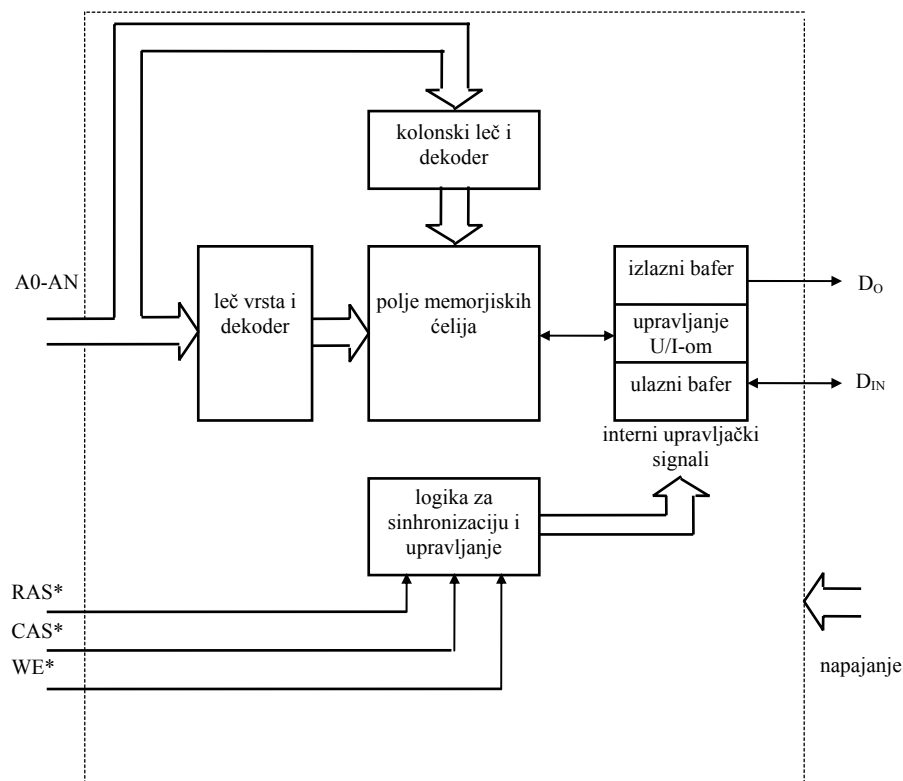
Tip RAM-a novije proizvodnje je NOVRAM (*Nonvolatile Random Access Memory*). NOVRAM predstavlja veoma brzi statički RAM kod koga je, kao rezerva, na istom čipu predviđeno identično EEPROM polje. Statički RAM se koristi za normalne operacije čitanja i upisa, a njegov sadržaj se može pročitati iz EEPROM-a kada sistem postane ponovo operativan.

3.5.3. Dinamički RAM

Dva glavna razloga popularnosti DRAM-ova su:

- veliki kapacitet, i
- relativno mala potrošnja.

DRAM-ovi su obično organizovani za smeštaj jednog bita po lokaciji, pa su poznati i kao *1 organizacija. Tipični primeri su 64k*1 i 256k*1 organizacija. Interna organizacija ovakvih tipova DRAM-ova je prikazana na slici 3.8.



Sl. 3.8. Tipična organizacija DRAM-a.

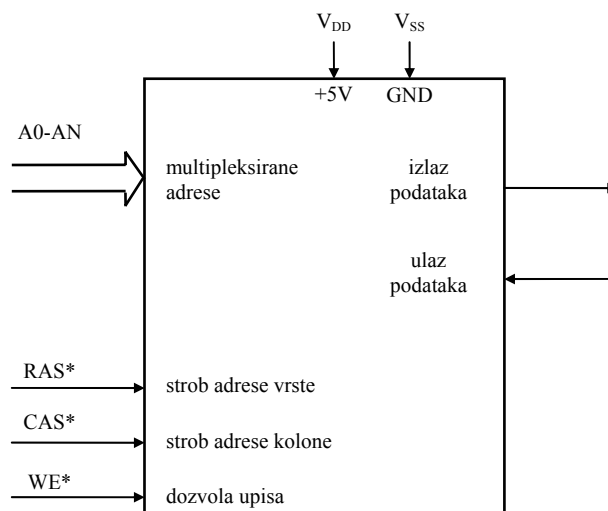
Glavni deo čipa je polje memorijskih ćelija u kojima se smeštaju nule i jedinice. Svaku ćeliju čine jedan ili više tranzistora i kondenzator male kapacitivnosti. Ćelije se adresiraju pomoću adresa vrsta i adresa kolona. Ove dve adrese se mogu zamisliti kao koordinate ćelija. Sa ciljem da se smanji broj spoljnih interfejs linija, ove dve adresne komponente se multipleksiraju na istim ulaznim linijama A0-AN. Prvo se na adresne linije postavlja adresa vrste, a zatim aktivira signal za njihovo pamćenje RAS* (*Row Address Strobe*). DRAM se odaziva pamćenjem prisutnih adresa vrste u interni leč vrste. Nakon toga se adresa vrste deaktivira, a na linije A0-AN postavlja adresa kolone. Sledi aktiviranje signala CAS* (*Column Address Strobe*). DRAM se odaziva pamćenjem adresa kolone u interni leč kolone. Shodno raspoloživoj informaciji (ukupnoj adresi) locira se adresa memorijske ćelije na osnovu dekodiranja zapamćenih adresa vrste i adresa kolone. Stanje na liniji WE* (*Write Enable*) ukazuje na to da li se pristup odnosi na operaciju čitanja ili operaciju upisa. D_{IN} i D_O su ulazna i izlazna linija podataka DRAM-a, respektivno.

Spoljne interfejs linije standardnog DRAM-a organizacije *1 su prikazane na slici 3.9.

U stanju pripravnosti (*standby*) snaga disipacije DRAM-a je deset ili više puta manja u odnosu na stanje kada je aktivan. Ovo u značajnoj meri smanjuje potrošnju.

Broj DRAM-ova koje je potrebno ugraditi kod implementacije memorije računara zavisi od:

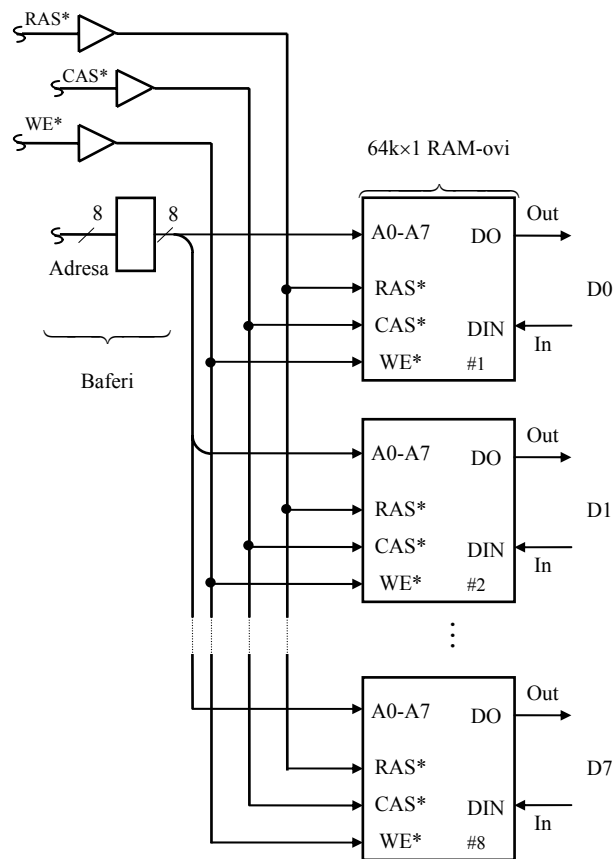
- obima spoljne magistrale podataka CPU-a,
- željenog kapaciteta memorije, i
- kapaciteta DRAM-a i njegove organizacije.



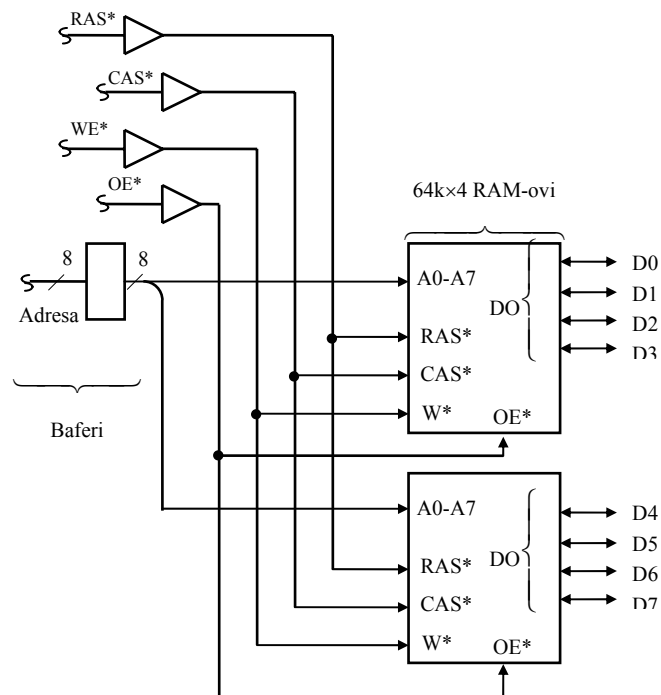
Sl. 3.9. Spoljne interfejs linije standardnog DRAM-a organizacije *1.

Implementacija 64kB memorije koja koristi osam 64k*1 DRAM-ova prikazana je na slici 3.10.

DRAM čipovi se mogu organizovati za čuvanje i više od jednog bita po lokaciji, kao što je to slučaj kod organizacija 16k*4, 8k*8 itd. Na primer, 256kb DRAM čipovi su dostupni u organizacijama 64k*4 i 32k*8. Ove alternativne organizacije u odnosu na organizaciju *1 često se zovu "wide-word" organizacije, a pogodne su samo za projektovanje memorija malih kapaciteta. Kod 8-bitnih mikror računarskih sistema potrebna su samo dva 16k*4 DRAM-a da bi se implementirala memorija kapaciteta 16kB (slika 3.11). Prvenstveno zbog manjeg broja izvoda po čipu, a saglasno tome i manjeg prostora koji zauzimaju na ploči, DRAM memorije većeg kapaciteta, "wide-word" organizacije DRAM-a, se izrađuju u organizaciji *1.



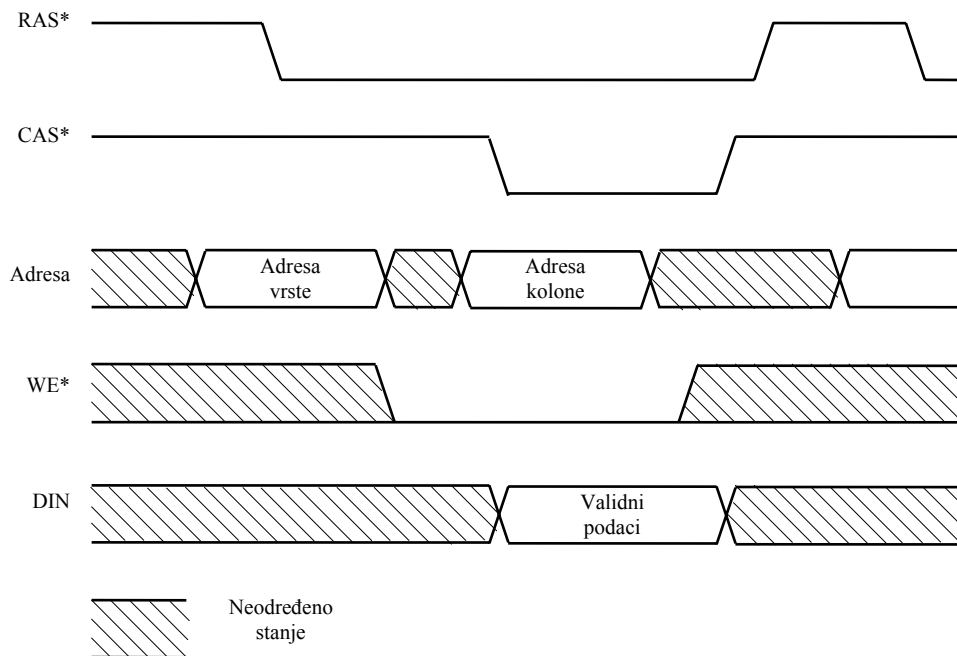
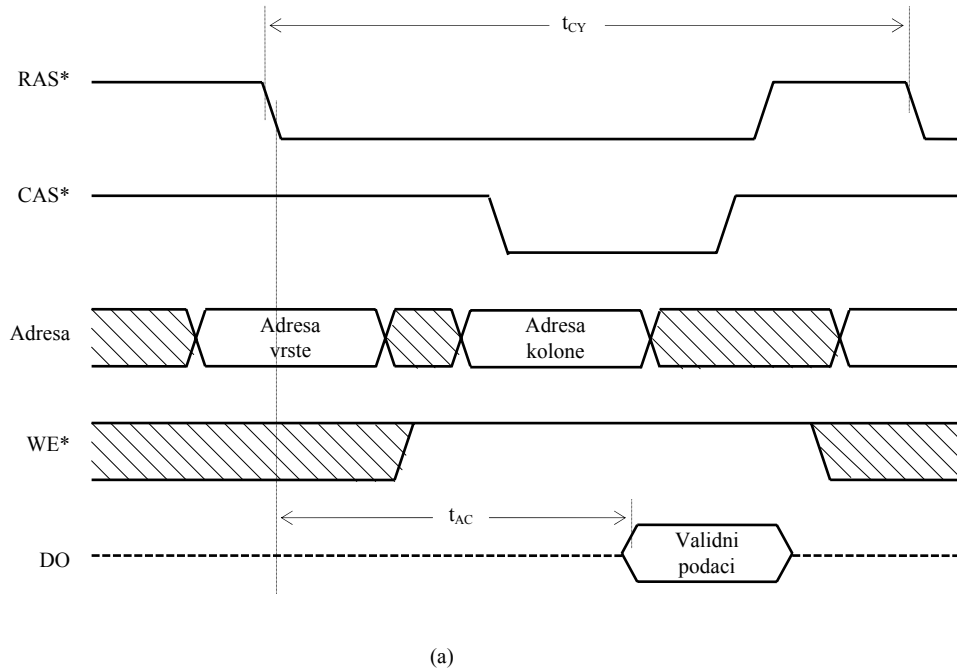
Sl. 3.10. Implementacija memorije kapaciteta 64kB pomoću osam 64k*1 DRAM čipova.



Sl. 3.11. Implementacija 16kB memorije pomoću dva 16k*4 DRAM-a.

3.5.4. Sinhronizacija rada DRAM-a

Na slici 3.12 je prikazan vremenski odnos između spoljnih interfejs signala DRAM-a važećih za operaciju čitanje i upis. Pre aktiviranja RAS*-a adresa vrste mora biti stabilna i mora da zadovoljava zahteve u pogledu postavljanja (*set-up requirements*) za dati čip. Adresa vrste mora biti važeća na ulaznim linijama, kada RAS* postane aktivan tokom perioda zadržavanja važnosti adrese (*address hold time*). Vreme trajanja aktivnog RAS* signala, kada je RAS* na nisko, je takođe ograničeno. Nakon toga RAS* prelazi na visoko tokom određenog vremenskog intervala sa ciljem da se u DRAM-u izvrše neke interne operacije koje se odnose na prebacivanje kapacitivnog tovara. Zbog minimalnih vremena kada je RAS* na nisko i visoko definiše ciklus čitanja, t_{CY} , DRAM-a.



Sl. 3.12.

Slična ograničenja u odnosu na vremena postavljanja i zadržavanja, kao i ograničenja koja se odnose na trajanja impulsa važe i za impuls CAS*. Vreme pristupa, t_{AC} , definiše se relativno u odnosu na prednju ivicu RAS* signala.

Sekvenciranje signala je slično i kod memorijskog ciklusa upis. U ovom slučaju je aktivan signal WE* što ukazuje da se izvršava operacija upis.

Drugi tip ciklusa je čitanje-modifikacija-upis. U toku ovog ciklusa RAS* i CAS* ostaju aktivni sve dok se modifikovani podatak ponovo ne upiše u memoriju.

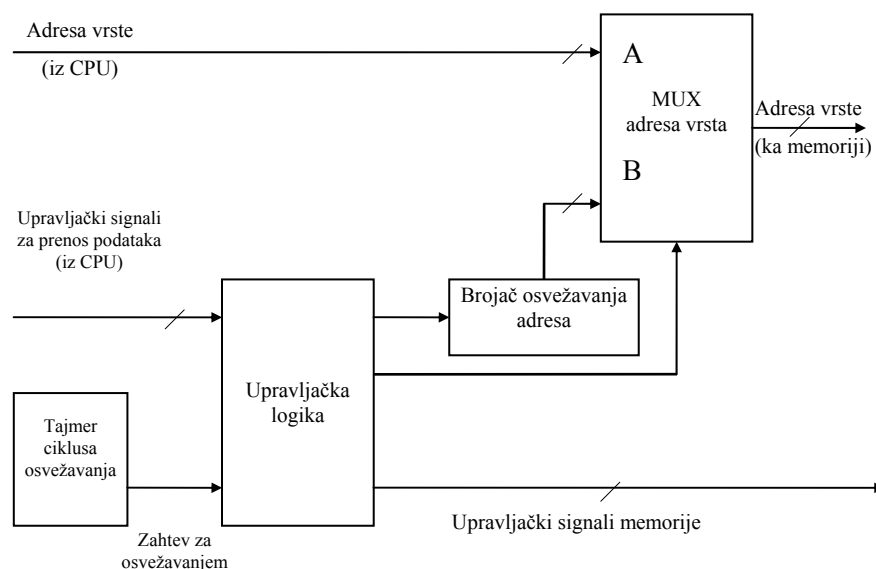
3.5.5. Osvežavanje

Opterećenje koje je akumulirano na malom kondenzatoru DRAM ćelije se brzo smanjuje zbog struje curenja. Kao rezultat toga napon na krajevima kondenzatora se smanjuje do vrednosti koja više ne odražava realno stanje zapamćene informacije. Da bi se izbegao gubitak podataka neophodno je da se:

- pročita vrednost napona svake ćelije,
- pojača,
- kondenzator ponovo napuni na početnu vrednost.

Ovaj proces mora periodično da se ponavlja i zove se *osvežavanje*. DRAM-ovi u toku jedne operacije obezbeđuju osvežavanje svih ćelija koje pripadaju jednoj vrsti. Obično DRAM-ovi zahtevaju osvežavanje po 128 vrsta svake 2ms, ili po 256 vrsta svake 4ms.

Da bi se logika za osvežavanje mogla realizovati neophodno je ugraditi tajmer koji periodično generiše zahteve za osvežavanjem i brojač na osnovu čijeg stanja se određuje redosled adresirane vrste koja se osvežava. Blok šema logike za osvežavanje DRAM-ova je prikazana na slici 3.13.



Sl. 3.13. Logika za osvežavanja DRAM-a.

Kada upravljačka logika primi zahtev za osvežavanjem, ona preduzima sledeće aktivnosti:

- Postavlja upravljački signal select u takvo stanje da MUX na svom izlazu usmerava ulaze B.
- Generiše memorijske upravljačke signale koji se sekvenciraju u saglasnosti sa zahtevima ciklusa za osvežavanje.
- Nakon završetka ciklusa za osvežavanje upravljačka logika inkrementira brojač. Novi sadržaj brojača ukazuje na adresu vrste koja se naredna osvežava.

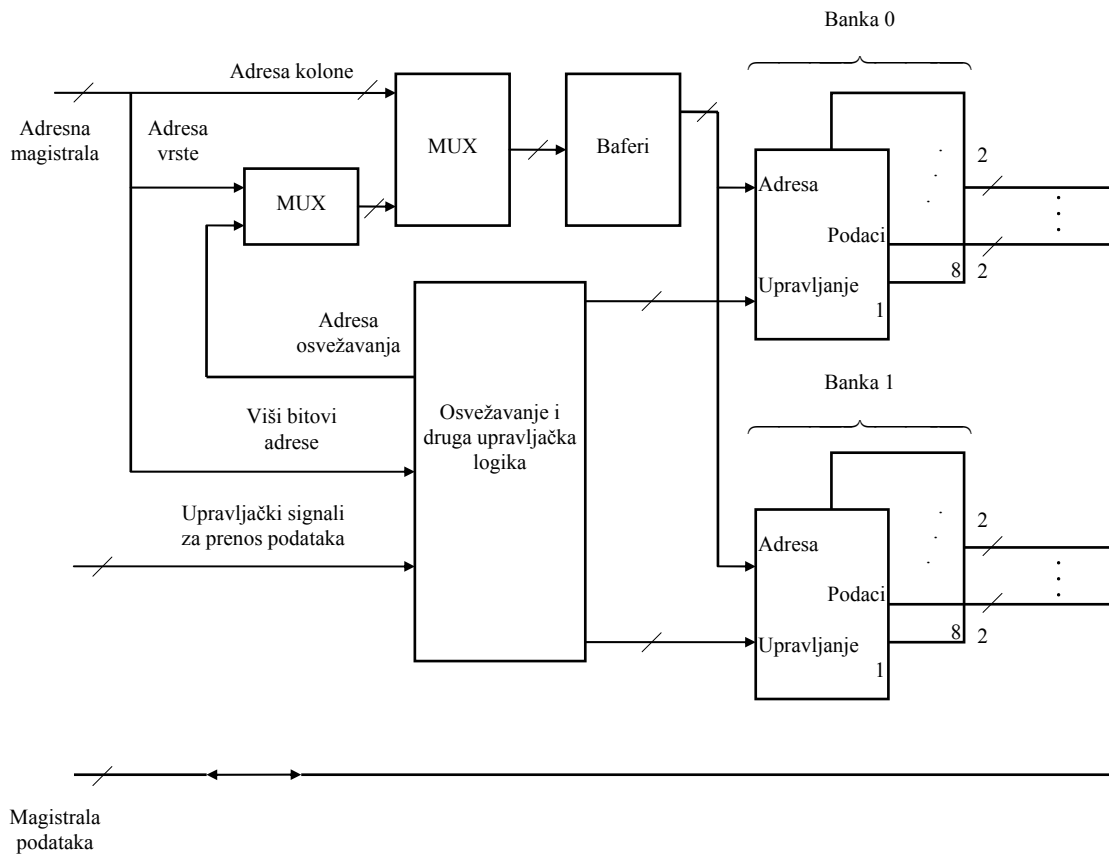
Kada je CPU izdao zahtev za memorijskim ciklusom, upravljačka logika preko MUX-a usmerava odgovarajuću adresu vrste prema memoriji.

Interesantna situacija se javlja kada upravljačka logika primi istovremeno dva zahteva. Jedan zahtev se odnosi na standardni pristup memoriji od strane CPU-a, a drugi na osvežavanje. Jasno je pri ovome da jedan od zahteva mora da sačeka na svoje izvršenje. Projektovanje kola koje će obavljati arbitražnu funkciju nije trivijalan zadatak. Ako ovo kolo nije korektno projektovano mogu se manifestovati nepredvidljiva ponašanja koja se veoma teško mogu logički interpretirati. Drugi aspekt na koji treba obratiti pažnju kod projektovanja ovih kola se ogleda u činjenici da ona ne smeju tokom arbitraže da unose značajna kašnjenja.

3.5.6. Memorijska interfejs kola

Na slici 3.14 su prikazani ostali blokovi koji se ugrađuju u sistem a neophodni su radi realizacije sprege između memorije i sistemske magistrale. MUX2 se koristi radi multipleksiranja adresa vrste i kolona. Naravno, pre distribucije adresnih signala ka DRAM-ovima adresni signali se baferuju.

Kao što se vidi sa slike 3.14, adresni signali se distribuiraju na 16 DRAM-ova. Postoje dve banke od kojih se svaka sastoji od po osam DRAM-ova. Linije D_{IN} i D_O se povezuje na magistralu podataka. Adrese manje težine dele se na dva dela. Jedan deo se koristi kao adresa vrste a drugi kao adresa kolone. Viši deo adrese se koristi za selekciju banaka. Dekodiranje se izvodi upravljačkom logikom koja generiše upravljačke signale za izbor memorijskih banaka.



Sl. 3.14. Struktura kompletnog memorijskog sistema dinamičkog RAM-a