

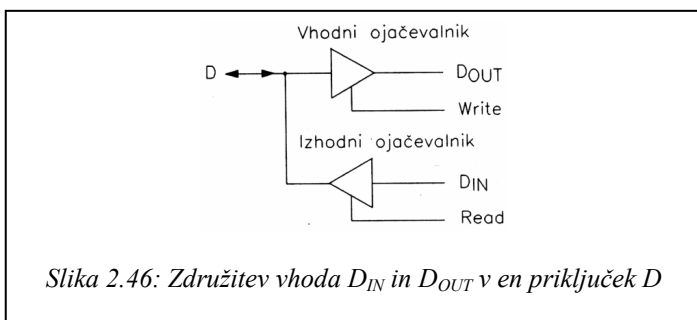
2.8 POMNILNIŠKA VEZJA

2.8.1 SPLOŠNO O POMNILNIKI

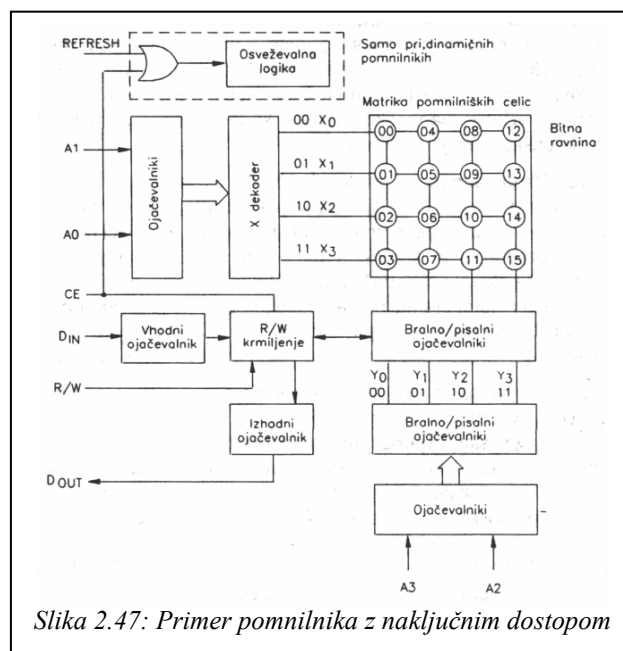
Mikroprocesorska vezja potrebujejo za hranjenje programov, instrukcijskih kod ali podatkov tudi integrirana vezja s katerimi lahko hitro komunicirajo. Pomnilnik je videti kot skladišče iz katerega mikroprocesor jemlje ali daje podatke, vendar govorimo raje o branju oz. vpisovanju podatkov. Pomnilnik se ne zaveda pomena podatkov in tako npr. ne razlikuje ukazov od operandov, prav tako pa ne raznih tipov operandov. Seveda lahko uporabnik s pomočjo napisanega programa določi na katerih lokacijah pomnilniške vsebine se bodo nahajali podatki, kje bodo instrukcijske kode, delovni program, itd... Običajno ima mikroprocesorski sistem več tipov pomnilnika, ki so namenjeni za specifični namen. Tako je npr. delovni program največkrat zapisan v pomnilniškem vezju tipa ROM ali EPROM, tekoči podatki se shranjujejo v pomnilnik tipa SRAM ali DRAM, spremenljivi parametri (npr. nastavitve režima delovanja) pa so najpogosteje zapisani v pomnilniku tipa EEPROM. Vsi ti pomnilniki so narejeni tako, da jih lahko med sabo povezujemo in poljubno razširjamo kapaciteto pomnilnika. Nekateri mikrokontrolerji imajo že v istem ohišju poleg CPU nekaj pomnilnika. Vsa ta vezja spadajo med notranje pomnilnike in jih bomo podrobneje spoznali, seveda pa pomnilniško kapaciteto lahko povečamo tudi z zunanjimi (npr. diskovnimi enotami), ki pa jih ne bomo obravnavali, ker presegajo okvir predmeta.

2.8.2 ZGRADBA POMNILNIKA Z NAKLJUČNIM DOSTOPOM

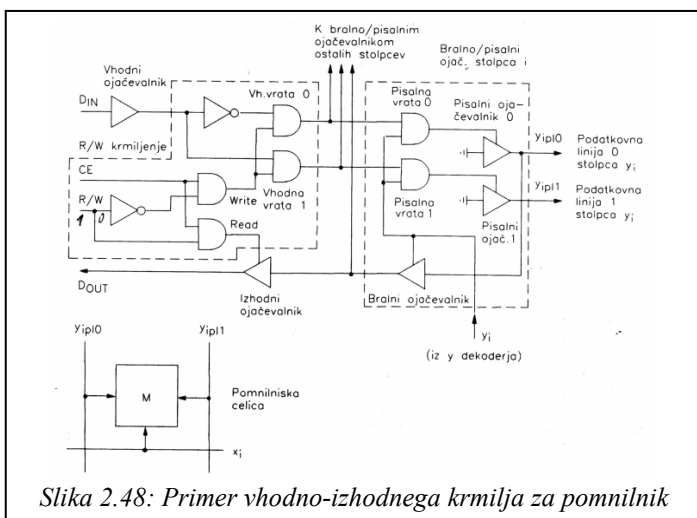
Na sliki je podan primer zgradbe pomnilniškega vezja z naključnim dostopom. Pomnilnik vsebuje večino funkcij, ki jih danes vsebujejo pomnilniška integrirana vezja. Čeprav imajo pomnilniška integrirana vezja mnogo več kot 16 pomnilniških celic, je razširitev pojmov na pomnilnike večje kapacitete preprosta.



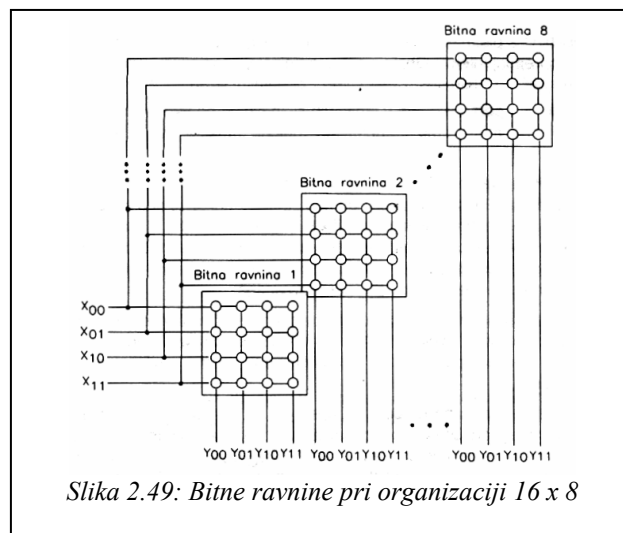
Slika 2.46: Zdržitev vhoda D_{IN} in D_{OUT} v en priključek D



Slika 2.47: Primer pomnilnika z naključnim dostopom

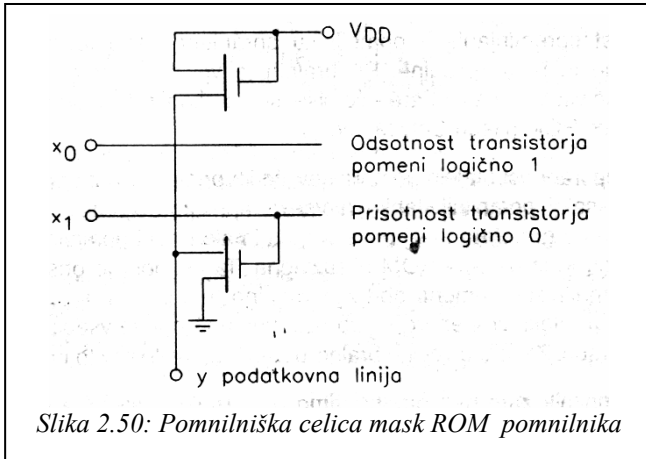


Slika 2.48: Primer vhodno-izhodnega krmilja za pomnilnik

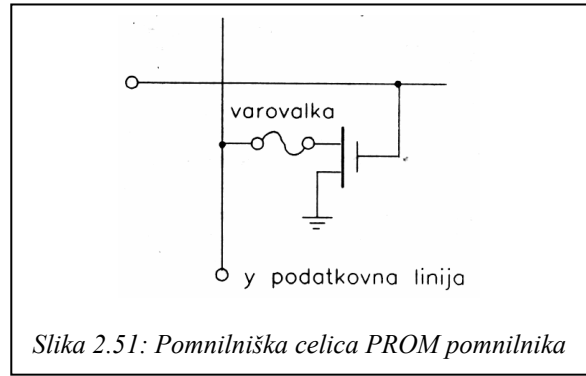


Slika 2.49: Bitne ravnine pri organizaciji 16 x 8

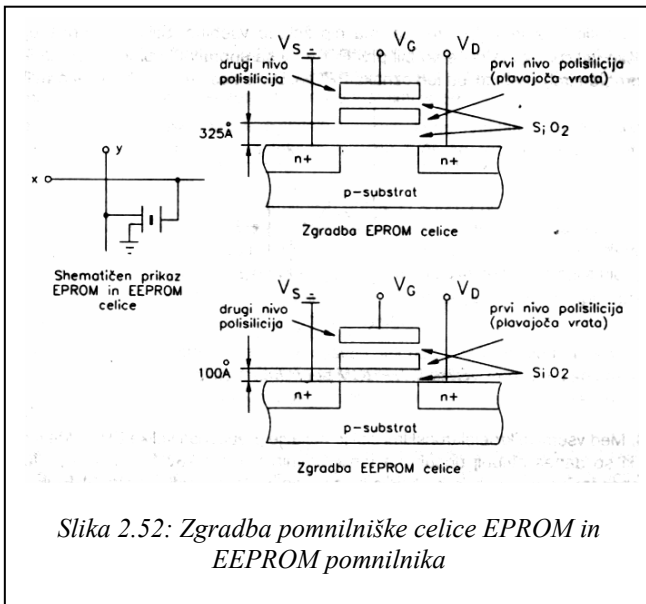
2.8.3 BRALNI POMNILNIKI (ROM, PROM, MASK ROM, EPROM IN EEPROM)



Slika 2.50: Pomnilniška celica mask ROM pomnilnika

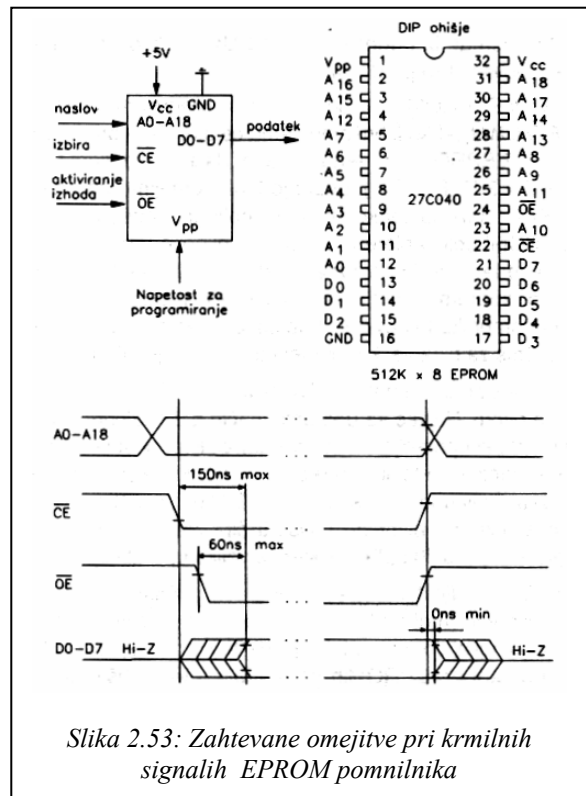


Slika 2.51: Pomnilniška celica PROM pomnilnika



Slika 2.52: Zgradba pomnilniške celice EPROM in EEPROM pomnilnika

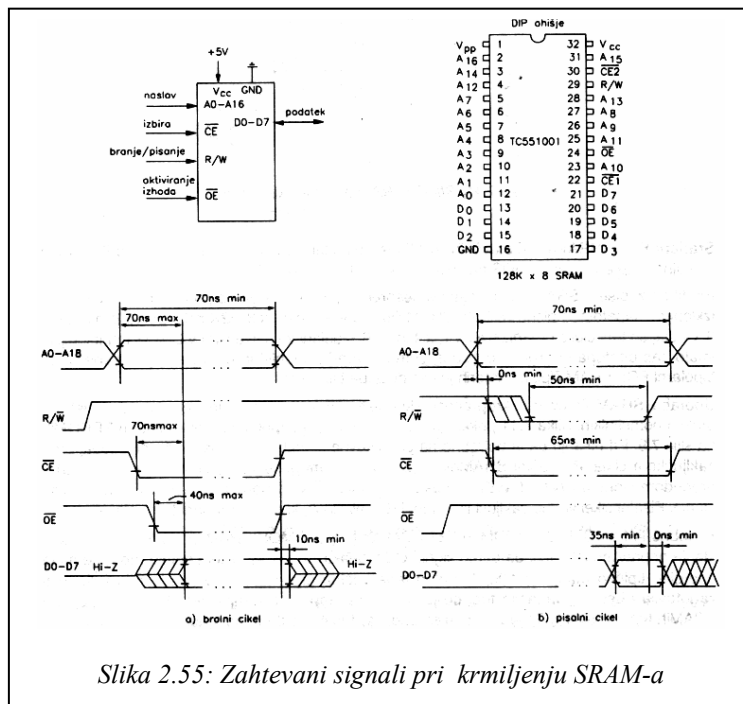
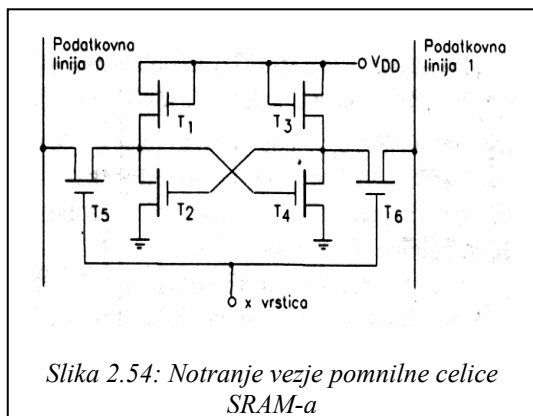
Pripadajoča pojasnitev! (študent)



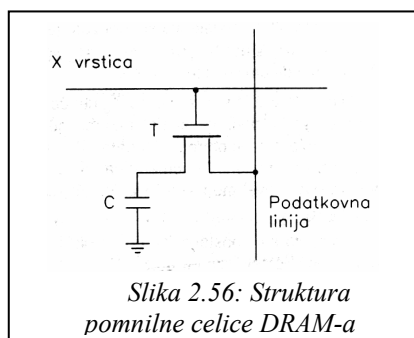
Slika 2.53: Zahtevane omejitve pri krmilnih signalih EPROM pomnilnika

2.8.4 Bralno-pisalni pomnilniki tipa RAM (statični RAM, dinamični RAM)

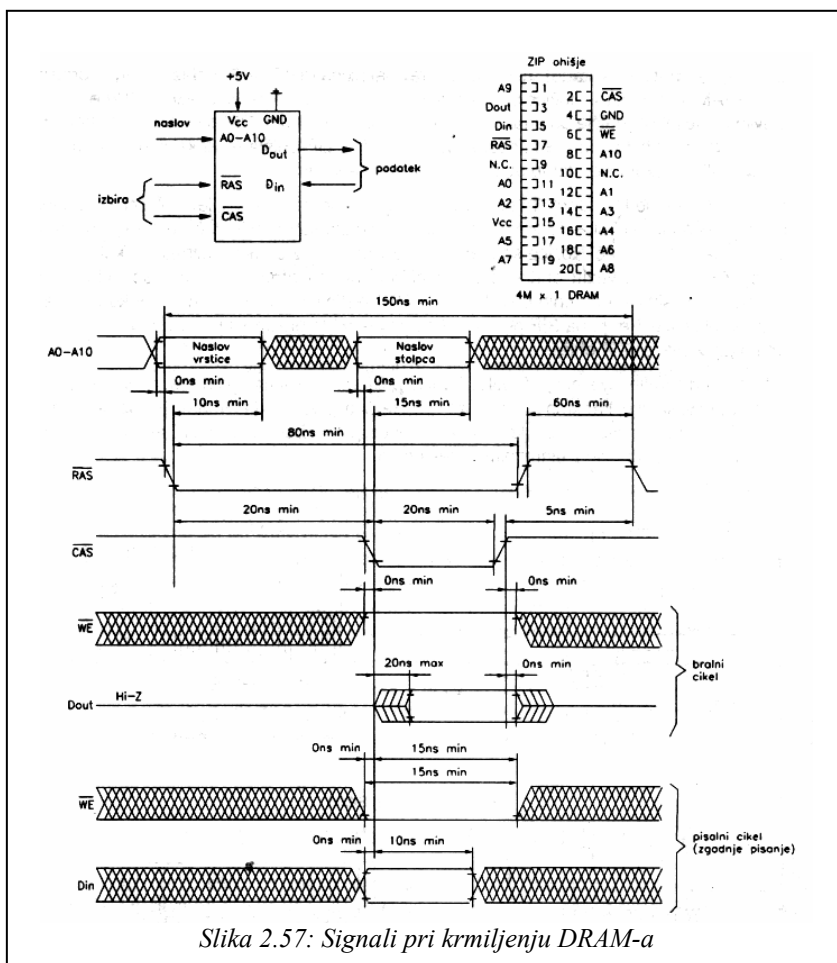
Statični RAM- SRAM



Dinamični RAM - DRAM



Opis! (študent)



2.8.4.1 Načini osveževanja dinamičnih RAM-ov.

Za DRAM pomnilnike je značilno, da za ohranjanje vpisane vsebine potrebujejo neprestano osveževanje. V praksi večinoma velja, da se spominska vsebina na neki lokaciji DRAM-a ohranja, če je zagotovljen postopek osveževanja v času osveževalne periode (15,6ms). To pomeni, da je potrebno najkasneje v času osveževalne periode opraviti osveževanje vseh spominskih celic v DRAM-u. Glede na to zahtevo lahko poteka osveževanje na več načinov, ki so odvisni od tipa in kapacitete DRAM-a, ter od strukture vezja in pripadajočega programa.

- **Osveževanje z bralnimi cikli**

Osveževanje z bralnimi cikli se izvaja s pomočjo ustreznega napisanega programa. Pri pisanju programa mora programer dodati v program tudi instrukcije, ki bodo zagotovile, da se bo v času vsake osveževalne periode vsaj enkrat prečitala vsaka vrstica. Za osveževanje celotne vrstice je že tudi dovolj, če se prečita samo en podatek v vrstici.

- **Osveževanje s periodičnimi vpisovalnimi cikli**

V primerih, kjer se podatki v DRAM nenehno vpisujejo in na ta način novi »prekrivajo« stare lahko izvedemo osveževanje z vpisovalnimi cikli. Pri tem načinu je potrebno zagotoviti, da se v času vsake osveževalne periode vpišejo v vse spominske lokacije novi podatki. Ta način se uporablja npr. pri digitalnih osciloskopih, EKG in podobnih medicinskih aparatih, digitalnih monitorjih in podobnih napravah.

- **Osveževanje s periodičnim naslavljanjem vrstic (RAS+ naslov vrstice)**

Pri DRAM-ih je za osvežitev posamezne vrstice večinoma že dovolj, če se vrstica samo naslovi, pri tem pa branje ali vpisovanje podatkov sploh ni potrebno. Vendar ta način zahteva dodatno vezje, ki mimo delovanja mikroprocesorja v času osveževalne periode, postopoma naslovi vse spominske vrstice in formira pripadajoči RAS signal. Takšno vezje imenujemo DMA krmilnik (Direct Memory Access) in ima svoj interni števec naslovov spominskih lokacij. V času osveževanja konkretne vrstice mikroprocesor seveda ne more dostopati do te spominske lokacije, zato DMA krmilnik dostop zavrne tako, da postavi kontrolni signal WAIT v aktivno stanje.

- **Osveževanje z naslavljanjem stolpcev pred vrsticami (CAS pred RAS)**

DRAM pomnilniki velikih kapacitet imajo že vgrajen tudi interni števec za naslavljanje vrstic tako, da je za vezje in programerja procedura poenostavljena. Za DRAM je značilno, da se mora v primeru branja ali vpisovanja pojaviti najprej RAS (Row Adress Select) signal z naslovom vrstice in šele zatem CAS (Column Adress Select) signal, ki mu sledi naslov stolpca. V primeru obratnega vrstnega reda pa notranje vezje »razume«, da gre le za postopek osveževanja. V tem primeru pride CAS signal pred RAS signalom, s čimer se aktivira notranji števec naslovov, ki v osveževalni periodi zagotovi postopno osveževanje vrstic.