
Organizacija memorije

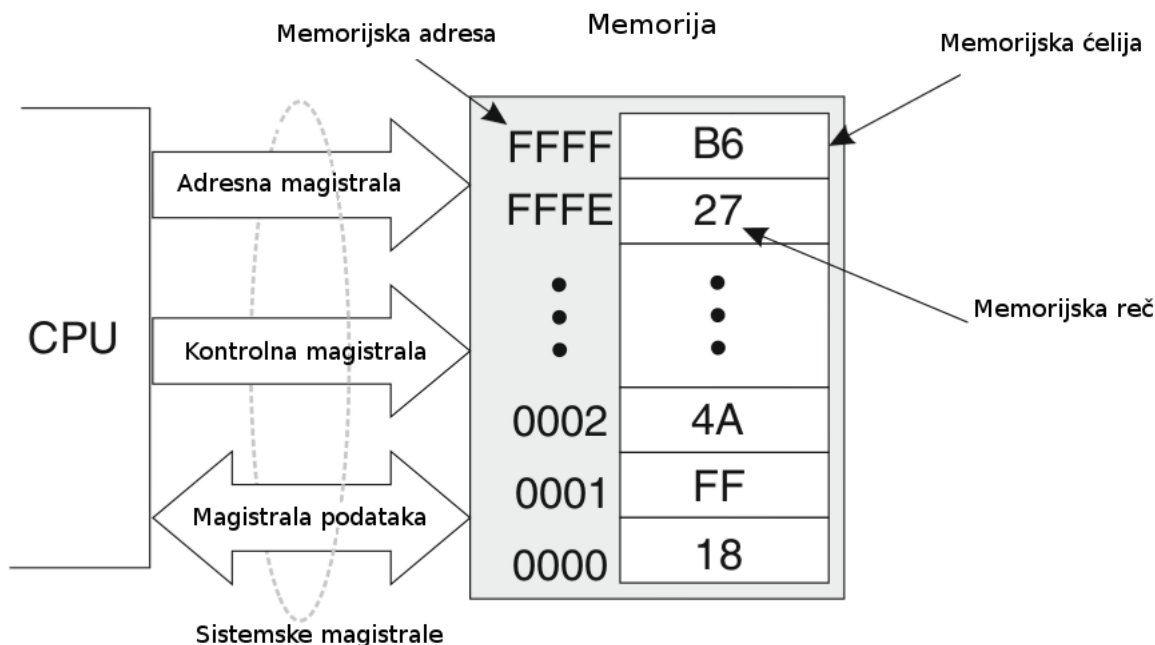
Memorije čine jedan od tri osnovna bloka mikroračunarskog sistema. Veza memorije sa centralnom procesorskom jedinicom je ostvarena putem adresne i magistrale podataka. U okviru ovog predavanja ćemo malo više pažnje posvetiti strukturi memorija kao i sprezi memorije sa centralnom procesorskom jedinicom.

1 Uvod - organizacija memorije

Memorijski podsistem je zadužen za smeštanje instrukcija i podataka. Memorija se sastoji od velikog broja hardverskih komponenti od kojih svaka može da skladišti jedan bit informacije. Ovi biti su organizovani u n -bitne reči, koje funkcionišu kao registar, a obično se nazivaju *memorijska ćelija* ili *memorijska lokacija*. Sadržaj memorijske ćelije je osnovna jedinica informacije i naziva se *memorijska reč*. Pored toga, svaka memorijska lokacija je identifikovana na jedinstven način, korišćenjem memorijske adrese, koja se koristi od strane CPU u cilju ili čitanja ili pisanja memorijske reči koja je skladištena na toj lokaciji. Generalno, za memorijsku jedinicu koja se sastoji od m ćelija, pri čemu je svaka veličine n bita, kaže se da je $m \times n$ memorija. Ukoliko je $n = 1$ i $n = 8$, ova informacija se dopunjuje sa b ili B , respektivno. U skladu sa tim, ukoliko govorimo o 1Mb (jedan Mega bit) ili 1MB (Mega Bajt) memorijama, govorimo o $1M \times 1$ ili $1M \times 8$ memorijama.

Uobičajeno je da su adrese sekvencijelni brojevi kao što je prikazano na slici 1. Ipak, za neke mikrokontrolere, određene adrese možda nisu stvarno prisutne. Primer sa slike pokazuje memorijski modul koji sadrži 64k ćelija, pri čemu svaka sadrži 8 bita (bajt), što ukupno čini 64kB memoriju. Na slici, na primer, ćelija na adresi 0FFFEh sadrži vrednost 27h. Centralna procesorska jedinica koristi adresnu magistralu kako bi selektovala jednu ćeliju koju želi da koristi. Sa druge strane, sama interakcija sa ćelijom je ostvarena korišćenjem magistrale podataka. Tokom operacije upisa, CPU modifikuje informaciju koja je sadržana u okviru ćelije, dok u sklopu operacije čitanja CPU preuzima njen sadržaj bez menjanje sadržaja. Pri tome, CPU koristi kontrolnu magistralu i njene signale kako bi odredio koji tip operacije želi da izvrši nad datom memorijom, kao i smer toka podataka na magistrali podataka (rečeno je ranije da je magistrala podataka bi-direkciona).

Važno je spomenuti još jedan detalj u vezi sa pristupom memoriji. CPU pristupa samo jednoj memorijskoj lokaciji u datom trenutku (ili jednoj perifernoj jedinici, kao što ćemo videti kasnije). Samim tim, svaka lokacija je jednoznačno određena svojom adresom. Skup linija (fizičkih) koju procesor koristi za postavljanje željene adrese naziva se adresna magistrala. Ove



Slika 1: Organizacija memorije

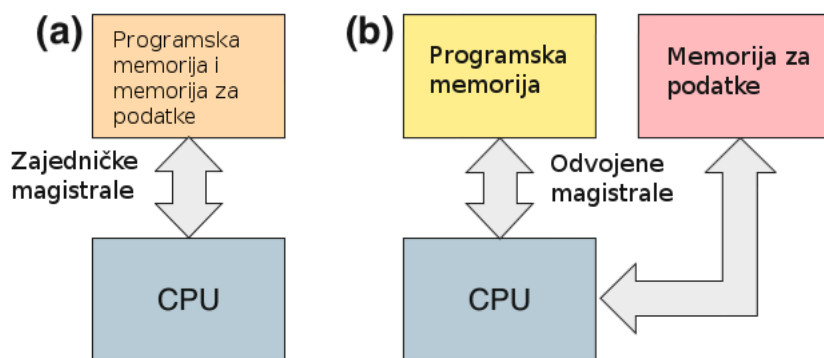
linije su tipično izlazne linije za CPU i ulazne za memoriju (ili periferijsku jedinicu) i kaže se da su uni-direkzione. Širina adresne magistrala određuje veličinu najvećeg memorijskog prostora koji CPU može da adresira. Ukoliko je adresna magistrala širine m bita, CPU može da adresira maksimalno 2^m različitih memorijskih lokacija, koje su uglavnom označene svojim heksadecimalnim vrednostima. Na primer, u slučaju 16-bitne magistrala, CPU može da adresira $2^{16}=64k$ lokacija tj. 0x0000, 0x0001, ..., 0xFFFF.

2 Programska memorija i memorija za podatke

Sa stanovišta sadržaja memorije, u okviru mikroračunarskog sistema mogu se naći dva različita tipa memorije: Programska memorija i memorija za podatke.

Programska memorija, kao što se naslućuje iz naziva, odnosi se na deo memorije koji je zadužen za smeštanje programa u formi koju CPU može direktno da koristi za njegovo izvršavanje. Program je, samim tim, sekvenca instrukcija koje opisuju funkcionalnost datog mikroračunarskog sistema.

Ako govorimo o mikrokontrolerima, programi su uglavnom smešteni na fiksnim memorijskim lokacijama, kako bi CPU mogao neometano da ga izvršava. To znači da u slučaju kada sistem ostane bez napajanja, nakon što se napajanje opet dovede, program i dalje mora biti na svom mestu kako bi sistem funkcionisao normalno. Da bi ovo bilo moguće, programi se smeštaju u memorije koje ne gube sadržaj nakon nestanka napajanja. Kapacitet programske memorije u ovakvim sistemima je



Slika 2: Von Neumann (a) i Harvard arhitektura (b)

tipično reda nekoliko kilo-reči. U nekim slučajevima, mikrokontroler može da izvršava program iz RAM memorije, dok se u većini slučaja program direktno izvršava iz ROM memorije.

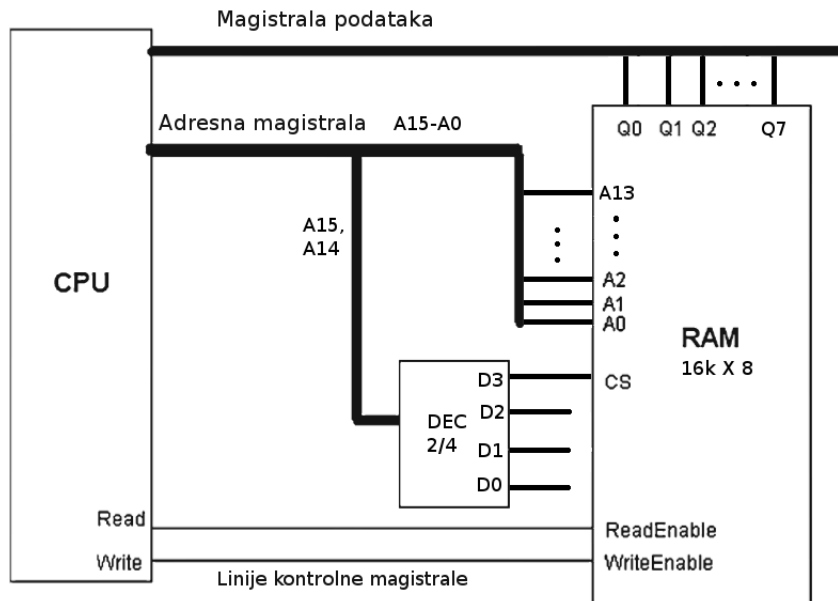
Memorija za podatke skladišti promenljive koje će se menjati tokom izvršavanja programa. Dakle, ovaj tip memorije mora omogućiti jednostavnu modifikaciju njenog sadržaja. U većini mikroracunarskih sistema, baziranih na mikrokontrolerima, ova memorija se nalazi u RAM bloku. Obzirom na činjenicu da je memorija za podatke predviđena za privremeno smeštanje podataka (potreba za njima postoji samo tokom izvršavanja programa), osobina da RAM memorija gubi svoj sadržaj sa nestankom napajanja ne predstavlja nikakav problem. Prosečna veličina RAM memorije koja je potrebna u nekom mikroracunarskom sistemu sa mikrokontrolerom je relativno mala (uglavnom samo nekoliko stotina reči).

Podaci koji su izuzetno značajni, i čiji se sadržaj ne sme izbugiti nakon nestanka napajanja, moraju biti skladišteni u ROM sekciji. Neki mikrokontroleri imaju mogućnost pristupa tim podacima direktno, iako ne mogu da im menjaju sadržaj tokom izvršavanja programa. U ostalim slučajevima, ovaj sadržaj se na početku kopira u RAM sekciju, nakon čega se kasnije može manipulirati tim podacima nesmetano.

Programska i memorija za podatke mogu da dele iste sistemske magistrale ili ne moraju, što zavisi isključivo od arhitekture mikrokontrolera. Sistemi koje karakteriše jedinstvena sistemska magistrala koju koiste kako programi tako i podaci, naziva se *Von Neumann*¹ arhitektura ili *Princeton* arhitektura. Alternativa ovakvoj organizaciji je *Harvard* arhitektura. Kod Harvard arhitekture su fizički odvojeni adresni prostori za programsku memoriju i memoriju za podatke. Samim tim, postoje fizički odvojene adresne i magistrale za podatke. Slika 2 pokazuje topološke razlike između Von Neumann i Harvard arhitekture. AVR mikrokontroleri koriste Harvard arhitekturu.

Širina IR i PC registara zavisi od arhitekture. U slučaju Von Neumann arhitekture, oni imaju istu širinu kao i ostali CPU registri koji mogu sadržati adrese, dok u slučaju Harvard arhitekture, ovi registri su potpuno nezavisni od ostalih registara, obzirom da magistrale ne moraju biti jednakih veličina (širina).

¹U samom začetku razvoja mikroracunarskih sistema, veliki doprinos ovoj oblasti pružio je profesor John Von Neumann sa univerziteta Princeton, po kome je ova arhitektura i dobila ime.



Slika 3: Primer povezivanja memorije sa centralnom procesorskom jedinicom

3 Primeri razmene podataka između CPU i memorije

Kako bismo ilustrovali mehanizam razmene podataka između CPU i memorije, posmatraćemo pojednostavljenu statičku RAM memoriju, kapaciteta 16k X 8, i CPU sa adresnim prostorom od 64kB. Slika 3 prikazuje koncept povezivanja CPU i ovakve RAM memorije.

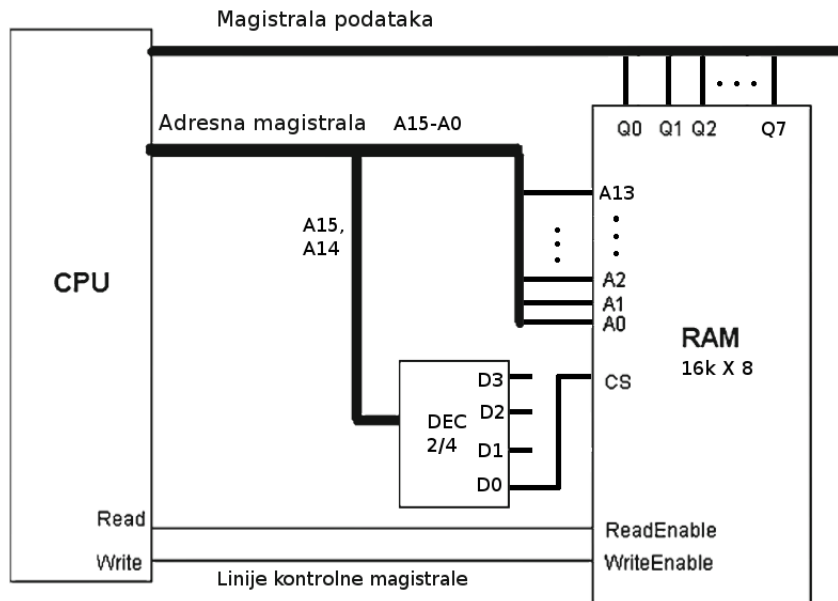
Ako posmatramo interfejs RAM memorije, on je podeljen u tri grupe:

1. ulazi/izlazi za podatke Q0, Q1, ... Q7 (obzirom da je širina memorijske reči 8 za memoriju kapaciteta 16k X 8)
2. (Interni) Adresni ulazi A0, A1, ... A13 koji se koriste kako bi seletovali jednu ćeliju unutar bloka. Ukupno postoji 14 adresnih ulaza jer sama memorija ima $16k = 16 * 1024 = 16384$ memorijskih lokacija, pri čemu se na svakoj lokaciji nalazi 1 bajt te je ukupan kapacitet memorije 16kB
3. selekcionni ulaz (Chip Select-CS) koji se koristi da aktivira/deaktivira RAM i kontrolni terminali (ReadEnable i WriteEnable) koji služe za upravljenje memorijom.

Što se tiče centralne procesorske jedinice (CPU na slici 3), ona ukupno sadrži 16 adresnih linija jer je

$$2^{16} = 65536 = 64 * 1024 = 64kB$$

Obzirom da je adresni prostor CPU 64kB, očigledno je da se u okviru njega može smestiti 4 bloka statičke RAM memorije kapaciteta 16k X 8. Da bi se to postiglo, neophodno je koristiti adresni dekodner koji će na adekvatan način rasporediti četiri RAM bloka u okviru raspoloživog adresnog prostora. U tu svrhu se koristi CS ulaz memorijskog bloka.



Slika 4: Povezivanje memorije u niži opseg adresnog prostora CPU

Kao što je rečeno, CS se koristi za aktivaciju bloka, u cilju postavljanja RAM bloka u pristupačno stanje kako bi CPU mogao da ga koristi. Sa druge strane, za aktiviran blok kod koga je vrednost ulaza CS = 1, ReadEnable i WriteEnable određuju operaciju koja će se izvršavati sa datom memorijom.

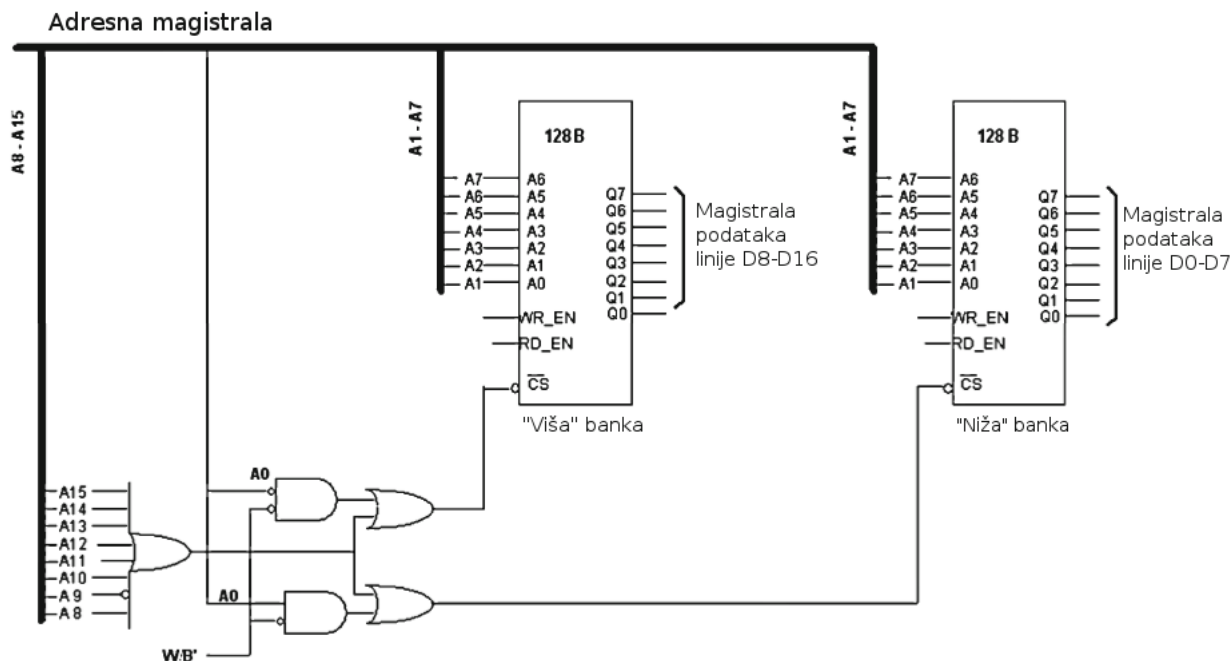
Linije magistrale podataka su spojene sa ulazno/izlaznim terminalima. Interno, unutar samog memorijskog bloka, ovi terminali su povezani korišćenjem tri-statičkih bafera. Ovi baferi omogućavaju kontrolu smera toka podataka (upis podataka u memoriju ili čitanje podataka iz memorije), ali takođe i da postave izlaze memorije u stanje visoke impedanse kako bi odspojili memoriju od magistrale u trenucima kada se magistralom podataka komunicira sa drugim memorijskim blokom ili nekom perifernom jedinicom (kada RAM nije aktiviran signalom CS). Upis i čitanje memorije su inicirani od strane CPU korišćenjem signala kontrolne magistrale. Adresna magistrala koja polazi od CPU se deli na dve grupe:

- skup linija koje su direktno povezane sa memorijom i njenim internim adresnim linijama i
- skup linija koje se koriste od strane dekodera kako bi aktivirao memorijski blok od interesa.

U primeru sa slike 3, adresne linije A0-A13 se koriste za povezivanje sa internom adresnom magistralom memorijskog bloka, dok se ostatak adresnih linija vodi na dekodera koji ima ulogu da aktivira RAM blok. Adresa memorijske lokacije je onda reč formirana od adresnih bita koji aktiviraju blok memorije, zajedno sa onima koji selektuju pojedinačne reči unutar same memorije.

Iz svega gore navedenog, može se zaključiti da je memorija smeštena u adresni prostor CPU u opseg adresa:

početna adresa : 0xC000



Slika 5: Primer povezivanja dva memorijska bloka

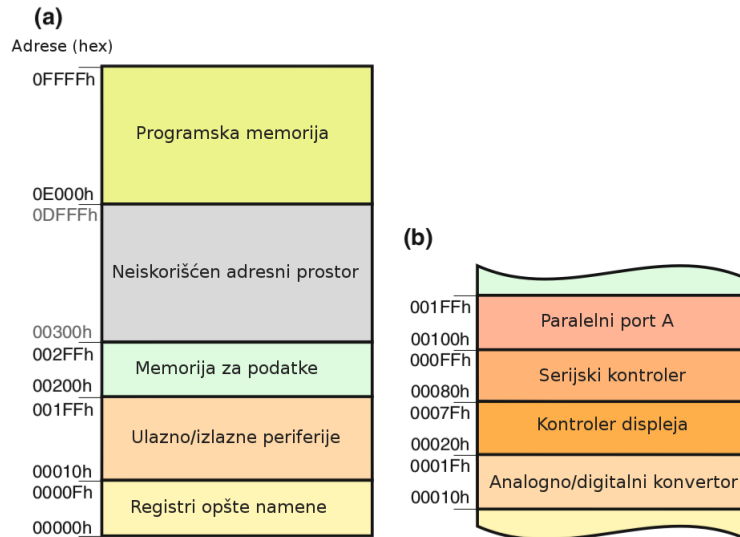
krajnja adresa : 0xFFFF

Ukoliko be se CPU i memorija povezali kao na slici 4 memorija bi bila smeštena u adresni prostor:

početna adresa : 0x0000

krajnja adresa : 0x3FFF

Na primeru prikazanom na slici 5, dva memorijska bloka kapaciteta 128B (bajta) koji su povezani na 16-bitnu magistralu podataka, sa 16-bitnom adresnom magistralom i kontrolnim signalom W/\overline{B} . Obzirom da svaki blok ima samo osam ulazno/izlaznih terminala, dva bloka mogu pokriti u potpunosti 16 bita magistralne podataka. Kontrolni signal W/\overline{B} kontroliše da li CPU čita/piše reč ($W/\overline{B} = 1$) ili bajt ($W/\overline{B} = 0$). Blokovi su aktivirani kada je logička "0" prisutna na \overline{CS} ulazima, koji su opet povezani na izlaze ILI kola koje se koristi za dekodovanje adresa. Dakle, ukoliko je izlaz ILI kola u stanju logičke "1", memorije su u isključenom stanju (neaktivne) i njihovi ulazi/izlazi za podatke su u stanju visoke impedanse. U skladu sa tim, potrebna je kombinacija adresnih linija $A15=A14=A13=A12=A11=A10=0$, $A9=1$ i $A8=0$, kako bi blokovi bili aktivni. Adresne linije A7-A1 aktiviraju interne adrese aktivnih RAM blokova. Na kraju, kada je signal $W/\overline{B} = 1$, tada su oba bloka aktivna, nezavisno od vrednosti A0 linije. Sa druge strane, ukoliko je $W/\overline{B}=0$, tada $A0=0$ aktivira desnu "nižu" memorijsku banku, dok $A0=1$ aktivira levu "višu" RAM banku. U skladu sa svim gore rečenim, adrese koje se koriste u ovom primeru je:



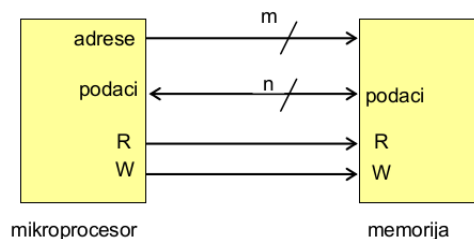
Slika 6: Memorijska mapa mikroračunarskog sistema zasnovanog na Von Neumann arhitekturi

- opseg: od 0000 0010 0000 0000 (binarno) do 0000 0010 1111 1111 (binarno), tj. od 0200h do 02FFh;
- niža banka: aktivirana kada je $A_0 = 0$, što znači da pokriva “parne” adrese u datom opsegu, tj. 0200h, 0202h, 0204h,...02FEh;
- viša banka: aktivirana kada je $A_0 = 1$, što znači da pokriva “neparne” adrese u datom opsegu, tj. 0201h, 0203h,...02FFh.

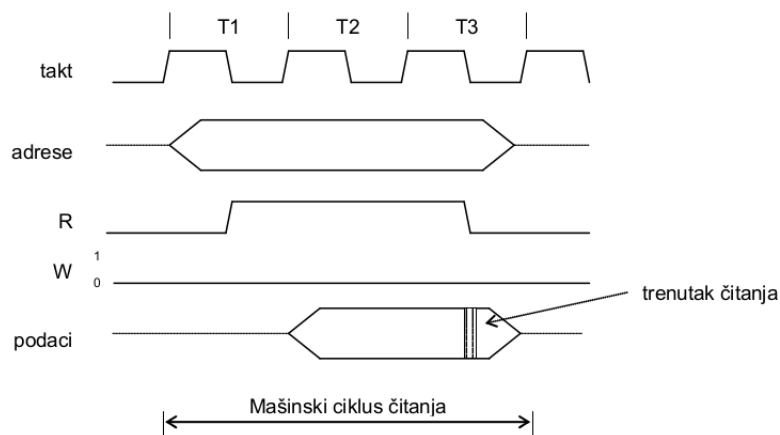
4 Memorijska mapa

Memorijska mapa predstavlja reprezentaciju iskorišćenosti adresnog prostora mikroprocesorskog sistema. Na primeru sa slike 6, prikazana je memorijska mapa mikroračunarskog sistema sa 16-bitnom adresnom magistralom, tj. adresnim prostorom kapaciteta 64k reči u slučaju Von Neumann modela. Na ovoj mapi, memorija je organizovana kao jednostruki niz. U ovom konkretnom primeru, ulazno/izlazne jedinice su mapirane zajedno sa registrima opšte namene, kao deo istog memorijskog prostora. U ovom primeru, prvih 16 memorijskih reči (sa adresama 0h do 0Fh) su alocirane za registre opšte namene. Narednih 496 lokacija su dodeljene ulazno/izlaznim perifernim jedinicama i samo 256 reči su rezervisane za memoriju za podatke (adrese koje počinju sa 0200h i završavaju na 02FFh). Programska memorija je locirana na vrhu adresnog prostorasa svojih 8k reči na adresama počevši od 0E000h do 0FFFFh. Adrese u opsegu 0300h do 0DFFFh pripadaju neiskorišćenom adresnom prostoru, i one bi se mogle koristiti u slučaju proširenja sistema (npr. dodavanja dodatnih perifernih jedinica, ili dodatne spoljašnje RAM memorije).

Memorijska mapa može biti globalna ili parcijalna. Globalna memorijska mapa prikazuje ceo adresni prostor kao što je prikazano na slici 6 a). Parcijalna memorijska mapa prikazuje detalje jednog dela adresnog prostora, pružajući dodatne informacije u vezi sa rasporedom u globalnoj memorijskoj mapi. Na primeru sa slike 6 b), prikazana je parcijalna memorijska mapa koja prikazuje



Slika 7: Veza mikroprocesora i memorije koji imaju isti broj adresnih signala (m) i signala za podatke (n)



Slika 8: Vremenski dijagram čitanja kod sinhronne sprege

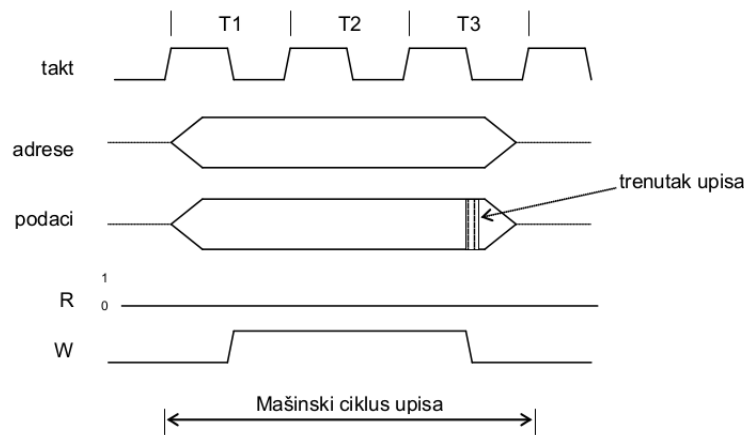
jednu moguću raspodelu ulazno/izlaznog opsega adresa.

5 Sprega mikroprocesora i memorije

Mikroprocesor i poluprovodnička memorija međusobno su kompatibilni u smislu da se mogu neposredno međusobno povezati. Neposredna veza znači da se adresni signali mikroprocesora vežu sa adresnim signalima memorije, signali podataka mikroprocesora sa signalima podataka memorije i upravljački signali mikroprocesora sa odgovarajućim signalima memorije. Slika 7 prikazuje neposrednu vezu mikroprocesora i memorije u slučaju da imaju isti broj adresnih signala (m) i signala za podatke (n) i da su upravljački signali mikroprocesora R i W.

Sprega mikroprocesora i memorije može biti sinhrona i asinhrona. Kod sinhronne sprege podrazumeva se da je memorija dovoljno brza da operacije čitanja i upisa obavi u predviđenom vremenskom intervalu. Slika 8 prikazuje vremenski dijagram ciklusa čitanja kod sinhronne sprege. Ciklus traje ukupno tri periode sinhronizacionog signala (signala takta). Na početku periode T1 mikroprocesor stavi na adresne linije signale adrese memorijske lokacije iz koje treba pročitati podatak, a zatim, kad su adresni signali dovoljno stabilni aktivira signal čitanja R. Za sve vreme ciklusa čitanja signal W je na logičkoj 0, odnosno u neaktivnom stanju.

U trenutku kada primi aktivan signal čitanja R, memorija dekoduje adresne signale, pristupi adresiranoj memorijskog lokaciji i sadržaj te lokacije prosledi na magistralu podataka. Svakako da



Slika 9: Vremenski dijagram upisa kod sinhronne sprega

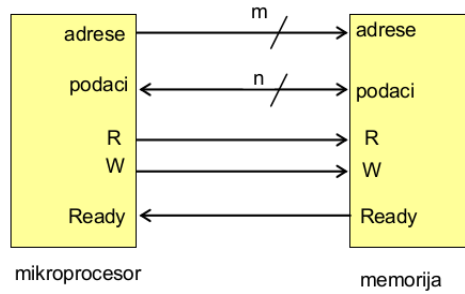
memorijskim kolima treba izvesno vreme da obavi sve ove operacije, tako da se sadržaj adresirane memorijske lokacije pojavljuje na linijama za podatke sa kašnjenjem. Mikroprocesor ostavlja memoriji tačno određeno vreme, u navedenom primeru to je do opadajuće ivice impulsa T3, kada mikroprocesor signale sa magistrale podataka učitava u interni registar. Posle toga adresni i upravljački signali (u ovom slučaju signal R) prevode se u neaktivno stanje, čime se završava ciklus čitanja. Ciklus čitanja naziva se još mašinski ciklus.

Ciklus upisa počinje tako što mikroprocesor stavlja na adresne linije adresne signale, a na linije za podatke signale podatka koji treba upisati u adresiranu memorijsku lokaciju. Posle toga aktivira se signal W za upis, dok signal R za čitanje ostaje na logičkoj 0. Po isteku dovoljno dugog vremenskog intervala, u ovom primeru do opadajuće ivice impulsa T3 memorija treba da podatak sa magistrale upiše u adresiranu memorijsku lokaciju. Posle tog trenutka, svi signali se vraćaju u početno stanje i sve je spremno za sledeći ciklus.

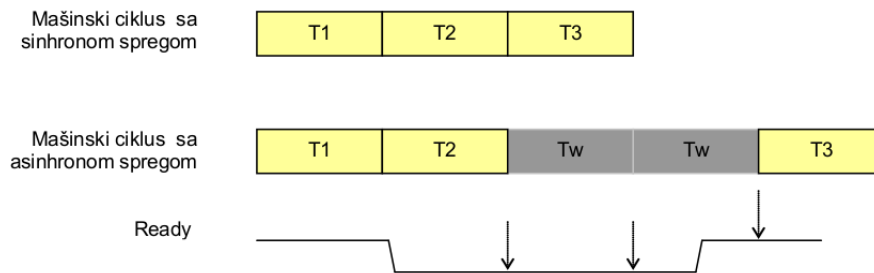
Ciklusi upisa i čitanja nazivaju se mašinski ciklusi. Izvršenje instrukcije sastoji se iz nekoliko mašinskih ciklusa: na primer jedan ciklus za čitanje koda operacije, drugi ciklus za čitanje operanda, treći ciklus za interne operacije i četvrti ciklus za upis rezultata u memoriju.

Sinhrona sprega mikroprocesora i memorije, na primer sprega koju prikazuje slika 7, ima sledeće karakteristike:

- Mikroprocesor nema povratnu informaciju da je memorija završila zahtevanu operaciju. Prema tome, mikroprocesor mora da memoriji ostavi dovoljno vremena da obavi operaciju čitanja ili upisa i to u najnepovoljnijem slučaju;
- Mikroprocesor može da radi onoliko brzo koliko dozvoljava najsporija memorijska jedinica u sistemu;
- Sinhrona sprega je jednostavna;
- Ako memorijski modul nije dovoljno brz, onda treba smanjiti frekvenciju takta mikroprocesora ili preći na korišćenje brže memorije.



Slika 10: Asinhrona sprega mikroprocesora i memorije



Slika 11: Primer asinhronog mašinskog ciklusa sa umetanjem perioda čekanja T_w

Kod asinhronne sprege memorija posebnim signalom (koritićemo engleski termin *Ready*) saopštava mikroprocesoru da je završila operaciju čitanja odnosno upisa, slika 10. Mikroprocesor nadgleda signal *Ready* i produžava mašinski ciklus sve do trenutka kada memorija završi zahtevanu operaciju.

Slika 11 prikazuje mašinski ciklus sa asinhronom spregom u poređenju sa istim ciklusom kod sinhronne sprege. Radi jednostavnosti, ovde su pravougaonicima sa upisanim oznakom perioda signala takta predstavljeni svi signali koji su navedeni u prethodnim vremenskim dijagramima. Tako na primer mašinski ciklus kod sinhronne sprege, sa slike 8 (gornji dijagram), traje tri perioda takta označena sa T1, T2 i T3.

Kod asinhronne sprege, na početku mašinskog ciklusa, na primer na kraju periode takta T1, memorija prevede signal *Ready* na logičku jedinicu i time obaveštava mikroprocesor da je započela zahtevanu operaciju, ali da je još nije završila. Na kraju periode T2 mikroprocesor proverava signal *Ready* i ako je na logičkoj nuli, produžava mašinski ciklus umetanjem periode čekanja T_w . Na kraju periode T_w , mikroprocesor ponovo proverava *Ready*, ako je i dalje na logičkoj 0 onda umetne još jedan period čekanja T_w . Ovaj postupak se ponavlja dok memorija ne završi zahtevanu operaciju i obavesti mikroprocesor tako što vrati signal *Ready* na logičku 1. Kada detektuje visoki nivo signala *Ready*, mikroprocesor završi mašinski ciklus tako što obavi mikrooperacije predviđene periodom T3.

Karakteristike asinhronne sprege:

- Mikroprocesor dobija obaveštenje da je memorija završila predviđene operacije, pa je sprega mikroprocesora i memorije pouzdanija.
- Mikroprocesor može da radi sa memorijama različite brzine i prilagođava trajanje mašinskog

ciklusa brzni svake memorijske jedinice.

- Asinhrona sprega je složenija u odnosu na sinhronu spregu.