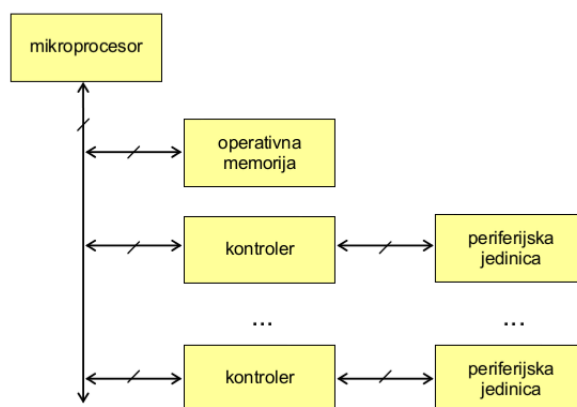

Ulazno/Izlazni podsistem

Osnovne funkcionalne jedinice mikroračunarskog sistema su centralni procesor, memorija i ulazno/izlazne jedinice. U ovom poglavlju ukratko su izloženi osnovni principi sprezanja ulazno/izlaznih jedinica sa centralnim procesorom i organizacije ulazno/izlaznog podsistema.

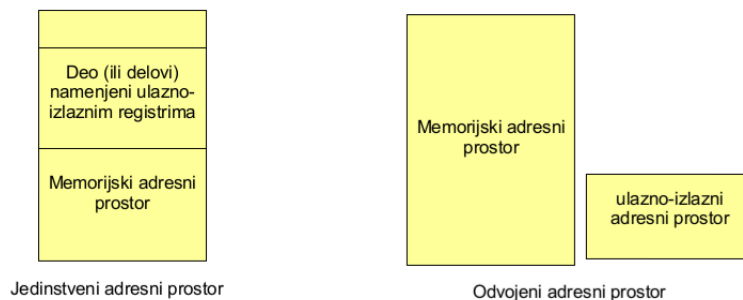
1 Organizacija ulazno/izlaznog podsistema

Slika 1 predstavlja jednostavnu organizaciju mikroračunarskog sistema sa centralnim procesorom, memorijom i ulazno/izlaznim podsistemom. Ulazno/izlazni podsistem sastoji se od perifernih jedinica i kontrolera. Perifernije jedinice su obično elektromehanički uređaji koji omogućavaju ulaz i izlaz podataka. Primeri izlaznih perifernih jedinica su monitor sa ekranom za prikazivanje teksturalnih i grafičkih informacija i štampač. Ulazne jedinice, na primer tastatura i miš, služe za unošenje podataka u računarski sistem.



Slika 1: Pojednostavljena organizacija ulazno/izlaznog podsistema

Kontroleri (ili adapteri) su elektronske komponente koje služe za prilagođenje ili interfejs između perifernih jedinica i mikroprocesora. Treba imati u vidu da su perifernije jedinice specifične i nisu prilagođene načinu i brzini rada mikroprocesora. Kontroleri obično sadrže sve neophodne elemente i logiku koja je neophodna za efikasno upravljanje perifernim jedinicama i prilagođenje između perifernih jedinica i mikroprocesora u pogledu brzine rada, naponskih



Slika 2: Adresni prostor

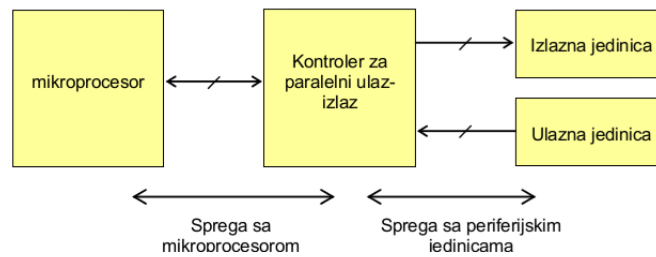
M	IO	R	W	Operacija
1	0	1	0	Čitanje iz memorije
1	0	0	1	Upis u memoriju
0	1	1	0	Čitanje iz ulazno-izlaznog registra
0	1	0	1	Upis u ulazno-izlazni registar
1	1	x	x	Nije dozvoljeno
x	x	1	1	Nije dozvoljeno

Slika 3: Operacije čitanja i upisa u adresnom prostoru

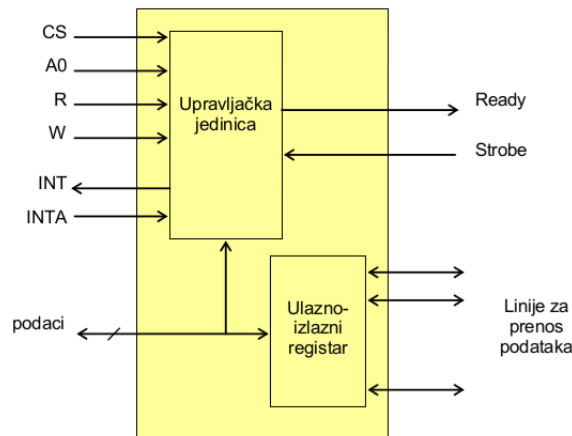
nivoa i sinhronizacije. Prema tome, mikroprocesor nema neposredan pristup perifernim jedinicama, već 'vidi' samo kontrolere i komunicira sa kontrolerima. Kontroleri su programabilne jedinice koje poseduju svoje upravljačke jedinice sposobne da obavljaju zadatke vezane za upravljanje i komunikaciju sa perifernim jedinicama. Kontroleri poseduju programabilne registre preko kojih se razmenjuju upravljačke informacije i podaci sa mikroprocesorom. Naravno, programabilni registri imaju svoje adrese i mikroprocesor mora da generiše tačnu adresu programabilnog registra kome pristupa. U jednostavnim primerima ulazno/izlaznih podsistema, programabilni registri mogu da se neposredno koriste za ulaz i izlaz podataka.

Stoga ćemo sve registre koji se nalaze u ulazno/izlaznom podsistemu nazivati ulazno/izlazni registri. Ulazno/izlazni registri mogu da se nalaze u jedinstvenom adresnom prostoru zajedno sa memorijom ili da budu odvojenom, ulazno/izlaznom adresnom prostoru (slika 2). Prednost jedinstvenog adresnog prostora je u tome što sve instrukcije koje se koriste za pristup memoriji mogu da se koriste i za pristup ulazno/izlaznim registrima i što se memorijski i ulazno/izlazni registri tretiraju na isti način. Prednost odvojenog adresnog prostora je što omogućava jednostavniju organizaciju sistema u kome se jedan adresni prostor koristi isključivo za memorijske, a drugi za ulazno/izlazne komponente.

Mikroprocesor koristi iste linije za prenos adresa i podataka kod memorijskog i ulazno/izlaznog adresnog prostora i zato mora da ima posebne signale kojima označava vrstu pristupa. U slučaju da signal M označava pristup memorijskom adresnom prostoru a signal IO pristup ulazno/izlaznom adresnom prostoru, sledeća tabela opisuje operacije mikroprocesora u zavisnosti od logičkih nivoa ovih signala. Signal R označava ciklus čitanja, a W ciklus upisa.



Slika 4: Sprega kontrolera za paralelni ulaz/izlaz sa mikroprocesorom i ulazno/izlaznim jedinicama



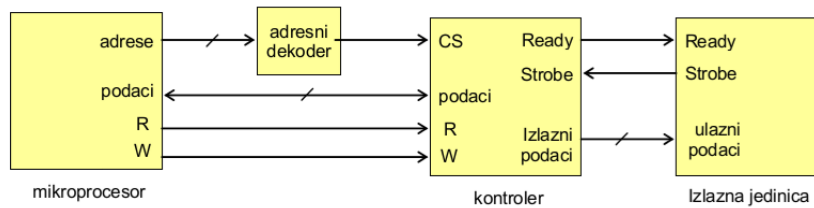
Slika 5: Struktura jednostavnog kontrolera za paralelni ulaz/izlaz

2 Kontroler za paralelni ulaz/izlaz

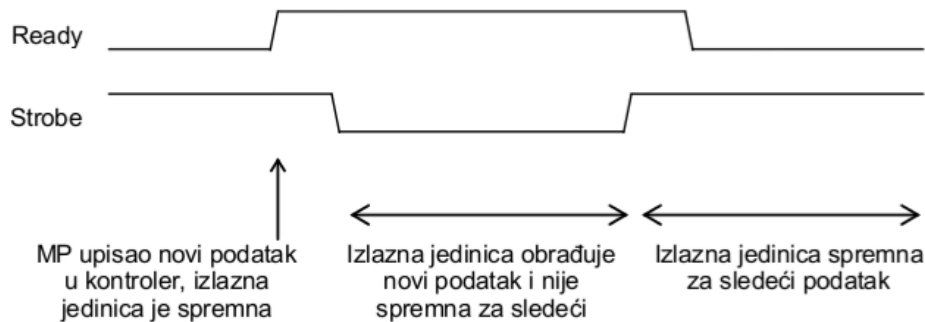
Jednostavni primer ulazno/izlaznog kontrolera je kontroler za paralelni ulaz i izlaz podataka. Ovaj kontroler sadrži registre za podatke koji se mogu konfigurisati tako da služe za paralelni ulaz ili izlaz podataka. Pored toga, ovi kontroleri poseduju programabilne registre koji se koriste za konfigurisanje i izbor načina rada kontrolera. Kontroler poseduje sa jedne strane signale za spregu sa mikroprocesorom, a sa druge strane signale za spregu sa ulazno/izlaznim jedinicama, kao što je prikazano na slici 4.

Slika 5 prikazuje strukturu jednostavnog kontrolera za paralelni ulaz/izlaz, gde su sa leve strane nacrtani signali za spregu sa mikroprocesorom, a sa desne signali za spregu sa ulazno/izlaznim jedinicama. U mikroprocesorskom sistemu mora da postoji dekodir ulazno/izlaznih adresa koji generiše signal CS za aktiviranje kontrolera. U slučaju da poseduje više ulazno/izlaznih registara kontroler mora da ima dodatne adresne signale. U primeru sa slike postoji samo jedan adresni signal A0. Signali R i W određuju operacije čitanja i upisa koje se obavljaju nad registrima kontrolera.

Kontroler ima linije za pristup magistrali podataka mikroprocesorskog sistema preko kojih mikroprocesor komunicira sa kontrolerom. Konačno, kontroler obično ima signal INT kojim zahteva prekid od mikroprocesora i signal INTA kojim mikroprocesor obaveštava kontroler da je prekid prihvaćen. Na strani perifernjske jedinice kontroler ima linije za prenos podataka i dve linije za prenos signala za sinhronizaciju, koji se obično nazivaju Ready i Strobe. Uloge ova dva



Slika 6: Sistem sa mikroprocesorom, kontrolerom za paralelni ulaz/izlaz i perifernom jedinicom



Slika 7: Postupak sinhronizacije (rukovanje) između kontrolera i izlazne jedinice

signala različite su kod ulaznih i izlaznih operacija. Na primer, kod izlaza paralelnih podataka prema perifernoj jedinici značenja mogu biti:

- $Ready = 0$, na izlaznim linijama prema perifernoj jedinici je 'stari' podatak,
- $Ready = 1$, na izlaznim linijama je novi podatak,
- $Strobe = 0$, periferna jedinica nije spremna da prihvati novi podatak,
- $Strobe = 1$, periferna jedinica je spremna da prihvati novi podatak.

Slika 6 prikazuje konfiguraciju sistema sa mikroprocesorom, kontrolerom za paralelni ulaz- izlaz (konfigurisan za izlaz podataka) i perifernom izlaznom jedinicom. Radi jednostavnosti nisu prikazane linije za prenos signala A0 i signala za prekide INT i INTA.

Postupak izlaza podataka prema perifernoj jedinici započinje tako što mikroprocesor upiše u kontroler podatak koji treba preneti perifernoj jedinici. Posle toga mikroprocesor može da radi neke druge operacije i prepušta kontroleru da prenese podatak do periferijske jedinice. Osnovni problem u prenosu podataka je što periferna jedinica može da bude spora, odnosno da zahteva neko vreme da bi obradila primljeni podatak. Na primer, ako je izlazna jedinica DA konvertor, onda je potrebno neko vreme za konverziju iz digitalnog u analogni oblik. Ako je izlazna jedinica štampač, onda je potrebno neko vreme za štampanje primljenog znaka. Prema tome, očigledno je da kontroler mora da prilagodi brzinu prenosa izlaznih podataka mogućnostima izlazne jedinice.

Postupak sinhronizacije između kontrolera i izlazne jedinice ilustruje vremenski dijagram, prikazan na slici 7.

1. U početku je izlazna jedinica spremna da prihvati novi podatak i stavlja signal Strobe na logičku 1;

2. Kada mikroprocesor upiše novi izlazni podatak, kontroler prenosi taj podatak na linije za prenos podataka do izlazne jedinice i stavlja signal Ready na logičku 1, čime obaveštava izlaznu jedinicu da je je novi podatak spreman;
3. Izlazna jedinica prevodi signal Strobe na logičku 0 i ostavlja ga na toj vrednost sve dok obrađuje novi podatak;
4. U trenutku kad je završila obradu novog signala (na primer kad je štampač odštampao primljeni znak) izlazna jedinica prevodi signal Strobe na logičku 1 i tako obaveštava kontroler da je spremna za naredni podatak;
5. Kontroler prevodi signal Ready na logičku 0 i tako stavlja do znanja izlaznoj jedinici da je na izlaznim linijama podatak koji je već obrađen.

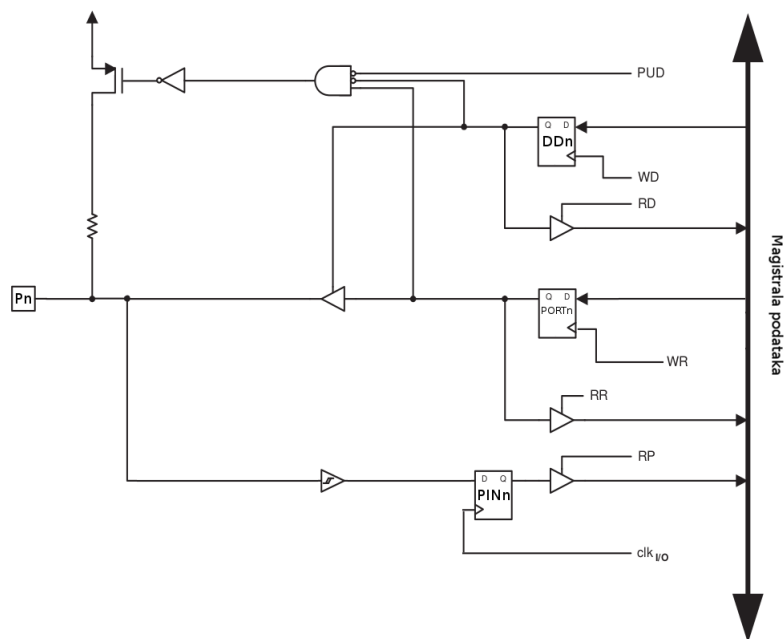
Dalji postupak zavisi od načina prenosa izlaznih podataka. Na primer, tipično bi bilo da kontroler generiše signal prekida i tako obavesti mikroprocesor da je izlaz prethodnog podatka završen. U programu za obradu prekida mikroprocesor može da odredi sledeći podatak i da ga upiše u kontroler čime se postupak izlaza podataka ponavlja. Ovakav način sinhronizacije između kontrolera i izlazne jedinice naziva se 'rukovanje' (engleski: *handshaking*).

2.1 Primer: Digitalni ulazno/izlazni port opšte namene kod AVR mikrokontrolera

U ovom primeru će biti prikazan ulazno/izlazni port kod AVR familije mikrokontrolera. Kada se port konfigurise kao ulazni sa njega je moguće čitati ulazne signale (npr stanje sa tastera ili prekidača, ulazni signal sa senzora itd). U slučaju da je port konfigurisan kao izlazni, na istim pinovima (nožicama mikrokontrolera) mogu se generisati signali koji će, na primer, uključivati ili isključivati LED diode, ili upravljati radom motora. Mikrokontroler ATmega328, koji se koristi na vežbama, ima 23 bi-direkciona ulazno izlazna pina, grupisana u okviru tri ulazno/izlazna porta. Na slici 8 je prikazana interna struktura jednog pina.

U okviru strukture se mogu zapaziti:

- flip-flop DDn koji služi za odabir smera, tj. on odlučuje da li će se pin koristiti kao ulazni ili kao izlazni
- flip-flop PORTn služi da skladišti bit koji će biti prosleđen u slučaju kada se pin konfigurise kao izlazni
- flip-flop PINn prihvata informaciju u slučaju kada je pin konfigurisan da bude ulazni
- pin Pn je nožica mikrokontrolera koja se spaja sa eksternim komponentama (otpornicima, sensorima, aktuatorima, LED diodama, prekidačima, tasterima, ...)
- signal PUD ("Pull-Up Disable" služi da isključi pull-up tranzistor kada je to potrebno
- signal WD ("Write Direction") služi za upis bita koji odabira smer u DDn flip-flop
- signal RD ("Write Direction") služi za čitanje bita koji odabira smer u DDn flip-flop



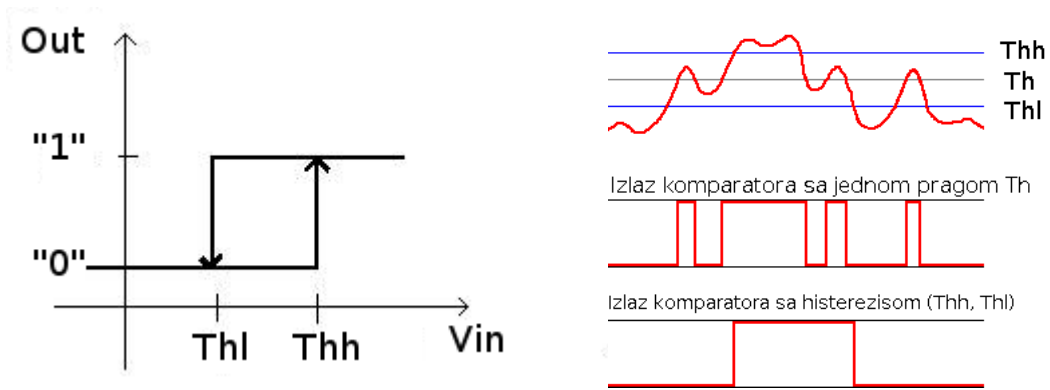
Slika 8: Uprošćena struktura pina kod ATMega familije mikrokontrolera

- signal WR (“Write port Register”) služi za upis bita koji će se poslati na pin Pn konfigurisan kao izlaz
- signal RR (“Read port Register”) služi za čitanje bita koji će se poslati na pin Pn konfigurisan kao izlaz
- signal RP (“Read Pin”) se koristi za upis novog ulaznog bita prisutnom na pinu Pn konfigurisanom da bude ulazni.

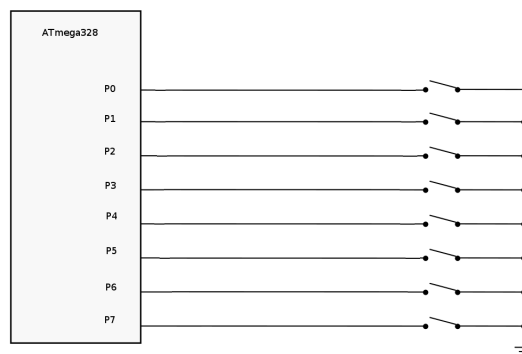
Najpre ćemo analizirati rad kola u slučaju kada je pin konfigurisan kao izlazna periferija. U tom slučaju se u flip-flop DDn upisuje “1”, aktiviranjem signala WD i dovodenjem logičke jedinice sa magistrale podataka na D ulaz flip-flopa DDn. Kao rezultat, tro-statički bafer na sredini slike propušta signal sa izlaza flip-flopa PORTn direktno na izlazni pin Pn, pri čemu je pull-up tranzistor isključen (“1” upisana u flip-flop DDn se vodi na ulaz I kola preko invertora i kao rezultat isključuje p-kanalni pull-up MOSFET, čija će uloga biti objašnjena kasnije). Aktivacijom signala RD može se, u svakom trenutku proveriti trenutno stanje registra DDn, tj. konfiguracije pina koji se posmatra.

Čitanje spoljašnjeg signala preko pina je znatno složenije. Da bi se uopšte mogao čitati spoljašnji signal na pinu Pn potrebno je postaviti flip-flop Dxn u stanje “0”. Ova “0” sa jedne strane uključuje pull-up tranzistor, dok istovremeno isključuje tri-statički bafer na sredini slike i na taj način “odspaja” pin Pn od flip-flopa PORTn.

Zašto je potreban pull-up tranzistor i koja je njegova uloga? Ovaj tranzistor obezbeđuje da će na izlaznoj liniji biti visok potencijal ukoliko ništa nije spojeno (otuda naziv pull-up), ali se neće protiviti ukoliko neki drugi izvor signala “obori” potencijal pina Pn na logičku nulu. Kao što ćemo videti kasnije, ovo znatno olakšava dizajn kola i broj potrebnih eksternih komponenti. Treba obratiti pažnju da se pull-up tranzistor može isključiti čak i u slučaju kada je pin Pn konfigurisan



Slika 9: Komparator sa histerezisom



Slika 10: Mikrokontroler sa prekidačima

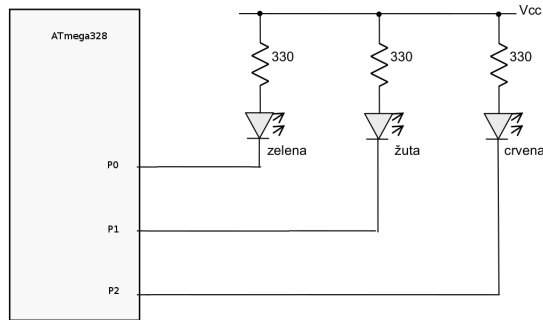
kao ulazni, a to se dešava kada je vrednost flip-flop-a $PORTn$ "0". Pošto ovaj flip-flop svakako nema uticaja u konfiguraciji kada je pin ulazni, iskorišćen je da se isključi pull-up tranzistor kada nije potreban (jednostavnim upisom logičke "0" u flip-flop $PORTn$).

Ostaje još jedino da prokomentarišemo donji deo slike i da objasnimo kako se zaista čita stanje sa ulaznog pina Pn . Signal se sa pina vodi na komparator koji ima ulogu da klasifikuje proizvoljan ulazni signal ili kao logičko "1" ili kao logičko "0". U najjednostavnijoj implementaciji, ovo se postiže tako što se signal poredi sa pragom Th i sve što je ispod praga se tretira kao logičko "1" dok je sve što je ispod praga prepoznato kao logičko "0". Problem sa ovim rešenjem nastaje u slučaju kada je ulazni signal blizu praga, pa zbog šuma signal bude kratkotrajno iznad, pa kratkotrajno ispod praga, što dovodi do nestabilnosti izlaza komparatora. Ovaj problem se rešava korišćenjem komparatora sa histerezisom koji ima dva praga (slika 9):

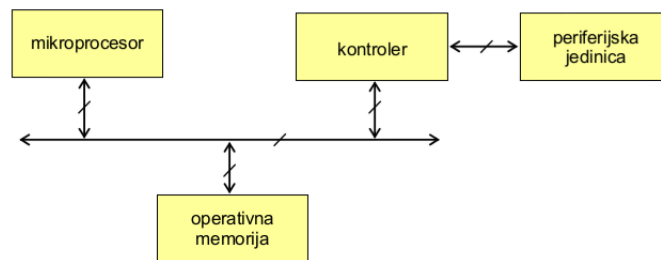
- Thh se koristi kada je ulazni signal nizak i raste
- Thl se koristi kada je ulazni signal visok i opada.

Na ovaj način je obezbeđeno da neće doći do nestabilnosti na izlazu komparatora jer se "aktivan" prag menja onog trenutka kada signal na ulazu dođe do njega.

Izlaz komparatora koji daje logičko "1" ili logičko "0" u zavisnosti od vrednosti signala na ulaznom pinu Pn se upisuje u flip-flop $PINn$ korišćenjem taktnog signala za ulazno/izlazni podsistem koji je



Slika 11: Mikrokontroler sa LED diodama



Slika 12: Blok dijagram jednostavnog mikroracunarskog sistema

obično znatno niže frekvencije od osnovnog takta centralne procesorske jedinice. Vrednost očitanoog spoljašnjeg signala može da se pročita korišćenjem RP signala.

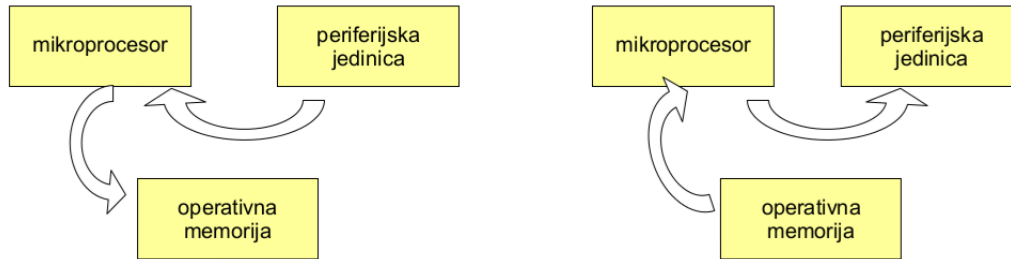
Na slikama 10 i 11 su prikazane dve konfiguracije u kojima se portovi koriste kao ulazi i izlazi mikrokontrolera. Zahvaljujući pull-up tranzistorima, potrebno je povezati samo prekidače na ulaze mikrokontrolera u slučaju kada su pinovi konfigurisani kao ulazni koji treba da detektuju stanje na prekidačima. U drugom slučaju, povezivanjem LED dioda na izlaz mikrokontrolera moguće je paliti/gasiti diode preko pinova konfigurisanih kao izlaz.

3 Direktan pristup memoriji

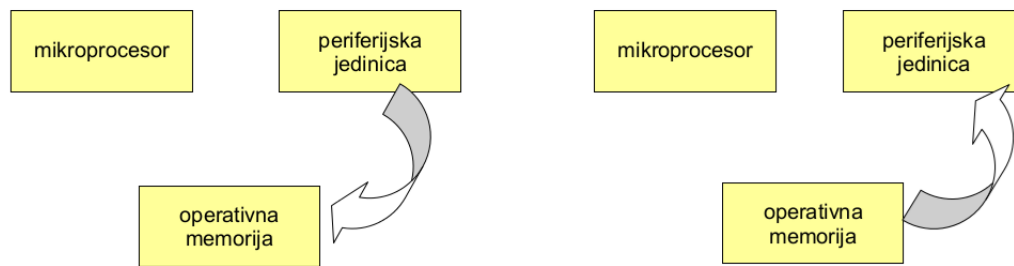
Posmatrajmo jednostavan mikroracunarski sistem sa mikroprocesorom, operativnom memorijom i jednom periferijskom jedinicom, videti sliku 12. Radi jednostavnijeg objašnjenja direktnog pristupa memoriji, posmatraćemo kontroler i periferijsku jedinicu zajedno, kao jedan blok.

Kod prenosa podataka sa periferijske jedinice u operativnu memoriju, svaki pojedinačni podatak mora prvo da se prenese u registar mikroprocesora, a zatim iz registra mikroprocesora u operativnu memoriju, slika 13 (levo). Naravno, kod prenosa u suprotnom smeru, podatak mora prvo iz memorije da se prenese u registar mikroprocesora, a zatim iz registra mikroprocesora u periferijsku jedinicu, slika 13 (desno).

Očigledno je ovakav način prenosa spor i neefikasan. Spor je zato što mikroprocesor mora da izvrši niz instrukcija koje pritupaju periferijskoj jedinici, izračunavaju adresu memorije, pristupaju memoriji i tako dalje. Prenos nije efikasan zato što mikroprocesor u toku prenosa ne može da obavlja bilo kakve druge operacije.



Slika 13: *Prenos podataka sa periferijske jedinice u operativnu memoriju (levo) i u suprotnom smeru (desno)*



Slika 14: *Ideja direktnog pristupa memoriji kod prenosa podataka iz periferijske jedinice u operativnu memoriju (levo) i u suprotnom smeru (desno)*

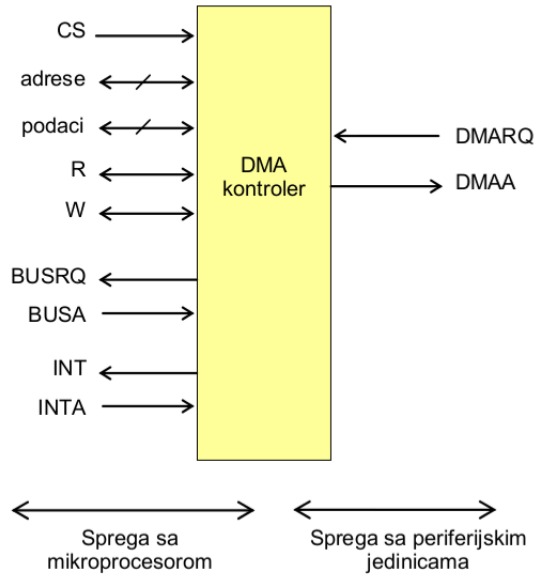
Ideja direktnog pristupa memoriji (Direct Memory Access, DMA) sastoji se u tome da se podaci neposredno prenose između periferijske jedinice i operativne memorije, bez ikakvog učešća mikroprocesora, slika 14. Naravno, mikroprocesorski sistem mora da ima poseban DMA kontroler koji organizuje i upravlja ovakvim načinom prenosa podataka.

Slika 15 prikazuje spoljne signale DMA kontrolera. Signal CS koristi se za aktiviranje DMA kontrolera kod pristupa mikroprocesora programabilnim registrima kontrolera. Signali adresne magistralne, magistralne podataka i upravljački signali R i W su dvosmerni i mogu se prevesti u stanje visoke impedanse. Signali INT i INTA koriste se za mehanizam prekida.

Da bi organizovao direktan pristup memoriji DMA kontroler mora prvo da preuzme upravljanje magistralama i generiše adresne signale i signale R i W. Pošto u svakom trenutku samo jedna jedinica može upravljati magistralama, jedinica koja hoće da preuzme upravljanje magistralama mora od mikroprocesora da traži dozvolu. Signali BUSRQ (Bus Request) i BUSA (Bus Acknowledge) služe za sinhronizaciju pristupa magistralama između DMA kontrolera i mikroprocesora. Logičkom 1 na liniji BUSRQ DMA kontroler traži od mikroprocesora upravljanje magistralama, a logičkom 1 na liniji BUSA mikroprocesor obaveštava DMA kontroler da dozvoljava upravljanje magistralama.

Dok mikroprocesor upravlja magistralama, DMA kontroler drži svoje linije za adresnu magistralu i linije R i W u stanju visoke impedanse. Sa druge strane, kada prepusti DMA kontroleru upravljanje magistralama, mikroprocesor adresnu magistralu i signale R i W prevodi u stanje visoke impedanse.

Signalom DMARQ (DMA Request) periferijska jedinica obaveštava DMA kontroler da je spremna za DMA prenos, a signalom DMAA (DMA Acknowledge) DMA kontroler obaveštava periferijsku jedinicu da je započeo ciklus DMA prenosa.



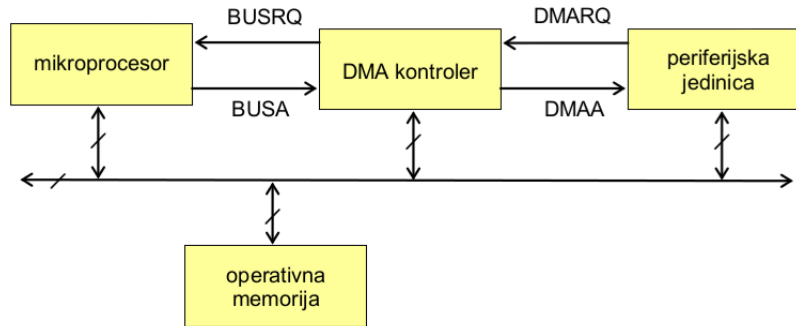
Slika 15: Osnovni spoljni signali DMA kontrolera

U postupku primene DMA razlikuju se tri perioda vremena: (i) pre DMA prenosa, (ii) u toku DMA prenosa i (iii) posle DMA prenosa. Pre DMA prenosa mikroprocesor programira periferijsku jedinicu i DMA kontroler. Na primer, u slučaju diska, mikroprocesor u kontroler diska upisuje sektore kojima treba pristupiti, smer prenosa podataka i količinu podataka koje treba preneti. U DMA kontroler mikroprocesor upisuje smer prenosa podataka, količinu podataka koje treba preneti i početnu memorijsku adresu zone u koju treba upisati podatke sa periferijske jedinice ili iz koje treba čitati podatke koji se prenose periferijskoj jedinici.

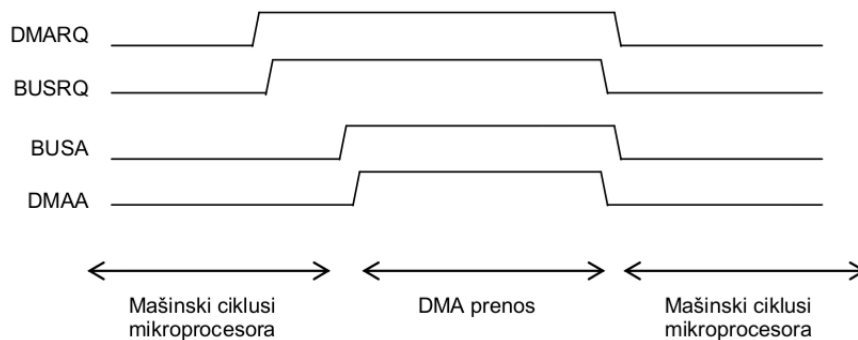
Jednom kada završi programiranje periferijske jedinice i DMA kontrolera, mikroprocesor ne učestvuje u DMA prenosu i može da obavlja druge operacije. DMA kontroler organizuje DMA prenos tako što za svaki podatak koji treba preneti između periferijske jedinice i memorije, traži od mikroprocesora pristup magistralama i kada dobije dozvolu pristupa, generiše sve signale koji su neophodni za pristup memoriji.

Postupak prenosa jednog podatka, na primer iz periferijske jedinice u memoriju, počinje tako što periferijska jedinica signalom DMARQ javlja DMA kontroleru da je spremna za prenos, slika 16. DMA kontroler zatim aktivira signal BUSRQ kojim od mikroprocesora traži upravljanje magistralama. Kada mikroprocesor završi tekući mašinski ciklus, prevodi svoje magistrale u stanje visoke impedanse i signalom BUSA obaveštava DMA kontroler da ima dozvolu pristupa magistralama. Sada DMA kontroler signalom DMAA obaveštava periferijsku jedinicu da započinje ciklus DMA prenosa i periferijska jedinica prenosi podatak na magistralu podataka. Istovremeno, DMA kontroler generiše adresne signale i signal W. Podatak sa magistrale podataka memorija smešta u adresiranu memorijsku lokaciju i time se završava DMA prenos jednog podatka.

Slika 17 prikazuje vremenske dijagrame signala koji učestvuju u sinronizaciji kod DMA prenosa. Po završetku ciklusa DMA prenosa, DMA kontroler svoje spoljne signale prevodi u stanje visoke impedanse a signale BUSRQ i DMAA u neaktivno stanje (logička 9). Periferijska jedinica svoj



Slika 16: Konfiguracija mikroprocesorskog sistema za DMA prenos

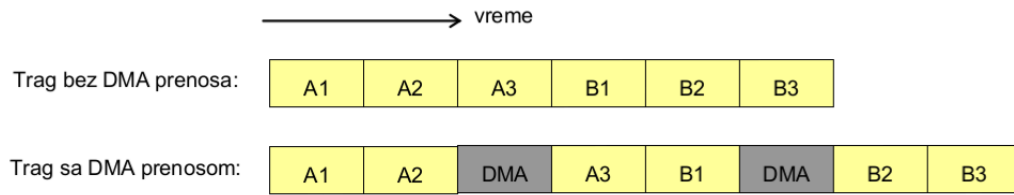


Slika 17: Vremenski dijagrami sinhronizacije kod DMA prenosa

signal DMARQ prevodi u logičku 0. Mikroprocesor signal BUSA prevodi u neaktivno stanje (logička 9) i nastavlja da izvršava svoje naredne mašinske cikluse. U ovom trenutku sve jedinice su spremne da ponove ciklus DMA prenosa.

Kod svakog DMA prenosa, DMA kontroler smanjuje za jedan sadržaj registra u kome se nalazi broj podataka koje treba preneti i inkrementira (ili dekrementira) adresu memorijske lokacije kojoj se pristupa u toku DMA prenosa. Kada se prenesu svi podaci DMA kontroler obično generiše prekid kojim obaveštava mikroprocesor da je DMA prenos je završen. Mikroprocesor može da preduzme dalje akcije, na primer da obradi prenesene podatke ili da pripremi novi DMA prenos. Efekat DMA prenosa može se posmatrati preko traga mikroprocesorskog sistema, slika 18. Pravougaonici predstavljaju mašinske cikluse, a radi jednostavnosti uzeto je da svi ciklusi imaju isto trajanja. Mašinski ciklusi mikroprocesora su na primer, čitanje iz memorije, upis u memoriju, interne operacije i pristup ulazno/izlaznim registrima. Gornji trag predstavlja izvršenje dve instrukcije, A i B, sa po tri mašinska ciklusa, označena sa A1, A2 i A3, odnosno B1, B2 i B3.

Donji trag predstavlja izvršenje instrukcija A i B u slučaju da su se dogodila dva mašinska ciklusa sa DMA prenosom, jedan između ciklusa A2 i A3 i drugi između B1 i B2. U toku mašinskog ciklusa A2 mikroprocesor je primio zahteva za DMA prenos, sačekao da završi tekući ciklus i predao DMA kontroleru upravljanje magistralama. DMA kontroler je obavio DMA prenos u toku jednog mašinskog ciklusa i vratio mikroprocesoru upravljanje magistralama. Mikroprocesor nastavlja izvršenje instrukcije A tako što izvrši naredni mašinski ciklus, A3, i time



Slika 18: *Trag mikroprocesorskog sistema bez DMA prenosa (gore) i sa DMA prenosom (dole)*

završava instrukciju A. Na sličan način obavlja se i drugi DMA prenos. Iz traga se vidi da je kod DMA prenosa trag isti kao i bez DMA prenosa, samo što s vremena na vreme DMA kontroler obavlja DMA prenos. Ovakav način korišćenja mašinskih ciklusa naziva se 'krađa ciklusa' (eng. *cycle stealing*).