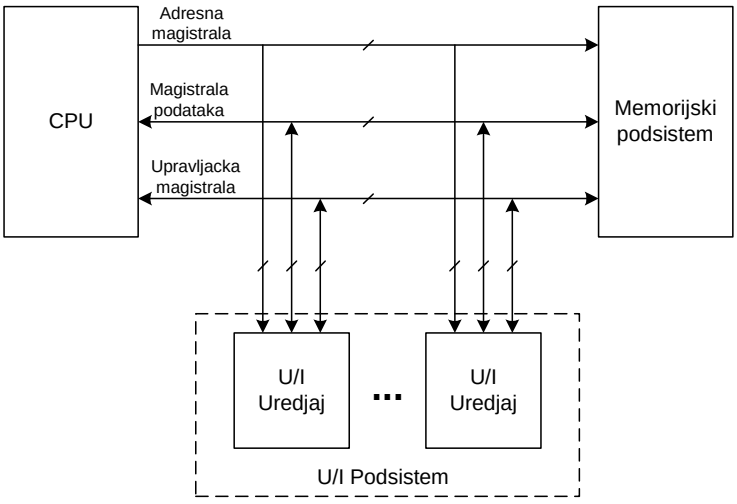
ARHITEKTURA MIKROPROCESORSKIH SISTEMA

Organizacija računara

Većina savremenih računarskih sistema, počev od jednostavnih mikrokontrolera do personalnih računara i radnih stanica velike moći izračunavanja, zasnovana je na istoj bazičnoj organizaciji koju čine sledeća tri glavna podsistema (Sl. 1):

- **Centralna procesorska jedinica** (CPU – *Central Processor Unit*). Interno, CPU se sastoji od upravljačke jedinice, koja upravlja radom celokupnog računara i staze podataka koja sadrži registre, aritmetičko-logičku jedinicu (ALU) i druge funkcijske jedinice specijalizovane za pojedine tipove obrade podataka (matematički koprocesori, akceleratori i sl.).
- **Memorijski podsistem**. Sačinjen od specijalizovanih kola sa mogućnošću memorisanja informacije. U memoriji računara čuvaju se program i podaci. CPU vidi memoriju kao skup memorijskih lokacija, gde svaka lokacija, u zavisnosti od organizacije memorije, sadrži 8/16/32/64 bita. Svakoj memoriskoj lokaciji pridružena je jedinstvena adresa.
- **Ulazno-izlazni podsistem**. Skup specijalizovanih kola koja služe za spregu CPU sa aktorima iz spoljnjeg sveta, kao što su tastatura, miš i displej. Ovo su integrisana kola koja, tipično, poseduju svoju upravljačku jedinicu i stazu podataka. CPU vidi U/I uređaj kao skup od nekoliko memorijskih lokacija (tzv. registara ili portova). Upisom/čitanjem u/iz ovih lokacija moguće je upravljati radom U/I uređaja, tj. inicira neke aktivnosti perifernog uređaja, postavi uređaj u željeni režim rada, pribavi informaciju o tekućem stanju uređaja, prenese/prihvati podatke u/od uređaja.



Sl. 1 Organizacija računarskog sistema.

CPU je u stanju da obavlja elementarne računske operacije i razmenjuje podatke sa memorijom i specijalizovanim interfejsnim kola za komunikaciju sa spoljnim svetom. Program se sastoji od niza instrukcija, od kojih svaka ukazuje na jednu elementarnu operaciju, a sve zajedno definišu izračunavanja koja računar treba da obavi kako bi se ostvario željeni algoritam. Program sastavlja čovok (programer) i zajedno sa ulaznim podacima smešta ih u memoriju računara. CPU pribavlja iz memorije instrukcije i podatke, i izvršava instrukcije, transformišući tako ulazne podatke u izlazne, koje, po okončanju rada,

predaje okruženju. Koncept računara je nezavisan od tehnologije. Prvi računari bili su elktro-mehaničke naprave. Međutim, danas, računarski sistemi se realizuju isključivo u digitalnoj integrisanoj tehnologiji. Ova tehnologija omogućava realizaciju računskih mašina koje se sastoje od više miliona elementarnih sklopova (gejtova) i u stanju su da obavljaju više stotina miliona elementarnih računskih operacija u sekundi.

Sistemske magistrale

Kao što se može videti sa Sl. 1, podsistemi računara međusobno su povezani uz pomoć magistrala. Magistralu čini skup veza (žica), od kojih se svaka koristi za prenos jednog bita informacije. Prenos informacije od jedne do neke druge komponete ostvaruje se tako što izvorišna komponenta postavlja podatak na magistralu, a odredišna komponenta očitava podatak sa magistrale. Korišćenje magistrala je ekonomičnije rešenje u odnosu na direktne veze između sistemskih komponenata, naročito u slučevima kada je broj komponenti veliki. Magistrale zauzimaju manje prostora na štampanoj ploči ili na čipu i zahtevaju manji utrošak energije u odnosu na veći broj direktnih veza. Takođe, magistrale zahtevaju manji broj pinova na čipu ili čipovima koji čine računarski sistem.

Sistem prikazan na Sl. 1 koristi tri magistrale:

(1) adresna magistrala,

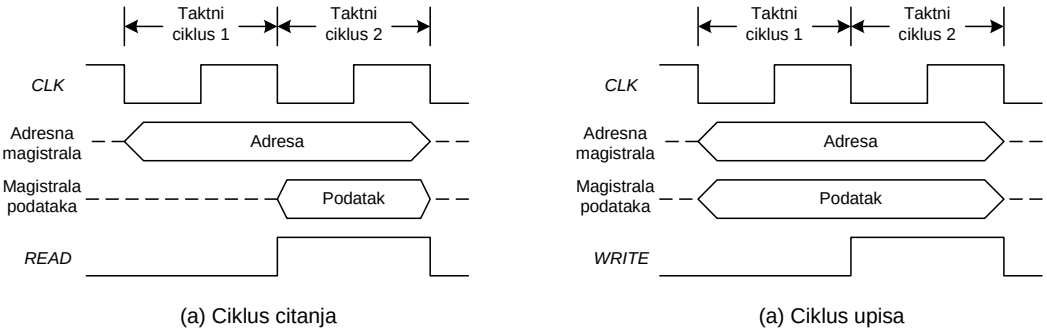
(2) magistrala podataka i

(3) upravljačka magistrala.

Adresna magistrala se koristi za prenos memorijskih i U/I adresa. Uvek kada CPU čita podatak ili instrukciju iz memorije ili upisuje podatak u memoriju, on mora da navede adresu memorijske lokacije kojoj želi da pristupi. CPU postavlja adresu pristupa na adresnu magistralu; memorija prihvata adresu sa adresne magistrale i koristi je za pristup odgovarajućoj memorijskoj lokaciji. Takođe, svaki U/I uređaj, kao što je tastatura, kontroler displeja, ili kontroler diska, kao i svaki port unutar U/I uređaja poseduje jedinstvenu adresu. Uvek kad CPU pristupa nekom U/I uređaju, on postavlja adresu uređaja na adresnu magistralu. Svaki uređaj očitava adresu sa magistrale i utvrđuje da li CPU pristupa baš njemu. Za razliku od ostalih sistemskih magistrala, podatke na adresnu magistralu uvek postavlja CPU (tj. CPU nikada ne preuzima podatak sa adresne magistrale). Drugim rečima, adresna magistrala je jednosmerna, usmerena od CPU ka ostalim sistemskim komponentama.

Podaci se prenose preko magistrale podataka. Prilikom pribavljanja podataka ili instrukcija iz memorije, CPU prvo postavlja memorijsku adresu na adresnu magistralu. Nakon toga, memorija postavlja traženi podatak na magistralu podataka koga, konačno, CPU preuzima. Kada upisuje podatak u memoriju, CPU postavlja adresu na adresnu magistralu, a podatak na magistralu podataka. Memorija preuzima podatak i smešta ga u odgovarajuću lokaciju. Proces čitanja/upisa podataka iz/u U/I uređaja odvija se na sličan način.

Upravljačka magistrala razlikuje se donekle od druge dve magistrale. Naime, adresna magistrala se sastoji od n linija koje se zajedno koriste za prenos jedne n-to bitne adrese. Slično, linijama magistrale podataka prenosi se jedinstveni, više-bitni podatak. Nasuprot prethodnom, upravljačka magistrala je skup nezavisnih upravljačkih signala i to je razlikuje od magistrale podataka i adresne magistrale. Na primer, signali upravljačke magistrale ukazuju da li se pristupa U/I uređajima ili memoriji, da li CPU zahteva upis ili čitanje, da li su U/I uređaj ili memorija spremni za prenos podataka i td. Iako je na Sl. 1, upravljačka magistrala predstavljena kao dvosmerna, praktično, ovu magistralu uglavnom čine logički nezavisni jednosmerni signali koji se kreću u različitim smerovima. Većina signala izlazi iz CPU jedinice, mada postoje i oni koji ulaze u CPU.



Sl. 2 Vremenski dijagrami za čitanje iz i upis u memoriju.

Na Sl. 2(a) prikazan je vremenski dijagram operacije čitanja iz memorije. Uočimo signal CLK. To je **sistemski taktni** (**klok**) **signal** koji sinhroniše sve operacije koje CPU obavlja. Na početku **taktnog ciklusa 1**, CPU postavlja adresu na adresnu magistralu. Memorijski podsistem prihvata adresu i dekodira je kako bi pristupio željenoj memorijskoj lokaciji. Jedan taktni ciklus kasnije, CPU aktivira upravljački signal READ, što uslovljava da memorija postavi traženi podatak na magistralu podataka. U toku ovog taktnog ciklusa, CPU očitava podatak sa magistrale i smešta ga u neki od svojih internih registara. Na kraju ovog taktnog ciklusa, CPU sklanja adresu sa adresne magistrale i deaktivira signal READ. Tada, memorija sklanja podatak sa magistrale podataka, čime je operacija čitanja iz memorije završena.

Vremenski dijagram operacije upisa u memoriju prikazan je na Sl. 2(b). U toku prvog taktnog ciklusa, CPU postavlja adresu i podatak na sistemske magistrale. Na početku drugog taktnog ciklusa, CPU aktivira upravljački signal WRITE. Kao što signal READ nalaže memoriji da pročita podatak, tako signal WRITE nalaže memoriji da prihvati podatak sa magistrale podataka i upiše ga u lokaciju čija je adresa prisutna na adresnoj magistrali. Na kraju drugog ciklusa, CPU uklanja adresu i podatak sa sistemskih magistrala i deaktivira signal WRITE. Na taj način, operacija upisa u memoriju je završena.

Brzina odziva memorije karakteriše se vremenom pristupa. To je vreme koje je neophodno da memorija nakon postavljanja nove adrese pronađe i pristupi adresiranoj lokaciji. Da bi sistem radio korektno, neophodno je da period sistemskog takta bude veći ili jednak vremenu pristupa memoriji. Ako to nije slučaj, neophodno je produžiti ciklus čitanja/upisa umetanjem jednog ili više taktnih ciklusa tzv. stanja čekanja.

Operacije čitanja/upisa iz/u U/I uređaja su slične operacijama memorijskog čitanja i upisa.

U/I podsistem može biti organizovan na jedan od dva standardna načina:

> (1) **memoriski mapirani U/I** – gde CPU tretira portove U/I uređaja kao memorijske lokacije i shodno tome, CPU je u obavezi da pristup U/I podsistemu tretira na isti način kako pristup memorijskom podsistemu;
>
> (2) **izolovani U/I** – pristup U/I podsistemu je identičan pristupu memorijskom podsistemu, s tom razlikom da CPU poseduje dodatni upravljački signal koji razdvaja pristup U/I uređajima od pristupa memoriji. Na primer, procesor 8085 poseduje upravljački signal IO/M. Kada želi da pristupi memoriji, bilo radi čitanja ili upisa, CPU drži signal IO/M na 0 za sve vreme trajanja operacije čitanja/upisa. Sa druge strane, za vreme trajanja operacije čitanja/upisa iz/u U/I podsistema, CPU postavlja signal IO/M na 1.

Za razliku od memorije čija je jedina funkciju da skladišti informacije, svaki U/I uređaj ima neku specifičnu funkciju. Obično, podatak koji se upisuje u U/I uređaj predstavlja instrukciju U/I uređaju koju akciju da preduzme.

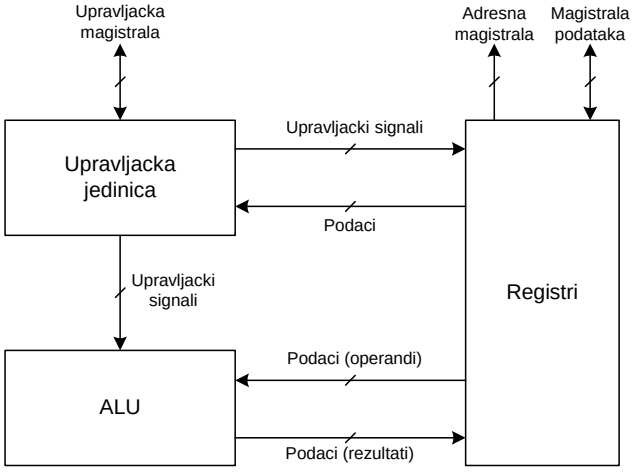
CPU jedinica

CPU jedinica upravlja radom računara tako što izvršava programske instrukcije. CPU izvršava program instrukciju-po-instrukciju pri čemu izvršenje svake pojedinačne instrukcije odgovara jednom instrukcijskom ciklusu. Svaki instrukcijski ciklus uključuje tri glavne aktivnosti (ili faze): (1) pribavljanje instrukcije, (2) dekodiranje instrukcije i (3) izvršenje instrukcije. Pribavljanje instrukcije odgovara čitanju instrukcije iz programske memorije. Da bi se instrukcija izvršila potrebno je obaviti neku specifičnu sekvencu operacija. Kada CPU dekodira instrukciju, on zapravo utvrđuje o kojoj instrukciji se radi, kako bi izabrao korektnu sekvencu operacija koju treba obaviti. Konačno, CPU izvršava instrukciju. Niz operacija koje treba obaviti da bi se izvršila neka instrukcija, se razlikuje od instrukcije do instrukcije. Izvršenje instrukciju može da uključi čitanje podatka iz memorije, upis podatka u memoriju, čitanje iz ili upis podatka u U*I* uređaj, obavljanje aritmetičkih ili logičkih operacija unutar CPU jedinice, ili neku kombinaciju ovih operacija.

Interno, CPU se sastoji iz tri dela (Sl. 3). Registarska sekcija, kao što samo ime kazuje, sadrži registre i magistralu ili neki drugi mehanizam za pristup registrima. Pored registara koji su vidljivi programeri, a čija se imena javljaju kao operandi u pojedinim instrukcijama, registarska sekcija sadrži i registre koji imaju posebnu namenu, a koji nisu direktno vidljivi programeru.

Kao što je već rečeno, u toku faze pribavljanja instrukcije, CPU postavlja adresu instrukcije na adresnu magistralu. CPU poseduje registar koji se zove **programski brojač** u kome se čuva adresa sledeće instrukcije koju treba pribaviti. Pre nego što CPU postavi adresu instrukcije na adresnu magistralu, CPU uzima tu adresu iz programskog bojača. Na kraju faze pribavljanja instrukcije, CPU pruzima instrukciju sa magistrale podataka i smešta je interni registar koji se zove **registar instrukcije**. Ni jedan od ova dva registra nije direktno vidljiv programeru, ali se oba koriste od strane CPU jedinice u toku svakog instrukcijskog ciklusa.

Aritmetičko/logička jedinica ili **ALU**, obavlja aritmetičke i logičke operacije, kao što su sabiranje i AND. Operandi za ove operacije stižu iz registarske sekcije, a rezultat se pamti, takođe, u registarsku sekciju. Tipično, za obavljanje bilo koje operacije u ALU jedinici potreban je jedan taktni ciklus. Na primer, instrukcija *ADD r*1,*r*2,*r*3 je ekvivalentna operaciji: *r*1=*r*2+*r*3, a izvršava se tako što se u jednom taktnom ciklusu pročita sadržaj registara *r*2 i *r*3 i postavi na ulaze ALU, a drugom taktnom ciklusu rezultat prisutan na izlazu ALU jedinice upiše u registar *r*1.



Sl. 3 Interna organizacija CPU jedinice

Kao što CPU upravlja radom računara, tako upravljačka jedinica upravlja radom CPU jedinice. Upravljačka jedinica generiše interne upravljačke signale na način kao kod bilo kog ASIC kola, tako što se ostatak CPU jedinice tretira kao staza podataka. Na primer, ovi upravljački signali iniciraju upis u i čitanje iz registara i konfigurišu ALU za obavljanje željene operacije. Prilikom genisanja upravljačkih signala, upravljačka jedinica koristi podatke iz registarske sekcije. Ovi podaci uključuju kod instrukcije i statusne informacije koje se čuvaju u registarskoj sekciji. Takođe, upravljačka jedinica generiše signale za upravljačku magistralu, kao što su READ, WRITE i IO/M. Da bi pribavio, dekodirao i izvršio instrukciju, CPU mora obaviti niz operacija. Postavljanjem internih i eksternih upravljačkih signala u pravilnom redosledu, upravljačka jedinica upravo inicira CPU i preostali deo računara da obave ovaj niz operacija.

Iako svaka instrukcija uključuje neku specifičnu kombinaciju elementarnih operacija, CPU izvršava sve instrukcije na, u suštini, identičan način. Procedura koja opisuje rad CPU jedinice u toku svakog instrukcijskog ciklusa zove se algoritam ¨*pribavi-izvrši*¨. Mada detalji ovog algoritma mogu značajno da variraju od računara do računara, bazični princip rada ovog algoritma je u sištini isti kod svih računara i može se razložiti na sledeće operacije:

1. Pribavi tekuću instrukciju iz memorije sa adrese na koju ukazuje programski brojač
2. Ako je potrebno, pribavi podatke iz memorije
3. Pripremi se za pribavljanje sledeće instrukcije (uvećaj programski brojač za 1)
4. Dekodiraj i izvrši tekuću instrukciju
    a. Interpretiraj šta tekuća instrukcija znači
    b. Obavi operacije koje su naložene tekućom instrukcijom i eventualno modifikuj memoriju.

Koraci 2. i 4. uključuju detalje koji zavise od načina implementacije CPU jedinice. Takođe, kod nekih CPU jedinica koraci 2., 3. i 4. mogu da slede u nekom drugačijem redosledu. Nakon koraka 4., CPU se vraća na korak 1. i ponavlja istu proceduru, ali sada za sledeću instrukciju.

Opisana varijanta algoritma pribavi-izvrši odgovara CISC (*Complex Instruction Set Computer*) modelu CPU jedinice. Kod ovog modela, svaka od faza u izvršenju instrukcije može trajati jedan ili više taktnih ciklusa, a operandi adresirani instrukcijom se mogu uzimati kako iz internih registara tako i direktno iz memorije. Slično, rezultat izvršenja instrukcije može biti smešten bilo u interni registar bilo u memoriju. CISC model omogućava realizaciju složenih instrukcija i složenih načina adresiranja.

Za razliku od CISC modela, RISC model CPU jedinice propisuje donekle drugačiji oblik algoritma pribavi-izvrši koji se sastoji iz 5 faza:

1. Pribavi tekuću instrukciju iz memorije sa adrese na koju ukazuje programski brojač i uvećaj programski brojač za 1. (*Instruction Fatch - IF*).
2. Dekodiraj instrukciju i pribavi (prikupi) operande specificirane instrukcijom. Operandi se pribavljaju iz registarskog fajla. (*Instruction Decoding - ID*).
3. Izvrši operaciju specificiranu instrukcijom. (*Execution - EX*)
4. Ako se radi o instrukciji za pristup memoriji (Load/Store), pristupi eksternoj memoriji radi upisa/čitanja podatka. (*Memory Access - MEM*).
5. Upiši rezultat operacije u odredišni registar registarskog fajla. (*Write Back - WB*).

Kod RISC modela sve faze traju tačno jedan taktni ciklus, a operandi se uvek uzimaju iz internih registara. Takođe, rezultat se uvek smešta u registar. Za razmenu podataka sa memorijom koriste se specijalizovane instrukcije, koje prebacuju podatak iz memorije u interni registar (instrukcija tipa *Load*), odnosno podatak iz internog registra u memoriju (instrukcija tipa *Store*). Kod CPU jedinica RISC modela instrukcije su jednostavnije i u mogucnosti su da obavljaju samo proste operacije.

Skup instrukcija

Specifičnosti u izvođenju algoritma ¨pribavi-izvrši¨ zavise od skupa instrukcija koje računarska mašina treba da implementira. *Skup instrukcija* je skup reči (binarnih kombinacija) mašinskog jezika koje je hardver mašine u stanju da interpretira. Celokupan softver koji se obavlja na konkretnoj mašini se u konačnoj instanci može razložiti na instrukcije iz skupa instrukcija. Hardver mašine nije u stanju da izvršava instrukcije koje ne pripadaju skupu instrukcija te mašine. Mada, konceptualno slični, različiti modeli računara zasnovani su na potpuno različitim skupovim instrukcija.

Performamnse rečunarskog sistema

Glavni performansni kriterijum računarskog sistema je vreme izvršenja programa. Kada se porede dva računara, brži (moćniji) je onaj koji je isti program izvrši za kraće vreme. Vreme izvršenja programa, T_p, može se izraziti sledećom jednačinom:

$T_p = (N_c * N_p)/F_c$, gde je

F_c – taktna frekvencija, N_c – broj taktnih ciklusa po instrukciji i N_p – broj instrukcija koje čine program.

Što je taktna frekvencija viša, to će računar biti u stanju da izvrši veći broj operacija u jedinici vremena. Taktna frekvencija zavisi od tehnologije u kojoj je realizovana CPU jedinica. Za izvršenje svake instrukcije potreban je jedan ili više taktnih ciklusa. Računar je brži, što je prosečan broj taktnih ciklusa po instrukciji, N_c, manji. Broj instrukcija koje CPU izvrši u toku trajanja programa, zavisi od kompozicije skupa instrukcija konkretne CPU jedinice. Ako skup instrukcija sadrži samo proste instrukcije, kao što je to slučaj kod RISC računara, biće potreban veći broj instrukcija za realizaciju istog posla, nego ako skup instrukcija sadrži složenije instrukcije, kao što je to slučaj kod CISC računara.

Dakle, pri konstantnoj taktnoj frekvenciji, performanse računara zavise od proizvoda vrednosti parametara N_c i N_p. Ova dva parametra su međusobno zavisni, jer ako želimo da samnjimo N_c nećemo moći realizovati složene instrukcije, pa će N_p biti veće. Ako želimo da smanjimo N_p biće neophodno u skup instrukcija uvrstiti i složene instrukcije, što će neminovno povećati N_c. Kod RISC računara, naglasak je na smanjenju parametra N_c, na račun povećanja parametra N_p. Kod CISC računara, koristi se suprotan pristup, tj. teži se minimizaciji parametra N_p na račun povećanja parametra N_c.

Opis CPU jedinice, kako je izložen u ovom poglavlju, je nepotpun. Savremeni računari poseduju još mnoštvo drugih, često veoma složenih funkcija i karakteristika koje doprinose poboljšanju njihovih performansi. Jedan od takvih mehanizama je, na primer, instrukciona protočnost koji omogućava da CPU pribavlja narednu instrukciju u isto vreme dok izvršava tekuću.

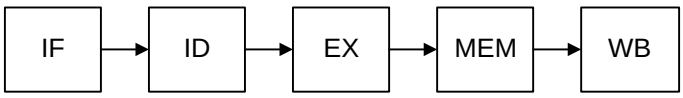
Instrukciona protočnost

Instrukcijski ciklus traje više taktnih ciklusa (mašinskih cilusa), pri čemu se u svakom mašinskom ciklusu obave aktivnosti karakteristične za jednu fazu. Kod klasične organizacije

CPU-a, CPU ne prelazi na izvršenje naredne instrukcije dok u potpunosti ne završi tekući (tj. instrukcijski ciklusi se ne preklapaju - sekvencijalno izvršenje instrukcija):

Instrukcija *n*					Instrukcija *n*+1				
IF	ID	EX	MEM	WB	IF	ID	EX	MEM	WE

Treba uočiti, da je kod sekvencijalnog izvršenja instrukcija efektivno iskorišćenje resursa CPU-a nisko, tj. dok traje jedna faze, resursi koji se koriste u nekoj drugoj fazi su neiskorišćeni. Na primer, ALU jedinica se koristi samo u toku faze izvršenja (EX), dok je u preostalim fazama neiskorišćena. Međutim, ako je CPU tako projektovan da svaku fazu obavlja fizički poseban stepen, onda se efektivna brzina izvršenja instrukcija može značajno povećati korišćenjem tehnike koja se zove protočnost (*pipelining*). Kod protočnog izvršenja instrukcija, svi stepeni su upošljeni u svakom mašinskom ciklusu, tako što svaki stepen radi na izvršenju različite instrukcije:



Instrukcijski ciklus	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10
n	IF	ID	EX	MEM	WB					
n+1		IF	ID	EX	MEM	WB				
n+2			IF	ID	EX	MEM	WB			
n+3				IF	ID	EX	MEM	WB		
n+4					IF	ID	EX	MEM	WB	
n+5						IF	ID	EX	MEM	WB

Na primer, u trenutku t5, stepen IF pribavlja instrukciju n+4, stepen ID dekodira instukciju n+3, stepen EX izvršava instrukciju n+2, stepen MEM obavlja pristupa memoriji za instrukciju n+1 i stepen WB upisuje rezultat instrukcije n u registarski fajl.

Kod protočne obrade, u idealnom slučaju, CPU svakog mašinskog ciklusa završi jednu instrukciju. To znači da je kod ovakve CPU jedinice parmetar N_c=1, što odgovara ubrzanju za faktor 5 u odnosu na sekvencijalno izvršenje. Međutim, idalno ubrzanje nije moguće postići u realnim uslovima. Razlog je pojava tzv. *hazarda*. To su sitacije kada izvršenje jedne instrukcije biva privremeno zaustavljeno zato što operandi koje ona zahteva još uvek nisu izračunati od strane prethodne instrukcije. Razmotrimo sledeću programsku sekvencu:

n+1) A=B+C

n+2) D=A*E

n+3) F=G*H

kod koje se javlja situacija da se rezultat instrukcije n+1 koristi kao operand instrukcije n+2. S obzirom da se rezultat instrukcije upisuje u registarski fajl tek u poslednjoj fazi (WB), dekodiranje instrukcije n+2 mora biti odloženo sve dok se instrukcija n+1 ne završi:

Instruction	1	2	3	4	5	6	7	8	9	10	11
n	IF	ID	EX	MEM	WB						
n+1		IF	ID	EX	MEM	WB					
N+2			IF	XXX	XXX	XXX	ID	EX	MEM	WB	
N+4							IF	ID	EX	MEM	WB

U nekim slučajevima, hazardi se mogu izbeći pažljivim struktuiranjem programske sekvence. Npr. ako se polazna sekvenca preuredi na sledeći način:

n+1) A=B+C

n+2) F=G*H

n+3) D=A*E

do hazarda ne dolazi, jer su konfliktne instrukcije (A=B+C i D=A*E) vremenski "razmaknute".

Opisani tip hazarda se naziva "hazard po podacima" (*data hazard*). Pored ovog tipa hazarda mogu se javiti drugi tipovi: strukturni hazardi i upravljački hazardi.

Superskalarno izvršenje instrukcija

Dodatno ubrzanje izvršenja programa postiže se tehnikom koja se zove *superskalarno* izvršenje. U ovom slučaju CPU istovremeno pribavlja i **paralelno** izvršava dve ili više instrukcija:

Instrukcijski ciklus	t1	t2	t3	t4	t5	t6	t7
n	IF	ID	EX	MEM	WB		
n+1	IF	ID	EX	MEM	WB		
n+2		IF	ID	EX	MEM	WB	
n+3		IF	ID	EX	MEM	WB	
n+4			IF	ID	EX	MEM	WB
n+5			IF	ID	EX	MEM	WB

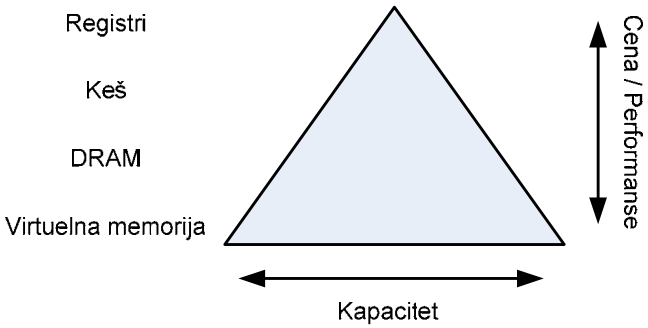
Kod superskalarnog načina rada, pored hazarda koji su karakteristični za protočno izvršenje, javljaju se i novi tipovi hazarda, što dodatno usložnjava upravljačku logiku CPU jedinice.

Protočna i superskalarna obrada se koristi kod većine savremenih mikroprocesora opšte namene.

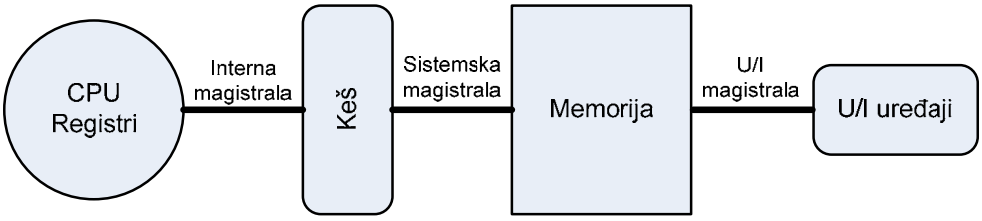
Organizacija memorijskog podsistema

Glavna (operativna) memorija računarskih sistema, koji zahtevaju veliku količinu memorije, realizuje se kao DRAM (dinamički RAM). DRAM obezbeđuje optimalan odnos između kapaciteta i cene memorije. Međutim, vreme pristupa DRAM-u je reativno veliko, tj. značajno duže od trajanja mašinskog ciklusa CPU-a. CPU je u čvrstoj sprezi sa memorijom; iz memorije čita program i čita i upisuje podatke. Ukoliko bi CPU prilikom svakog obraćanja memoriji morao da čeka na spru memoriji, brzina rada sistema bi bila značajno redukovana.

Da bi se ovaj problem rešio (ublažio) pribegava se tzv. hijerarhijskoj organizaciji memorije:



Na vrhu hijerarhije, najbliže CPU-u, nalaze se registri iz registarskog fajla. CPU pristupa registrima direktno, bez kašnjenja. Kod savremenih mikroprocesora, broj registara se kreće u granicama od desetak do nekoliko stotina. Na sledećem nivou je tzv. keš (*cache*) memorija. To je SRAM (statički RAM) kapaciteta tipično 64-512KB i vremenom pristupa od 10ns. Sledeći nivo je zauzet glavnom DRAM memorijom (tipičan kapacite 64-265MB i vreme pristupa od 100ns). Na najnižem nivou je sekundarna memorija (hard-disk), sa kapacitetom od više desetina GB i vremenom pristupa od 5ms.



Sa tačke gledišta programa (programera), keš memorija je "nevidljiva" (transparentna). U keš memoriji se čuvaju nedavno korišćeni podaci i instrukcije. Kada CPU zahteva podatak/instrukciju koja se trenutno nalazi u kešu, CPU se obraća brzom kešu, a ne sporoj memoriji, tako da je vreme pristupa efektivno skraćeno. Ako se traženi podatak/instrukcija ne nalazi u kešu, tada se CPU obraća sporoj memriji (što uključuje i privremeno zaustavljanje CPU-a). Pri tome, podatak/instrukcija koja je uzeta iz memorije smešta se i u keš, sa "nadom" da će uskoro ponovo biti potrebna. Takođe, kod prenošenja podatka iz memorije u keš, ne prenosi se samo jedan podatak, već celokupan blok podataka (tipična veličina do 1KB) kome pripada traženi podatak. Ovakva organizacija daje dobre rezultate s obzrom da se kod najvećeg broja programa uočavaju tzv. *vremenska* i *prostorna lokalnost*. Ako je CPU izvršio instrukciju sa adrese A, velika je verovatnoća da će u bliskoj budućnosti ponovo izvršiti tu istu instrukciju (zbog prgramskih petlji). Slično, ako je CPU pristupio podatku sa adrese P, velika je verovatnoća da će uskoro pristupiti i podatku sa adrese P+1 (nizovi podataka).

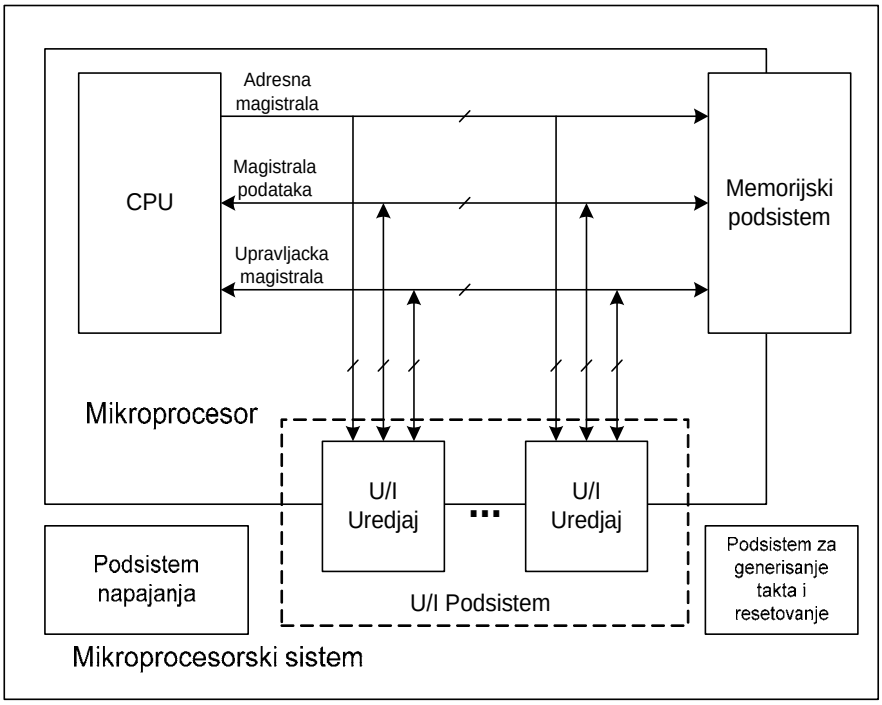
Hijerarhijska organizacija memorije omogućava da ukupna cena memorije bude određena memorijom najniže cene (DRAM), a vreme pristupa memorijom najveće brzine (keš). Napomenimo, da se manipulacija kešom vrši hardverski (automatski), tako da CPU (a time ni programer) nije "svestan" njenog prisustva.

Keš memorija je standardni sastavni deo savremenih mikroprocesora opšte namene.

Pojam mirkorprocesora

Mikroprocesor je integrisano kolo (IC) koje u sebi sadrži CPU zajedno sa dodatnim hardverom koji omogućava direktnu spregu CPU sa drugim resursim računarskog sistema (memorija, U/I uređaji) (Sl. 4). Osim CPU jedinice, mikroprocesor može sadržati i izvesnu količinu memorije i neke U/I uređaje kao što su tajmeri i brojači događaja, komunikacioni kontroleri i sl. koji se koriste kod velikog broja primena.

Mikroprocesorski sistem (MPS) je računarski sistem realizovan na bazi mikroprocesora. Pored mikroprocesora MPS sadrži podsistem napajanja i podistem za generisanje takta i resetovanje. Takođe, u zavisnosti od zahteva konkretne primene, MPS može sadržati, u vidu zasebnih komponenti, dodatnu memoriji i specifične U/I uređaje.



Sl. 4 Organizacija mikroporcesorskog sistema.

Klasifikacija mikroprocesora

Danas, na tržistu postoji veliki broj tipova mikroprocesora, koji se razlikuju po sepenu integracije, brzini rada, unutrašnjoj strukturi, broju i raznovrsnosti internih resursa (registri, ALU, tajmeri..), veličini memorije koju mogu da adresiruju. Generalno, savremeni mikoprocesori se mogu svrstati u sledeće tri kategorije:

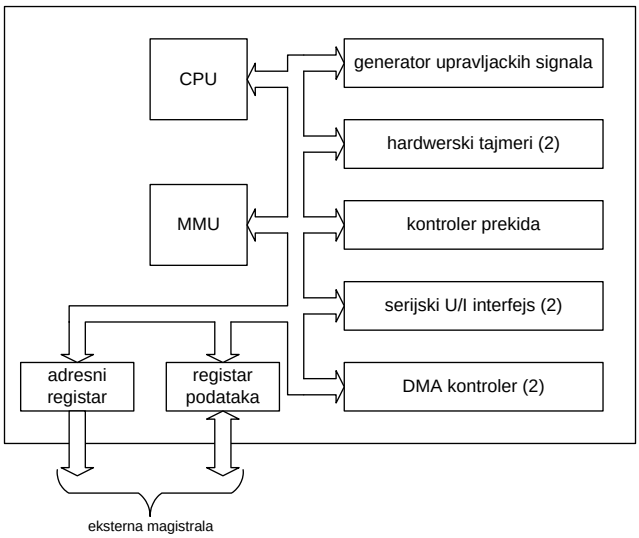
1. **Mikroporcesori** (u užem smislu)
    a. Mikroprocesori opšte namene
    b. Mikroprocesori visoke integracije
2. **Mikrokontroleri**
    a. Mikrokontroleri na čipu
    b. Mikroračunari na čipu
3. **Specijalizovani procesori**
    a. DSP procesori
    b. Paralelni procesori

Mikroprocesori opšte namene

Mikroprocesori opšte namene predstavljaju najbrojniju pojedinačnu grupaciju mikroprocesora. U ovu grupu spadaju mikroprocesori kao što su Intelova familija 80X86, Motorolina familija 68000 itd. To su 16/32/64-bitni mikroprocesori prevashodno namenjeni ugradnji u računare opšte namene (PC mašine, radne stanice). Izrađuju se u vrhunskoj tehnologiji visokog stepena integracije i rade na vrlo visokim frekvencijama. Koriste napredne tehnike za ubrzanje izvršenja programa, kao što su protočnost, superskalarno izvršenje. Obezbeđuju visoke performase za široku klasu zadataka. Sadrže specijalizovane resurse kao što je keš memorija, MMU (*memory management unit*), matematički koprocesor. Pojedini mikroprocesori poseduju direktnu podršku za multitasking. U poređenju sa ostalim kategorijama mikroprocesora, mikroprocesori opšte namene su najmoćniji, ali u isto vreme i najbrže zastarevaju.

Mikroprocesori visoke integracije

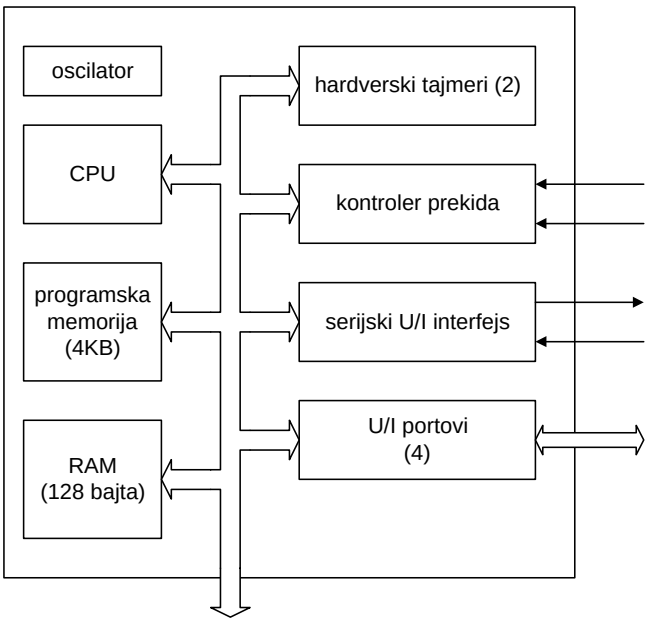
Kod ovog tipa mikroprocesora, veći broj standardnih elemenata MPS-a (tajmer, kontrololer prekida, serijski U/I interfejs, DMA konktroler,..) je integrisan zajedno sa CPU-on u isti čip. Primeri mikroprocesora ovog tipa su Intel-ovi mikroprocesori 80186 i 80188, Hitachi 64180 (Sl. 5). Treba napomenuti, da mikroprocesori visoke integracije ne sadrže memoriju (RAM ili ROM). Mada je cena mikroprocesora visoke integracije viša u poređenju sa odgovarajućim mikroprocesorom opšte namene, integracijom većeg broja dodatnih komponenata u jedinstveni čip, tipično se smanjuje ukupna cena sistema, kao i povrđina štampane ploče koja je nephodna za realizaciju MPS-a. Takođe, smanjena je cena proizvodnje i testiranja MPS-a.



Sl. 5 Mikroprocesor visoke integracije - Hitachi 641800.

Mikroračunari na čipu

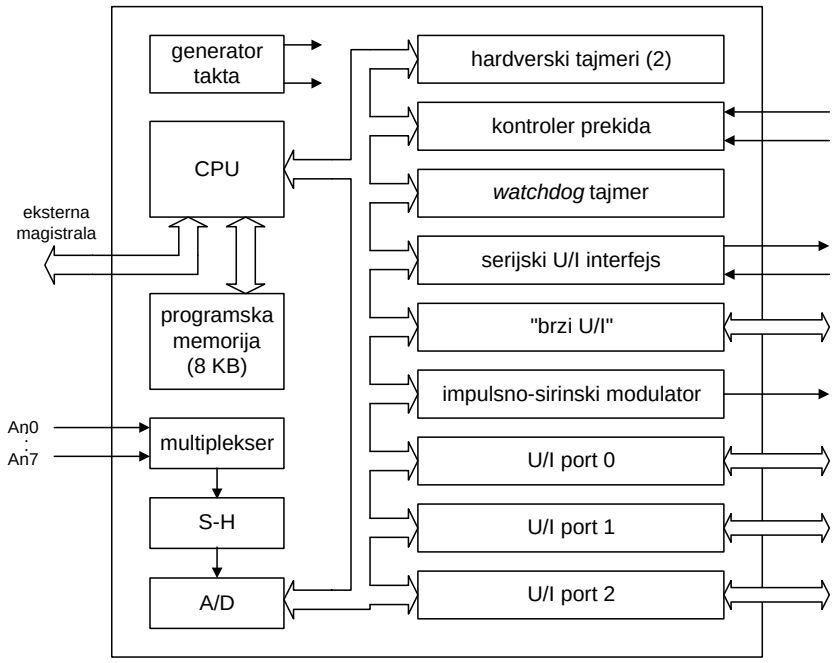
Arhitektura mikroračunara na čipu je takva da se celokupan MPS može realizovati bez dodatnih komponenata (čipova). To znači da mikroprocesori ovog tipa sadrže i izvesnu količinu programske memorije (ROM) i memorije podataka (RAM). U poređenju sa mikroprocesorima visoke integracije, računarski resursi (CPU, ALU) mikroračunara na čipu su manje složenosti (8/16-bitna CPU, redukovani skup instukcija). Tipičan primer mikroračunara na čipu je Intelov mikroprocesor 8751 (Sl. 6). Ovo je 8-bitni mikroprocesor koji u sebi sadrži 4KB EPROM-a (za smeštanje programa) i 128 bajta RAM-a, dva 16-bitna hardverska tajmera/brojača, jedan serijski U/I interfejs, četiri 8-bitna U/I porta (koji imaju ulogu paralelnog U/I interfejsa), i podržava dva eksterna prekida. Arhitektura mikroračunara na čipu je veoma fleksibilna i namenjena je realizaciji MPS male složenosti. Napomenimo da se u MPS zasnovan na mikroračunaru na čipu, ukoliko je to potrebno, može ugraditi dodatna memorija (eksterni RAM i ROM, tipično do 64KB), kako i dodatne periferne komponente.



Sl. 6 Arhitektura mikroračunara na čipu 8751.

Mikrokontroleri na čipu

Mikrokontroleri na čipu su derivati mikroračunara na čipu namenjeni ugradnji u specijalizovane elektronske uređaje. Slično mikroračunarima na čipu, oni su opremljeni svim hardverskim resursima potrebnim za realizaciju MPS-a za primenu kod sistema za akviziciju (prikupljanje) podataka, sistema za upravljanje i nadzor, aparata za domaćinstvo, merno-instrumentacionoj opremi, telekomunikacionim uređajima... Tipičan primer mikrokontrolera na čipu je Intelov mikroprocesor 80196 (Sl. 7).



Sl. 7 Arhitektura mikrokontrolera na čipu 80196.

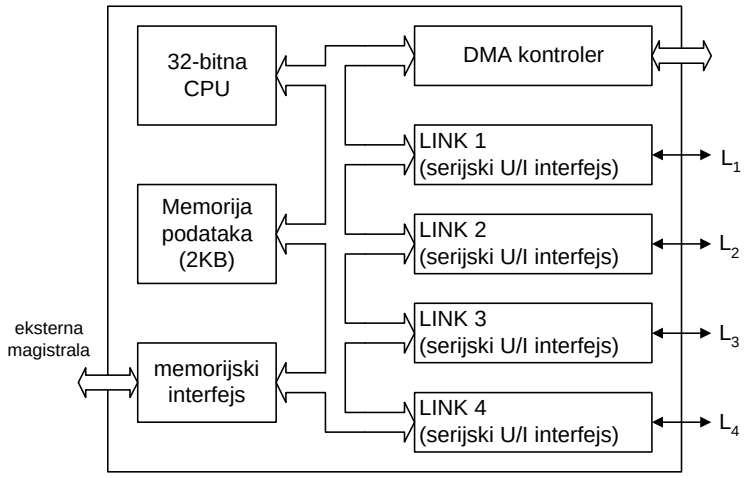
Unutrašnja hardverska strukutura, skup naredbi, broj i raznovrsnost integrisanih perifernih interfejsa, optimizovani su za primenu kod "brzih" sistema za rad u realnom vremenu. Sa hardverske tačke gledišta, glavna razlika u odnosu na prethodnu kategoriju mikroprocesora odnosi se na U/I podsistem. U okviru U/I podsistema implementiran je sistem za obradu analognih signala, koga čini 10-bitni A/D konvertor, kolo uzorkovanja i držanja (*semple and hold - S-H*) i 8-kanalni analogni multiplekser, što omogućava digitalizaciju sa 10-bitnom rezolucijom do 8 ulaznih analognih signala. Standardni digitalni ulazi/izlazi obezbeđeni su preko tri 8-bitna U/I porta. Dodatno mikroprocesor sadrži sklop za generisanje širinsko-impulsno modulisanih signala. Podsistem "Brzi U/I" obezbeđuje podršku za precizno merenje frekvenciji i periode ulaznih digitalnih signala. Prekidni podsistem obezbeđuje podršku za čak 21 izvor prekida, od toga su 8 eksterna. CPU je 16-bitni, a u čipu je integrisano 8KB programske memorije i 232 bajta RAM-a. Treba napomenuti, da se mikrokontroleri na čipu proizvode u velikom broju varijanti koje se razlikuju po "kombinaciji" ugrađenih elemenata, količini memorije, tipu memorije (OTP, EPROM) i sl. Tipično, proizvođači mikroprocesora nude "familije" mikrokontrolera na čipu koje se sastoje od većeg broja (i do nekoliko desetina) različitih varijanti istog bazičnog tipa mikroprocesora (programski kompatabalni). Na taj način, projektantu se pruža mogućnost da izabere varijantu mikroproceosora koji je optimalan sa stanovišta njegove ciljne primene.

Paralelni procesori

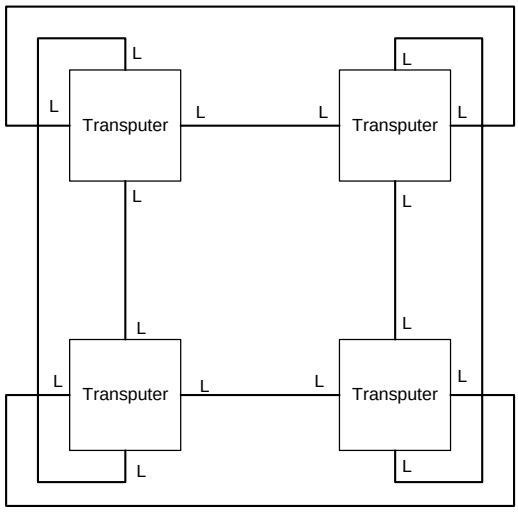
Standardni računarski sistemi se karaktrišu tzv. *sekvencijalnim* načinom rada, tj. u stanju su da izvršavaju samo jedan *tok* instrukcija (izvršavaju program instrukciju-po-instrukciju).

Performase sekvencijalnog računarskog sistema određene su brzinom sa kojom procesor izvršava tok instrukcija (tj. trajanjem instrukcijskog ciklusa). S obazirom da je trajanje instrukcijskog ciklusa određeno maksimalnom taktnom učestanošću, mogućnosti za povećanje performansi sekvencijalnih procesora su ograničene. Alternativno, performanse računarskog sistema se mogu poboljšati ugradnjom u sistem više od jednog procesora, koji će istovremeno (paralelno) i koordinirano izvršavati različite delove istog programa (tj. zajdnički raditi na rešavanju istog problema). Ovakvi sistemi se zovu paraleni procesori, ili paralaleni računarski sistemi. U domenu hardvera, ključni problem koga treba rešiti a koji je u vezi sa koordiniranim radom procesora je obezbeđivanje podrške za brzu interprocesorsku komunikaciju (tj. razmenu podataka između procesora). U domenu softvera, glavni problem se odnosi na optimalnu podelu programa na paralelne programske sekcije i dodelu sekcija na izvršenje pojedinačnim procesorima.

Jedan od mikroprocesora koji je namenski projektovan za ugradnju u paralelne računarske sisteme je *Transputer*. Bazična struktura ovog mikroprocesora, kako i način integracije Transputer-a u paralelni računarski sistem, tipa polje procesora (*processor array*) prikazan je na slici 11.



(a)



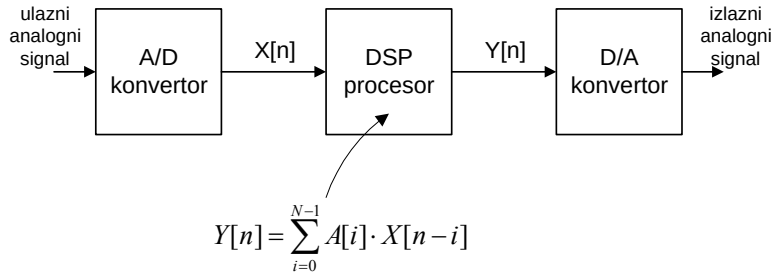
(b)

Sl. 8 Transputer. (a) unutrašnja struktura; (b) polje procesora.

Ključna karakteristika Transputer-a je hardverska podrška za četiri brze serijske komunikacione linije (LINK 1-4), koje omogućavaju brzinu prenosa podataka od 20Mbit/s. Ove linije se koristi kao komunikacioni kanali za spregu procesora u strukturu procesorskog polja.

DSP procesori

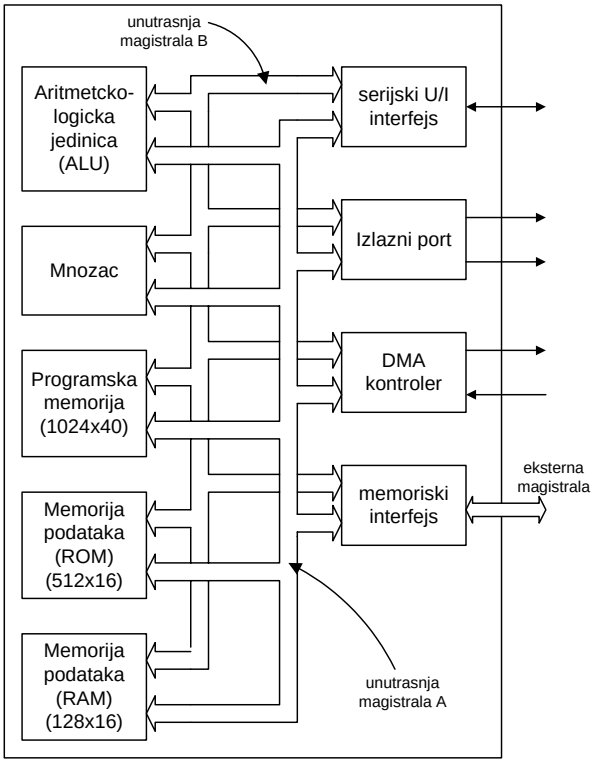
DSP (*Digiral Signal Processor*) su mikroprocesori ili mikroračunrai čiji su hardver, softver i skup instrukcija optimizovani za brza numerička izračunavanja nad digitalnim podacima koji reprezentuju analogne signale. Oblasti primene DSP procesora su: obradu govornog i video signala, obradu signala kod radar, sonara, instrumentacione tehnike, upravljačkih i telekomunikacionih sistema. Za obradu signala, u prošlosti, korišćena je analogna tehnika (pasivne i aktivne električne mreže). Međutim, zbog ograničenih mogućnosti analognog procesiranja signala, projektanti su prešli na digitalnu obradu, koja pored veće fleksibilnosti omogućava realizaciju daleko složenijih tipova obrade signala. Tipične primene DSP su: digitalni filtri i FFT (brza furijeova transforamcija). Princip digitalne obrade signala prikazan je na Sl. 9. Uz pomoć A/D konvertora, ulazni, analogni signal se digitalizuje tj. konvertuje u niz brojnih vrednosti, X[n], n=1,2,… . Pri tome, frekvencija odmeravanja je jednaka dvostrukoj graničnoj frekvenciji ulaznog signala. Digitalni filtar, računskim putem, ulaznu sekvencu X[k] transformiše u izlaznu sekvencu brojnih vrednosti Y[k]. Konačno, uz pomoć D/A konvertora, sekvenca brojnih vrednosti Y[k] se konvertuje u izlazni, analogni signal.



Sl. 9 Realizacija digitalnog filtra pomoću DSP procesora.

Tipična operacija koja se sreće kod većine algoritama iz oblasti digitalne obrade signala je suma proizvoda. Da bi se obezbebilo brzo izračunavanje, arhitektrura DSP procesora je zasnovana na aritmetičkoj jedinici (kombinacija sabirača i množača), koja ima mogućnost direktnog izračunavanja izraza A=A+B*C. Takođe, skup instrukcija DSP procesora sadrži specijalizovane instrukcije za efikasnu manipulaciju podacima. Izračunavanje se obavlja nad celobrojnim podacima i podacima u formatu fikasnog zareza, dok kod pojedinih tipova DSP procesora postoji mogućnost direktne manipulacije podacima dostupnim u formatu pokretnog zareza.

Tipičan predstavnik DSP procesora je mikroprocesor TMS320 firme Texas Instruments (Sl. 10). Za prenos podataka između internih funkcijskih blokova mikroprocesora koriste se dve unutrašnje magistrale, što omogućava da se u jednom taktnom ciklusu prenesu dva podatka između bilo koja dva bloka. Mikorprocesor ima mogućnost sprege sa eksternom memorijom, međutim, optimalne performanse se postižu u slučaju kada je celokupan programa i podaci smešteni u unutrašnjoj memoriji.



Sl. 10 Unutrašnja struktura DSP procesora (TMS320).

Elementi mikroprocesorskih sistema

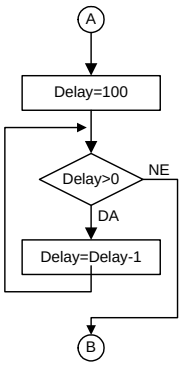
Mikroprocesor, ili jedan od njegovih derivata, je sastvani deo svakog mikroprocesorskog sistema (MPS). Međutim, da bi bio operativan, MPS pored mikroprocesora mora da sadrži elemente kao što su: podsistem napajanja, memorija, tajmeri, komunikacioni kontroleri, kontroleri perifernih jedinica, itd. Memorija služi za smeštanje programa (programska memorija) koga mikroprocesor izvršava, i podataka (memorija podataka) sa kojima mikroprocesor manipuliše u toku izvršenja progama. Mikroprocesor u spoju sa programskom i memorijom podataka, zajedno sa adresnim dekoderom, koji obezbeđuje selekciju elementa kome mikroprocesor pristupa, predstavlja "jezgro" (minimalnu konfiguraciju) svakog MPS. Mikroprocesorski sistem minimalne konfiguraciju u stanju je da izvršava program, ali ne poseduje mogućnost bilo kakve interakcije sa okruženjem niti ima "pojam" o protoku vremena. Da bi se obezbedile ove mogućnosti u MPS je neophodno, u zavisnosti od potrebe, ugraditi dodatne hardverske elementi.

Sat realnog vremena

Informacija o protoku vremena je od ključne važnosti kod mikroprocesorskih sistema koji upravljaju fizičkim procesima (to su tzv. sistemi za rad u realnom vremenu). Kod takvih sistema, za obezbeđivanje precizne vremenske reference koriti se sat realnog vremena. Sat realnog vremena se realizuje kao generator signala pravougaonog talasnog oblika stabilne učestanosti (*clock* signal). Trajanje periode *clock* signala se zove "tick" i predstavlja osnovnu vremensku jedinicu u MPS. Tipično, trajanje tika iznosi 0.1ms, 1ms, ili 10ms. Brojanjem tikova, mikroprocesor dobija informaciju o vremenu proteklom od trenutka startovanja (tj. resetovanja). Ovaj signal se obično koristi kao signal prekida mikroprocesora. Svaki impuls *clock* signala inicira prekid mikroprocesora, a u okviru odgovarajuće prekidne rutine, programska promenjiva, rezervisana za tu namenu, se uvećava za jedan.

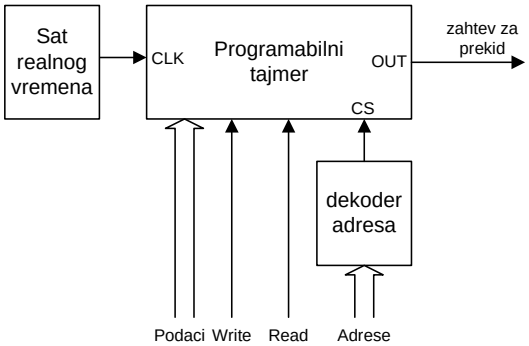
Hardverski tajmeri i brojači događaja

Kod MPS, hardverski tajmeri se koriste za merenje dužih vremenskih intervala ili realizaciju funkcije vremenskog kašnjenja. Vremensko kašnjenje se često koristi kod realizacije upravljačkih zadataka kada je potrebno ostvariti vremenski razmak između dve uzastopne aktivnosti. Funkciju vremenskog kašnjenja je moguće realizovati i softverski, ukoliko se poznaju vremena izvršenja mašinskih instrukcija. Princip softverske rezalizacije vremenskog kašnjenja ilustrovan je na Sl. 11.



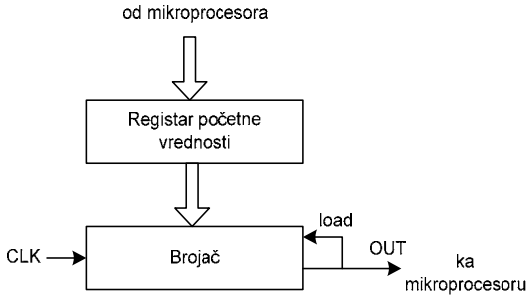
Sl. 11 Softverska realizacija funkcije vremenskog kašnjenja.

Za realizaciju vremenskog kašnjenja koristi se programska petlja sa unapred zadatim brojem ciklusa (prolazaka). Poznavanjem vremena izvršenja mašinskih instrukcija koje realizuju programsku petlju moguće je odrediti ukupno vreme izvršenja petlje. Međutim, ovakvo rešenje ima sledeće nedostatke: (1) nije univerzalno; promenom taktne frekvencije mikroprocesora, neophodno je menjati softver, tj. vrednost konstante koja reguliše broj prolaska kroz petlju. (2) ukoliko se u toku izvršenja petlje javi prekid, vreme izvršenja petlje biće produženo za iznos vremena izvršenja prekidnog programa; (3) u toku izvršenja petlje, mikroprocesor je onemogućen da obavlja druge aktivnosti. Iz ovih razloga, kod sistema koji ne poseduju sat realnog vremena, za realizaciju vremensog kašnjenja koriste se hardverski tajmeri. Čak i slučajevim kada u MPS postoji sat realnog vremena, hardverski tajmeri pružaju veću fleksibilnost prilikom realizacije vremenskih kašnjenja. Hardverski tajmeri se realizuju kao programabilna periferijska kola, kod kojih je moguće softverski podesiti kako vremenske parametre tako i način rada. (Tipičan primer komponente ovog tipa je kolo sa standardnom oznakom 8254 (programabilni interval-tajmer), koga čine tri nezavisna 16-bitna programabilna brojača (tajmera)). Način ugradnje programabilnog hardverskog tajmera u MPS prikazan je na Sl. 12. Posredstvom sistemske magistrale, mikroprocesor ima pristup internim registrima kola u cilju zadavanja početnog stanja internog brojača, a kada brojač dođe u stanje 0, aktivira se izlazni signal OUT što inicira zahtev za prekid mikroprocesora.



Sl. 12 Način ugradnje programabilnog tajmera u MPS.

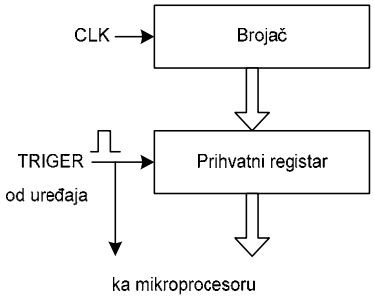
Tajmer koristi princip rada delitelja učestanosti (Sl. 13). Interno, tajmer sadrži dva registra: registar početne vrednosti i brojački registar. Registar početne vrednosti je dostupan za upis od strane mikroprocesora, a njegov sadržaj određuje trajanje vremenskog perioda. Brojački registar broji unazad i taktuje se spoljnim taktnim signalom (CLK) fiksne frekvencije. Kada brojač dostigne vrednost nula, generiše se signal OUT koji se vodi ka mikroprocesoru. Isti signal inicira paralelni upis u brojač sadržaja registra početne vrednosti. Dakle, ako je sadržaj registra početnog stanja N, tada će perioda signal OUT biti N puta duža od periode taktnog signala.



Sl. 13 Princip rada tajmera

Marker vremenskih intervala

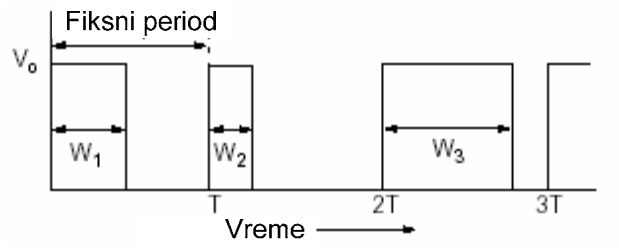
Konfiguracija sa Sl. 14 odgovara markeru vremenskih intervala. Ovakvo kolo omogućava precizno registrovanje vremenskog trenutka nekog događaja. Marker vremenskih intervala se sastoji iz brojača, koji se taktuje signalom CLK stabilne učestanosti i prihvatnog registra. Tekuće stanje ukazuje na proteklo vreme (izraženo celobojnim umnoškom perode taktnog signala) od trenutka resetovanja brojača. Pod dejstvom impulsa TRIGER, koji je posledica nekog spoljnjeg događaja, u prihvatni registar se upisuje tekuća vrednost brojača i inicira prekid mikroprocesora. Naknadnim očitavanjem prihvatnog registra, mikroprocesor može doći do informacije o vremenu kada se događaj desio.



Sl. 14 Marker vremenskih intervala

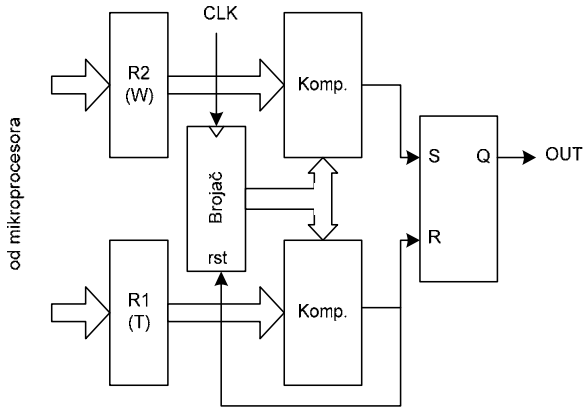
Impulsno-širinski modulator

Impulsno-širinski modular je kolo koje na svom izlazu generiše periodičnu povorku impulsa zadatog trajanja (Sl. 15).



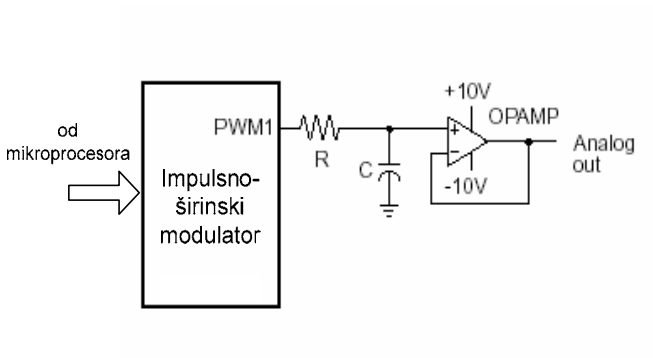
Sl. 15 Impulsno-širinski modulisani signal.

Perioda impulsa, T, je fiksna, dok je trajanje impulsa, W, srazmerno upravljačkoj veličini. Kod impulsno-širinskih modulatora sa analognim ulazom trajanje impulsa je srazmerno ulaznom naponu, dok je kod digitalnih impulsno-širinskih modulatora srazmerno zadatoj digitalnoj reči. Princip rada digitalnog impulsno-širinskog modulatora ilustrovan je na Sl. 16. Kolo se sastoji iz brojača koji se pobuđuje taktnim signalom, dva prihvatna registra i dva komparatora. Sadržaj registra R1 određuje periodu, a sadržaj registra R2 trajanje impulsa izlaznog signala.



Sl. 16 Struktura digitalnog impulsno-širinskog modulatora.

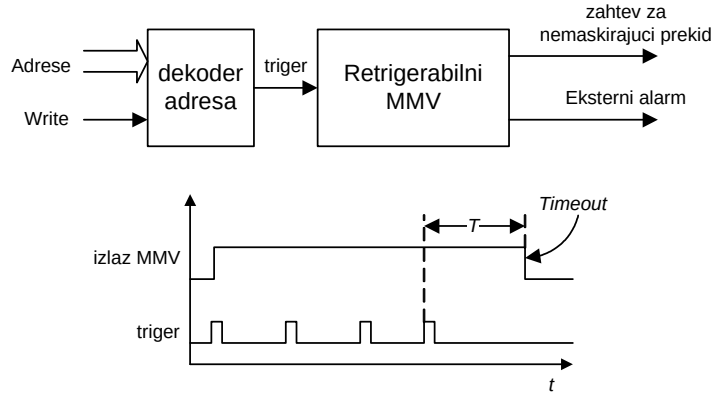
Impulsno-širinski modulator se može koristiti za realizaciju D/A konvertora, kao na Sl. 17. Izlazni signal impulsno-širinskog modulatora se filtrira pomoću RC kola, čime se odstranjuju sve prostoperiodične komponente i propušta samo jednosmerna komponenta koja je srezmerna srednoj vrednosti izlaznog signala. Srednja vrednost izalaznog signala srezmerna je odnosu trajanja impulsa i periode. Dakle, promenom trajanja impulsa direktno se reguliše izlazni napon.



Sl. 17 Realizacija D/A konvertora pomoću impulsno-širinskog modulatora.

Watchdog tajmer

Watchdog je uobičajen koncept koji se koristi kod visoko-pouzdanih mikroprocesorskih sistema za detekciju i signalizaciju neispravnog ponašanja programa. *Watchdog* tajmer se realizuje kao retrigerabilni monostabilni multivibrator (MMV), kod koga se trigerovanje obavlja pod programskom kontrolom. MMV je impulsno kolo koje može biti u jednom od dva stanja: stabilno i kvazistabilno (ili pobudjeno). Pod dejstvom trigerskog (okidnog) impulsa, MMV iz stabilnog prelazi u pobuđeno stanje i ovom stanju ostaje fisksno vreme *T*, a zatim se spontano vraća u stabilno stanje. Retrigerabilni MMV ima osobinu da svaki trigerski impuls koji se javi dok je MMV u pobuđenom stanju resetuje MMV, tj. "vraća" MMV na početak pobuđenog stanja. Da bi retrigerabilni MMV bio stalno u kvazistabilnom stanju, neophodno je da se konstantno pobuđuje okidnim impuslima, ali tako da vremenski interval između svaka dva uzastopna impulsa bude kraći od *T*.

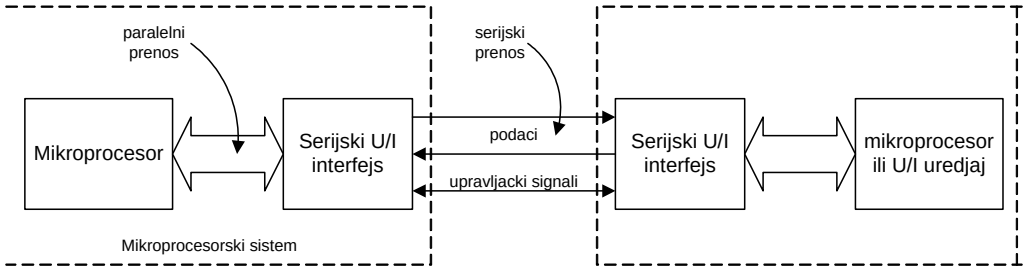


Sl. 18 Watchdog tajmer.

Način ugradnje *watchdog* tajmera u mikroprocesorski sistem prikazan je Sl. 18. *Watchdog* tajmeru je pridružena adresa koja se dekodira uz pomoć dekodera adresa. Uvek kada mikroprocesor izvrši fiktivni upis na adresu *watchdog* tajmera formira se okidni implus koji resetuje MMV. Mikroprocesor ima "obavezu" da periodično, sa periodom manjom od *T*, vrši upis na adresu *watchdog* tajmera. To se postiže tako što programer, nakon što je završio razvoj programa, na pojedinim mestima u programskoj sekvenci ubacuje instrukcije upisa na adresu *watchdog* tajmera, tako da se u slučaju korektnog izvršenja programa okidni impusli generiću regularno, a MMV stalno ostaje u pobuđenom stanju. Ukoliko, zbog otkaza nekog dela hardvera MPS-a ili nekorektnog izvršenja programa, upis izostane u vremenu dužem od *T*, MMV prelazi u stabilno stanje, a izlaz MMV-a se aktivira što inicira zahtev za nemaskirajući prekid mikroprocesora i eksterni alarm. Do nekorektnog rada progama može doći usled greške u programu koja nije otkrivena u toku testiranja programa, a nalazi se u segmentu programa koji se poziva u nekim retkim i specifičnim situacijama u toku rada MPS. Posledica takve greške može biti programski skok na "nepostojeću" instrukciju, tj. lokaciju u programskoj memoriji koja nije zauzeta mašinskim kodom, nakon čega sledi nekontrolisani rad mikroprocesora (kaže se da je program "zalutao"). Do slične situacije može doći ukoliko mikroprocesor, pod dejstvom smetnji iz okurženja, pogešno pročita instrukciju. Zahtev za nemaskirajući prekid inicira proceduru za "oporavak" sistema, čiji je zadatak da postavi sistem u inicajalno i bezbedno stanje. Međutim, ako je razlog otkaza krav mikroprocesora ili neke druge hardverske komponente (kristal, logika za dekodiranje adresa…), oporavak je nemoguć. Iz tog razloga, koristi se i signal eksternog alarma kojim se operater obavećtava o pojavi otkaza.

Serijski U/I interfejs

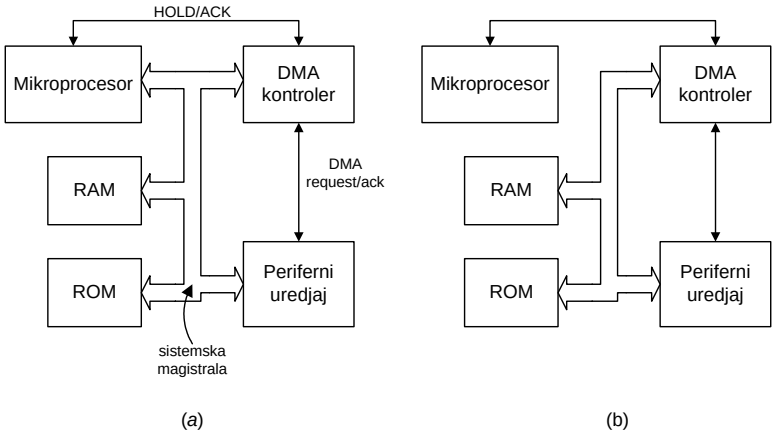
Za razmenu podataka između računara tipično se koristi serijska komunikacija. Takođe, serijska komunikacija se koristi za spregu računara i pojedinih U/I uređaja (terminali, tastature, miš, modem,..). Za podršku serijske komuikcije kod mikroprocesorskih sistema koristi se serijski U/I interfejs. To su specijalizovano kola koja obavljaju konverziju formata podataka iz paralelnog u serijski oblik i obrnuto, proveru ispravnosti prenosa, baferovanje podataka i sl. Tipično kolo ovog tipa je USART (*Universal Asynchronous Receiver Transmitter*). Način ostvarivanja serijske komunikacije pomoću USART-a prikazan je na Sl. 19.



Sl. 19 Serijska komunikacija između mikroprocesorskog sistema i U/I uređaja (ili drugog mikroprocesorskog/računarskog sistema).

DMA kontroler

Kod mikroprocesorski sistema često postoji potreba brzog prenosa velike količine podataka između memorije MPS-a i perifernih uređaja, kao što su diskovi ili interfejsi za serijsku komunikaciju velike brzine prenosa podataka. Ovaj, u suštini, jednostavan zadatak kopiranja podataka, može obavljati mikroprocesor, uzastopnim izvršenjem instrukcija čitanja/upisa. Međutim, ovakvo rešenje ima sledeće nedostatka. Prvo, efektivna brzina prenosa je relativno niska. Naime, da bi mikroprocesor obavo prenos jednog podatka, pored instrukcije čitanja podatka iz memorije (perifernog uređaja) i instrukcije upisa podatka u periferni uređaj (memoriju) treba da izvrši još nekoliko dodatnih instukcija kao bi uvećao adresu pristupa memoriji i proverio da li je prenet zadati broj podataka. Drugo, ukoliko se potreba za ovakvom vrstom prenosa podataka često javlja, efektivno vreme u kome je procesor angažovan na prenosu potadaka moze biti značajno. Da bi se prenos podataka između brzih perfernih uređaja i memorije MPS-a ubrzao i u isto vreme mikroprocesor rasteretio ovog zadatka, koristi se DMA kontroler. Način ugradnje DMA kontrolera u MPS i princip DMA prenosa prikazan je na Sl. 20.



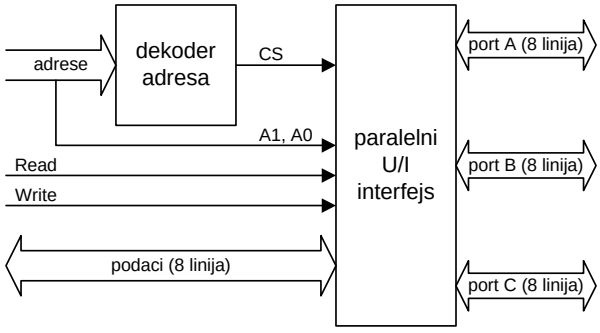
Sl. 20 DMA. (a) normalni režim rada; (b) DMA prenos;

Kada nema potrebe za prenosom podataka, DMA kontroler, gledano sa strane mikroprocesora, vidi se kao i svako drugo programabilno periferno interfejsno kolo, preko skupa internih upravljačkih, statusnih i konfiguracionih registara (Sl. 20(a)). Upisom u ove registre, mikroprocesor programira DMA kontroler za prenos podataka, tako što zadaje početnu adresu oblasti memorije u koju će biti smeštani (tj. iz koje će biti čitani) podaci u toku prenosa, zajedno sa brojem podataka koje treba preneti. Periferni uređaj, u trenutku kada je spreman da preda ili prihvati nove podatke, generiše zahtev DMA kontroleru (tzv. *DMA request*). Nakon toga DMA kontroler preuzima od mikroprocesora pravo korišćenja sistemske magistrale (aktivan je signal *HOLD*) i obavlja zadati prenos podataka tako što direktno, bez ikakvog angažovanja mikroprocesora, pristupa memoriji. Za vreme trajanja prenosa, mirkoprocesor je neaktivan ("zamrznut"), izvršenje programa je zaustavljeno, a linije mikroprocesora za spregu sa sistemskom magistralom su postavljene u stanje visoke impedanse (Sl. 20(b)). Nakon obavljenog prenosa, DMA kontroler deaktivira signal HOLD, mikroproceosor se ponovo priključuje na sistemsku magistralu i nastavlja izvršenje programa.

Paralelni U/I interfejs

Paralelni U/I interfejs se koristi da obezbedi jednostavnu spregu između mikroprocesora i pojedinih tipova perifernih uređaja (štampači, tastature, displej…). Paralelni U/I interfejs obezbeđuje određeni broj U/I linija (tipično, 8, 16, 24, 32), koje se programski mogu konfigurisati kao ulazi ili izlazi (Sl. 21). Nakon izvršene konfiguracije, mikroprocesor ima mogućnost da postavlja stanje izlaznih linija, odnosno da očitava stanje prosutno na ulaznim linijama. Paralelni U/I interfejs se može koristiti za brzu paralelnu komunikaciju sa uređajima kao što je štampač, ali i za direktno upravljanje komponentama kao što su rele, LED indikator, ili za spregu sa senzorima/davačima ON/OFF tipa, kao što su prekidači, tasteri i sl..

Na Sl. 21 je prikazan način ugradnje programabilnog paralelnog U/I interfejsa u MPS. Interno, U/I interfejs sadrži četiri 8-bitna registra: tri 8-bitna dvosmerna registra, koji odgovaraju portovima A, B i C, i upravljački registar preko koga se vrši konfigurisanje portova (U ili I). Izbor internog registra kome se pristupa određen je stanjem adresnih linija A1 i A0.

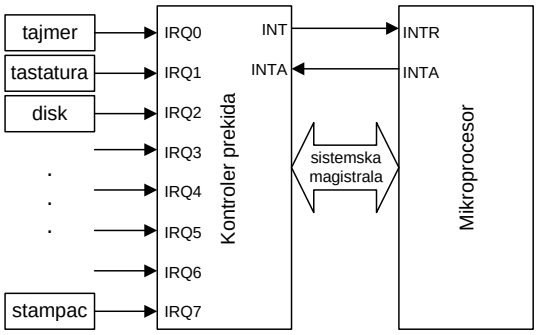


Sl. 21 Paralelni programabilni U/I interfejs.

Programabilni kontroler prekida

Mehanizam prekida je standardna funkcija svih savremenih mikroprocesora. Prekidi se koriste za dojavu asinhronih događaja koji se javljaju unutar mirkroprocesorskog sistema ili u njegovom okruženju, a koji zahtevaju odgovarajuću programsku obradu. Tipično, zahtev za prekid inicira periferni uređaj tražeći, na taj način, da bude "opslužen" od strane mikroprocesora. Na primer, zahtev za prekid inicira serijski U/I interfejs kada primi novi podatak. Pritisak dirke tastature inicira zahtev za prekid. Nakon prijema zahteva za prekid,

mikroprocesor prekida izvršenje programa, pamti status (adresa sledeće instrukcije prekinutog programa, sadržaj internih registart), i prelazi na izvršenje prekidnog potprograma koji je pridružen prihvaćenom zahtevu za prekid. Prekidni potprogrami su obično kraće programske sekvence kojima se "opslužuje" uređaja ili kolo koje je iniciralo prekid (prenos primljenog podatak iz serijskog U/I interfejsa u memoriju, "skaniranje" tastaturu da bi utvrdio koja dirka je pritisnuta…). Nakon završetka prekidnog potprograma, mikroprocesor nastavlja izvršenje prekinutog programa. Tipično, mikroprocesori poseduju mali broj linija za prihvatanje zahteva za prekid (od jedne do maksimalno četiri). Problem se javlja u situacijama kada u MPS postoji više "izvora" prekida od broja raspoloživih "prekidnih" linija mikroprocesora. Ovaj problem se rešava ugradnjom specijalizovanog kola, tzv. programabilnog kontrolera prekida. Način sprege perifernih uređaja i kontrolera prekida, kao i kontrolera prekida i mikroprocesora prikazan je na Sl. 22.



Sl. 22 Sprega kontrolera prekida i mikroprocesora.

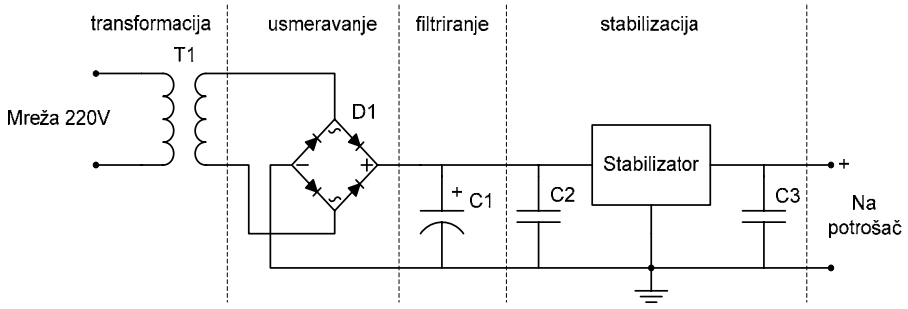
Zadatak kontrolera prekida je da zahtev za prekid upućen od strane nekog od perifernih uređaja (signali IRQ0-7) prosledi mikroprocesoru (signal INT) i da nakon prihvatanja zahteva za prekid od strane mikroprocesora (aktivan je signal INTA), preko sistemske magistrale preda mikroprocesoru informaciju o tome koji od uređaja je inicirao prekid. Pored toga, programabilni kontroler prekida omogućava selektivnu dozvolu/zabranu pojedinih zahteva za prekid (tj. maskiranje prekida), pridruživanje prioriteta svakom od zahteva za prekid (ako je više od jednog uređaja izdalo zahtev za prekid, prihvata se zahtev najvišeg prioriteta).

Napajanje mikroprocesorskih sistema

Za rad mikroprocesorskih sistema koriste se jednosmerni izvori za napajanje koji obezbeđuju napone i jednosmerne struje potrebne za rad mikroprocesora i ostalih komponenti mikroprocesorskog sistema. Za prenosne elektronske uređaje kao izvori za napajanje koriste se baterije i akumulatori, dok se za stacionarne uređaje jednosmerno napajanje obezbeđuje iz komercijalne mreže naizmeničnog napona. Mikroprocesori, kao i druga digitalna integrisana kola, zahtevaju stabilan izvor napajanja koji će obezbediti zahtevani napon koji neće zavisiti kako ni od veličine potrošača tako ni od varijacija napona mreže ili baterije.

Napon napajanja digitalnih integrisanih kola se specificira od strane proizvođača. Standardno se koristi naponi napajanja od 5V, sa dozvoljenim varijacijama između 4.75V i 5.25V. Prekoračenje gornje granice napona napajanja može dovesti do trajnog oštećenja kola, dok napajanje naponom koji je manji od minimalno dozvoljenog može usloviti pogrešan rad kola. Osim napona napajanja od 5V, u sve široj upotorebi je i napon od 3.3V. Napajanje malim naponom obezbeđuje manju potrošnju energije kola, što je naročito bitno kod prenosnih uređaja koji se napajaju baterijom. Pojedine familije digitalnih integrisanih kola izrađenih u CMOS tehnologiji mogu se napajati naponom iz šireg opsega, npr. 2.7-6V. Pri tome, postoji direktna veza između napona kojim se kolo napaja i maksimalne dozvoljene taktne učestalosti. Na primer, kod mikrokontrolera koji se pri naponu napajana od 5V može

taktovati maksimalnom frekvencijom od 20MHz, pri smanjenom naponu napajanja od 2.7V frekvencija taktnog signala ne sme biti veća od 4MHz.



Sl. 23 Realizacija izvora jednosmernog napajanja.

Svako jednosmerno napajanje koje se dobija iz mreže naizmeničnog napona zahteva četiri koraka (Sl. 23):

- *Tranformacija* – uz pomoć tranformatora napon se smanjuje na određenu vrednost.
- *Usmeravanje* – naizmenični napon se pretvara u jednosmerni. Za tu namenu koriste se poluprovodničke diode vezane u most tako da propuštaju pozitivne, a preusmeravaju negativne poluperiode naizmeničnog napona. Dobijeni napon je talasastog oblika.
- *Filtriranje* – obavlja se sa ciljem da se smanji talasnost jednosmernog napona. Filterska mreža može sadržati samo jedan kondenzator dovoljno velike kapacitivnosti, kao na Sl. 23, ali može biti i složenije topologije sačinjena od više kondenzatora i kalema. Na izlazu filtra dobija se gotovo ravan napon.
- *Stabilizacija* – obavlja se sa ciljem da se napon stabilizira na određeni željeni nivo. Za stabilizaciju se koriste elektronska kola koja se zovu stabilizatori napona.

Stabilizacija jednosmernih napona

Stabilizacija predstavlja najbitniji deo izvora za napajanje, jer direktno utiče na kvalitet napona napajanja i ukupnu potrošnju sistema. U upotrebi su dva tipa stabilizatorskih kola:

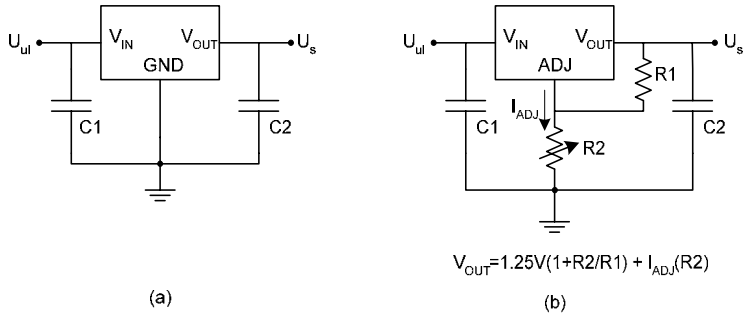
- linearni regulatori napona i
- prekidački regulatori napona

Linearni regulatori napona

Linearni regulatori napona generišu stabilan jednosmerni izlazni napon na bazi manje stabilnog ulaznog jednosmernog napona. U normalnom radnom režimu, linearni regulator obezbeđuje konstantan izlazni napon nezavisno od varijacija ulaznog napona i/ili izlazne struje (tj. struje opterećenja). Takođe, linearni regulator eliminiše kratkotrajne pikove koji se eventualno javljaju u ulaznom naponu. Linearni regulatori su dostupni u obliku diskretnih elektronskih komponenti, koja osim osnovne funkcije stabilizacije napona obezbeđuju i temperatursku stabilizaciju, zaštitu od preoterećenje i kratkog spoja na izlazu.

Postoje dve vrste linearnih regulatora: fiksni i promenljivi. Fiksni regulatori (Sl. 24(a)) su deklarisani za određeni napon. Tako, postoje fiksni regulatori za 5V, 9V, 12V i td. Kod promenljivih regulatora izlazni, stabilisani napon se može podešavati u određenim granicama. Na Sl. 24(b) je prikazano stabilizatorsko kolo koje koristi promenljivi linearni regulator. Vrednost izlaznog napona je određena odnosom otpora R1 i R2.

Danas su na tržištu dostupne brojne varijante linearnih regulatora. Najčešće korišćena serija linearnih regulatora je LM78XX za fiksnu i LM317 za promenljivu regulaciju.



Sl. 24 Linerani regulatori: (a) fiksni; (b) promenljivi.

Linearni regulatori ispoljavaju svojstvo stabilizacije samo ako je pad napona između ulaza i izlaza regulatora veći od neke određene vrednosti, tipično 2-3V. Na primer, da bi se dobio stabilan napon od U_s=5V, ulazni nestabilisani napon mora biti U_{ul}>7.5V. Pošto su regulator i potrošač vezani na red (otuda i naziv redna stabilizacija), celokupna struja potrošača, I_p, protiče i kroz regulator, tako da je ukupna snaga sistema jednaka $P_s=U_{ul}*I_p$, a ne samo U_s*I_p. To praktično znači da se na samom regulatoru javlja značajna disipacija snage od $P_r=(U_{ul}-U_s)*I_p$. Efikasnost regulatora se definiše kao odnos snage potrošača i ukupne snage, što iznosi $E=U_{ul}/U_s$. Za navedeni primer, U_{ul}=7.5V i U_s=5V, efikasnost je E=66.6%, što znači da cena koja se plaća za stabilizaciju napona iznosi čak 33.3% ukupne potrošnje sistema. Mala efikasnost je glavni nedostatak linearnih regulatora, koji naročito dolazi do izražaja kod sistema sa baterijskim napajanjem. Zato se kod sistema sa baterijskim napajanjem koristi jedna posebna vrsta linearnih regulatora koji se zovu *regulator sa malim odstupanjem* (LDO – *Low Dropout*). Ovi regulatori su tako konstruisani da tolerišu malu razliku između ulaznog i izlaznog napona, tipično 0.5-1V. Međutim, u odnosu na klasične linearne regulatore, LDO regulatori su manje stabilni i zahtevaju veći broj dodatnih eksternih komponenti.

Prekidački regulatori napona

Prekidački regulatori napona (ili DC/DC konvertori) su stabilizatorska kola visoke efikasnosti. Osnovni princip rada DC/DC konvertora je da se ulazni jednosmerni napon analognim prekidačima pretvori u promenenjivi napon koji će biti pogodan za tranformaciju pomoću magnetnog elemenata (kalemovi i transformatori).

Postoji više različitih tipova DC-DC konvertora koji se razlikuju po konstrukciji i osobinama. Međutim, bazični princip rada svih DC-DC konvertora je u suštini isti. Pojednostavljena blok šema DC-DC konvertora prikazana je na Sl. 25. Za razliku od linarnih regulatora, kod kojih se energija iz izvora za napajanje neprekidno prenosi na potrošač, kod prekidačkih regulatora energija se prenosi u diskretnim jedinicima. Kao što se može videti sa Sl. 25, kod DC-DC konvertora, naizmenično se odvijaju dva procesa: najpre se iz izvora za napanje uzima određeni iznos energije i privremeno smešta u kalem u vidu magnetskog polja (prekidač P1 je zatvoren, a prekidač P2 otvoren), a zatim se energija nagomilana u kalemu prebacuje u izlazni kondenzator koji napaja potrošač (prekidač P1 je otvoren, a prekidač P2 zatvoren). Prekidačima upravlja impulsno-širinski modulator (PWM) koji generiše binarni signal konstantne periode T sa promenljivim trajanjem impulsa D*T, 0<D<1 (Sl. 25(c)). Takođe, kod praktičnih realizacija, prekidač P2 se realizuje pomoću diode. Naime, sve dok je prekidač P1 zatvoren napon na anodi diode je 0V i dioda D1 ne vodi (odgovara otvorenom prekidaču P2). U trenutku otvaranja prekidača P1, a zbog tendencije nagle promene struje kroz kalem, napon na kalemu naglo raste, što uslovljava da dioda D1 provede (odgovara zatvorenom prekidaču P2).

U toku prve faze, kada je prekidač P1 zatvoren, napon na kalemu je konstantan i iznosi V_{in}, a struja kroz kalem linearno raste. Priraštaj struje u toku prve faze iznosi:

$$\Delta I1=V_{in}*D*T / L$$

U toku druge faze, kada je prekidač P1 otvoren, napon na kalemu je V_{out}-V_{in}, a struja kroz kalem linearno opada. Pad struje u toku druge faze iznosi:

$$\Delta I2=(V_{out}-V_{in})*(1-D)*T / L$$

Prilikom proračuna kola treba uzeti u obzir činjenicu da struja kroz kalem ne može trenutno da se promeni. To znači da će u ustaljenom stanju apsolutne vrednost promene struje kroz kalem u toku obe faze biti identične:

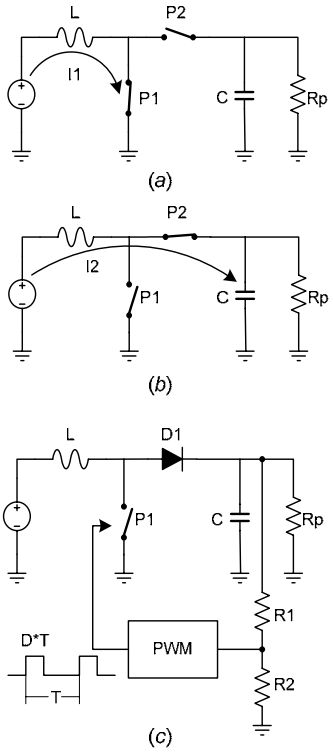
$$\Delta I1=\Delta I2$$

Odakle sledi:

$$V_{out}=V_{in}/(1-D)$$

Dakle izlazni napon je veći od ulaznog, i njegova vrednost se može podešavati parametrom D (*Duty Ratio*). Zadatak impulsno širinskog modulatora je da nadgleda izlazni napon i podešava parametar D tako da izlazni napon uvek ima željenu vrednost. Ako se otpornost potrošača smanji, povećava se struja pražnjenja kondenzatora, potrošač počinje da preuzima veći iznos energije iz kondenzatora i napon na kondenzatoru pada. PWM detektuje pad izlaznog napona i povećava faktor D. Pošto prekidač P1 ostaje duže vremena zatvoren iznos energije koja se prenosi na kondenzator biće veća, što dovodi do povećanja izlaznog napona. Slično, ako se otpornost potrošača poveća, smanjuje se iznos energije koji se predaje potrošaču, a izlazni napon počinje da raste. PWM reaguje tako što smanjuje D, što ima za posledicu smanjenje izlaznog napona.

Opisana konfiguracija DC-DC konvertora odgovara tzv. *step-up* konvertoru, s obzirom da omogućava konverziju nižeg ulaznog u viši izlazni napon.



Sl. 25 **Princip rada *step-up* DC-DC konvertora.**

Na Sl. 26 je prikazana principijena šema tzv. *step-down* DC-DC konvertora. Osnovna osobina *step-down* konvertora je da omogućava transformaciju višeg ulaznog jednosmernog napona u niži izlazni jednosmerni napon.

U prvoj fazi, za vreme dok je prekidač P1 zatvoren (Sl. 26(a)) struja I1 teče iz izvora napajanja, prolazi kroz kalem do kondenzatora i potrošača. Struja I1 raste sa protokom vremena, a u kalemu se gomila energija. U drugoj fazi (Sl. 26(b)), prekidač P1 je otvoren, a P2 zatvoren. Iako je sada izvor isključen i ne daje struju, potrošač nastavlja da se napaja strujom I2 koju daje kalem koji se rasterećuje nagomilane energije. Struja I2 opada sa protokom vremena.

S obzirom da struja kroz kalem ne može trenutno da se promeni, priraštaj struje kroz kalem u toku prve faze mora biti jednak iznosu za koji se struja kalem smanji u toku druge faze. Ako je trajanje prve faze D*T, tada je priraštaj struje:

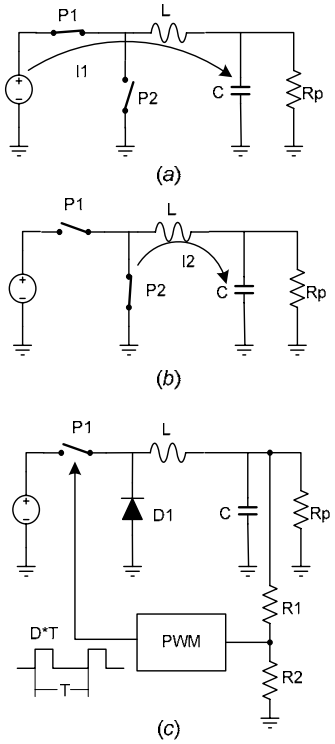
$$\Delta I1 = (V_{in}-V_{out})*D*T/L$$

Trajanje druge faze je (1-D)*T, a za to vreme napon na kalemu je V_{out} tako da pad struje u toku ove faze iznosi:

$$\Delta I2 = V_{out}*(1-D)*T/L$$

Iz uslova $\Delta I1 = \Delta I2$ nalazimo:

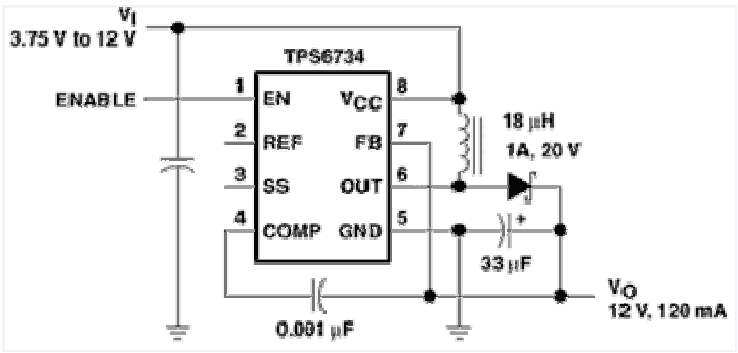
$$V_{out}=D*V_{in}$$



Sl. 26 Princip rada step-down DC-DC konvertora.

Kao i kod *step-up* konvertora, faktor D direktno utiče na vrednost izlaznog napona. Ovo omogućava da se ugradnjom u kolo impulsno-širinskog modulatora kao na Sl. 26 omogući stabilizacija izlaznog napona.

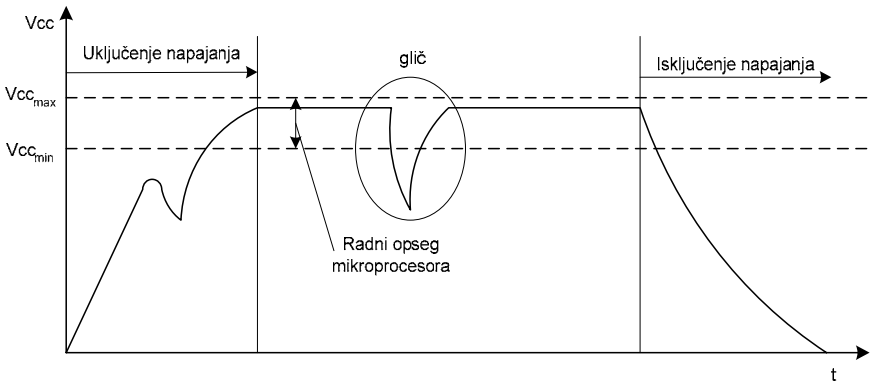
Integrisana kola koja upravljaju radom konvertora, danas prave brojni proizvođači poluprovodnickih komponenti. Na Sl. 27 je prikazan jedan *step-up* konvertor firme Texas Instrumens. Integrisano kolo objedinjuje impulsno-širinski modulator, oscilator, i prekidački tranzistor. Za konstrukciju kompletnog DC-DC konvertora neophodno je ugraditi kalem, prekidačku diodu i izlazni kondenzator.



Sl. 27 Realizacija DC-DC konvertora pomoću specijalizovanog integrisanog kola.

Resetovanje mikroprocesora

Mikroprocesorski sistemi se napajaju regulisanim i naponski stabilisanim izvorom napajanja koji obezbeđuje da napon napajanja mikroprocesora bude u propisanom opsegu. Međutim, i pored toga, u izvesnim okolnostima mogu se javiti varijacije u naponu napajanja što može imati za posledicu neregularan rad sistema. Varijacije napona napajanja mogu biti posledica otkaza izvora napajanja, kada dolazi do potpunog prestanka napajanja, smanjenja napona ili pojave dužih intervala u prekidu napajanja ili mogu biti prouzrokovane naglom promenom opterećenja izvora napajanja, kada dolazi do krakotrajnih gličeva u naponu napajanja. Takođe, pri svakom uključenu izvora napajanja, napon napajana ne uspostavlja se trenutno, već je za dostizanje konačne vrednosti uvek potrebno neko vreme u toku koga napon raste i to ne uvek linearno. Slično, pri isključenu naponskog izvora, napon napajanja ne pada trenutno na 0V, već se postepeno smanjuje. Bez obzira na uzrok, uvek kada napon napajanja padne ispod propisane minimalne vrednosti (ispod Vcc_{min}), ponašanje mikroprocesora se više ne može predvideti. Najčešće posledice izlaganja mikroprocesora smanjenom naponu napajanja su: pogrešno izvršenje programa, nekontrolisana promena stanja izlaznih portova i neželjeni upis u internu ili eksternu memoriju. Pri opadanju napona napajanja ne isključuju se sva interna kola mikroprocesora u usto vreme. Tipično, najpre dolazi do isključenja internog RAM-a i registara koji počinju da gubi memorisani sadržaj. Usled toga, na primer, sadržaj programskog brojača može biti promenjen što dovodi do nekontrolisanih programskih skokova i izvršenja instrukcija izvan programskog redosleda. Instrukcije koje se tom prilikom izvrše mogu da promene stanje izlaznih portova što može da izazove akcije kod drugih kola koja su spregnuta sa mikroprocesorom, a koja su još uvek u radnom režimu. Na primer, EEPROM memorije su karkteristične po tome da mogu reagovati na signale upisa/brisanja i pri veoma niskom naponu napajanja (1.2V). Ukoliko u toku perioda nekontrolisanog rada mikroprocesora dođe do aktiviranja signala upisa u EEPROM, sadržaj EEPROM-a može biti promenjen.

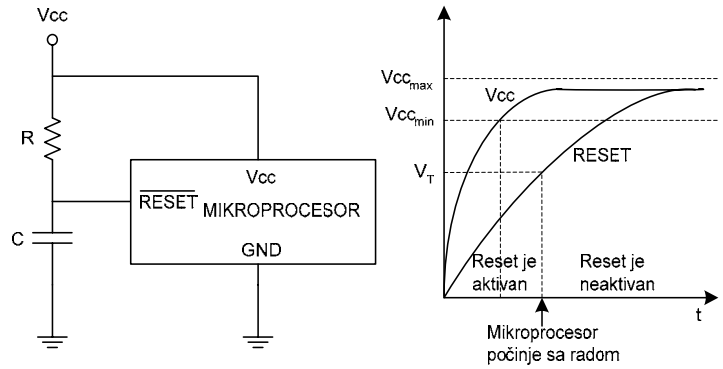


Sl. 28 Varijacije napona napajanja

Ponašanje mikroprocesora, čak i kada je napon napajanja ispod donje granice radnog opsega, može se kontrolisati aktiviranjem *reset* ulaza. Aktiviranjem reset signala, mikroprocesor se postavlja u definisano početno stanje (npr. resetuju se programski brojač, akumulator i statusni registri, izlazni portovi se postavljaju u definisano stanje i td.). Za sve vreme dok je reset signal aktivan rad mikroprocesora je blokiran, a mikroporcesor ostaje u početnom stanju sve dok se reset signal ne deaktivira. Nakon deaktiviranja reset signala, mikroprocesor počinje da izvršava program. Da bi se obezbedio pouzdani rad mikroprocesora, reset signal mora biti aktivan za sve vreme dok se napon napajanja nalazi izvan dozvoljenog radnog opsega.

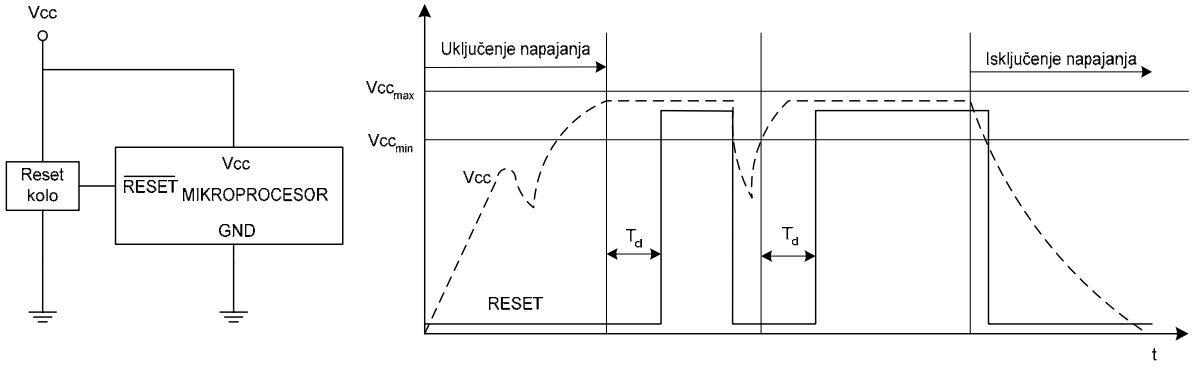
Uobičajena praksa je da se reset signal generiše RC kolom koje je direktno povezano sa izvorom napajanja (Sl. 29). RC kolo unosi kašnjenje tako da po uključenju napajanja, reset signal ostaje aktivan i neko vreme nakon što je napon napajanja dostigao konačnu vrednost. Naime, u trenutku uključenja napajanja kondenzator C je prazan, napon na reset ulazu mikroprocesora je 0V, što znači da je reset ulaz aktivan. Vremenom, kako se kondenzator puni, napon na njemu raste. U trenutku kada napon na kondenzatoru dostigne napon praga reset ulaza V_T reset se deaktivira i mikroporcesor počinje sa radom. Brzina porasta napona na kondenzatoru zavisi od vrednosti vremenske konstante RC. Što je proizvod RC veći to će napon sporije da raste, a reset ulaz će biti duže vremena aktivan. Potrebno trajanje reset signala zavisi od mikroprocesora, a mora biti dovoljno dugo kako bi se obezbedilo da se oscilator, koji takođe počinje sa radom, stabilizuje, a zatim i obave sve interne aktivnosti u mikroprocesoru koje ga postavljaju u iniciajalno stanje.

Opisano rešenje je zadovoljavajuće za najveći broj primena. Međutim, kod sistema kod kojih se ne može tolerisati nekontrolisano ponašanje mikroprocesora, potrebno je preduzeti dodatne mere zaštite od varijacija napona napajanja.



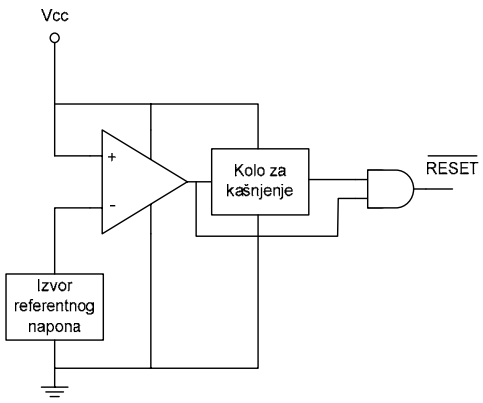
Sl. 29 Generisanje reset signala RC kolom.

Pouzdano resetovanje mikroprocesora se postiže ako se reset signal generiše korišćenjem tzv. *reset kola*. Reset kolo je specijalizovano integrisano kolo koje prati promene napona napajanja i aktivira reset signal uvek kada napon napajanja padne ispod donje granične vrednosti radnog opsega. Reset signal se deaktivira tek nakon nekog fiksnog vremena T_d pošto je pri porastu napona napajanja dostignuta vrednost donje granice radnog opsega (Sl. 30). Vreme T_d tipično iznosi 100-150ms i uvedeno je kako bi se obezbedilo da u trenutku puštanja mikroprocesora u rad napon napajanja bude u potpunosti stabilizovan.



Sl. 30 Princip rada reset kola.

Na **Sl. 31** prikazana je pojednostavljena unutrašnja struktura reset kola. Centralni deo reset kola je naponski komparator koji poredi napon napajanja (Vcc) i interno generisani referentni napon, koji je podešen tako da odgovara donjoj graničnoj vrednosti napona napajanja mikroprocesora. Kolo za kašnjenje može biti realizovano u obliku RC kola, koje kasni premene signala sa izlaza komparatora, a čija je vremenska konstanta podešena na vreme T_d. Uvek kada napon napajanja postane manji od referentnog napona, na izlazu komparatora se postavlja nizak naponski nivo koji se trenutno prenosi na izlaz reset kola (odgovara aktivnom reset signalu). Sa druge strane, visok naponski nivo na izlazu komparatora koji se postavlja u trenutku kada napona napajanja postane veći od referentnog napona, ne uslovljava trentuno deaktiviranje reset signala, već se to dešava tek nakon vremena T_d koje je potrebno da se naponska promena na ulazu kola za kašnjenje prenese na njegov izlaz.



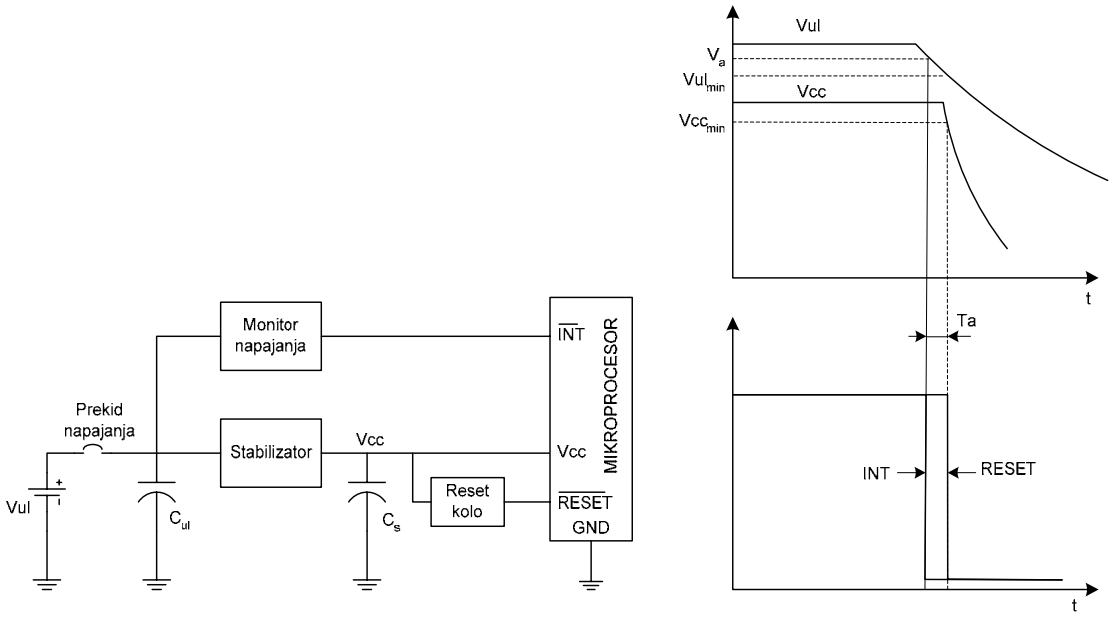
Sl. 31 Unutrašnja struktura reset kola.

Reset kolo predstavlja efikasnu tehniku zaštite mikroprocesorskog sistema od neregularnog rada mikroprocesora pri smanjenom naponu napajanja. Međutim, u mnogim slučajevima poželjno je obaviti neke kontrolisane aktivnosti koje će postaviti sistem, kao celinu, u bezbedno stanje, pre nego što napajanje u potpunosti otkaže. Zamislimo, na primer, mikroprocesorski sistem koji broji proizvode na pokretnoj traci. Broj izbrojanih proizvoda se čuva u internoj memoriji mikroprocesora i prikazuje na displeju. Ukoliko dođe do prekida napajanja mikroprocesor će biti resetovan, a informacija o broju proizvoda izgubljena. Kada se napajanje ponovo uspostavi na displeju će biti prikazana nula. Da bi se informacija o broju proizvoda sačuvala i nakon prestanka napajanja, neophodno je da bude upisana u permanentnu memoriju. Međutim, rešenje kod koga se nova vrednost broja proizvoda upisuje u permanentnu memoriju (npr. tipa EEPROM) uvek kada se na pokretnoj traci detektuje novi proizvod nije dobro. Broj upisa u EEPROM je ograničen (npr. na 10.000), a kada se ovaj broj prekorači može doći do trajnog fizičkog oštećenja EEPROM-a. Optimalno rešenje je ono kod koga se informacija o broju proizvoda čuva u internoj RAM memoriji mikroprocesora, a prebacuje u EEPROM neposredno pre nestanka napajanja. Nakon resetovanja, mikroprocesor čita EEPROM i prebacuje zatečenu vrednost broja proizvoda iz EEPROM-a u interni RAM. Da bi ovakvo rešenje bilo moguće neophodno je detektovati trenutak nestanka napajanja.

Monitor napajanja

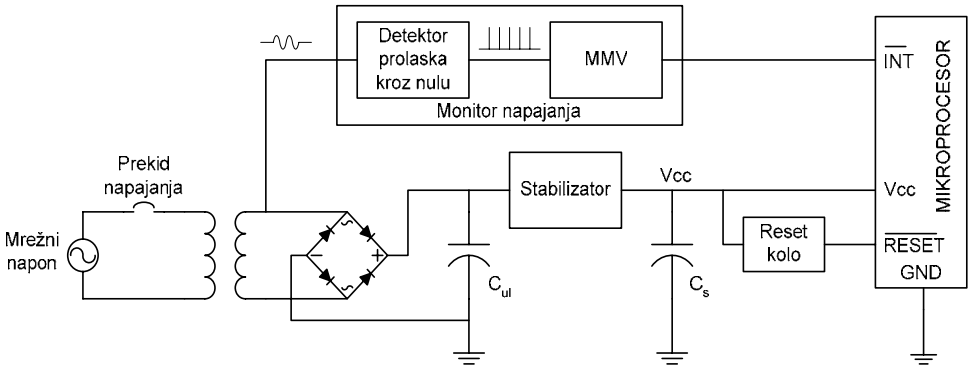
Za detekciju prestanka napajanja koristi se kolo koje se zove *monitor napajanja*. Ovo kolo nadgleda ulazni napon napajanja i aktivira signal alarma u trenutku kada ulazni napon padne ispod zadate granične vrednosti (Sl. 32). Čak i kada se izvor napajanja isključi, sistem nastavlja sa radom jer se se napaja elekticitetom koji je nagomilan na oblogama ulaznog kondenzatora C_{ul}. Vremenom, kondenzator C_{ul} se prazni, a ulazni napon smanjuje. Brzina smanjenja ulaznog napona zavisi od kapacitivnosti ulaznog kondenzatora i struje potrošnje sistema. Što je kapacitivnost ulaznog kondenzatora veća i struja potrošnje manja, to će proteći više vremena do trenutka konačnog prekida rada sistema. U trenutku kada ulazni napon padne ispod minimalnog napona regulacije, stabilizatorsko kolo se isključuje, a napon napajanja Vcc brzo opada. Granični napon monitora napajanja (V_a) podešen je na vrednost koja je veća od minimalnog napona regulacije. U trenutku kada je uslov $Vul=V_a$ ispunjen, monitor napajanja aktivira signal alarma koji prekida rad mikroprocesora. U odgovarajućem

prekidnom potprogramu mikroprocesor obavlja aktivnosti koje su neophodne da bi se sistem postavio u bezbedno stanja (npr. upisuje kritične podatke u EEPROM, zatvara ventile, zaustavlja motore i sl.). Za obavljanje ovih aktivnosti mikroprocesor ima na raspolaganju ograničeno vreme T_a posle koga, zbog pada napona napajanja Vcc ispod minimalne dozvoljene vrednosti, reset kolo generiše signal reset koji konačno zaustavlja rad mikroprocesora.



Sl. 32 Princip rada monitora napajanja.

U slučajevima kada se sistem napaja mrežnim naponom, detekcija nestanka napajanje može se ostvariti direktnim nadgledanjem ulaznog naizmeničnog napona (Sl. 33). Monitor napajanja, u ovom slučaju, sadrži detekor prolaska kroz nulu i retrigerabilni monostabilni multivibrator (MMV). Uvek kada ulazni naizmenični napon prođe kroz nulu, detektor prolaska kroz nulu generiše kratkotrajni impuls. U normalnom režimu rada frekvencija ovih impulsa iznosi 100Hz, što odgovara periodi od 10ms. Impulsi okidaju monostabilni multivabrator čija je vremenska konstanta podešena na vreme koje je tek nešto veće od 10ms, npr. 11ms. Sve dok je mrežno napajanje uključeno, monostabilni multivibrator ne uspeva da izađe iz kvazistabilnog stanja i na njegovom izlaz prisutnan visok naponski nivo. Kada se mrežno napajanje isključi, prekida se povorka okidnih impusla i 11ms nakon poslednjeg okidnog impulsa, MMV se vraća u stabilno stanja, na njegovom izlazu se uspostavlja nizak naponski nivo što inicira prekid mikroprocesora.



Sl. 33 Direktno nadgledanje ulaznog naizmeničnog napona.