

Ugradbeni računalni sustavi - ispitna pitanja s odgovorima 2 ciklus; by Čule

Preuzeto s <http://www.fer.hr/predmet/purs>, stara ispitna pitanja iz pursa, pokrivena sva teorijska pitanja za drugi ciklus URS-a. Pitanja odgovaraju pitanjima za provjeru znanja na kraju svakog predavanja.

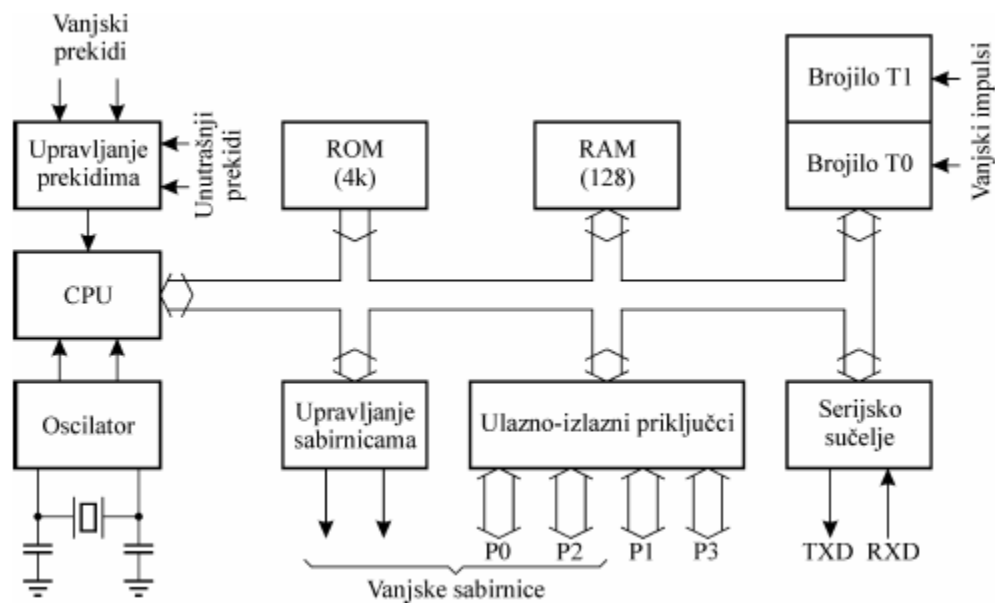
- 1. Koje je razlika između mikroprocesora, procesora, mikrokontrolera i kontrolera? Što oni sadrže, a što ne sadrže? Kakvi mikrokontroleri postoje na tržištu, obzirom na organizaciju memorije, tehnologiju, primjenu, izvedbu napajanja, cijenu i temperaturno područje rada?**

Mikrokontrolerom se obično naziva komponenta koja osim CPU može u istom kućištu imati sadržati programsku i podatkovnu memoriju, te neke periferne jedinice. Mikroprocesor osim CPU-a ima i priručnu memoriju (cache) u istom kućištu, dok se programska i podatkovna memorija (rom i ram) kao i periferne jedinice spajaju izvana. Mikroprocesori se obično susreću stolnim računalima a mikrokontroleri s URSima. Granica između kontrolera i mikrokontrolera i procesora i mikroprocesora postavlja se na 16 bitova (niži od 17 su mikro).

Prema organizaciji memorije dijele se na: von Neumann, harvard, modificirani Harvard. Prema tehnologiji izvedbe: CMOS i NMOS. Prema primjeni dijele se na: mikrokontrolere opće namjene (general purpose) i namjenske mikrokontrolere (dedicated). Prema izvedbi napajanja: opća, s malim naponom napajanja, s malom potrošnjom. Prema cijeni: srednja i visoka, pristupačna (general purpose) i niska cijena (low cost). S obzirom na temperaturno područje: komercijalno (0 do 70), industrijsko (-25 do 85), vojno (-55 do 125) te prošireno industrijsko (-40 do 125).

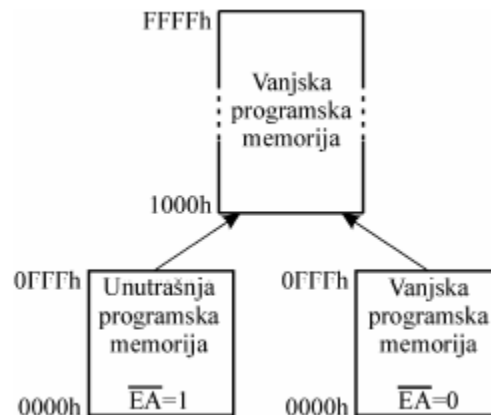
2. Kakve unutrašnje programske memorije nalazimo kod mikrokontrolera? Kako izgledaju najraširenije izvedbe kućišta? Nacrtati blokovsku shemu mikrokontrolera 8051 i objasniti funkcije pojedinih sklopova. Kakva je arhitektura mikrokontrolera 8051 obzirom na memoriju?

Unutrašnja programska memorija može biti izvedena kao PROM, UV-EPROM, EEPROM, FLASH EPROM, „ROMLESS“. Najraširenije izvedbe kućišta su: DIL (dual in line), LCC (lead chip carrier), QFP (quad flat pack), BGA (ball grid array) te smart card. 8051 ima modificiranu harvard arhitekturu memorije – programski i podatkovni prostor su odvojeni ali postoji mogućnost čitanja iz programskog djela.

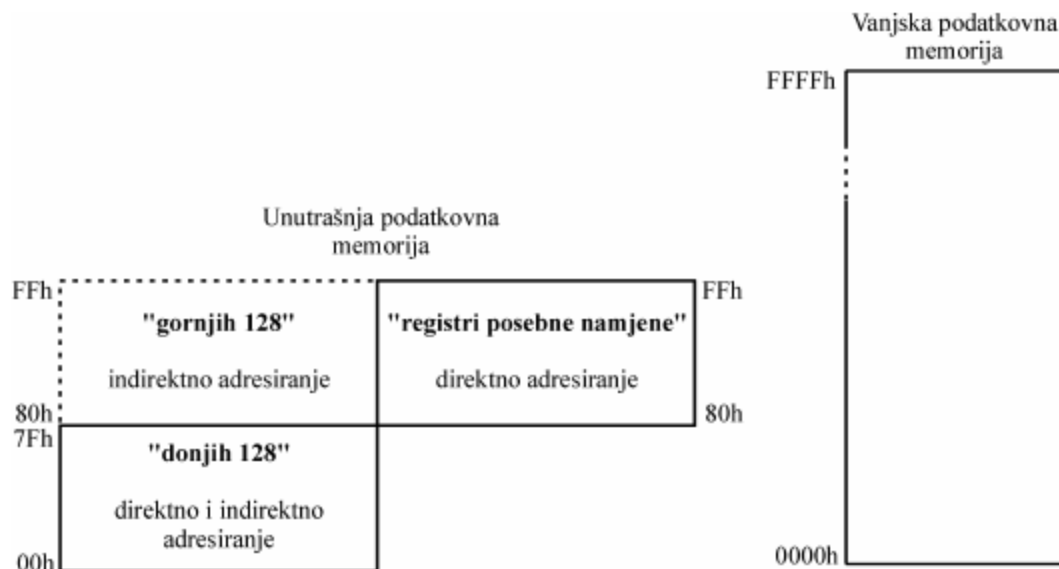


8 bitni CPU s izvorom takta i sklopovljem za upravljanje prekidima, programska i podatkovna memorija te programibilne vanjske jedinice. Četiri skupa ulazno izlaznih priključaka (port) služe za komunikaciju s vanjskim svijetom. Unutar integriranog kruga nalazi se i serijsko sučelje te dva brojila koja mogu brojati ili impulse izvedene iz signala takta ili dovedene izvana.

3. Nacrtati i opisati mapu unutrašnje programske i podatkovne memorije. Što su registri posebne namjene? Gdje se oni nalaze u memoriji?

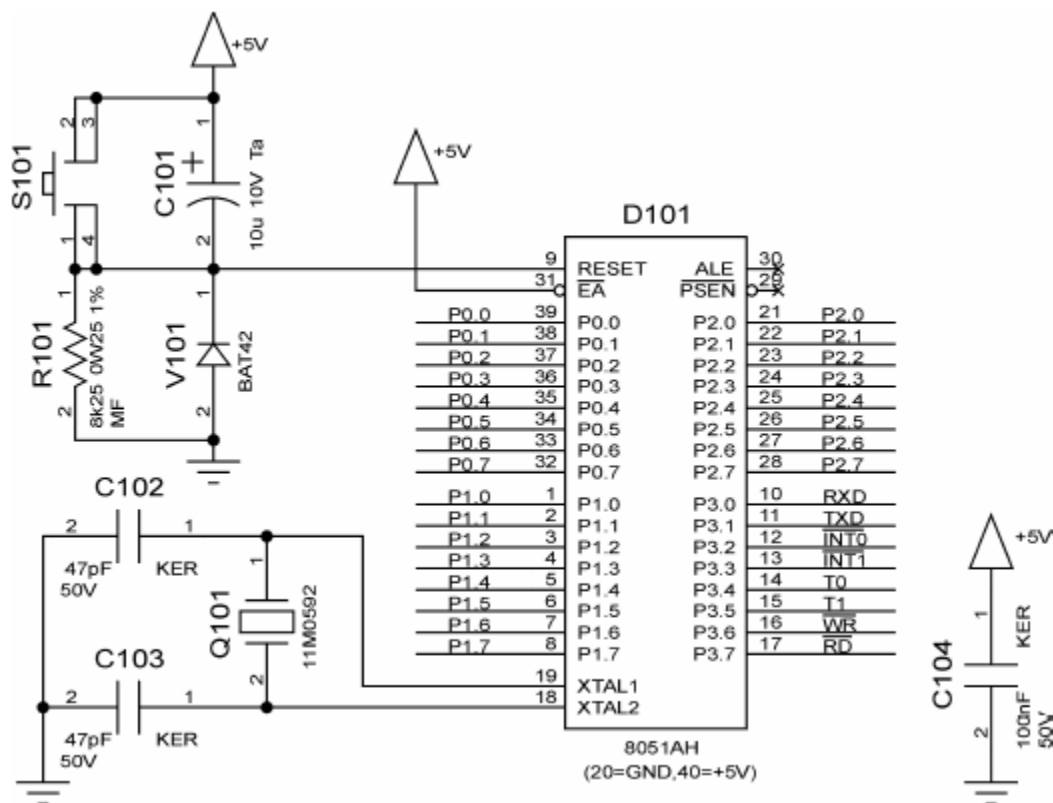


Programska memorija ima kapacitet 4 kilobajta. Unutrašnja programska smještena je na adresama 0000h do 0FFFh. Može se proširiti vanjskom ROM memorijom na adresama 1000h do FFFFh. Može se koristiti isključivo vanjska programska memorija od 0000h do FFFFh. Odabir se izvodi priključkom EA kako je pokazano na slici.



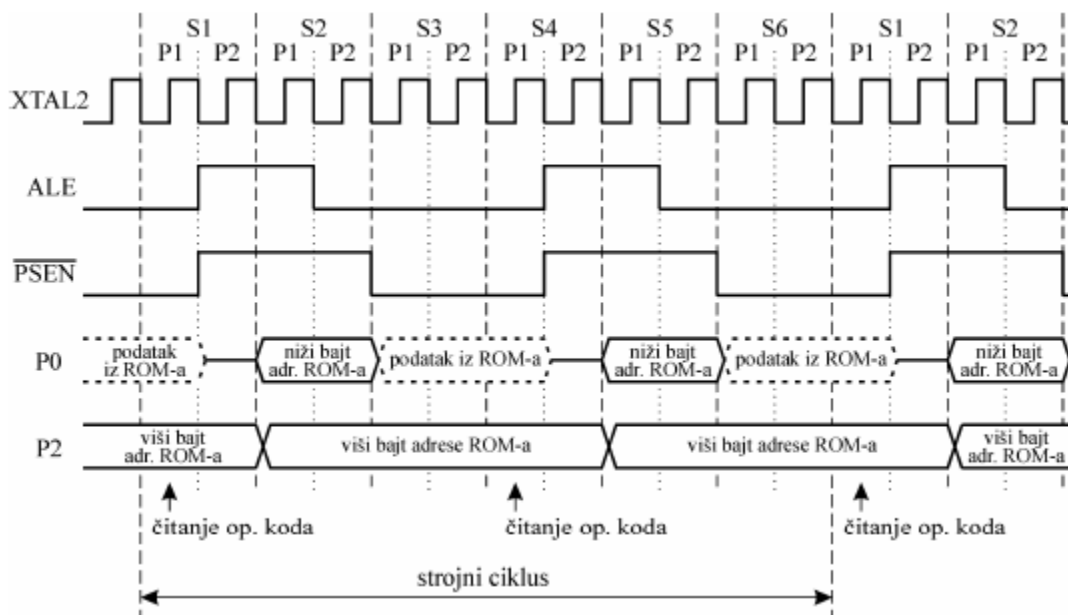
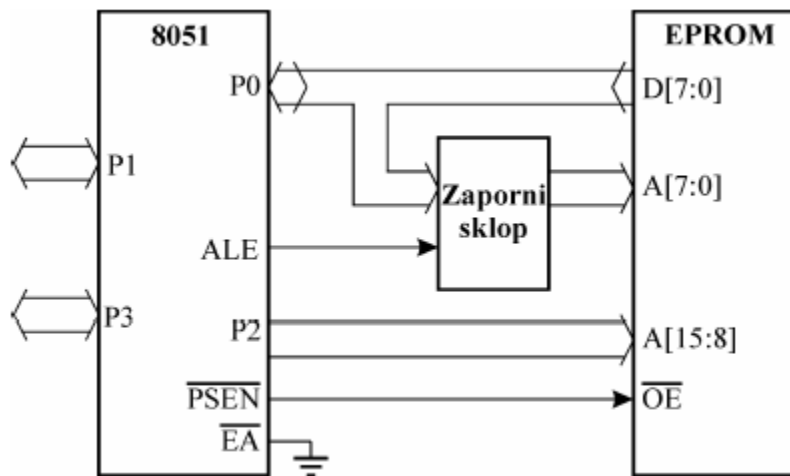
Unutrašnja podatkovna memorija sadrži 128 bajtova. Neki mikrokontroleri iz ove familije imaju 256 bajtova (8052). Tada postoji podjela na donjih 128 (lower 128) od 0 do 127 te na gornjih 128 (upper 128) od 128 do 255. Razlog ove podjele su registri posebne namjene (special function registers) koji se nalaze paralelno gornjih 128 bajtova te čine SFR memorijski prostor. Obzirom da se SFR i gornjih 128 bajtova nalaze na istim adresama registrima posebne namjene se pristupa direktnim adresiranjem a gornjih 128 bajtova memorije indirektnim. Može se dodati i vanjska podatkovna memorija (do 64 kilobajta) koja se zbog konflikta adresa adresiraju posebnim instrukcijama indirektnog adresiranja. Registri posebne namjene su razni registri koje koristi mikrokontroler. Među njima se nalaze i registri P0,P1,P2,P3 priključaka. Upisom konstante u taj registar ona se proslijeđuje na vanjske priključke.

4. Nacrtati osnovni sklop računala s mikrokontrolerom 8051 bez vanjskih memorija (ne treba navoditi brojeve priključaka integriranog kruga). Objasniti funkcije pojedinih sklopova. Objasniti sekundarne funkcije skupa priključaka P3.



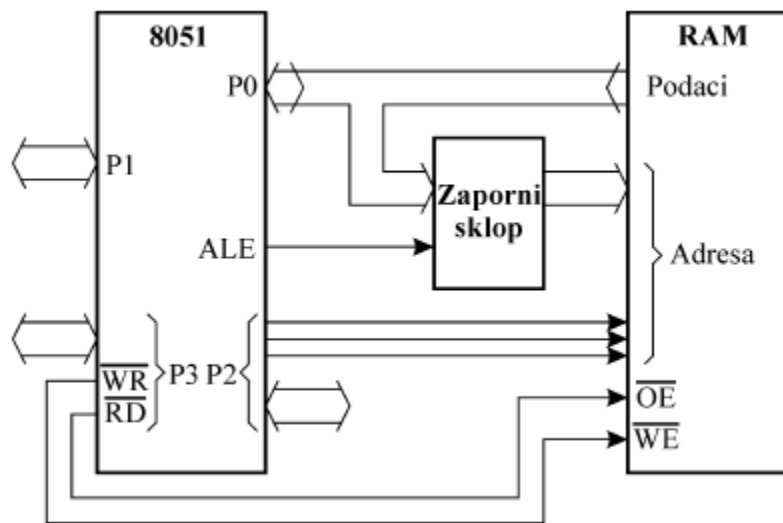
Sklopovlje za reset koje je spojeno na reset ulaz mikrokontrolera. Reset signal mora omogućiti pravilno pokretanje sustava nakon priključenja napajanja ili kratkotrajnog pada napajanja. Reset od vanjske jedinice ili korisnika u toku rada uređaja mora pokrenuti program od početka. Reset u nekim sustavima prekida idle ili power down režim rada. Reset kod 8051 mora trajati barem dva strojna ciklusa tj 24 perioda signala takta. Donji sklop služi za generiranje signala takta (xtal1 i xtal2). Sekundarne funkcije P3 priključaka izvedene su korištenjem NI sklopa, ako se koristi sekundarna funkcija odgovarajući zaporni sklop treba upisati 1. Sekundarne funkcije P3 su: serijska veza, vanjski prekidi, timeri, write i read signali kod pisanja na vanjsku podatkovnu memoriju.

5. Nacrtati blokovsku shemu i objasniti princip spajanja vanjske programske memorije na mikrokontroler 8051. Nacrtati i objasniti ciklus dohvata instrukcije iz vanjske programske memorije.

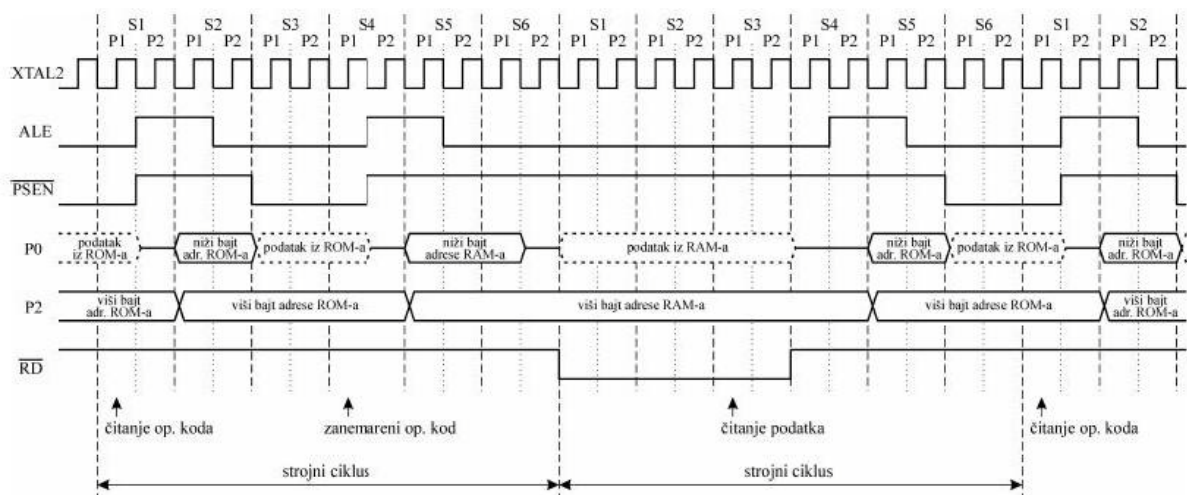


Za generiranje 16 bitne adrese i prijenos 8 bitnog podatka potrebno je ukupno 24 bita. Da se ne koristi svih 24 izvodi se multipleksiranje podatkovne sabirnice (D0 do D7) i nižeg bajta adresne sabirnice (A0 do A7). U prvoj polovici instrukcijskog ciklusa P0 emitira donji bajt adrese. U odgovarajućem trenutku mikrokontroler daje signal ALE (adress latch enable) koji upravlja zapornim sklopom. Kad je ALE=1 latch prihvaća donji bajt adrese, ALE=0 donji bajt adrese ostane zapisan u latchu. u drugoj polovici instrukcijskog ciklusa P0 emitira podatak. P2 cijelo vrijeme emitira viši bajt adrese. Nakon što je adresa postala stabilna, signal PSEN (program store enable) otvara izlaze ROM memorije koja emitira traženi podatak.

6. Nacrtati blokovsku shemu i objasniti princip spajanja vanjske podatkovne memorije na mikrokontroler 8051. Nacrtati i objasniti ciklus čitanja/pisanja iz vanjske podatkovne memorije.



Analogno vanjskom ROM-u korišten je latch za pamćenje donjeg bajta adrese. Signali WR i RD (sekundarne funkcije P3) služe za upravljanje pisanjem odnosno čitanjem iz memorije. Najveći kapacitet memorije je određen širinom adresne sabirnice 2^{16} tj 64 kilobajta. Ako se ne koristi max iznos preostali P2 priključci mogu se koristiti kao obični ulazno izlazni priključci.



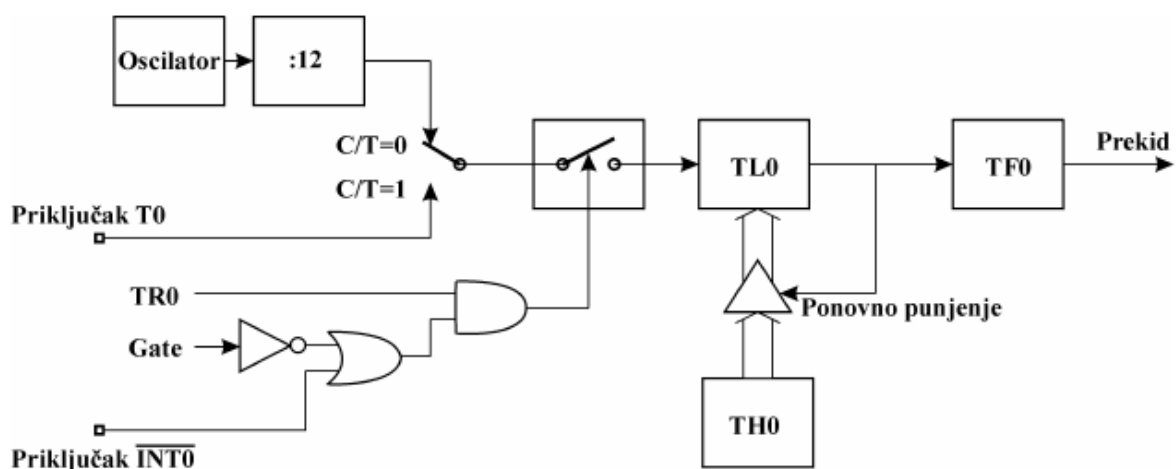
Instrukcije za čitanje iz vanjske podatkovne memorije su jednobajtna dvociklusne. Operacijski kod čita se iz ROMa. Operacijski kod čita se u stanju S1/P1 dok se čitanje u S4 zanemaruje. Tokom S5 na sabirnicu se postavlja adresa iz koje se čita/piše podatak. U sljedećem ciklusu S1, S2 i S3 generira se signal RD/WR te se podatak pročita/upiše iz RAM-a. u tom periodu ne javlja se signal ALE. U stanju S5 na sabirnicu se postavlja adresa novog operacijskog koda itd.

7. Nabrojiti i opisati izvore prekida kod mikrokontrolera 8051. Kakve ima prioritete prekida? U čemu je razlika kod prekida "na brid" i prekida "na razinu"? Koji registri sadrže parametre prekida i što u njima postavljamo (ne treba navoditi položaj pojedinih bitova)? Gdje se u programskoj memoriji nalaze lokacije na kojima se moraju nalaziti prekidne rutine?

8051 ima 5 izvora prekida: 2 vanjska izvora (INT0 i INT1), 2 preljeva iz brojača (TF0 i TF1), serijsko sučelje (RI i TI). Razina prioriteta prekida može biti niska ili visoka. Visoka razina može prekinuti prekid niske razine dok prekidnu rutinu visoke razine se ne može prekinuti. Prekidi se poslužuju od najvišeg prioriteta do najnižeg redom: IE0, TF0, IE1, TF1, RI+TI. Prekid na brid (IT0 i IT1=1 za prekid na brid) je prekid na prijelaz varijable, program će samo jednom ući u prekidnu rutinu dok je prekid na razinu prekid na stanje i u prekidnu rutinu će se ulaziti dok god je prisutno to stanje. Na prijelaz postavlja IE0 u stanje 0 po ulasku u rutinu, dok na razinu ne dira tu zastavicu. Parametre prekida sadrže IE, IP, TCON registri, u njima postavljamo jesu li prekidi omogućeni, postavljamo prioritet prekida, postavljamo jeli prekid na brid ili na razinu. Prekidne rutine nalaze se na sljedećim adresama: 0023h serijsko sučelje; 001Bh brojilo T1, 0013h vanjski prekid 1; 000Bh brojilo T0; 0003h vanjski prekid 0. (adrese prekidnih rutina nalaze se u službenom šalabahteru.

8. Nacrtati blokovsku shemu i opisati rad brojila? Koji načini rada brojila postoje? Opisati načine rada 1 i 2.

- način rada 2



brojilo može raditi kao counter (brojenje impulsa) ili kao timer (mjerjenje vremenskih intervala). Impulsi za brojenje mogu se dovesti iznutra ili izvan mikrokontrolera (T0 – sek priključak P3). Brojila mogu raditi u 4 moda (00,01,10,11 m1 m0 bitovi unutar TMOD registra). Bitom C/T odabire se izvor impulsa. Brojenje se omogućuje TR0=1 (TCON). Upisivanjem broja u (TL0 TH0) počinje brojanje, broji se od 00h do FFh. Kad se postigne stanje 00h postavlja se bit TF0 te timer daje zahtjev za prekid. (Mode 1). Mode 0 i mode 1 su isti osim što nula ima 13 bitova a 1 16 bitova. Mode 2 – brojilo je veličine 8 bitova te se kod preljeva TL0 automatski puni vrijednošću iz TH0 te se analogno modu 2 postavlja bit TF0 za prekid.

9. Opisati registar "Program Status Word". Koje bitove on sadrži i čemu oni služe? Što su to registrarske banke i kako ih postavljamo (dati primjer u assembleru)? Što je to bit adresabilni memorijski prostor, gdje se on nalazi i koja mu je namjena?

Program status word tj PSW registar sadrži zastavice koje opisuju trenutno stanje mikrokontrolera. C bit carry flag, zastavica za preljev te za rad s nekim bool operacijama; AC – auxiliary carry flag, F0 i '1' zastavica koju korisnik postavlja po vlastitoj želji; OV overflow flag, preljev; P parity flag paritet akumulatora (1 ako je paran broj jedinica). RS1 i RS0 određuju područje adresiranja registrarskih instrukcija. Donjih 128 bajtova unutrašnje podatkovne memorije podijeljeno je na 4 registrarske banke, bit adresibilni prostor, te prostor u kojem se pristupa samo cijelim bajtovima. Bit adresabilni prostor je skup memorijskih lokacija kod kojih se može pristupiti tj upisati ili pročitati sadržaj pojedinim bitovima bez utjecaja na preostale bitove te lokacije. Nalazi se u donjih 128 bajtova unutarnje programske memorije (20h do 2Fh).

SETB PSW.3; postavi RS0 u jedinicu
MOV R0, #22h; R0 postavi #22 hex

10. Nabrojiti i opisati tipove adresiranja koje podržava mikrokontroler 8051. Za svaki tip navesti primjer instrukcije (ne treba navoditi operacijski kod)? Opisati koje tipove instrukcija ima mikrokontroler 8051. Navesti primjer za svaki tip.

Direktno adresiranje:

ADD A, 6Fh; (a)←(a)+(6Fh); akumulatoru pribroji sadržaj memorijske lokacije 6Fh iz unutrašnje podatkovne memorije.

Registrarske instrukcije:

ADD A, R1; (a)←(a)+(R1); akumulatoru pribroji sadržaj R1;
ako je aktivna banka 0 to je isto ka ADD A, 01h; ako je aktivna 3 ADD A, 19h

Indirektno adresiranje: (R1=63h)

ADD A, @R1; (a)←(a)+((R1)); akumulatoru pridružuje vrijednost mem lokacije na koju pokazuje R1

Konstane:

ADD A, #1Ah; (a)←(a)+1Ah; akumulatoru pridružuje vrijednost #data

Instrukcije vezane za posebne registre (s točno određenim registrima), te instrukcije ne trebaju adresu jer uvijek rade s istim registrima.

CLR A; akumulator postavlja na 00h

Indeksno adresiranje samo za programsku memoriju:

MOVC A, @A+DPTR; (a)←((a)+(DPTR)); dptr 16 bitan

Tipovi instrukcija:

Aritmetičke:

ADD A, <byte>

Logičke

ANL A, <BYTE>

Instrukcije za prijenos podataka pristup unutrašnjoj podatkovnoj memoriji:

MOV A, <SRC>

Instrukcije za prijenos podataka pristup vanjskoj podatkovnoj memoriji; 8 bita ili 16 bita

MOVB A, @DPTR; za 8 bitovnu adresu

Instrukcije za čitanje podataka iz tablica: tablice isključivo iz programske memorije

MOVC A, @A+DPTR

Boolove instrukcije- rade samo s jednim bitom, uloga koju kod logičkih ima akumulator kod ovih ima C zastavica, naredbe skokova su također boolove.

ANL C,BIT

11. Napisati program koji bi bio pogodan za prvo puštanje mikrokontrolera u rad, tokom uhodavanja. Kakva svojstva mora imati takav program? Zašto?

CSEG AT 00h

Petlja: inc A
 mov P1, A
 sjmp petlja

 End

Program koristi vrlo malo unutrašnjeg i vanjskog sklopovlja. Provjera se izvodi osciloskopom. Dobije se pravokutni signal pri čemu svaki bit ima frekvenciju dva puta manju od prvog najnižeg bita.

12. Opisati kako se vanjski prekid "na razinu" može iskoristiti tokom uhodavanja uređaja. Napisati podršku u assembleru.

na priključku 0 skupa priključaka P1 radimo pravokutni signal perioda 200micro sekundi. Frekvencija mikrokontrolera = 12MHz.

CSEG AT 00h

JMP GLAVNI

CSEG AT 0Bh

CPL P1.0 ;KOMPLEMENTIRAJ BIT

RETI

GLAVNI: MOV TMOD, #00000010b ; BROJLO T0 U NAČIN 2
 MOV TH0, #156d ; BROJI OD 100 MIRKOSEK
 MOV TL0, #156d
 SETB TCON.4 ;BROJLO T0 ON TR0=1

 SETB IE.1
 SETB IE.7

```

PETLJA:    SJMP PETLJA

            END

```

Brojilo T0 radi u načinu 2, broji impulse od 1 mikrosekunde čije trajanje odgovara 12 perioda signala takta. Brojilo broji do 100 mikro sekundi i onda zatraži prekid, prekidna rutina mijenja stanje priključka p1.0 i nastaje pravokutni signal.

13. Napisati program pogodan za uhodavanje sustava s mikrokontrolerom 8051 pomoću osciloskopa, pod pretpostavkom da je neispravna vanjska podatkovna memorija. Kakva svojstva program mora imati? Zašto? Kako se spaja osciloskop?

```

CSEG AT 00h

Mov DPTR, #2AFFh
MOV A, #4Dh

```

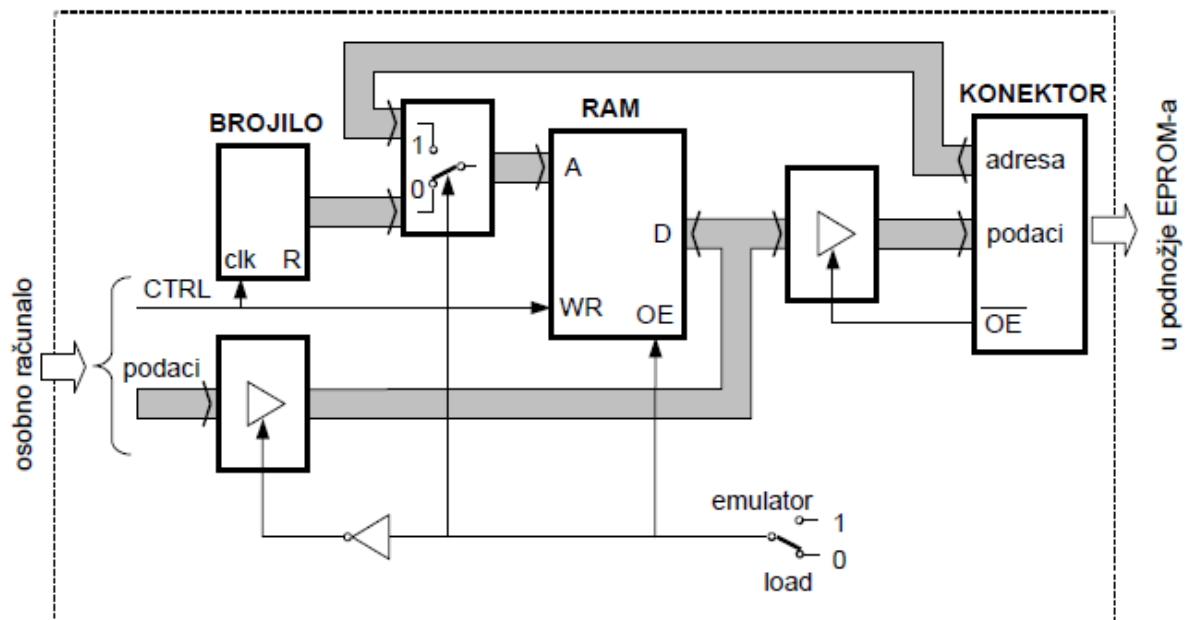
```

PETLJA:    MOVX @DPTR, A
            JMP PETLJA
            END

```

program neprestano upisuje broj 4Dh na mem lokaciju 2AFFh u vanjsku podatkovnu memoriju. Program mora pisati u memoriju da bi testirali ispravnost vanjske podatkovne memorije. Osciloskop se spaja na signal koji ima najmanju frekvenciju, u ovom slučaju to je signal WR. Aktivno stanje jednom u svakom prolazu petlje.

14. Nacrtati blokovsku shemu i objasniti princip rada EPROM emulatora.



Da se skрати postupak programiranja mikrokontrolera u fazi razvoja umjesto EPROM-a u njegovo podnožje spaja se EPROM emulator. Postupak punjenja EPROM emulatora traje nekoliko sekundi. Postoje 2 načina rada: load (podaci se iz osobnog računala (program za mikrokontroler) prenose u

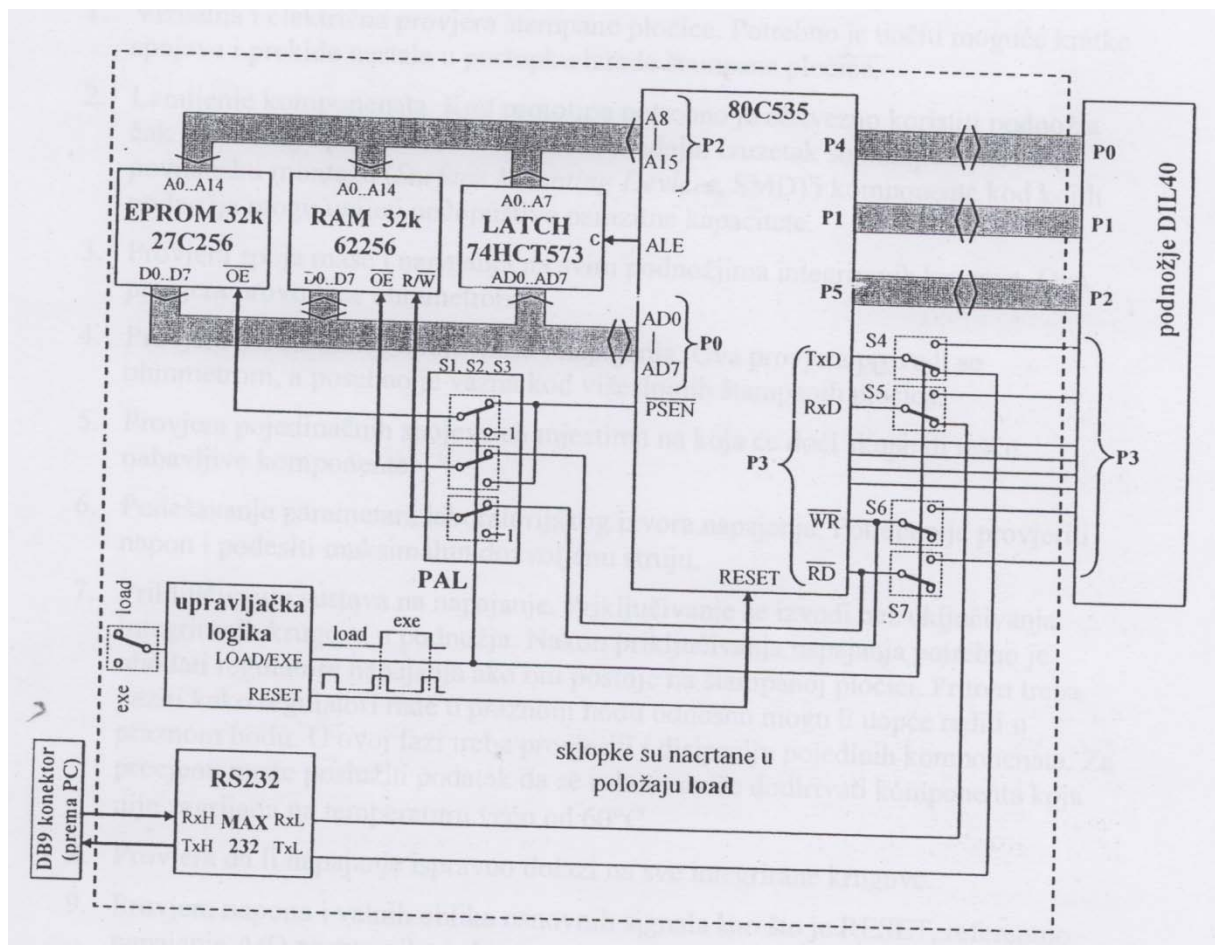
RAM) te emulator (mikrokontroler čita sadržaj RAM-a). Brojilo pri loadu daje adrese za RAM na koje se upisuju podatci iz računala. Upis u RAM i rad brojila kontrolira se signalom CTRL. Podatci se iz memorije prenose na podnožje mikrokontrolera kad je OE aktiviran.

15. Nacrtati blokovsku shemu i objasniti princip emulacije "single chip" izvedbe mikrokontrolera 8051 pomoću mikrokontrolera 80535.

Ukoliko računalni sustav nema vanjsku programsku memoriju, programski kod se nalazi u unutrašnjoj memoriji mikrokontrolera. Umjesto mikrokontrolera koristi se emulator mikrokontrolera (single chip emulator), on emulira sve funkcije mikrokontrolera kojem se sadržaj njegove programske memorije učitava iz osobnog računala.

80535 emulira 8051 mikrokontroler jer ima dodatno sklopovlje (p4 i p5). Skupove 80535 P4, P1, P5, P3 spajamo umjesto P0, P1, P2, P3 mikrokontrolera 8051. P0 i P2 mikrokontrolera 80535 iskorišteni su za spoj vanjske programske i podatkovne memorije.

Cilj opisanog spajanja mikrokontroler je postići da on u fazi emulacije izvršava program smješten u podatkovnoj a ne u programskoj memoriji. Da bi se to postiglo dodan je niz sklopki čiji je zadatak zamjena uloga ovih memorija. Sklopka ima položaj load ili exe. Load – memorije spojene na uobičajan način. Dok je load, program se čita iz eproma, taj program podatke koje prima sa serije i sprema ih u RAM. U EXE program se izvršava iz RAM-a



16. Navesti i opisati faze postupka prvog puštanja uređaja u rad tokom razvoja.

Predugo da bi došlo u ispit (18 točaka)

17. Koji postupci zaštite razvoja se u praksi koriste za zaštitu sklopovlja, a koji za zaštitu programske podrške.

Za zaštitu sklopovlja koriste se idući postupci: brisanje oznaka sa svih integriranih sklopova, upotreba PLA sklopova s sigurnosnim bitom u 1, FPGA sklop koji se jednom napuni te održava pod baterijskim napajanjem, fixno ožičenje FPGA sklopova kod velikih serija.

zaštita programske podrške dijeli se na zaštitu od kopiranja i zaštitu od mijenjanja programskog koda. Dio koda koji se želi zaštititi se pohranjuje u šifriranom obliku, dešifriranje se radi tokom izvršavanja programa. ROM memorije u kojima su pomiješane adresne i podatkovne linije (zaštita od promjene, kopiranje lako). Ako se program nalazi u unutrašnjoj programskoj memoriji mikrokontrolera a bit za zaštitu od čitanja je aktiviran, program je teško pročitati i kopirati.

18. Objasniti razliku između "Idle" i "Power Down" načina rada. Što se u tim načinima rada događa s oscilatorom, registrima, brojilima, serijskim sučeljem i unutrašnjom memorijom. Kako se prekida stanje "Idle", a kako "Power Down"? Kada se ti načini rada koriste?

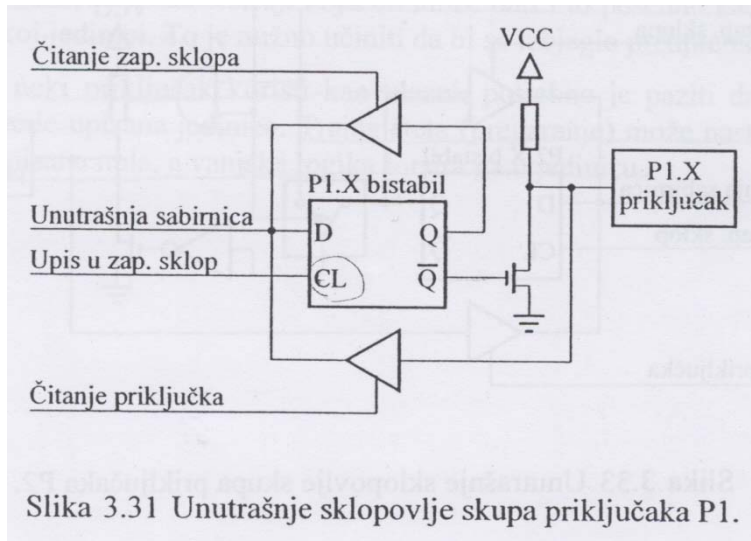
Idle i power down su režimi smanjene potrošnje.

U idle (IDL=1 u PCONu) načinu rada cpu prestaje raditi ali zadržava stanje svih registara (SP, PC, PSW); oscilator, serija, brojila i sklopovlje za posluživanje prekida i dalje rade, stanje unutrašnje podatkovne memorije ostaje sačuvano, vanjski ulazno izlazni priključci zadržavaju stanje koji su imali. Signali ALE i PSEN prelaze u stanje 1. Izlazak iz idle načina rada: zahtjev za prekid, izvršava se prekidna rutina te nakon nje nastavlja se izvršavanje instrukcija; signal RST nakon reseta nastavlja se izvršavanje gdje je program stao (dakle ne od 0 kao kod pravog reseta).

U power down načinu radu (PD=1 u PCONu) svi sklopovi prestaju raditi, oscilator ne titra; stanja unutrašnje podatkovne memorije ostaju sačuvana; vanjski ulazno izlazni priključci imaju stanje koje su imali prije ulaza u power down stanje; signali ALE i PSEN prelaze u stanje 0. Nakon reseta mikrokontroler izlazi iz power downa, registri posebne namjene uđu u inicijalne vrijednosti ali se dira sadržaj unutrašnjeg RAM-a.

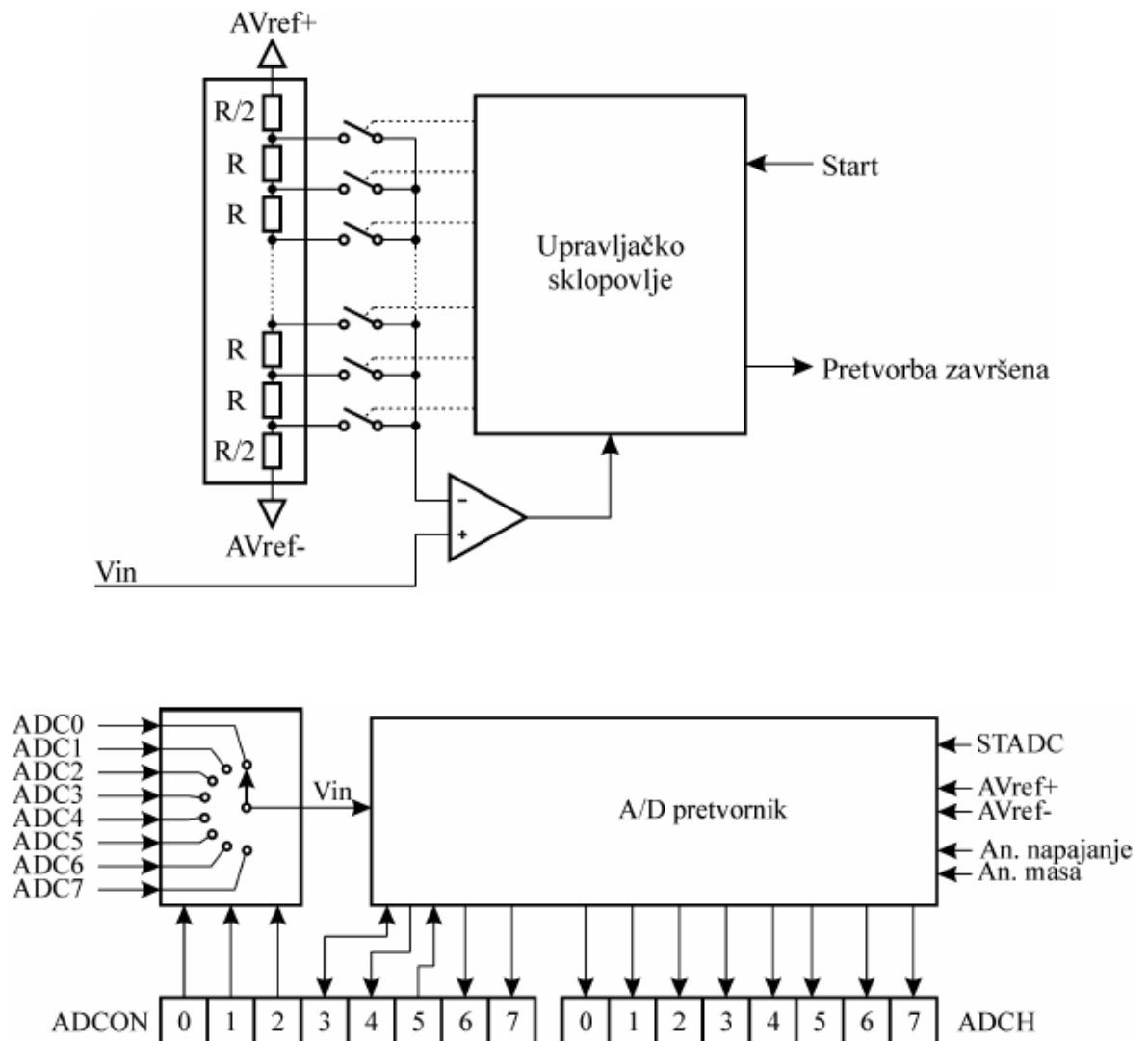
idle način rada koristi se kad je mikrokontroler besposlen a power down se koristi kad želimo smanjiti napajanje rada (npr. Želimo sačuvati sadržaj podatkovne memorije uz odspajanje glavnog napajanja).

19. Nacrtati shemu izlaznog sklopa skupa priključaka P1. Kako se ovi priključci koriste kao ulazni, a kako kao izlazni? Što su to "read-modify-write" instrukcije i po čemu se one razlikuju od ostalih instrukcija? Koje dodatne funkcije imaju pojedini skupovi priključaka? Nabrojiti sekundarne funkcije skupa priključaka P3.



Podatak koji želimo vidjeti na priključnicama upisuje se u P1 zaporni sklop (latch) koji se nalazi u SFR adresnom prostoru. Izlaz iz zapornog sklopa spojen je na upravljačku elektrodu izlaznog tranzistora, pa se upisana jedinica odnosno nula na izlaznom priključku pojavljuje kao niska odnosno visoka razina. Na ovaj način P1 se koristi kao izlaz. Ako je u neki zaporni sklop upisana 1 priključak se koristi kao ulazni. Read-modify-write instrukcije su instrukcije koje čitaju stanje zapornih sklopova a ne stanje izlaznih priključaka. Te instrukcije tokom izvođenja moraju pročitati stanje, napraviti operaciju i vratiti stanje natrag u zaporni sklop (anl, cpl, set). Skupovi priključaka P0 i P2 imaju sekundarne funkcije da služe kao vanjska adresna i podatkovna sabirница ukoliko je spojena vanjska memorija. Sekundarne funkcije P3 su: serijska veza, vanjski prekidi, timeri, write i read signali kod pisanja na vanjsku podatkovnu memoriju.

20. Na primjeru opisati izvedbu A/D pretvornika i ulaznog analognog preklopnika kakve srećemo kod mikrokontrolera familije 8051.

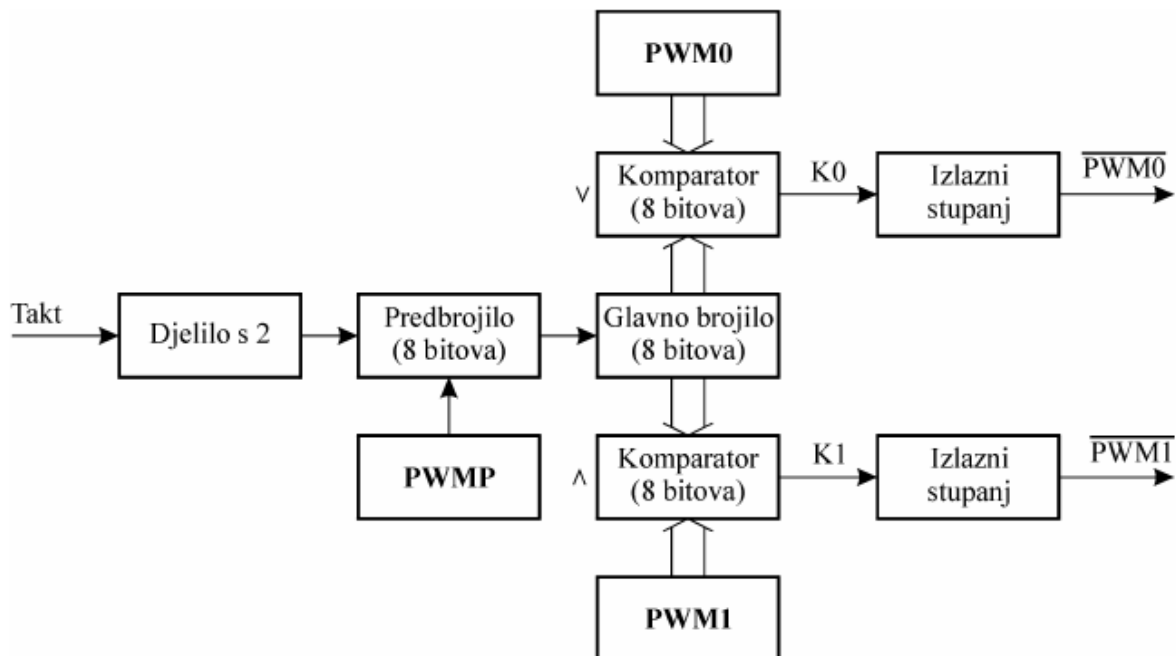


8051 koristi naponska dijelila izvedena otpornicima. Pretvornik ima 10 bitova što daje 1024 diskretne razine. Te razine su predstavljene naponima koji se dobivaju dijeljenjem referentnog napona $AVref$ pomoću dijelila sastavljenog od 1023 otpornika R te dva otpornika od $R/2$. Ulazni signal dovodi se na ulaz komparatora zajedno sa signalom iz jednog čvora otporničkog dijelila. Komparator daje informaciju kojije signal veći, ukoliko je ulazni veći upravljačko sklopovlje uključuje sklopku koja na ulaz komparatora daje veći napon nego što je bio u prethodnom koraku i obrnuto. Nakon nekoliko iteracija dobije se najbolje moguće poklapanje napona. Odabrani čvor djelila tada reprezentira diskretnu razinu koja se prikazuje binarnim kodom na izlazu pretvornika.

U sklopu 80552 osim opisanog AD pretvornika postoji i analogni preklopnik (multiplexer) koji omogućuje da se na ulaz preklopnika dovede jedan od 8 različitih ulaznih signala. ADC0 do ADC7 izvedeni su kao sekundarna funkcija priključaka P5. Koji od ulaza je spojen određuju najniža tri bita ADCON registra. Bistabil 3 tog registra znači start pretvorbe (upisom 1) dok 4 predstavlja kraj (upis 1 u 4 i 0 u 3). 4 bistabil ujedno predstavlja i zahtjev za prekid. Znak za početak može se dovesti izvana

(STADC). ADCH predstavlja 8 važnijih bitova pretvorbe a gornja dva bita ADCON predstavljaju donje najmanje važne bitove pretvorbe.

21. Nacrtati blokovsku shemu i opisati princip rada pulсно širinskog modulatora, te navesti odgovarajuće izraze za izlazni napon. Dati primjer primjene PWM modulatora.



Signal f_{osc} dovodi se na 8 bitno programibilno predbrojilo koje služi za dijeljenje frekvencije signala prije glavnog brojila (prescaler brojilo). PWMP registar određuje broj s kojim se dijeli frekvencija ulaza. Signal se dovodi na glavno brojilo koje ima frekvenciju na ulazu:

$$f_{presc} = \frac{f_{osc}}{2 * (1 + PWMP)}$$

Glavno brojilo ima stanja od 0 do 254 pa puni ciklus napravi nakon 255 ulaznih impulsa. Period izlaznog PWM signala tada je:

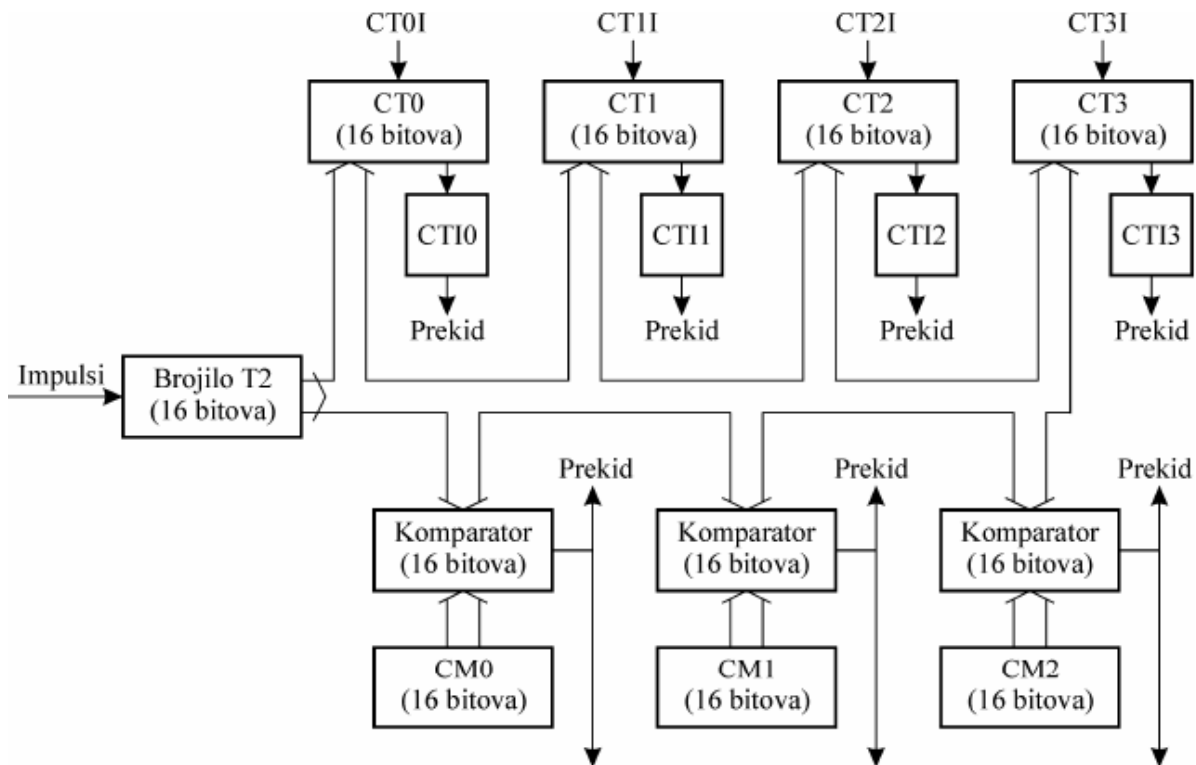
$$T = \frac{255}{f_{presc}}$$

Izlaz iz brojila dovodi se na dva jednaka digitalna komparatora. Na drugi ulaz komparatora dovode se registri (PWM0 i PWM1) koji određuju trajanje visoke razine izlaznog signala. Komparator daje 0 kad je sadržaj PWM registra manji ili jednak sadržaju glavnog brojila a 1 inače. Na taj način sadržajem registra PWM određen je faktor vođenja t/T , odnosno srednja vrijednost signala U_{av} na izlazu K0:

$$U_{AV} = U_0 \frac{t}{T} = U_0 \frac{PWM0}{255}$$

Pwm modulator se koristi za upravljanje rasvjetom, grijanjem te DC motorima.

22. Nacrtati blokovsku shemu i opisati princip rada sklopovlja za hvatanje i usporedbu. Navesti primjer primjene ovog sklopovlja.

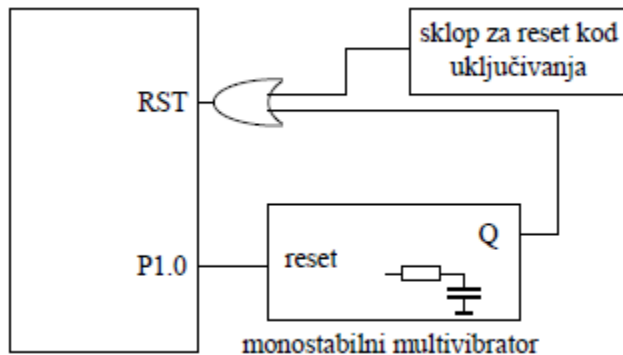


Sklopovlje za hvatanje (capture) služi za mjerenje vremenskih intervala dok sklopovlje za usporedbu (compare) služi za postavljanje stanja nekog signala u točno određenom trenutku. Kod 80552 ova dva sklopa objedinjena u jedan sklop (capture and compare logic). Brojilo T2 (slično brojilima T1 i T2) neprestano broji impulse takta ili vanjske. 16 bitna vrijednost brojila dovodi se na registre CT0 do CT3, ti registri upravljani su CT0I do CT3I (sekundarne funkcije P4). Kad se na CT0I do CT3I pojavi odgovarajući impuls registri CT0 do CT3 zapamte vrijednost brojila te se postavi vrijednost CTI0 do CTI3 što predstavlja zahtjev za prekid.

16 bitna vrijednost brojila dovodi se na 3 komparatora zajedno s izlazima registara (CM0 do CM2). U te registre programski se upisuju konstante s kojima se uspoređuje sadržaj brojila. Kad se sadržaj brojila izjednači s konstantom provode se iduće radnje: za CM0 jednako brojilu odgovarajući bitovi P4 postavljaju se u 1; za CM1 postavljaju se u 0; a za CM2 stanje se invertira (toggle). Koje bitove P4 želimo koristiti određujemo upisom u STE i RTE upravljačke registre.

Sklopovlja za hvatanje i usporedbu koristimo kad želimo mjeriti vremenske intervale, relativne vremenske odnose ili faze, kada želimo postaviti vrijednost signala u točno određenom trenutku.

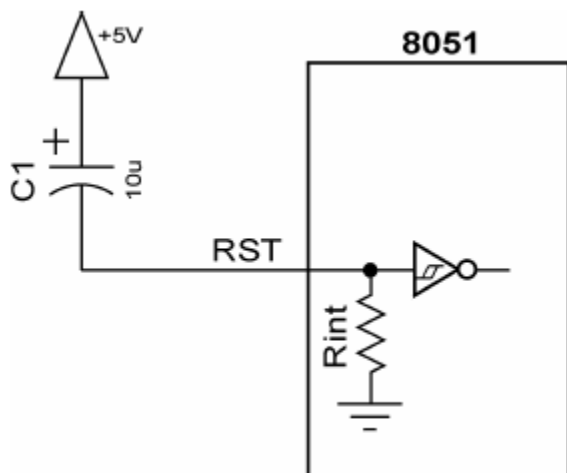
23. Objasniti princip rada "watch-dog" sklopa. Kako i kada se on koristi? Na što treba paziti kod pisanja programa kad sustav sadrži "watch-dog" sklop. U osnovnim crtama Objasniti princip rada "watch-dog" sklopa mikrokontrolera 80552.



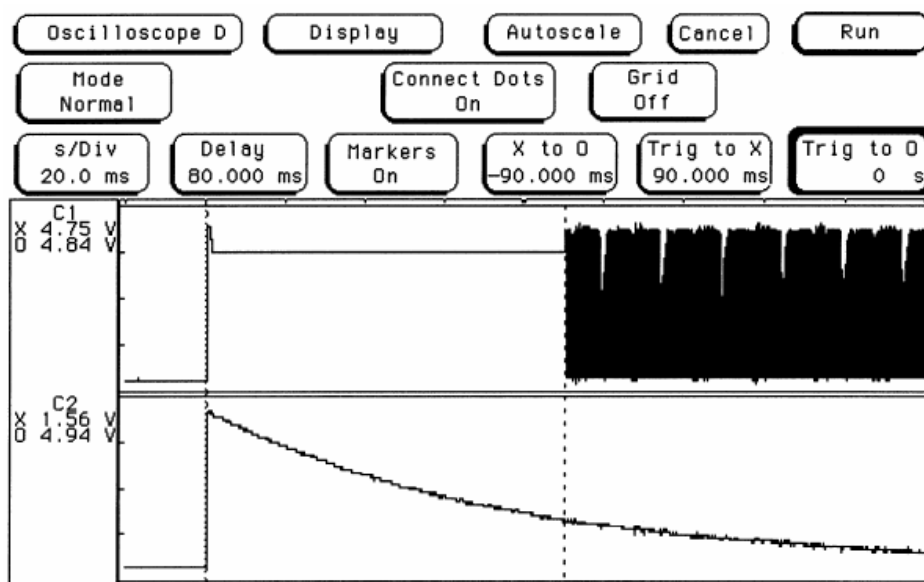
Watch dog sklop je sklop za detekciju pogrešnog rada. Program pod nadzorom watch dog sklopa mora svako neko periodično vrijeme (ne veće od nekog unaprijed određenog T) inicijalizirati watch dog sklop. Ukoliko program to ne napravi watch dog sklop će resetirati mikrokontroler. Program nadziran watch dog sklopom treba pisati vrlo pažljivo uz točno poznavanje vremena izvršavanja svih grana programa. Watch dog sklop koristi se kada se traži velika pouzdanost elektroničkog uređaja. 80552 ima 8 bitno brojilo na čiji ulaz se dovodi impulsi iz 11 bitnog predbrojila. Brojilo broji od B do FFh, za to je potrebno vrijeme T (watch dog interval). Kad brojilo izbroji do FFh dolazi do preljeva koje uzrokuje RESET mikrokontrolera. Da se to ne bi dogodilo program svako manje od T mora inicijalizirati vrijednost B (reload brojila). Prvo se upisuje 1 u bit 4 registra PCON (time se omogući upis u brojilo) nakon toga se upiše broj B opet. Priključak EW služi za omogućavanje watch dog sklopa, kod testiranja se ne koristi.

24. Koje zahtjeve mora zadovoljiti signal RESET? Nacrtati sklop za reset kod uključivanja te sklop koji ima dodatnu tipku za reset. Objasniti ulogu pojedinih komponenata. Što sadrže integrirani sklopovi za reset?

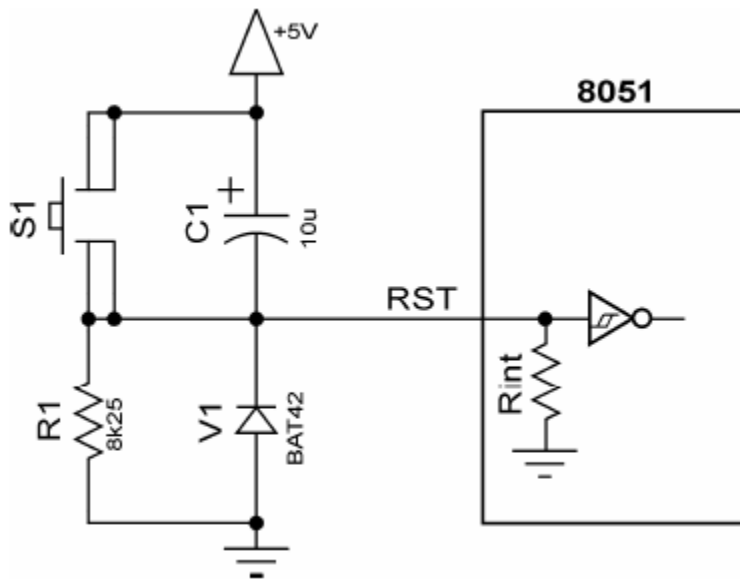
Reset signal mora omogućiti pravilno pokretanje sustava nakon priključenja napajanja ili kratkotrajnog pada napajanja. Reset od vanjske jedinice ili korisnika u toku rada uređaja mora pokrenuti program od početka. Reset u nekim sustavima prekida idle ili power down režim rada. Reset kod 8051 mora trajati barem dva strojna ciklusa tj 24 perioda signala takta.



Sklop za generiranje reset signala integriran u 8051 sadrži otpornik i invertor sa Schmittovim okidnim sklopom. Izvana se dodaje kondenzator spojen prema izvoru napajanja. Ukoliko je bi kratak pad napajanja kondenzator se ne isprazni te se mikrokontroler ne resetira, te obično počne nepravilno raditi.



Slika prikazuje signal reset (kanal c2) te ALE (kanal c1) prilikom uključivanja sustava.

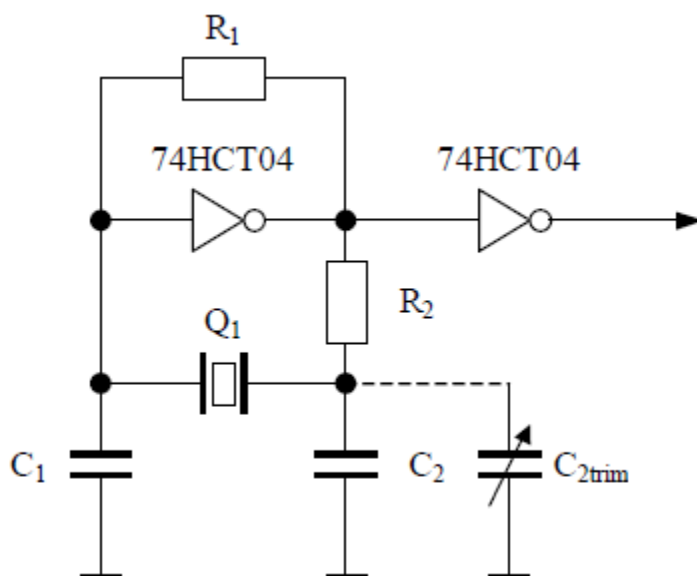
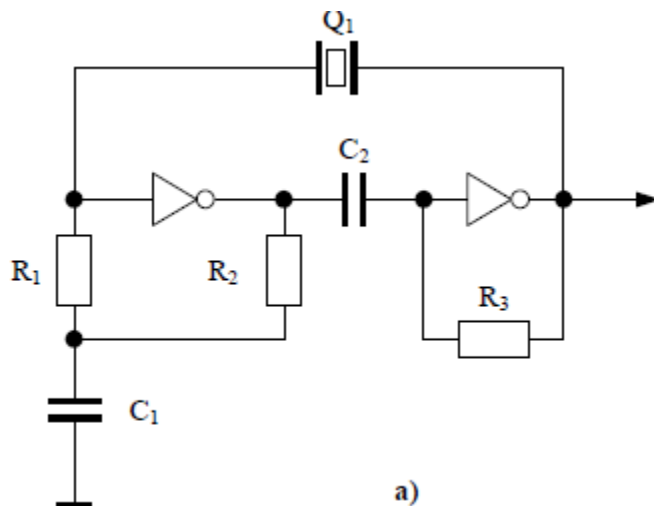


Sklop sadrži tipku za reset te diodu V1 kroz koju se kondenzator brzo isprazni te se time rješava problem kratkog pada napona jer će se i pri malom padu napona mikrokontroler uspješno resetirati. Razlog za upotrebu scmitovog okidnog sklopa leži u činjenici da se na eksponencijalu određenu vremenskom konstantom RC često superponiraju različite smetnje pa bi bez schmitovog okidnog sklopa dobilo mnoštvo impulsa u trenutku kad napon na ulazu invertora postigne napon praga ttl sklopa.

Integrirani sklopovi za reset (npr max 699) sadrže sklopovolje za reset kod uključivanja napajanja, wath dog sklop, komparator napona napajanja (daje reset kad napon padne ispod 4.65V).

25. Kad se kod oscilatora s digitalnim sklopovima koristi serijska, a kad paralelna rezonancija kristala? Nacrtati sheme odgovarajućih sklopova i objasniti princip rada. Koje sklopovlje za izvor takta sadrži mikrokontroler 80C51? Koja se rezonancija kristala koristi u tom slučaju?

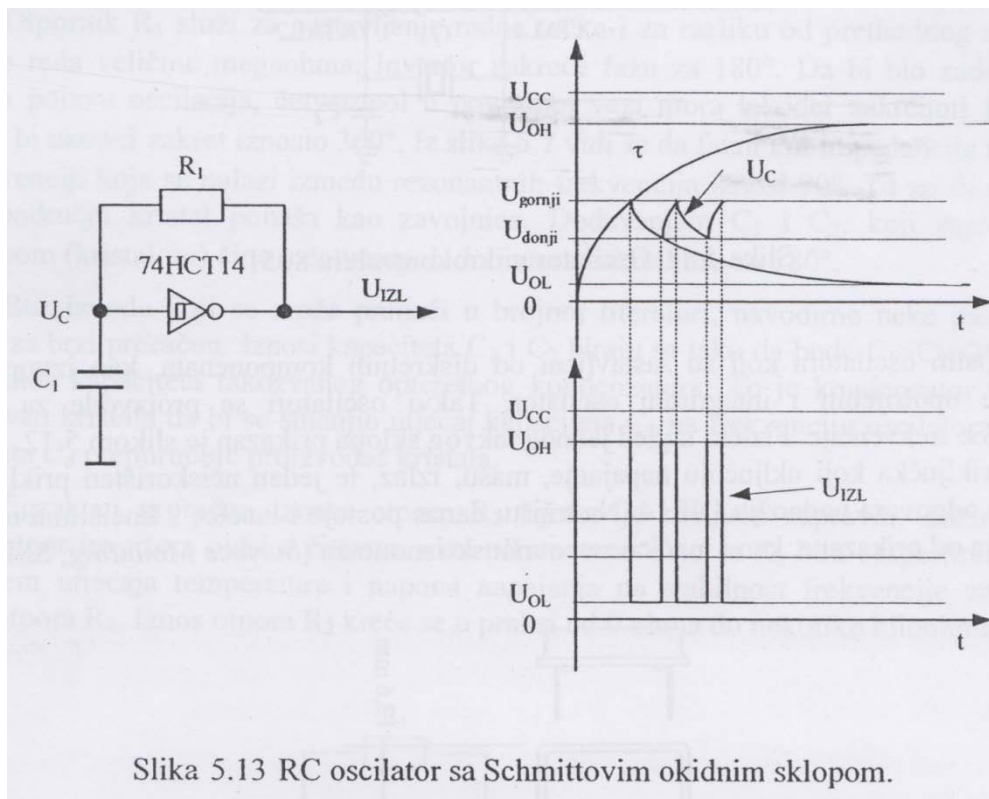
Serijska rezonancija koristi se kod sklopova s malim ulaznim otporom dok paralelna za sklopove s velikim ulaznim otporom u oba slučaja radi očuvanja faktora dobrote Q.



R₁ i R₂ su za pozicioniranje radne točke u linearnom radnom području, invertori da bude fazni zakret za 360° (svaki daje 180°), za serijsku R₁ i C₁ su da bi smanjili opterećenje na invertoru (da R i C ovi mogu biti sto manji, a Q ostati isti), C₂ služi za istosmjerno odvajanje invertora, kapaciteti C₁ i C₂ moraju biti sto veći da se impedancija njih zanemara. R₁ je reda veličine nekoliko stotina ohma, a R₂ i R₃ reda veličine kiloohma kod serijske. Kod paralelne R₁ je veličine megaohma te služi za postavljanje radne točke.

Unutar 80C51 postoji sklopovlje koje zajedno s vanjskim kristalom i kondenzatorima čini oscilator. Radi se o CMOS sklopu koji koristi paralelnu rezonanciju kristala.

26. Nacrtati shemu i pripadni vremenski dijagram te opisati rad RC oscilatora s invertorom koji sadrži Schmittov okidni sklop. Koje su prednosti i nedostaci ovog sklopa u odnosu na oscilator s kristalom?



Nakon priključenja napajanja kondenzator se puni od 0V do prema naponu izlaza invertora U_{OH} . Kad se kondenzator napuni do U_{gornji} (napon schmittovog okidnog sklopa) dolazi do preklapanja i na izlazu invertora pojavljuje se napon iznosa U_{OL} . Kondenzator se sada počne prazniti do napona U_{gornji} prema naponu U_{OL} . Kad se napon na kondenzatoru izjednači s U_{donji} schmittovog sklopa inverter opet preklapa i proces se ponavlja. Prednost RC oscilatora je brzo utitravanje, stacionarno stanje se postiže već nakon prvog impulsa. Nedostatak je manja točnost i stabilnost frekvencije oscilatora.

27. Navesti kada se za napajanje koriste serijski regulatori, a kada preklapajući izvori. Opisati sklopovlje za zaštitu koje se obično dodaje sklopovlju za napajanje tokom razvoja uređaja.

Preklapajući izvori (switching power supply) koriste se za sustave koji trebaju veliku snagu tj struje od nekoliko ampera, imaju veću učinkovitost i manje dimenzije od serijskih iste snage. Serijski regulatori koriste se za napajanje malenih računalnih sustava. Dioda V101 štiti sklop od obrnutog polariteta ulaznog napona. Također se dodaje zaštita od prevelikog iznosa ulaznog napona (dodaje se zenerova dioda koja provede na 20V te time progori osigurač). Ukoliko koristimo sklop u kojem se izvana dovodi već regulirani napon 5V ne smije se izgubiti 0.7V na diodi da to riješimo koristimo diodu koja provede na krivi predznak napona te tako uzrokuje pregaranje osigurača.

Sklopovlje za zaštitu dakle: sklopovi za zaštitu od krivog polariteta, te sklopovi za zaštitu od prevelikog ulaznog napona.

28. Objasniti razliku između sinkronog i asinkronog serijskog prijenosa. Opisati princip prijenosa podataka preko serijske veze RS232. Navesti redoslijed bitova i naponske razine signala. Opisati sklop MAX232?

Asinkroni prijenos je takav prijenos u kojem između jedinici koje komuniciraju ne prenosi takt za sinkronizaciju podataka, sinkroni prijenos ima takt za sinkronizaciju. Podatci se prenose u nizu jedan za drugim. Prvo se prenosi start bit (uvijek 0), sedam ili osam bitova podataka, stop bit na kraju (uvijek 1). Još se između podatka i stop bita može koristiti i bit pariteta. Naponske razine za RS232 su: 0- 3V do 15V, 1 -15V do -3V.

Podatci se unutar 8051 prenose preko dvije komunikacijske linije (RXD i TXD) a naponske razine su TTL kompatibilne za logičku nulu od 0 do 0.8V a za jedinicu od 2.4V do 5V.

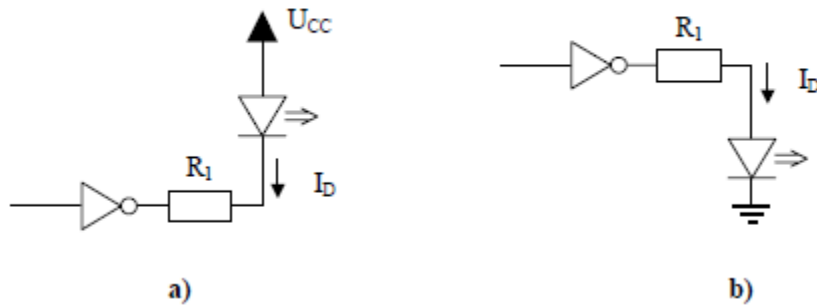
Sklop MAX232 služi za prilagodbu razina RS232 na TTL razine koje se koriste na mikrokontrolerima. Max232 je dakle integrirani pretvornik. Unutar sklopa MAX232 nalazi se udvostručivač napona i inverter napona koji rade na principu nabojske pumpe. Sadrži dva prijemna i dva odašiljačka kabela. Na izlazu daje +-10V a na ulazu je prilagođen za TTL/cmos razine.

29. Opisati ulogu brojila T1 kod komunikacije preko serijskog sučelja.

Brzina prijenosa (baud rate, Bit/s) u načinu rada 1 određena je brojiлом T1 koje radi u načinu rada 2. Podaci za inicijalizaciju brojila su izvedeni u tablici u službenom šalabahteru.

30. Nacrtati sklopovlje i pokazati kako se proračunavaju komponente sklopa za pogon svjetleće diode logičkim sklopom. Nacrtati sklop za pogon trošila pomoću tranzistora i MOSFET-a. Opisati način spajanja jedne znamenke 7-segmentnog pokazivača.

• pogon logičkim sklopom

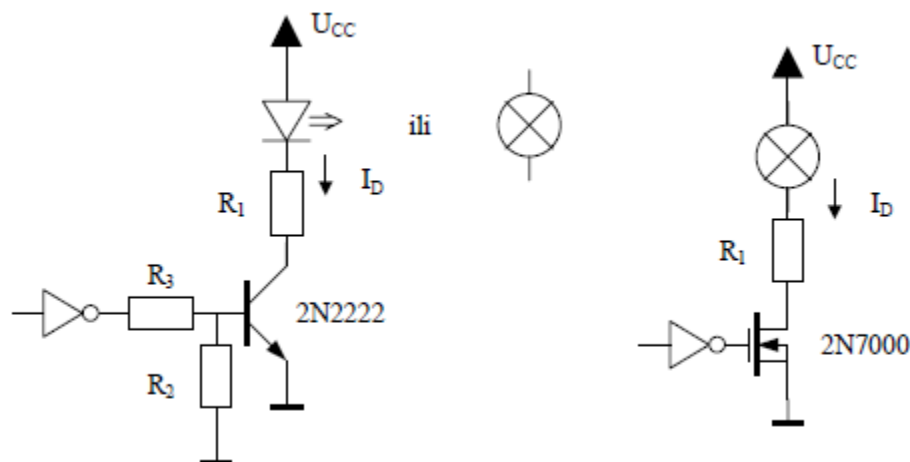


Struja koja teče kroz diodu ima iznos:

$$I_D = \frac{U_{CC} - U_D - U_{OL}}{R_1}$$

U_{CC} napon napajanja; U_D pad napona na diodi; U_{OL} napon na izlazu logičkog sklopa u stanju 0. Kod logičkih sklopova redovito je $U_{OL} > U_{OH}$ pa se češće koristi spoj diode prema napajanju nego prema masi (slika a). Pad napajanja na diodi obično iznosi oko 1.8V, struja od 3mA do 10mA, vrlo učinkovite led diode (haj efišnsi LED) od nekoliko desetaka do nekoliko stotina mikroA.

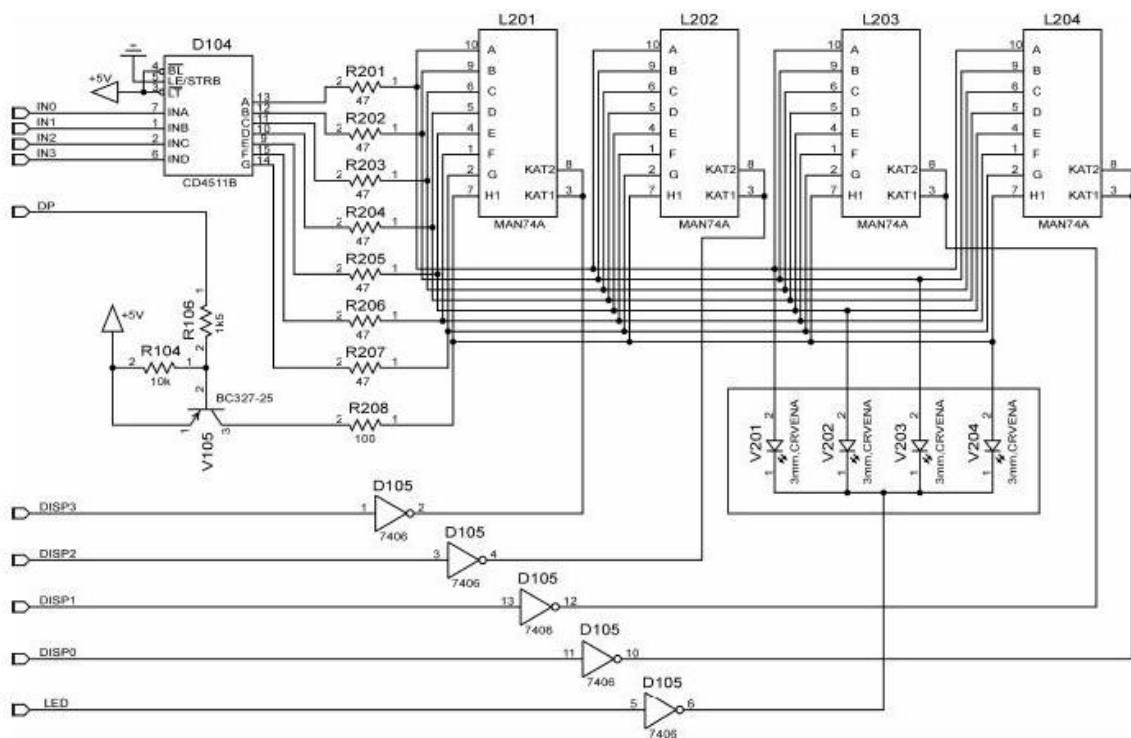
Na opisani način mogu se spajati i jača trošila, ograničenje predstavlja najveća struja koju sklop može primati I_{OL} odnosno davati I_{OH} . U tom slučaju pogon se izvodi pomoću bipolarnog tranzistora ili mosfeta.



Desno bipolarni tranzistor, lijevo mosfet. MOSFET tranzistori se proizvode za struje od desetak miliampera do nekoliko desetaka ampera.

Jedna znamenka se spaja na način kako je opisano. Pojedine znamenke pokaznika pogone se izravno iz digitalnog sklopa. Za pogon se može koristiti zaporni sklop (memorijski element, u njega upisujemo da li svijetli ili ne). Ukoliko zaporni sklop ne daje dovoljno veliku struju između njega i pokaznika stavlja se pojačalo (driver).

31. Nacrtati blokovsku shemu i pripadajući vremenski dijagram spoja 7-segmentnog pokazivača sa četiri znamenke koji radi u preklapajućem načinu rada. Što je ciklus osvježavanja. Koliko u praksi iznosi frekvencija osvježavanja? Zašto?

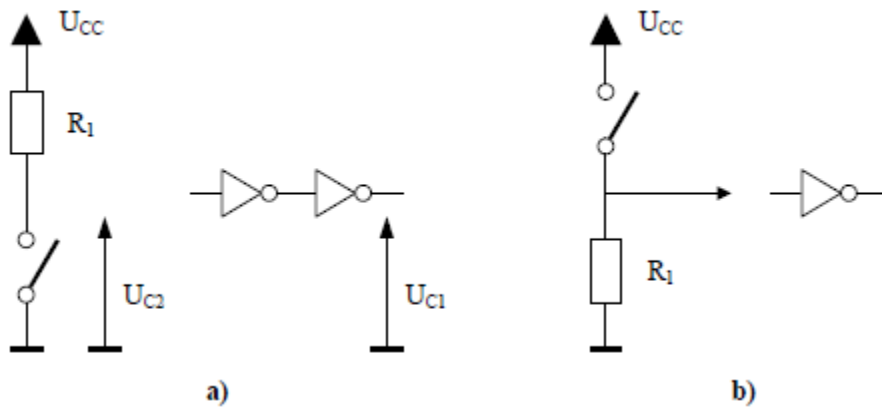


Sklopom se upravlja signalima IN0 do IN3, DP, DISP0 do DISP3 i LED. Ispis počinje od lijeva, na IN0 do IN3 dovodimo BCD broj koji treba pokazati lijeva znamenka, ma izlazi A do G pojavljuje se stanje visoke razine koje odgovara unesenom BCD broju. Nakon što su signali na A do G postali stabilni liniju DISP3 postavlja se u stanje 1, uključuje se lijevi pokaznik. Nakon nekog vremena (3.3ms) liniju DISP3 postavlja se u 0 te se postupak ponavlja za DISP2 DISP1 DISP0.

Vrijeme potrebno da se svih 5 znamenaka (4 znamenke i diskretna dioda) upale i ugase predstavlja ciklus osvježavanja. U našem slučaju to je $3.3 \times 5 = 16.7\text{ms}$. Nakon što se završi ciklus osvježavanja postupak se ponavlja. Ciklus od 16.7ms odgovara $f=60\text{Hz}$. U praksi se uzima frekvencija između 50 i 75 Hz. Te frekvencije uzimaju se zbog tromosti oka, frekvencije niže od 50Hz bi uzrokovala treperenje brojeva na pokazniku a frekvencije veće od 75Hz bi nepotrebno trošilo vrijeme procesora koji postavlja podatke na ulaze sklopa.

32. Nacrtati shemu osnovnog spoja tipke. Što je to istitravanje kontakata kod tipke? Koliko traje ta pojava? Kakve probleme izaziva i kako se ti problemi rješavaju? Dati primjer sklopovskog rješenja. Nacrtati dijagram toka programa osnovnog postupka za uklanjanje istitravanja.

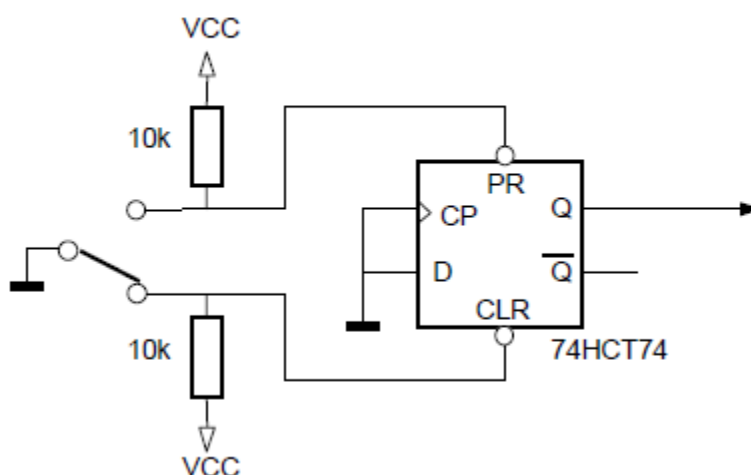
- osnovni spoj tipke



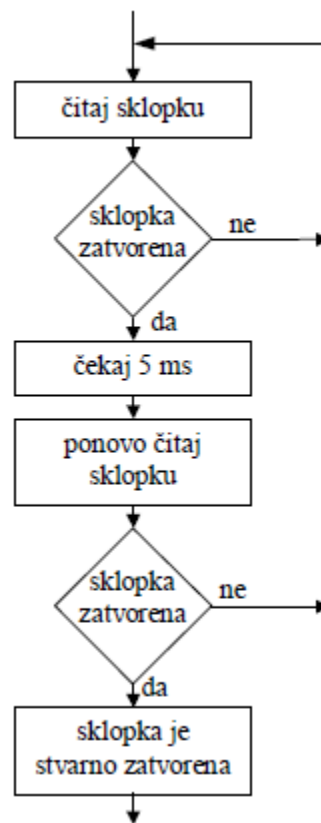
Spoj a) daje jedinicu kad je otpuštena tipka, 0 kad je pritisnuta, spoj b) obrnuto. Istitravanje (bouncing) je pojava kod koje dolazi u trenutku kad se kontakti tipke počinju spajati ili razdvajati. Nekoliko desetaka mikrosekundi (ponekad i duže) nakon uspostavljanja prvog kontakta pojavljuje se niz lažnih impulsa koji nastaju između početka uklapanja i trenutka potpunog spajanja kontakata. Kad bi ovakav signal bio spojen npr za ulaz za prekid nekog mikrokontrolera, kod svakog pritiska na tipku došlo bi do prekida nekoliko puta. Problem se rješava sklopovski i programski.

Bistabil zahtjeva sklopku s dva položaja, kod preklapanja sklopke bistabil mijenja stanje već nakon prvog kontakta. Vrijeme T u praksi se uzima između 2ms i 10 ms.

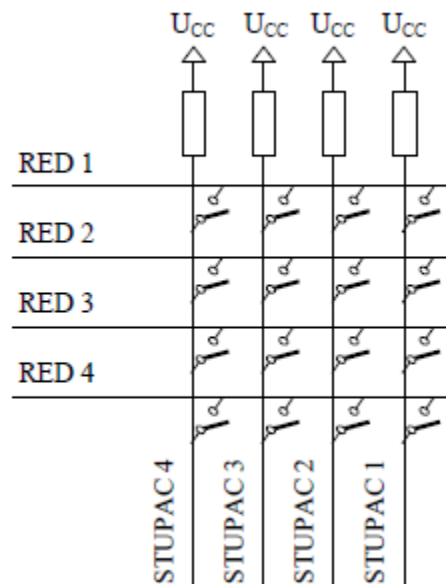
uklanjanje istitravanja sklopke pomoću bistabila



Programsko uklanjanje istitravanja



33. Nacrtati blokovsku shemu spoja matrične tipkovnice. Nacrtati dijagram toka programa koji rješava problem istitravanja pomoću prekida brojila T0.



Redovi se spajaju na izlazne priključke, a stupci na ulazne priključke. Na izlaz tj red damo logičku 0 a na ostale 1 te čitamo stupac po stupac, ako je tipka pritisnuta pročitat ćemo 0 a u ostalim stupcima 1.

Čitanje stanja s tipkovnice izvodi se u prekidnoj rutini koju pokreće prekid brojila T0 (svakih 50ms). U prekidnoj rutini brojila brojilo se opet inicijalizira te provjerava jeli tipka pritisnuta (gornji već objašnjeni algoritam), ukoliko je tipka prošli prekid bila pritisnuta, tipka je pritisnuta, inače nije.

34. Objasniti razliku između sinkronog i asinkronog serijskog prijenosa. Opisati I2C sabirnicu (sklopovlje, standardne brzine, osnovne pojmove, itd.). Nacrtati vremenski dijagram i opisati ciklus prijenosa podataka na sabirnici.

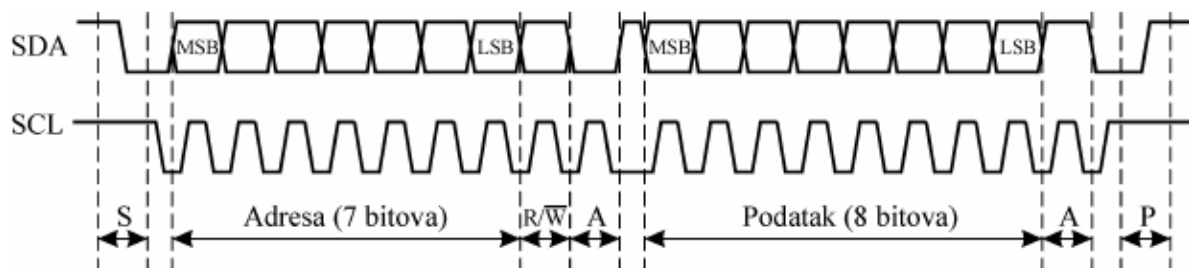
Asinkroni prijenos je takav prijenos u kojem između jedinici koje komuniciraju ne prenosi takt za sinkronizaciju podataka, sinkroni prijenos ima takt za sinkronizaciju. I2C sabirnica je sinkrona serijska dvosmjerna sabirnica namijenjena međusobnom povezivanju raznih digitalnih i upravljivih analognih sustava. primjer: mikrokontroleri, LCD, EEPROM memorije, RAM memorije i dr, pojačala s programibilnim pojačanjem, sintezatore frekvencije, separatori video signala, uređaji široke potrošnje. Broj komponenata koje se mogu izravno spojiti na sabirnicu ograničen je samo najvećim kapacitetom na svakoj liniji 400pF. Nije potreban glue logic – komponente se spajaju izravno.

Sva komunikacija ide preko SDA (serial data) i SCL (serial clock) linija. Brzine prijenosa su 100kbit/s u standardnom načinu rada, odnosno do 400kbit/s u brzom načinu rada. Izlazni stupnjevi komponenata koji pogone sabirnicu su tipa otvorenog kolektora.

Kod komunikacije postoji upravljač (master) – odašiljač (transmitter) te izvršilac (slave) – prijemnik (receiver). Postupak kojim se dodjeljuje sabirnica zove se arbitracija.

Prijenos podatka ide sljedećim redoslijedom: start bit (SDA iz 1 u 0 dok je SCL=1) → SCL prelazi u 0, dok je SCL=0 na SDA se postavlja MSB, SCL ide u 1 te opet u 0, za to vrijeme na SDA ide idući bit, postupak se ponavlja dok se ne prenese ukupno 8 bitova. Stop bit (SDA iz 0 u 1 dok je SCL=1). Jedino kod start i stop bita SDA se mijenja dok je SCL=1.

Start bit → 7 bitova adrese → 1 bit za R/W → SDA=1 → ako je izvršilac primio adresu postavit će SDA=0 taj bit naziva se potvrda (acknowledge) → ako je izvršilac primio adresu prenosi se idućih 8 bitova podataka.



35. Opisati protokol na primjeru pisanja u I2C memoriju. Opisati protokol na primjeru čitanja iz I2C memorije. Kad se koristi kombinacija čitanja i pisanja?

upravljač piše u izvršioca

	adresa	R/ \overline{W}	A	podatak	A	podatak	A	
S	1010110	0	0	10101101	0	01001101	0	P



upravljač (*master*)



izvršilac (*slave*)

Start bit -> adresa -> bit 0 (za upis) -> izvršilac potvrđuje s A=0 -> bajt podatka -> A=0 -> dok se ne pošalju svi podatci; stop bit koji daje upravljač

upravljač čita iz izvršioca

	adresa	R/ \overline{W}	A	podatak	A	podatak	A	
S	1010110	1	0	10101101	0	01001101	1	P



upravljač (*master*)



izvršilac (*slave*)

Start bit -> adresa -> bit 1 za čitanje -> izvršilac potvrđuje s A=0 -> izvršilac šalje bajtove podatka te upravljač daje A=0 -> izvršilac prestaje slati podatke kad upravljač da A=1 i stop bit.

Kombinacija čitanja i pisanja koristi se kad treba čitati iz neke serijske memorije. Prvo se šalje adresa lokacija s koje se čita a nakon toga se pročita jedan ili više podataka. Dakle upravljač prvo u memoriju upiše podatak s adresom memorijske lokacije te nakon toga vrši operaciju čitanja.

kombinacija čitanja i pisanja

	adresa	R/ \overline{W}	A	podatak	A		adresa	R/ \overline{W}	A	podatak	A	
S	1010110	0	0	10101101	0	S	1010110	1	0	01001101	1	P

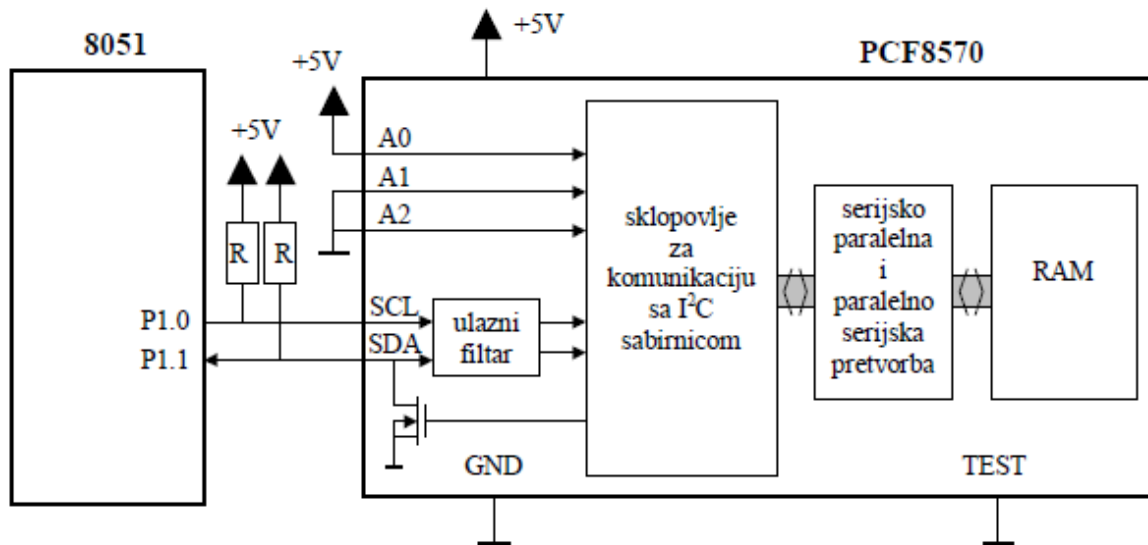


upravljač (*master*)



izvršilac (*slave*)

36. Nacrtati i opisati shemu spajanja I2C memorije na sustav s mikrokontrolerom. Objasniti kako se memorija adresira tokom komunikacije.



Kapacitet memorije iznosi 256 bajtova. Memorija se nalazi u kućištu s 8 priključaka. Masa, napajanje, SDA, SCL te tri adresne linije. Sedam bitna adresna sabirnica sastoji se od 4 fiksna bita i 3 bita koja možemo definirati po vlastitoj želji. 1010001. Adresa= 1010A2A1A0. Memorija se sastoji od sklopovlja za komunikaciju s I2C sabirnicom, sklopovlja za paralelno serijsku i serijsko paralelnu pretvorbu.

· iznos otpora otpornika za zaključenje sabirnice

$$R = \frac{t_r}{C_{BUS}}$$

Tr najveće vrijeme porasta signala porasta na sabirnici, Cbus najveći dopušteni kapacitet na pojedinim linijama sabirnice. R=2.5Kiloohma.

- Protokol za upis dva bajta u RAM

	adresa RAM-a	$\overline{R/W}$	A	početna adresa u RAM-u	A	prvi podatak	A	drugi podatak	A	
S	1010001	0	0	00000010	0	10100000	0	10101111	0	P

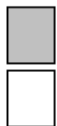


8051

PCF8570

- Protokol za čitanje dva bajta iz RAM-a

Serijski zapis podataka iz memorije														
	adresa RAM-a	$\overline{R/W}$	A	početna adresa u RAM-u	A		adresa RAM-a	$\overline{R/W}$	A	prvi podatak	A	drugi podatak	A	
S	1010001	0	0	00000010	0	S	1010001	1	0	10100000	0	10101111	1	P



8051

PCF8570

Možemo upisivati ili čitati s neke adrese između 0 i 255. Upisujemo tako da prvo šaljemo adresu memorije, nakon toga početnu adresu u RAM-u te onda podatke. Čitamo na kombinirani način: prvo šaljemo adresu RAM-a, nakon toga početnu adresu u RAM-u, opet adresu RAM-a te nakon toga čitamo podatke s adrese.

37. Objasniti ulogu tipa memorije u deklaraciji varijable. Čime je određen tip memorije kad u deklaraciji varijable on nije eksplicitno naveden. Kako se deklariraju registri posebne namjene, bitovi u registrima posebne namjene i bitovi podatkovnoj memoriji? Kada koristimo deklaraciju varijable tipa "volatile"? (Pitanje se odnosi na programski paket Keil uVision2.)

Sustavi s mikrokontrolerom sadrže razne memorijske prostore, ovisno o tome koliko memorije postoji, da li su one vanjske ili unutrašnje itd.

Memorijski prostori

Tip memorije	Opis
data	Direktno adresabilni unutrašnji RAM (donjih 128 bajtova unutrašnjeg RAM-a).
bdata	Bit adresabilna unutrašnja memorija (adrese od 20h do 2Fh)
idata	Indirektno adresabilni unutrašnji RAM (gomjih 128 bajtova unutrašnjeg RAM-a).
pdata	Vanjska memorija od 256 bajtova (dostupna preko MOVX @Ri).
xdata	Vanjski RAM od 64k (dostupan preko MOVX @A+DPTR).
code	Programska memorija (može se samo čitati pomoću MOVC @A+DPTR).

Kroz deklaraciju varijable određuje se u kojem memorijskom prostoru će se varijabla nalaziti. Data najbrži, idata sporiji, xdata najsporiji.

Npr:

Unsigned int xdata pero; vanjski ram
signed char dana a12; donjih 128 bajtova unutrašnjeg rama

Ako se u deklaraciji ne navede tip memorije, odabrati će se memorijski prostor ovisno o zadanom memorijskom modelu koji može biti SMALL, COMPACT, LARGE. SMALL model varijable se po defaultu stavlja u data prostor, COMPACT model varijable stavlja u idata dok LARGE stavlja u xdata.

Deklaracija registra posebne namjene (SFR registar):

Sfr P0=0x80; deklaracija SFR registra
sbit TI=0x99; deklaracija bit u sfr registru

Ili

Sfr P0=0x80;
sbit P0_1=P0^1;

Ukoliko je neka varijabla promjenjivog sadržaja, ona se deklarira kao varijabla tipa volatile. To je potrebno u situacijama u kojima više uzastopnih čitanja iste memorijske lokacije daje različite rezultate.

```
Volatile unsigned char xdata *ad_convertor=0xFFFF;
```

38. Objasniti razliku između generičkih i memorijski određenih pokazivača. Dati primjere deklaracija. (Pitanje se odnosi na programski paket Keil uVision2.)

Memorijski određene pokazivače koristimo kad znamo na koji memorijski prostor oni pokazuju i kad se taj prostor ne mijenja tokom izvođenja programa.

```
Char data *pok1; troši 1 bajt  
char xdata *pok2; troši 2 bajt  
pok1=&pero; dozvoljeno samo ako je pero u dana prostoru
```

Generički pokazivači mogu pokazivati na bilo koji memorijski prostor.

```
Char *pok3; troši 3 bajta
```

Sam pokazivač nalazi se u prostoru koji određuje model ali možemo to odrediti i sami

```
Data int xdata *pok5; pokazivač pok5 nalazi se u data prostoru  
int xdata *data pok5; ista stvar
```

39. Objasniti razliku između funkcija tipa "reentrant" i "nonreentrant". Kada se one koriste i kako se definiraju? (Pitanje se odnosi na programski paket Keil uVision2.)

Reentrant funkcije su funkcije u koje se može ponovo ulaziti; mogu se pozivati rekurzivno; mogu se pozivati iz prekidnih funkcija i glavnog programa; lokalne varijable se pohranjuju na simuliranom stogu s pokazivačem ?C_IBP

```
Char funk5(int a) reentrant
```

Non-reentrant funkcije ne mogu se pozivati rekurzivno, ne mogu se pozivati iz prekidnih funkcija i glavnog programa istovremeno; vrlo su učinkovite; podatci se drže u registrima ili fiksnim memorijskim lokacijama.

```
Char funk6 (int a); default je non reentrant
```


40. Kako se u funkcije prenose argumenti? Kad se koristi promjena registarske banke? Kako se definiraju funkcije koje koriste drugu registarsku banku? Na što treba paziti prilikom rada sa više registarskih banaka? (Pitanje se odnosi na programski paket Keil uVision2.)

u funkciju se argumenti prenose preko registara ili memorijskih lokacija, a iz funkcije preko registara.

Prijenos ulaznih argumenata preko registara.

	char, 1-bajtni pok.	int, 2-bajtni pok.	long, float	generički pokazivač
1. parametar	R7	R6, R7	R4 do R7	R1, R2, R3
2. parametar	R5	R4, R5	R4 do R7	R1, R2, R3
3. parametar	R3	R2, R3	-	R1, R2, R3

Prijenos izlazne vrijednosti funkcije registara.

Izlazna vrijednost	Registar	Opis
bit	Carry-Flag	
char	R7	
int	R6, R7	MSB u R6, LSB u R7
long	R4 do R7	MSB u R4, LSB u R7
float	R4 do R7	32 bita IEEE format (eksponent u R7)
pointer	R1, R2, R3	selektor u R3, MSB u R2, LSB u R1

primjer

```

funk1 (int a)    /* prenosi argument a preko registara R6 i R7 */
funk2 (int b,int c, int *d)
                /* prenosi argument b preko reg. R6 i R7 */
                /* argument c preko R4 i R5, */
                /* a argument d preko R1, R2 i R3 */
funk3 (float g, char h)
                /* prenosi argument g preko registara R4 do R7, */
                /* a argument h preko mem. lokacije u RAM-u */

```

Kad se na mikrokontroleru paralelno obavljaju dva zadatka (glavni program i prekidni), pojedini zadatci ne smiju kvariti posao drugih. Prekidni program ne smije promijeniti sadržaj trenutno aktivne registarske banke. To se može izbjeći PUSH POP naredbama kojima na stog pohranjujemo sve registre. Drugi način čuvanja sadržaja registra je promjena aktivne registarske banke.

Void funk10(void) using 3; funk10 radi s bankom b3

Prilikom rada sa više registarskih banka treba paziti jer funkcije koje se pozivaju iz drugih funkcija trebaju raditi s istim bankama kao i funkcije iz kojih su pozvane. Promjena banke nije uvijek korisna kad funkcija vraća podatak već je pogodnija funkcija tipa void.

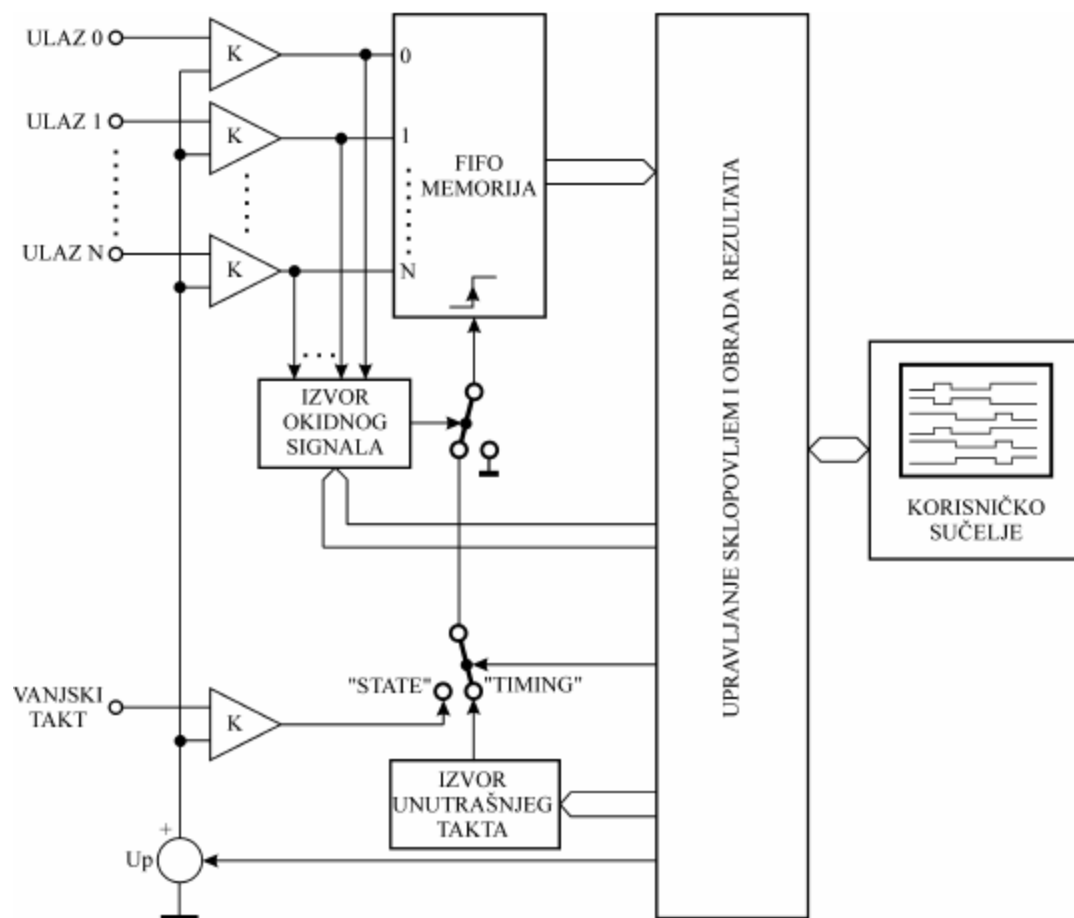
41. Kako se definira funkcija za prekidnu rutinu. Po čemu se ona, tj. kod koji napravi prevodilac, razlikuje od koda ostalih funkcija? Kakav učinak na kod ima promjena registrarske banke? (Pitanje se odnosi na programski paket Keil uVision2.)

Funkcija za prekidnu rutinu deklarira se:

```
Void f_prekid (void) interrupt n using m
```

Po tome što prekidna funkcija sprema sadržaj svih registara koji će biti korišteni u funkciji (ACC, B, DPH, DPL, PSW). Ukoliko ne postoji atribut using m spremi će se i cijela registrarska banka.

42. Nacrtati blokovsku shemu logičkog analizatora i opisati njegov princip rada.



Logički analizator možemo promatrati kao N-kanalni digitalni osciloskop čiji analogno digitalni pretvarač ima rezoluciju 1 bit (stanja 0 i 1). Ta pretvorba ostvaruje se komparatorima na čiji jedan ulaz dovodimo promatrani signal a na drugi ulaz praga komparacije. Prag je promjenjiv i može se prilagoditi tehnologiji logičkih sklopova koje analiziramo. Izlazi se dovode na ulaze FIFO memorije. Trenutci upisa određeni su taktom koji može biti doveden izvana ili iznutra. Takt se dovodi na memoriju (time upravlja sklopovlje za okidanje), u određenom trenutku takt se prekida te se time zamrzava stanje memorije. Čitanje iz memorije te upravljanje pojedinim dijelovima sustava obavlja računalni sustav. ukoliko je izabran unutarnji izvor takta ulazni signali bit će uzorkovani u jednakim

vremenskim razmacima (timing način rada). Vanjski izvor koristi se kad se želi analizirati ponašanje sklopa u diskretnim trenucima karakterističnim za sklop (state način rada).

Okidni signal formira se od na osnovi ulaznih signala (izlaza iz komparatora).

Kad ulazni signal zadovolji uvjet okidanja uzorkovanje se može prekinuti (vidimo signale prije pojave okidnog), propustiti još toliko podataka da se napuni pola memorije (snimljeni signali prije i poslije okidnog), propustiti još toliko podataka da se napuni cijela memorija (uzorci signala snimljeni nakon pojave okidnog signala)

Zadatak 2 (3 boda)

Napisati program u assembleru koji neprekidno čita stanje skupa priključaka P1, te ovisno o pročitanoj podacima izvodi slijedeće akcije

- ako je pročitana podataka 0, ne čini ništa
- ako je pročitana podataka 1, postavlja bit 1 skupa priključaka P3 u stanje 1
- ako je pročitana podataka 2, postavlja bit 1 skupa priključaka P3 u stanje 0
- inače, upisuje pročitani podatak u unutrašnju podatkovnu memoriju na adresu 7F

```
CSEG AT 00H
MOV R1, #7Fh

PETLJA:    MOV A, P1
           CJNE A, #0d, JEDAN
           JMP PETLJA

JEDAN:     CJNE A, #1d, DVA
           SETB P3.1
           JMP PETLJA

DVA:       CJNE A, #2d, INACE
           CLR P3.1

INACE:     MOV @R1, A
           JMP PETLJA

           END
```

Zadatak 2 (4 boda)

Napisati program u assembleru koji neprekidno čita stanje skupa priključaka P1, te svaki put kad se promijeni podatak, ovisno o pročitanoj podatku izvodi slijedeće akcije:

- ako je pročitani podatak 0, ne čini ništa
- ako je pročitani podatak 1, povećava sadržaj registra R1
- inače, upisuje pročitani podatak na skup priključaka P3

```
                                CSEG AT 00H

PETLJA:                        MOV A,P1
                                CJNE A,#0d,JEDAN
                                JMP PETLJA

JEDAN:                          CJNE A,#1d, INACE
                                INC R1
                                JMP PETLJA

INACE:                          MOV P3, A
                                JMP PETLJA

                                END
```

Zadatak 2 (4 boda)

Na računalni sustav s mikrokontrolerom 8051 spojene su dvije 8-bitne vanjske jedinice iz kojih se čita. Jedinice se vide u vanjskom podatkovnom prostoru, i to prva na adresi 0xFFFF i druga na adresi 0xFFFE. Osim vanjskih jedinica, na priključku 0 skupa P1 spojena je svjetleća dioda koja svijetli dovođenjem visoke razine napona na priključak mikrokontrolera. Potrebno je neprekidno čitati podatke iz jedinica, te ako je podatak iz prve jedinice veći ili jednak od podatka iz druge jedinice upaliti svjetleću diodu. Potrebno je napisati program u assembleru.

```
CSEG AT 00H

PETLJA:      MOV DPTR, #FFFEh
              MOVX A, @DPTR
              MOV R0, A

              MOV DPTR, #FFFFh
              MOVX A, @DPTR

              SUBB A, R0

              JC MANJI
VECI:        SETB P3.0
              JMP PETLJA

MANJI:       CLR P3.0
              JMP PETLJA

END
```

Zadatak 3 (4 boda)

Na računalni sustav s mikrokontrolerom 8051 spojena je tipka kod koje je istitravanje riješeno sklopovski. Tipka je spojena na priključak 0 skupa P1. Potrebno je svakim pritiskom tipke invertirati stanje priključka 1 istog skupa 300 μ s nakon što je tipka pritisnuta. Pretpostaviti da se tipka pritiska u intervalima mnogo većim od 300 μ s. Za generiranje vremenskog intervala potrebno je koristiti prekid koji daje brojilo T0, uz pretpostavku da frekvencija takta iznosi 12 MHz. Napisati program i odgovarajuću prekidnu funkciju u C jeziku.

```
#include <reg51.h>
#include <stdio.h>

void prekid (void)

sbit tipka = P1^0;
sbit iz = P1^1;

void main(void){
/*Inicijalizacija brojila*/
    TMOD = 0x01;
    IE = 0x82;

    while(1){
        if(tipka==1){
            TL0 = 0xD4; 2^16-300=65236 pa to pritvorit u hex
            TH0 = 0xFE;
            TR0 = 1;
        }
    }
}

void timer() interrupt 1 {
    TR0 = 0;
    if(iz==0) iz = 1;
    else iz = 0;
}
```

Zadatak 3 (4 boda)

Na računalni sustav s mikrokontrolerom 8051 spojeni su 8-bitni AD pretvarač i DA pretvarač koji se vide u vanjskom memorijskom prostoru na adresama 0xFFFF i 0xFFFE. Na ulaz za vanjski prekid 0 spojen je izvor takta uzorkovanja. Potrebno je napisati program u C jeziku, te odgovarajuću prekidnu funkciju koja kad dođe zahtjev za prekid pročita podatak iz AD pretvarača i upiše ga u DA pretvarač.

```
#include<stdio.h>
#include<reg51.h>
```

```
Void prekidna(void)
```

```
main () {
    IE=0x81;
    while(1);
}
```

```
void prekidna (void) interrupt 0 {
    volatile unsigned char xdata *ulaz,*izlaz;
    ulaz=0xFFFF;
    izlaz=0xFFFE;
    *izlaz=*ulaz;
}
```

Zadatak 3 (3 boda)

Napisati program koji na priključku 0 skupa priključaka P1 daje pravokutni signal perioda 200 μ s. Pretpostaviti da mikrokontroler radi na taktu frekvencije 12 MHz.

Program treba napisati u jeziku C, uz korištenje brojila i prekida.

```
#include<stdio.h>
#include<reg51.h>
Void prekidna (void)
    Sbit izlaz=P1^0;

main () {
    TMOD=0x02;
    TH0=56d;
    TL0=56d;
    IE=0x82;
    IP=0x02;
    TR0=1;

    while(1);
}

void prekidna (void) interrupt 1 {
    TF0 = 0;
    if(izlaz==0) izlaz = 1;
    else izlaz = 0;
}
```