Mikroprocesorska elektronika

Predavanje IX

Sadržaj predavanja

- Struktura ulazno/izlaznog porta opšte namene
- Povezivanje uređaja pomoću ulazno/izlaznog porta opšte namene
- Povezivanje prekidača i tastatura
- Povezivanje displeja

Struktura ulazno/izlaznog porta opšte namene

Struktura ulazno/izlaznog porta opšte namene I

- Nakon što je uspostavljena osnovna struktura embeded sistema (povezan je procesor sa memorijskim modulima i eventualno nekim unutrašnjim periferijama pomoću jedne ili više sistemskih magistrala) sledeći korak je da se uspostavi veza sa okruženjem

- Ova veza sa spoljašnjim svetom obično se ostvaruje korišćenjem ulazno/izlaznih (I/O) portova opšte namene (GPIO)

- U okviru ovog predavanja upoznaćemo se sa osnovnom strukturom, mogućnostima, konfiguracijom i električnim karakteristikama GPIO portova

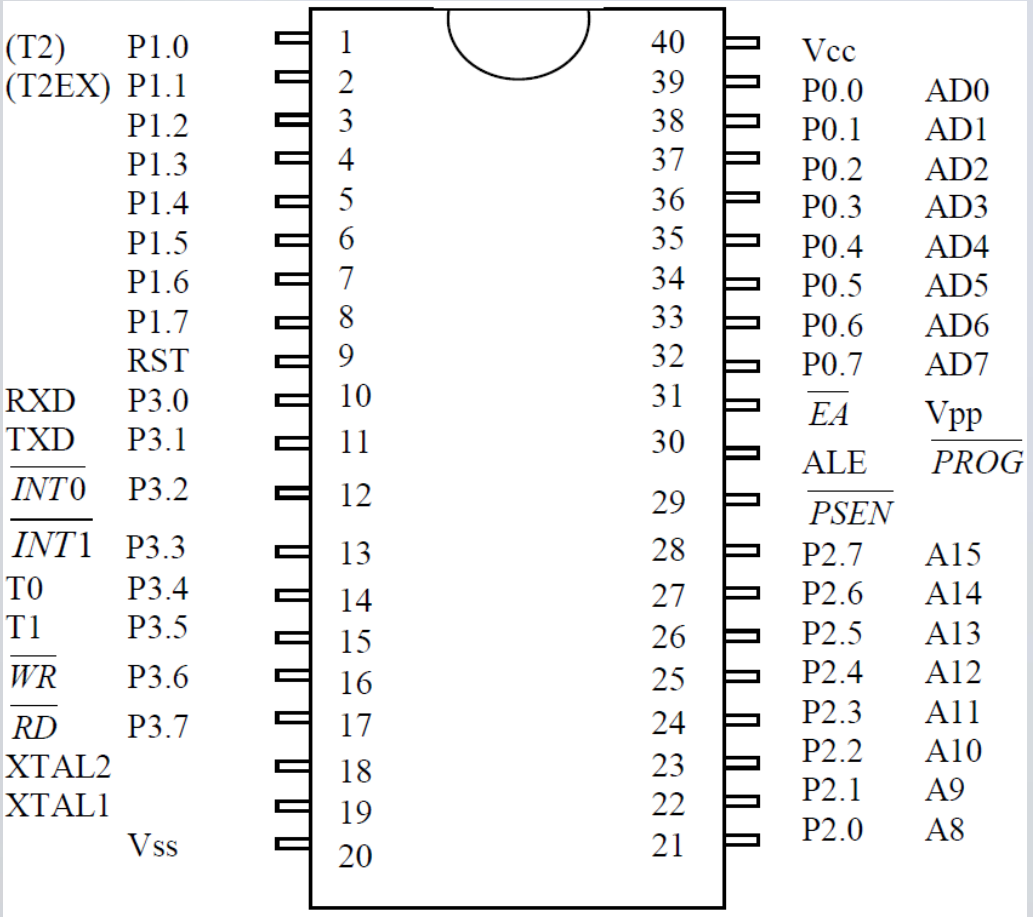
- Nakon toga biće analizirani načini povezivanja nekih najčešće korišćenih klasa periferijskih jedinica sa mikrokonotrolerom pomoću GPIO portova

Struktura ulazno/izlaznog porta opšte namene II

- Jedni od najvrednijih resursa mikrokontrolera su njegovi ulazno/izlazni portovi opšte namene (GPIO)
- GPIO portovi pružaju mogućnost da mikrokontroler komunicira sa svojim okruženjem preko skupa digitalnih linija koje mogu biti konfigurisane da rade kao ulazni ili izlazni kanali
- GPIO port sastoji se iz grupe ulazno/izlaznih linija koje se mogu, pojedinačno ili grupno, konfigurisati kao ulazni ili izlazni kanali
- Većina mikrokontrolera grupiše po 8 ulazno/izlaznih linija u jedan GPIO port, iako ovo nije generalno pravilo, pa se mogu naći mikrokontroleri koji grupišu po 4, 6, 16 ili 32 linije u jedan GPIO port
- GPIO linije postaju dostupne „spoljašnjem svetu“ preko posvećenih ili multipleksovanih nožica kućišta u koji se smešten mikrokontroler

Struktura ulazno/izlaznog porta opšte namene III

- U slučaju mikrokontrolera 8051 postoje 4 GPIO porta, označeni sa P0, P1, P2 i P3
- Svaki od ovih GPIO portova je 8-bitni
- U okviru svakog GPIO porta, svaka od linija može se individualno konfigurisati kao ulazni, odnosno izlazni kanal
- Neke od linija GPIO portova P0-P3 direktno se vode na odgovarajuće nožice kućišta (na primer linije P1.2 do P1.7)
- Većina linija GPIO portova P0-P3 je dostupna „spoljašnjem svetu“ preko multipleksiranih nožica
- Na ovaj način izvršena je redukcija broja potrebnih nožica na kućištu, što u znatnoj meri utiče na konačnu cenu mikrokontrolera



Struktura ulazno/izlaznog porta opšte namene IV

- I/O linija koja je konfigurisana kao ulazni kanal omogućava mikrokontroleru da prihvata binarne informacije o statusu spoljašnje periferijske jedinice koja je poveza na I/O liniju

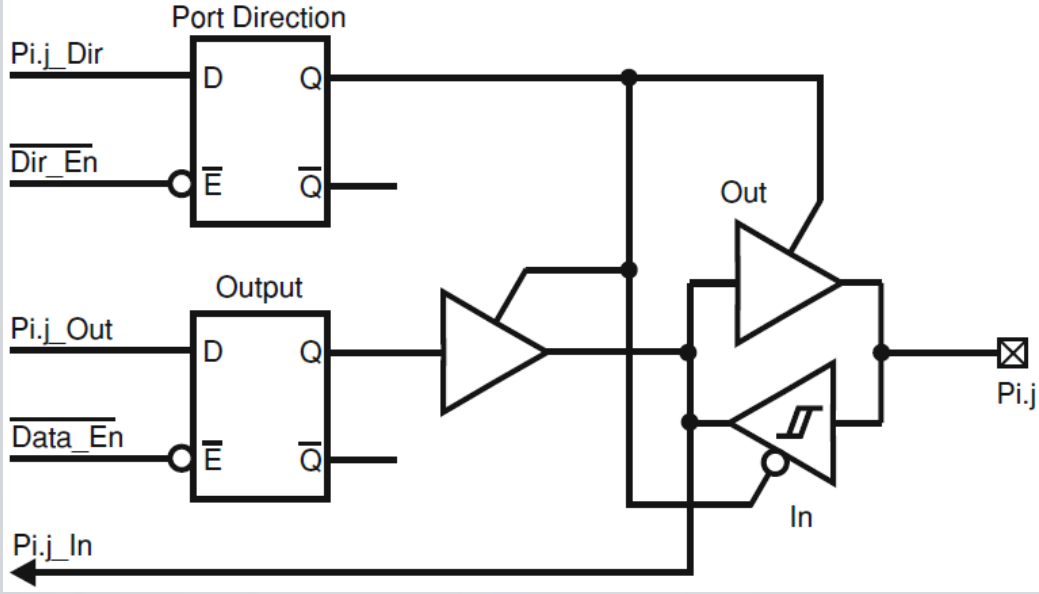
- Svaki put kada želimo da saznamo status nekog spoljašnjeg događaja, status nekog spoljašnjeg uređaja ili status nekog binarnog signala čija se vrednost može reprezentovati ili kao logička jedinica ili logička nula, ulazni I/O portovi predstavljaju pravi izbor za povezivanje

- Kada je I/O linija konfigurisana kao izlazna, I/O linija pruža mikrokontroleru mogućnost kontrole binarnog statusa neke spoljašnje periferije ili signala

- Akcije kao što je uključivanje ili isključivanje LED indikatora, zujalica, ili bilo koja akcija koja se može realizovati postavljanjem naponskog nivoa na visoku ili nisku vrednost može se realizovati korišćenjem izlazne I/O linije

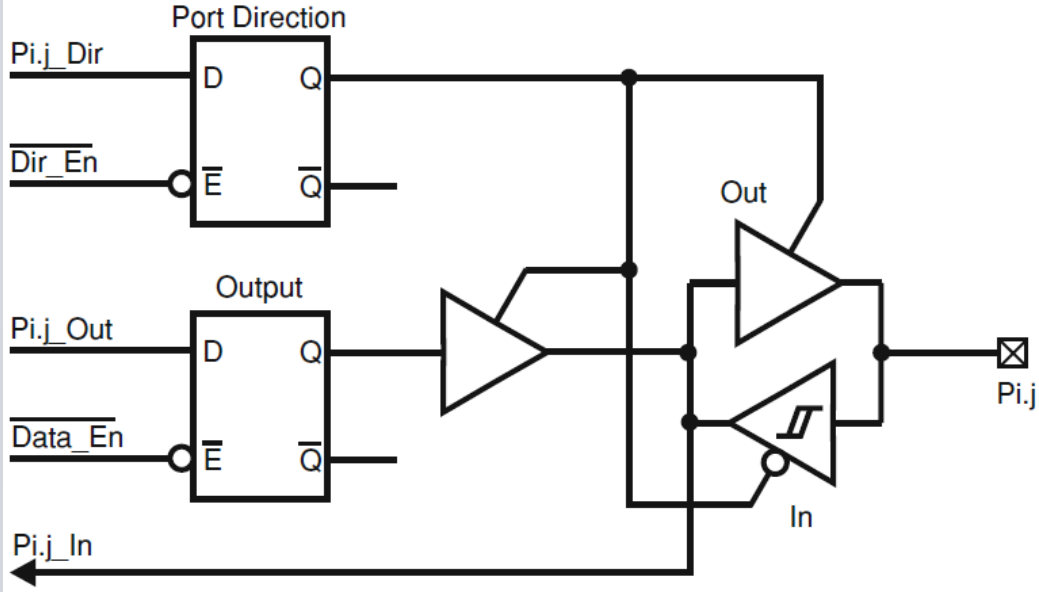
Generička struktura ulazno/izlazne nožice I

- Da bi se omogućila komunikacija procesora sa okruženjem korišćenjem digitalnog kanala, neophodan je odgovarajući interfejs koji omogućuje povezivanje fizičkog I/O pina sa sistemskim magistralama procesora
- Uopštena struktura ovog interfejsa prikazana je na slici desno
- Da bi se ostvarila **ulazna operacija (In)**, minimalno je postojanje nekog bafera, označenog sa „In“ na slici desno
- Ulazni bafer je obično realizovan kao standardni logički bafer sa “tri-state” kontrolom pomoću koje je moguće izlaz bafera postaviti u stanje visoke impedanse
- **Tri-state opcija** omogućava da se nožica konfiguriše kao ulazna ili izlazna, a da se pri tome izbegne konflikt između različitih signala



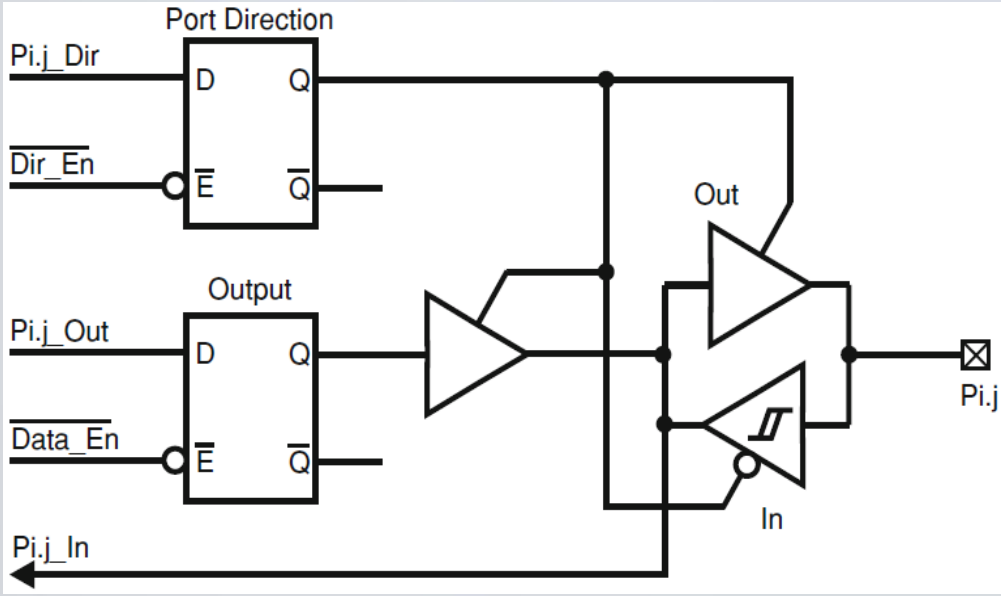
Generička struktura ulazno/izlazne nožice II

- Kada je u leč „Port Direction“ upisana logička nula „In“ bafer će biti aktiviran, dok će „Out“ bafer biti deaktiviran, te će se nožica ponašati kao ulazni port
- U ovom modu, **ulazni podaci ($P_{i,j}$)** teku kroz „In“ ulazni bafer preko $P_{i,j}$_*In* linije ka **unutrašnjoj magistrali podataka**
- Često je signal koji se dovodi na ulazni port sporo promenljiv (ima blage ivice) ili je zašumljen
- Ovakvi ulazni signali mogu izazvati neželjeno ponašanje digitalnih električnih kola
- U ovim slučajevima, ulazni bafer „In“ često se realizuje kao Šmit-triger ulazni bafer, koji ubrazava tranzicije signala i uklanja šum prisutan u signalu



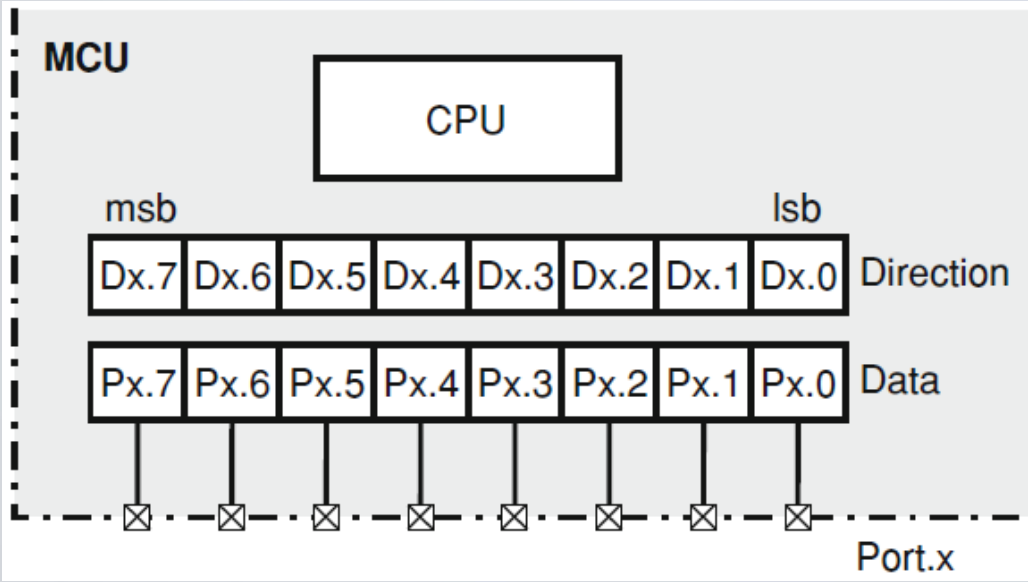
Generička struktura ulazno/izlazne nožice III

- U slučaju **izlazne transakcije** koja treba da se realizuje preko nožice, vrednost upisana od strane procesora preko ***P*$_{i,j}$*_Out* porta** mora biti prisutna na nožici $P_{i,j}$ sve dok procesor ne upiše novu vrednost koja treba da se pojavi na nožici
- Ovakva funkcionalnost zahteva postojanje još jednog **leča „Output“**
- Izlaz ovog leča je takođe baferovan korišćenjem dodatnog tri-state bafera
- **Smer toka podataka**, ulazni ili izlazni, u kojem se trenutno koristi nožica kontroliše se pomoću uključivanja i isključivanja odgovarajućih tri-state bafera pomoću sadržaja **„Port Direction“ leča**
- Na slici nije prikazana dodatna logika koja je neophodna da bi se ***P*$_{i,j}$*_Dir***, ***P*$_{i,j}$*_In*** i ***P*$_{i,j}$*_Out*** linije povezale na unutrašnju magistralu mikrokontrolera i generisali ***Dir_En*** i ***Data_En*** **upravljački signali**



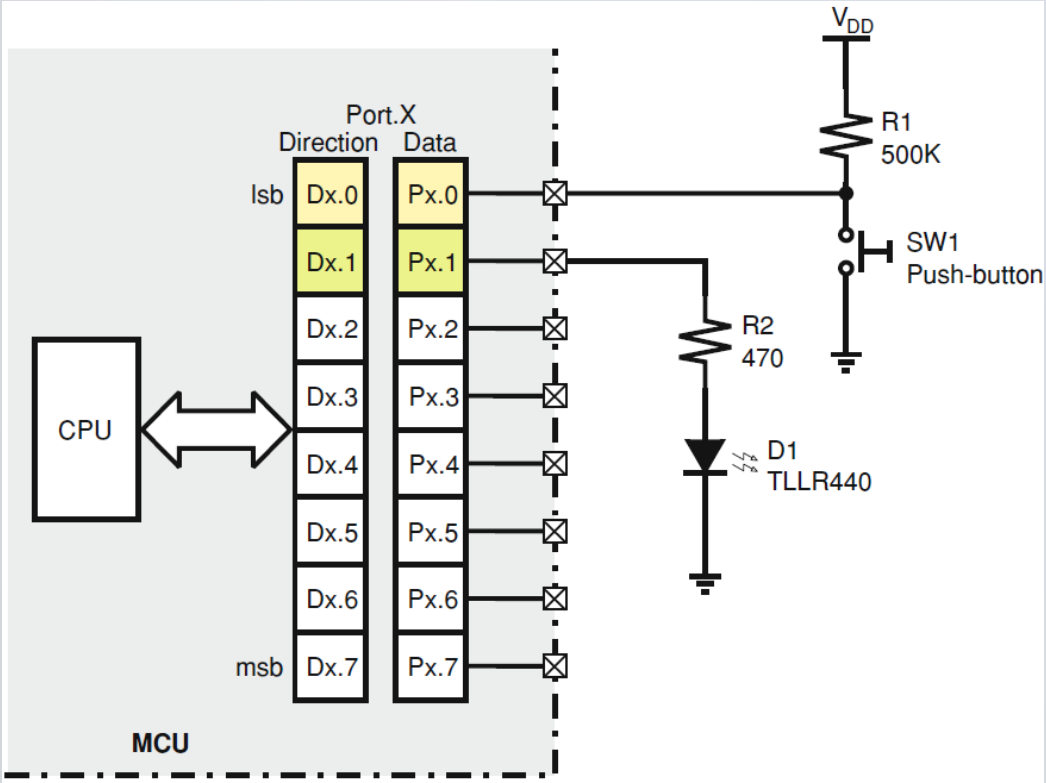
Konfigurisanje GPIO portova

- Da bi smo koristili GPIO port potrebno je da:
  - Konfigurišemo smer toka podataka (kao ulazni ili izlazni)
  - Upišemo ili pročitamo podatke sa porta
- Obzirom da su I/O linije grupisane unutar GPIO porta, svim I/O linijama u datom portu pristupa se pomoću jedne, zajedničke adrese
- Isto pravilo važi i u slučaju kontrole smera toka podataka kroz individualne I/O linije
- Na ovaj način je rad sa GPIO portovima maksimalno olakšan jer se oni tretiraju kao obični **registri** od strane softvera



Primer korišćenja GPIO porta I

- Posmatrajmo 8-bitni GPIO port, Port.X (Px), na koji su priključeni taster SW1 (preko pina 0, Px.0) i LED dioda D1 (preko pina 1, Px.1)
- Pretpostavimo da su direkcioni i registar podataka, asocirani GPIO portu Port.X, smešteni na **adresama PxDir** i **PxDat**, respektivno
- Takođe prepostavimo da direkcioni bit koji je postavljen na vrednost „1“ konfiguriše odgovarajuću nožicu GPIO porta kao izlaznu, a ako je postavljen na vrednost „0“ konfiguriše nožicu kao ulaznu
- Napisati kodni fragment koji će na pravilan način konfigurisati GPIO port Port.X i koji paliti/gasiti LED diodu D1 sa svakim pritiskom tastera SW1

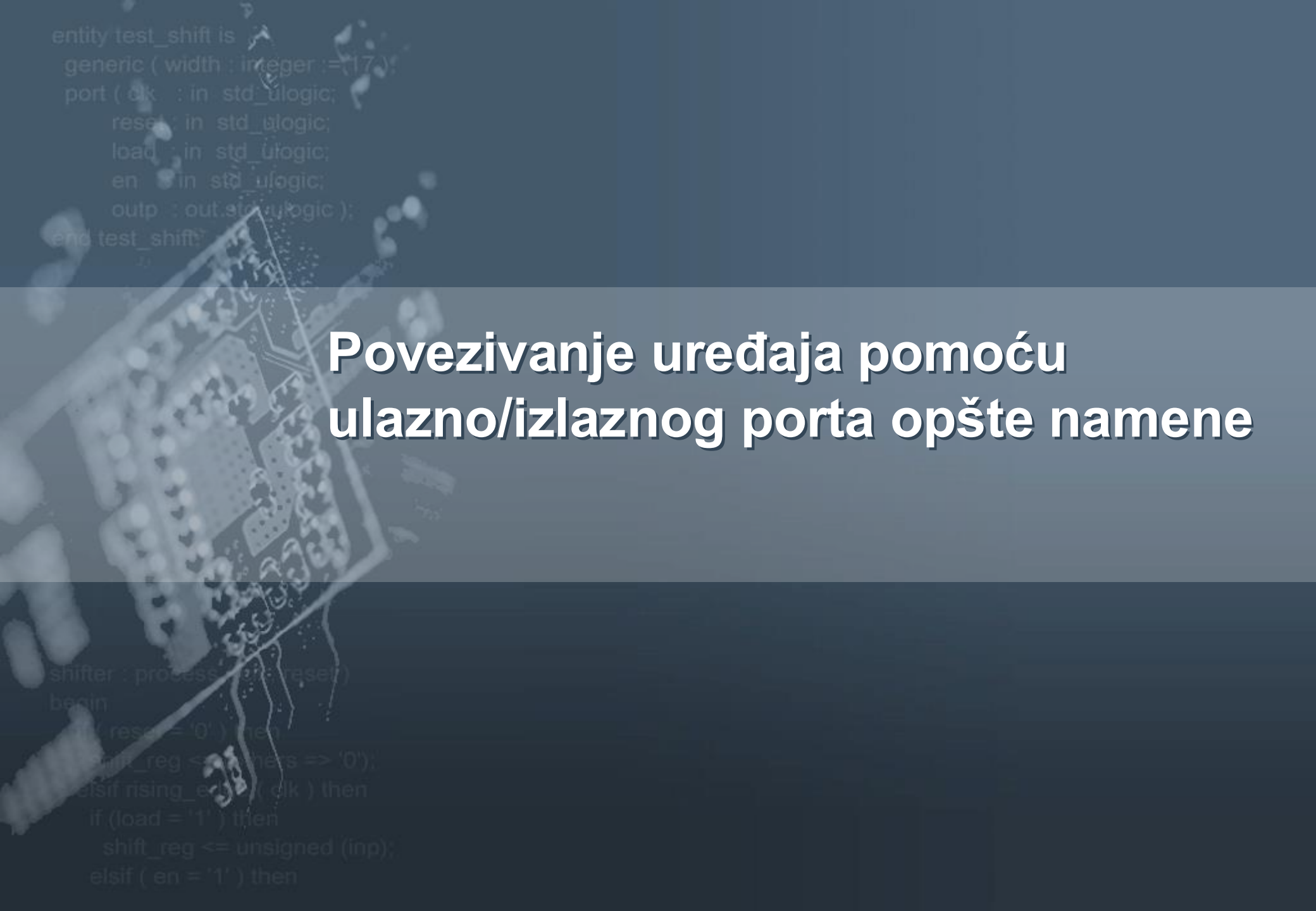


Primer korišćenja GPIO porta II

- Na početku, potrebno je nožicu **Px.0** konfigurisati kao **ulaznu** a Px.1 kao **izlaznu**
- Taster SW1 priključen je na nožicu Px.0 na takav način da se generiše nizak napon kada je taster pritisnut, a visok kada je otpušten
- Program na početku treba da konfiguriše **smer nožica** Px.0 i Px.1
- Zatim treba da uđe u beskonačnu petlju u kojoj će „prozivati“ taster SW1 da bi utvrdio trenutak kada je taster pritisnut
- Nakon detekcije pritiska tastera SW1, vrši se inverzija trenutnog stanja LED diode i ulazi se u sledeću beskonačnu petlju koja detektuje trenutak kada je taster otpušten
- U trenutku kada detektuje da je taster otpušten, program se vraća u prvu petlju gde ćeka na naredni pritisak tastera

```
            …
            AND #0FEh, &PxDir    ; Postavi Px.0 kao ulazni
            OR  #002h, &PxDir    ; Postavi Px.1 kao izlazni
            AND #0FDh, &PxDat    ; Na početku je D1 isključena
            …
loop_low:   TEST #001h, &PxDat   ; Proveri vrednost Px.0
            JNZ loop_low         ; Ako je PX.0=1 nastavi sa
                                   proverom
            XOR #002, &PxDat     ; U protivnom invertuj Px.1
loop_high:  TEST #001, &PxDat    ; Proveri da li je SW1 otpušten
            JZ loop_high
            JMP loop_low         ; Kada se SW1 otpusti vrati se
                                   na proveru da li je pritisnut

            END
```



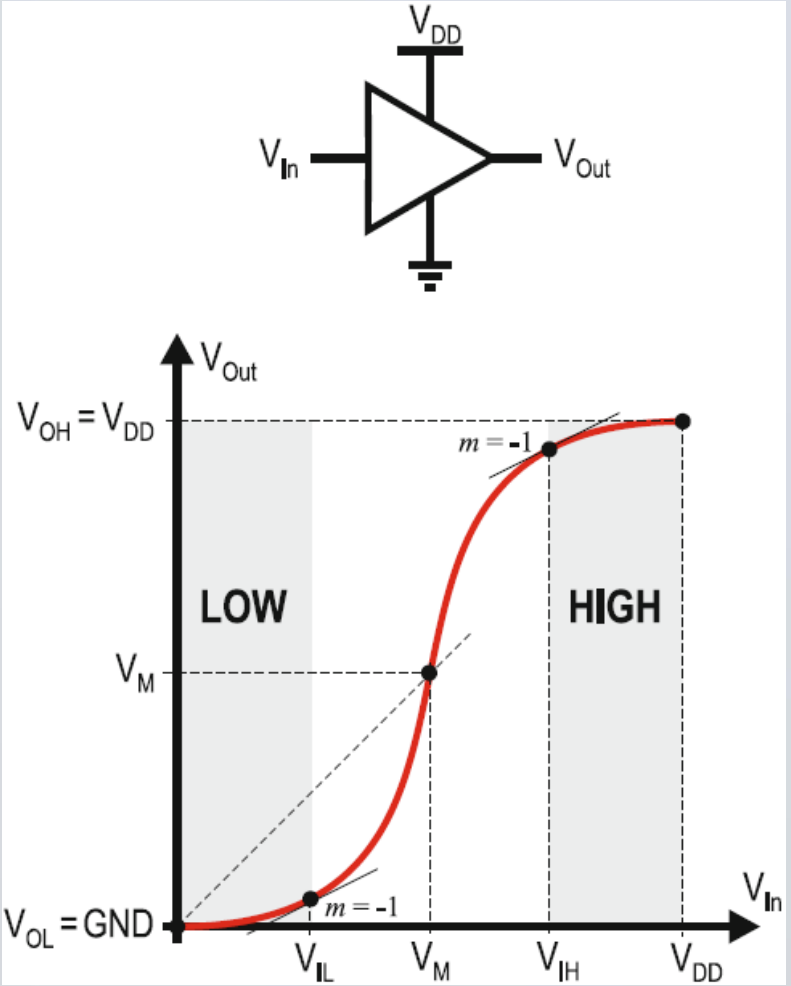
Povezivanje uređaja pomoću ulazno/izlaznog porta opšte namene

Povezivanje uređaja pomoću ulazno/izlaznog porta opšte namene

- Prilikom povezivanja mikrokontrolera sa svojim okruženjem, često se javlja problem da se električne karakteristike nožica periferijskih jedinica i nožica mikrokontrolera ne poklapaju
- U tom slučaju potrebno je koristiti odgovarajuća prilagodna kola koja će izvršiti konverziju sa jednog električnog protokola na drugi
- Prvi korak u utvrđivanju da li su prilagodna kola neophodna ili ne jeste da se pažljivo prouči dokumentacija mikrokontrolera i periferijskih jedinica koje planiramo da povežemo sa mikrokontrolerom
- GPIO portovi, kao i svako električno kolo, imaju ograničenja u pogledu minimalnih i maksimalnih vrednosti napona i struje koje se mogu pojaviti na njihovim nožicama
- Pored električnih, svaki GPIO port takođe ima i odgovarajuće vremenske karakteristike koje definišu maksimalnu brzinu promene signala na nožicama

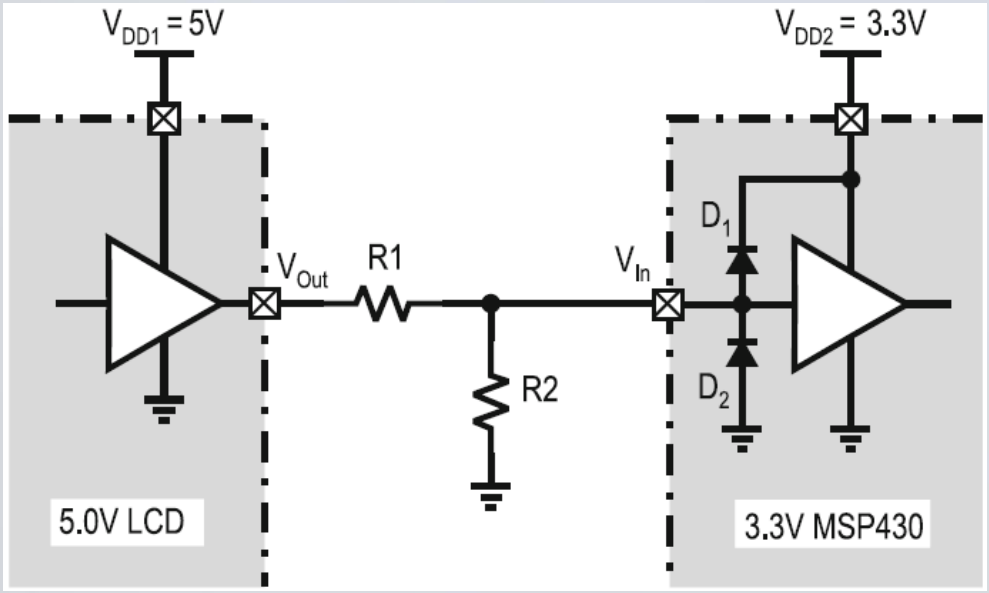
Električne karakteristike I/O nožica I

- **Električne karakteristike I/O nožica** definišu minimalne i maksimalne vrednosti ulaznih, odnosno izlaznih, napona:
  - V_{IL} – maksimalna vrednost ulaznog napona koja će biti interpretirana kao logička nula
  - V_{IH} – minimalna vrednost ulaznog napona koja će biti interpretirana kao logička jedinica
  - V_{OH} – maksimalna vrednost izlaznog napona koja se može pojaviti na izlazu
  - V_{OL} – minimalna vrednost izlaznog napona koja se može pojaviti na izlazu
- Oblasti vrednosti ulaznog napona koji će pravilno biti prepoznat kao logička nula, odnosno jedinica, označene su sa LOW i HIGH
- Oblast koja leži između LOW i HIGH regiona predstavlja **tranzicionu oblast**



Električne karakteristike I/O nožica II

- U normalnom režimu rada vrednosti ulaznog napona nikada ne bi trebalo da napuštaju LOW i HIGH oblasti
- Ukoliko je to slučaj, stabilan rad sistema ne može se garantovati te je neophodno koristiti konvertore naponskih nivoa (level-shifter)
- Ova situacija se često javlja ako se naponi napajanja mikrokontrolera i periferijskih jedinica ne poklapaju
- Danas većina mikrokontrolera i periferijskih jedinica radi na naponu napajanja od 3.3V, ali znatan broj sistema i dalje koristi napon napajanja od 5V koji je bio uobičajen kod TTL sistema
- Ukoliko generator signala radi na 5V a prijemnik radi na 3V, konverzija naponskog nivoa je neizbežna



Električne karakteristike I/O nožica III

- U opštem slučaju, konverzija naponskih nivoa biće neophodna ukoliko važe sledeće relacije

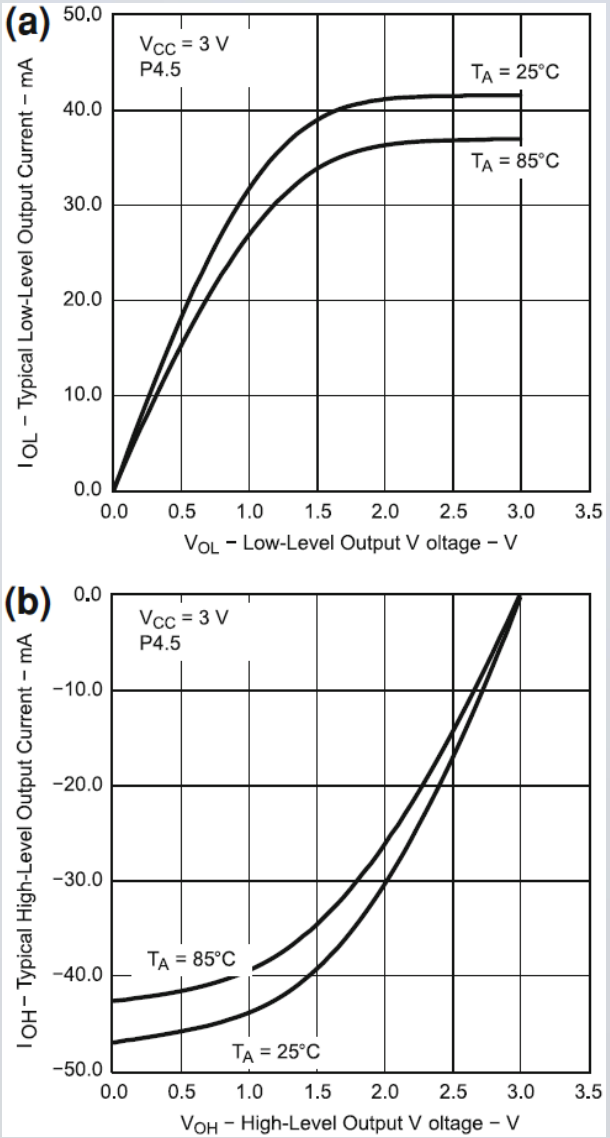
$$V_{OH\text{driver}} < V_{IH\text{load}} \text{ ili } V_{OH\text{driver}} > V_{DD\text{load}}$$

Ili

$$V_{OL\text{driver}} > V_{IL\text{load}} \text{ ili } V_{OL\text{driver}} > V_{SS\text{load}}$$

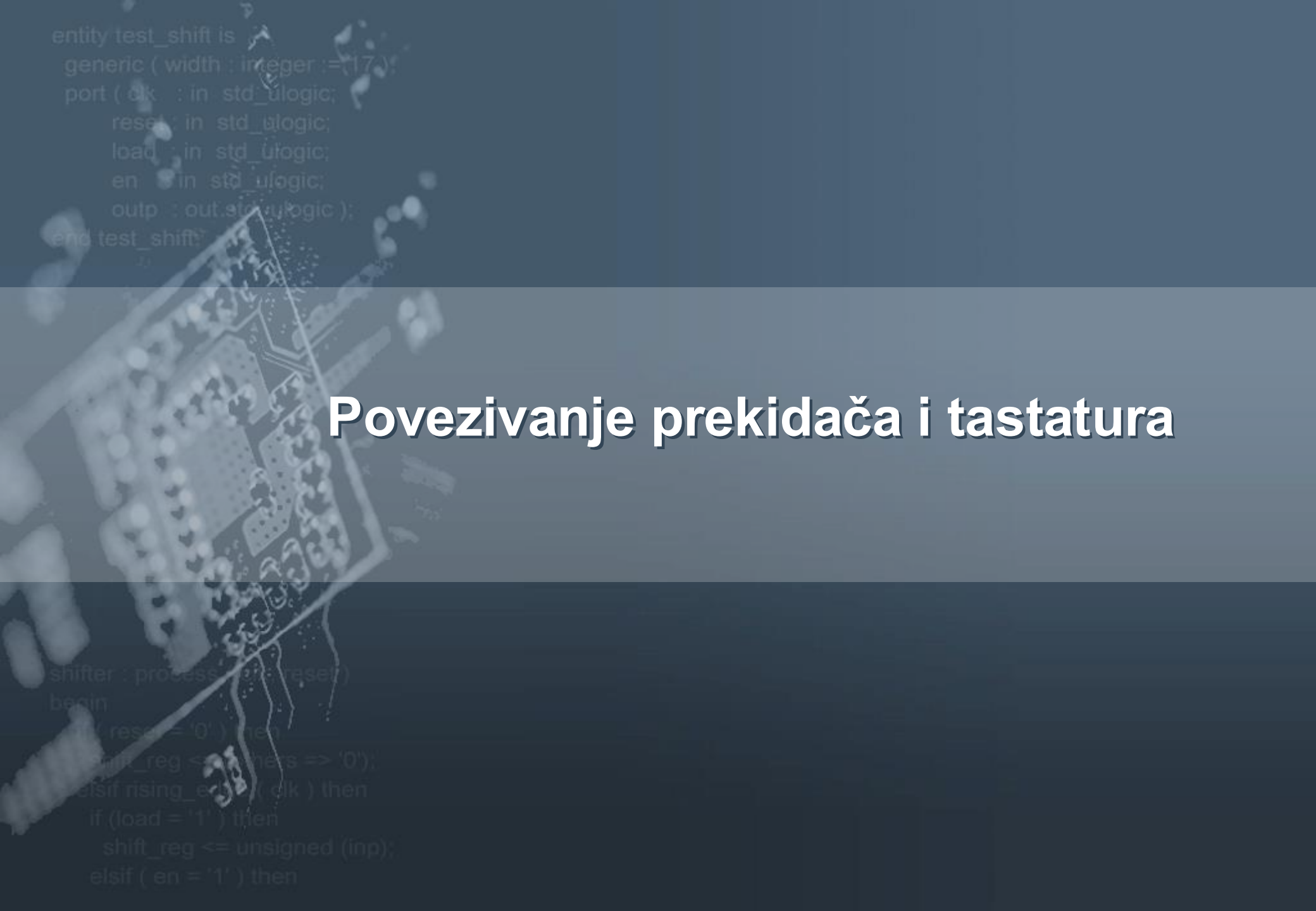
Izlazni limiti I/O nožica

- V_{OL} i V_{OH} vrednosti na izlazima nožice menjaju se u zavisnosti od intenziteta struje koja protiče kroz nožicu
- Tipična promena V_{OL} i V_{OH} vrednosti u zavisnosti od **jačine struje** koja protiče kroz nožicu prikazana je na graficima desno
- Prema konvenciji, jačina struje je pozitivan broj ukoliko struja utiče u nožicu, a negativan broj ukoliko struja ističe iz nožice
- Prilikom projektovanja interfejsa mora se voditi računa da usled velike jačine struje koja protiče kroz nožicu ne dođe do značajne degradacije naponskog nivoa koja bi mogla da naruši pravilan rad sistema
- Pored toga, ukoliko jačina struje dostigne ili čak premaši maksimalne dozvoljene vrednosti može doći do trajnog oštećenja komponente



Vremenske karakteristike I/O nožica

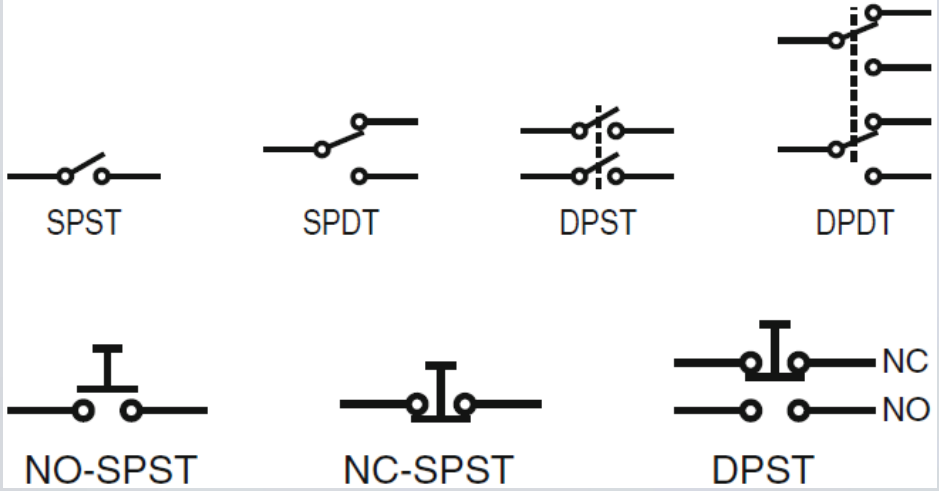
- Pod vremenskim karakteristikama I/O nožica podrazumeva se maksimalna brzina promene vrednosti signala sa logičke nule na jedinicu, ili obrnuto
- Ova **brzina promene** zavisi od napona napajanja kola, kapacitivnosti koja opterećuje nožicu i učestanosti sistemskog kloka signala
- Kako se smanjuje napon napajanja i povećava kapacitivnost koja je povezana na nožicu dolazi do **povećanja vremena** potrebnog za uspostavljanje nove stabilne vrednosti što dovodi do smanjenja maksimalne brzine promene signala
- U slučaju ulaznih nožica, obično se definiše i minimalna širina impulsa koji se može detektovati na ulazu



Povezivanje prekidača i tastatura

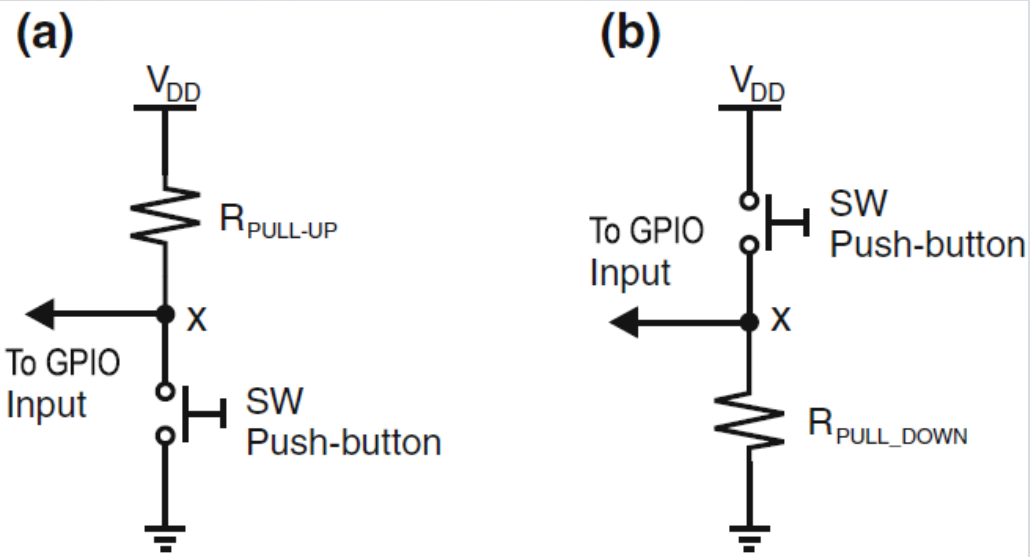
Povezivanje prekidača i tastatura

- Jedna od najčešćih periferijskih jedinica koja se povezuje sa mikrokontrolerom je mehanički kontaktni prekidač
- Prekidači predstavljaju vrlo intuitivan interfejsa za krajnjeg korisnika
- Pošto se svaki prekidač može naći samo u dva stabilna stanja, otvoren ili zatvoren, on se savršeno uklapa sa mogućnostima detekcije GPIO linije
- U zavisnosti od aplikacije, prekidači se mogu koristiti inidividualno ili u sistemu nizova odnosno **matrica prekidača (tastatura)**
- Posebnu klasu prekidača čine tasteri, koji imaju samo jedan stabilan položaj, dok se drugi položaj zauzima samo ako je taster pritisnut



Rad sa prekidačima i tasterima

- Povezivanje običnog prekidača ili tastera sa GPIO nožicom je vrlo jednostavno
- Jedina stvar koju je potrebno obezbediti je da se dva moguća stanja u kojima se može naći prekidač, OTVOREN i ZATVOREN, reprezentuju sa odgovarajućim logičkim nivoima (logička nula i logička jedinica)
- Na slici desno prikazana su dva načina povezivanja tastera sa GPIO nožicom
- Na slici a) čvor X se nalazi na niskom logičkom nivou kada je taster pritisnut, a na visokom logičkom nivou kada je taster otpušten
- Na slici b) situacija je obrnuta. Čvor X se nalazi na visokom logičkom nivou kada je taster pritisnut, a na niskom logičkom nivou kada je taster otpušten



Pravilno dimenzionisanje R_{PULL_UP} i R_{PULL_DOWN} otpornika I

- R_{PULL_UP} i R_{PULL_DOWN} otpornici su neophodni iz dva razloga:
  - Da bi se obezbedio odgovarajući **naponski nivo** na ulazu GPIO nožice kada je **taster otpušten** – u slučaju da nema otpornika ulaz X bi praktično bio slobodan, te naponski nivo na GPIO ulazu najverovatnije ne bi bio pravilno protumačen
  - Da bi **ograničio jačinu struje** koja protiče kroz taster kada je on pritisnut

- Prilikom dimenzionisanja otpornika njegova vrednost mora biti odabrana na takav način da se zadovolje dva navedena uslova

- U slučaju R_{PULL_UP} otpornika, da bi zadovoljili prvi uslov mora da važi sledeća relacija

$$V_{DD} - R_{PULL\_UP} \cdot I_{IH} \geq V_{IH} \Rightarrow R_{PULL\_UP} \leq \frac{V_{DD} - V_{IH}}{I_{IH}}$$

gde je I_{IH} nominalna jačina ulazne struje u GPIO nožicu pri visokom logičkom nivou, a V_{IH} minimalna vrednost napona na GPIO nožici koja će i dalje biti prepoznata kao logička jedinica

Pravilno dimenzionisanje R_{PULL_UP} i R_{PULL_DOWN} otpornika II

- Da bi drugi uslov bio zadovoljen, maksimalna struja koja protiče kroz zatvoreni taster mora biti manja od maksimalno dozvoljene vrednosti, I_{SWmax}

$$I_{SW} = \frac{V_{DD}}{R_{PULL\_UP}} \leq I_{SWmax} \Rightarrow R_{PULL\_UP} \geq \frac{V_{DD}}{I_{SWmax}}$$

- Uobičajena pretpostavka za vrednost I_{SWmax} je da je $I_{SWmax} = 10I_{IH}$

Primer

- Pretpostavimo da koristimo mikrokontroler koji radi na naponu napajanja od 5V, $V_{DD} = 5V$.
- Nominalna vrednost ulazne struje na GPIO nožicama ovog mikrokontrolera iznosi 1uA, $I_{IH} = 1uA$.
- Minimalna vrednost ulazno napona koji će i dalje biti prepoznat kao logička jedinica iznosi $0.7V_{DD}$, $V_{IH} = 0.7V_{DD}$.
- Dimenzionisati vrednost pull-up otpornika da bi se na GPIO nožicu pouzdano mogao povezati taster.

Pravilno dimenzionisanje R_{PULL_UP} i R_{PULL_DOWN} otpornika III

Rešenje:

- U našem proračunu uzećemo strožiju vrednost za V_{IH}, $V_{IH} = 0.9V_{DD}$, kako bi smo bili još sigurniji da će logički nivou na GPIO ulazu u slučaju kada je taster otpušten biti prepoznat kao logička jedinica

- Na osnovu nejednačine sa slajda 25 dobijamo gornju granicu za vrednost pull-up otpornika

$$R_{PULL\_UP} \leq \frac{V_{DD} - 0.9V_{DD}}{I_{IH}} = \frac{V_{DD} - 0.9V_{DD}}{I_{IH}} = \frac{0.1V_{DD}}{1\mu A} = 500K\Omega$$

- Koristeći pretpostavku da je $I_{SWmax} = 20I_{IH} = 20uA$ i jednačinu sa slajda 26, dobijamo donju granicu za vrednost pull-up otpornika

$$R_{PULL\_UP} \geq \frac{V_{DD}}{I_{SWmax}} = \frac{5.0V}{20\mu A} = 250K\Omega$$

Rad sa realnim prekidačima i tasterima

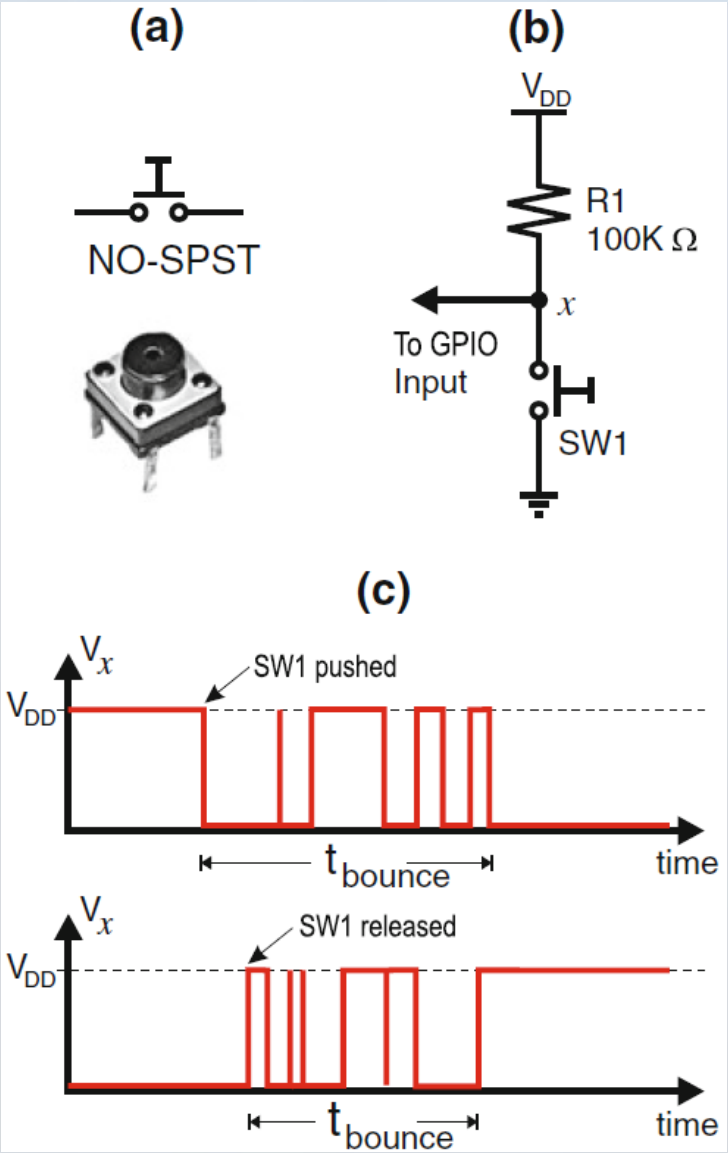
- U dosadašnjim razmatranjima pretpostavljali smo idealne karakteristike prekidača

- Realni prekidači imaju čitav niz neidealnih karakteristika:
  - **Otpornost u "uključenom" stanju koja je različita od nule**
  - **Maksimalnu jačinu struje koja može da protiče kroz zatvoreni prekidač** – ukoliko se ova jačina struje premaši može doći do oštećivanja prekidača
  - **Ograničenu dielektričnu jačinu** –maksimalna veličina napona koji se može pojaviti na kontaktima prekidača kada je on u otvorenom stanju, a da ne dođe do varničenja
  - **Ograničen životni vek** – broj otvaranja i zatvaranja prekidača je ograničen
  - **Konačno vreme smirivanja (bounce time)** – vreme potrebno da se prekidač smiri u novom stanju nakon što je došlo do promene stanja

- Najvažnija neidealna karakteristika prekidača prilikom razvoja embeded sistema je njegovo vreme smirivanja, koje je uvek različito od nule

Vreme smirivanja prekidača

- Slika desno ilustruje fenomen “odskakivanja” prekidača (switch bounce) prilikom njegovog uključivanja, odnosno isključivanja
- Odskakivanje prekidača manifestuje se kao višestruka promena logičkih nivoa na GPIO ulazu
- Obzirom da embeded sistem očekuje samo jednu tranziciju prilikom svakog pritiska prekidača ili tastera “odskakivanje” prekidača može izazvati probleme u normalnom radu sistema
- Vreme smirivanja prekidača tipično iznosi oko 10 ms, ali postoje prekidači kod kojih vreme smirivanja iznosi i preko 150 ms



Problemi izazvani “odskakivanjem” prekidača

- “Odskakivanje” prekidača rezultuje u tome da se svaki pritisak na prekidač ili taster u okviru embeded sistema interpretira kao čitav **niz uzastopnih pritisaka**, što može izazvati neispravan rad sistema
- Na primer, razmotrimo rad daljinskog upravljača televizora
- Ukoliko pritisnemo taster za promenu kanala, a on nije zaštićen od “odskakivanja”, svaki pritisak će biti interpretiran kao čitav niz pritisaka te će se umesto pomeranja na sledeći kanal zapravo izvršiti pomeranje za nekoliko kanala unapred ili unazad
- “Odskakivanje” prekidača je posebno problematično ukoliko se pritisak na prekidač interpretira kao zahtev za prekidom
- U ovom slučaju svaki pritisak će zapravo da generiše čitav niz zahteva za prekidom što može izazvati gotovo haotičan rad softvera

Tehnike za elemininaciju efekta “odskakivanja” prekidača

- Problem “odskakivanja” prekidača može se rešiti hardverskim ili softverskim tehnikama

- U slučaju hardverskih tehnika u sistem se dodaju moduli čiji je zadatak da potisnu višestruke tranzicije u logičkim nivoima koji su izazvani “odskakivanjem” prekidača

- Softverske tehnike dozvoljavaju da ove višestruke tranzicije “uđu” unutar mikrokontrolera, ali se zatim one uklanjaju programski korišćenjem različitih pristupa koji će biti izloženi u nastavku

Hardverske “Debouncing” tehnike

- Sve hardverske tehnike pokušavaju da “filtriraju” pojavu višestrukih tranzicija kao rezultata “odskakivanja” prekidača prilikom njegovog pritiska, kako bi svaki pritisak rezultovao pojavom samo jedne tranzicije sa jednog logičkog nivoa na drugi

- U praksi se koriste različite tehnike za postizanje ovog cilja:
  - Tehnike bazirane na korišćenju SR leča
  - Tehnike bazirane na korišćenju RC mreža i Šmit-triger bafera
  - Tehnike bazirane na korišćenju “debouncing” integrisanih kola, na primer, ELM310 proizvođača Eml Electronics ili MC14490 proizvođača ON-Semiconductor

Softverske “Debouncing” tehnike

- Softverske “debouncing” tehnike su privlačne jer ne zahtevaju nikakve dodatne hardverske komponente

- Međutim, dodatni CPU ciklusi će biti potrošeni da bi se iz ulaznog kontrolnog signala uklonile višestruke tranzicije

- U nastavku će biti predstavljene dve često korišćene tehnike:
  - Softverski “Debauncing” baziran na periodičnom prozivanju
  - Softverski “Debauncing” baziran na brojanju

Softverski “debouncing” baziran na periodičnom prozivanju

- Kod ove tehnike prekidač se periodično “proziva”, sa konstantnim periodom prozivke koji je izabran na takav način da bude **duži** od vremena potrebnog za smirivanje prekidača

- Na primer, ukoliko je prosečno vreme smirivanja prekidača 15 ms, onda bi softver mogao da proverava (“proziva”) stanje u kome se nalazi prekidač svakih 20 ili 25 ms i na taj način izbegne probleme izazvane “odskakivanjem” prekidača

- Da bi se izbeglo uzaludno trošenje CPU ciklusa dok se čeka da prođe vreme smirivanja prekidača, mogao bi se koristiti tajmer koji bi periodično generisao zahtev za prekidom u okviru kojega bi se izvršila provera stanja prekidača

- Glavni problem ovog pristupa je u tome što on neće raditi pouzdano ukoliko se koristi prekidač čije je vreme smirivanja duže od pretpostavljenog

Softverski "debouncing" baziran na brojanju

- Bolje rešenje jeste da se prepostavi da se prekidač smirio ukoliko nije bilo promene na GPIO ulazu tokom poslednjih *n* provera
- Ukoliko tokom *n* sukcesivnih provera uvek čitamo istu vrednost sa prekidača možemo pretpostaviti da se on smirio i koristiti očitanu vrednost u ostatku programa
- Ukoliko se desi promena sa jedne vrednosti na drugu tokom ovih *n* sukcesivnih čitanja, proces brojanja se reinicijalizuje i započine iz početka
- Moguća implementacija bi mogla koristiti tajmer za generisanje trenutaka u kojima bi se periodično proveravalo stanje prekidača
- Tipično vreme između dve uzastopne provere kreće se od 1 do 10 ms, dok se za *n* obično uzima vrednost veća od 10

Povezivanje nizova prekidača

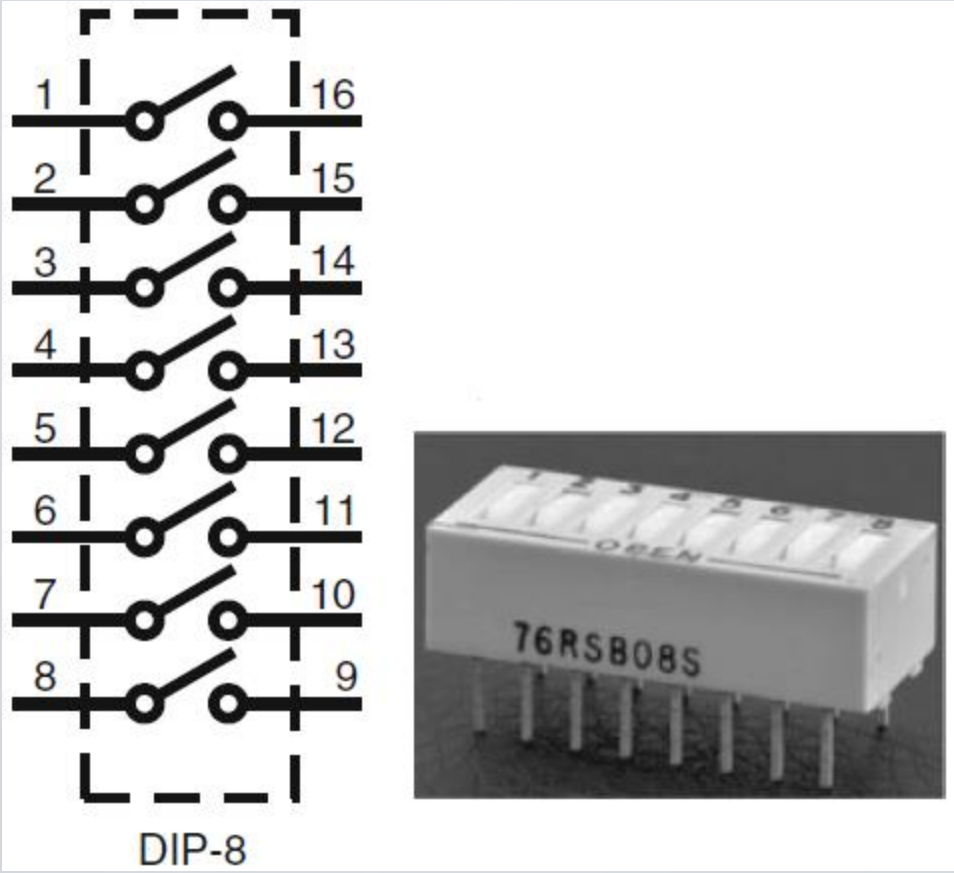
- Mnoge aplikacije zahtevaju postojanje velikog broja prekidača/tastera koje je potrebno povezati sa mikrokontrolerom

- U ovim situacijama zgodno je koristiti nizove prekidača

- Postoje dva tipa nizova prekidača:
  - Linearni nizovi prekidača, poznati pod nazivom DIP prekidači (DIP Switch)
  - Matrični nizovi prekidača, tastature

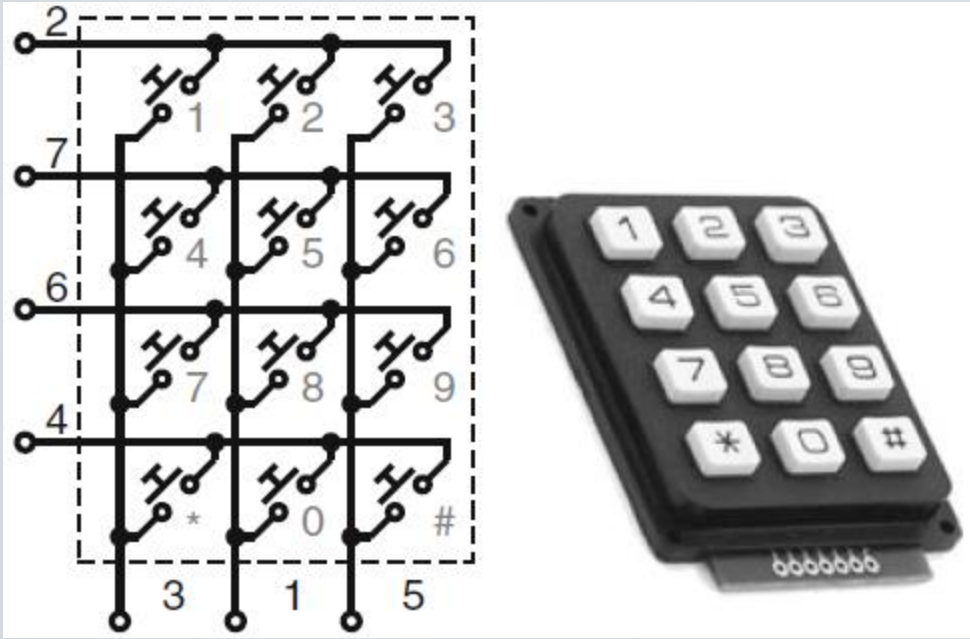
DIP prekidači

- DIP prekidač sastoji se od linearnog niza individualnih prekidača
- DIP prekidači obično imaju od 2 do 12 individualnih prekidača, pri čemu se najčešće koriste DIP-8 prekidači
- DIP prekidači često se koriste za konfigurisanje režima rada individualnih periferijskih jedinica
- Svaki prekidač iz linearnog niza mora se individualno povezati preko posebne GPIO nožice sa mikrokontrolerom
- Prilikom povezivanja DIP prekidača takođe se moramo primeniti “debouncing” tehnike kako bi se obezbedio pouzdan rad sistema



Tastature I

- Tastature se sastoje od većeg broja tastera, organizovanih u matričnu formu
- Status individualnog tastera može se dekodovati na osnovu **očitanih vrednosti sa kolona i vrsta matrice**
- Glavna prednost organizovanja tastera u matričnu formu leži u smanjenju potrebnog broja I/O linija u odnosu na broj koji bi bio potreban u slučaju organizacije matrice u jedan linearan niz
- Uzmimo kao primer tastaturu od 64 tastera
- Organizujući ovih 64 tastera u matricu 8x8 potrebno je samo 8+8=16 I/O linija za očitavanje svih tastera, što je 4 puta manje u odnosu na 64 I/O linije ukoliko bi svaki taster bio individualno vezan na mikrokontroler
- Generalno, niz od *N* tastera organizovanih u matričnu formu, može se interfejsovati pomoću $2\sqrt{N}$ I/O linija



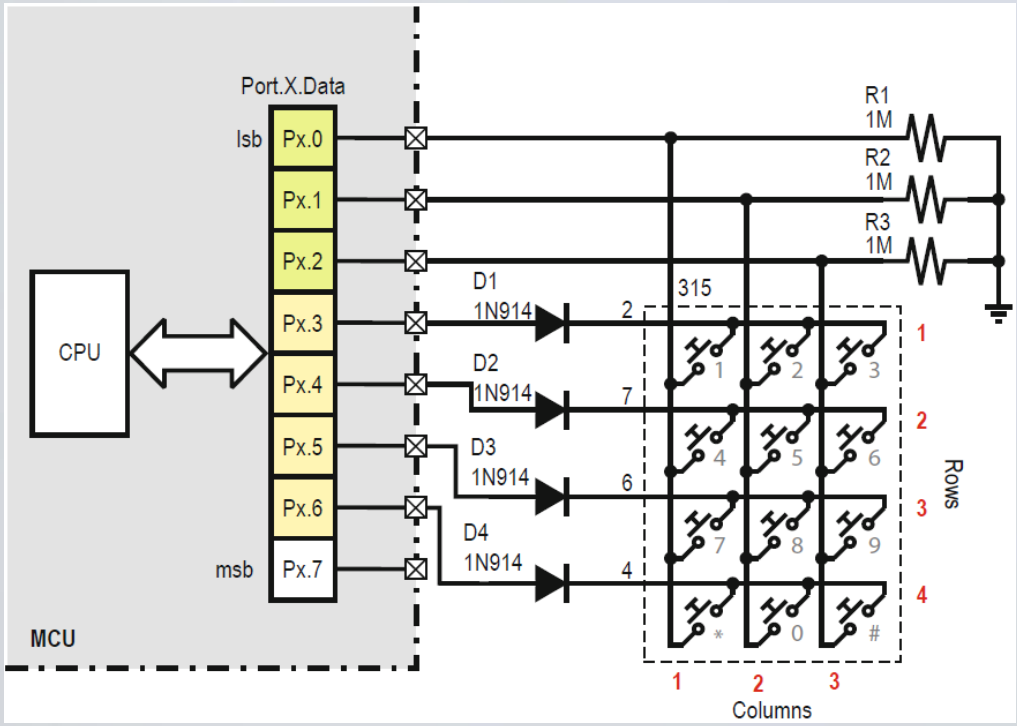
Tastature II

- Iako matrična organizacija tastera pojednostavljuje njihovo povezivanje sa mikrokontrolerom, u smislu potrebnog broja I/O linija, **utvrđivanje** koji je od **tastera pritisnut** zahteva složeniji algoritam od onog koji bi se koristio u slučaju linearne organizacije

- U praksi se koriste dva pristupa rešavanju problema dekodovanja:
  - Softversko skeniranje tastature
  - Korišćenje hardverskih modula za dekodovanje tastature

Softversko skeniranje tastature I

- Interfejs potreban da bi procesor mogao da skenira tastaturu uključuje **alociranje I/O portova** koje će biti povezane sa vrstama i kolonama tastature
- I/O portovi koji su povezani na **vrste** konfigurisani su kao **izlazni**, a I/O portovi povezani na **kolone** kao **ulazni**, mada je moguća i obrnuta konfiguracija
- Ukoliko je broj vrsta i kolona tastature različit, **ulazni I/O portovi** povezuju na onu grupu koja se sastoji od **manjeg broja signala**
- Ovakva konfiguracija omogućava da se **kod** za skeniranje tastature pošalje preko **izlaznih I/O portova a povratni kod pročita preko ulaznih** I/O portova

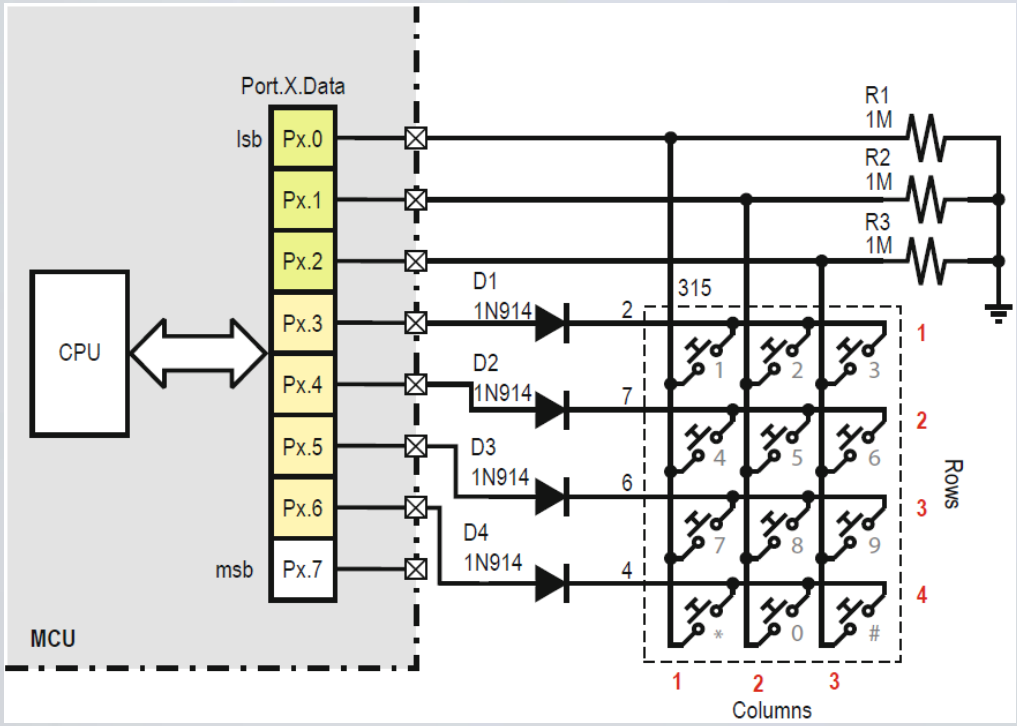


Softversko skeniranje tastature II

- Na slici desno prikazan je način povezivanja tastature, organizovane u matricu **3x4 tastera**, na GPIO port hipotetičkog mikrokontrolera

- Obzirom da je broj kolona manji od broja vrsta, kolone su povezane na ulazne I/O pinove (Px.0-Px.2) dok su vrste povezane na izlazne I/O pinove (Px.3-Px.7)

- Da bi čitav sistem radio pouzdano potrebno je:
  - Povezati **"pull-down" otpornike** na ulazne I/O pinove da bi se obezbedila **logička nula** na odgovarajućem I/O pinu ako u datoj koloni nije pritisnut ni jedan taster. Da ovog otpornika nema, vrednost napona na tom I/O pinu bila bi nedefinisana.
  - Povezati diode na svaki izlazni I/O pin, kako bi se sprečila pojava kratke veze između izlaznih I/O pinova u slučaju da su u istoj koloni pritisnuta dva ili više tastera.
  - Izvršiti "debouncing" ulaznih I/O pinova

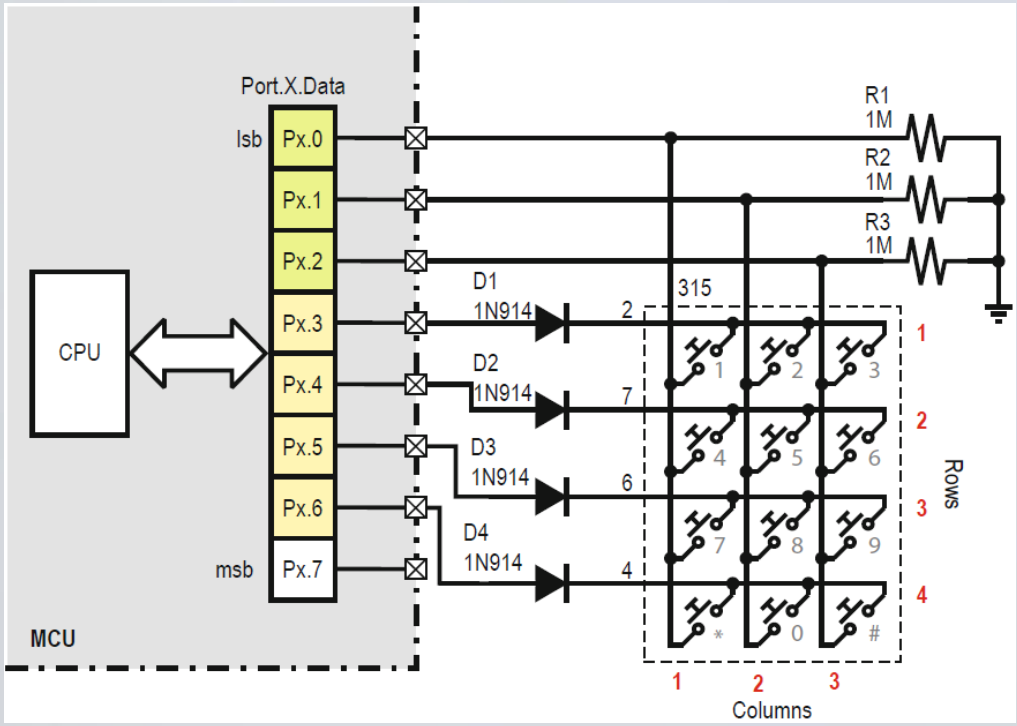


Softversko skeniranje tastature III

- Algoritam skeniranja tastature dovodi visoki napon na **samo jedan od izlaznih I/O pinova** (Px.3 – Px.7) dok ostale drži na niskom nivou

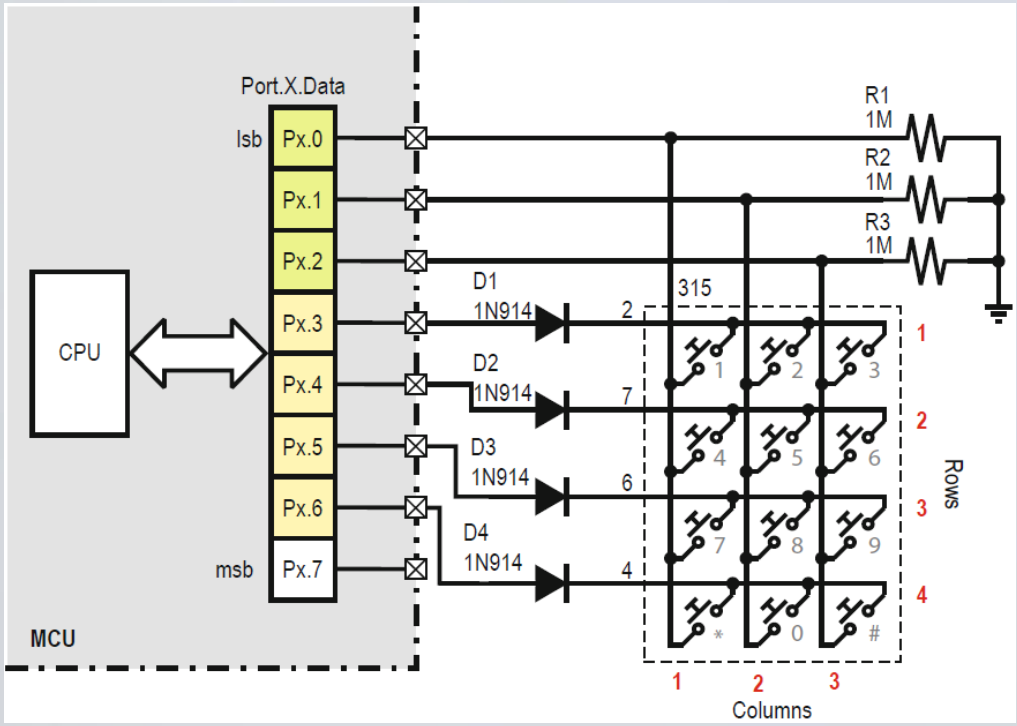
- **Očitavajući vrednosti signala na ulaznim I/O pinovima** (Px.0 – Px.2) može se jednoznačno odrediti pozicija pritisnutog tastera, jer će odgovarajući I/O pin na koji je povezana kolona u kojoj se nalazi pritisnuti taster biti na visokom nivou dok će ostali biti na niskom nivou

- Na primer, ako vrstu 2 dovedemo na visoki napon (postavljajući I/O pin Px.4 na visoki napon), ukoliko je u tom trenutku pritisnut taster 5, povratni kod pročitan na I/O pinovima Px.0-Px.2 biće “010”



Softversko skeniranje tastature IV

- Generišući sekvencu kodova za skeniranje tastature “0001”, “0010”, “0100”, “1000” aktiviraćemo jednu po jednu vrstu iz tastature
- Čitajući povratne kodove nakon primene svakog od **skenirajućih kodova** možemo jednoznačno utvrditi koji od tastera je pritisnut
- Algoritam za skeniranje tastature ovu proceduru treba kontinualno da ponavlja
- Da bi se obezbedilo da se ni jedan pritisak tastera ne propusti, algoritam za skeniranje mora da vrši skeniranje tastature **brže nego što korisnik može da pritiska tastere**



Povezivanje displeja

Povezivanje displeja

- Drugi najčešći tip periferije koji se koristi u embeded sistemima su displeji
- U kombinaciji sa tasterima i tastaturama, displeji formiraju **najčešći tip korisničkog interfejsa** pomoću kojega korisnici mogu da interaguju sa embeded sistemom
- Danas postoji čitav niz različitih displej uređaja koje je moguće koristiti unutar embeded sistema, počevši od diskretnih LED (Light Emitting Diode) dioda koje se koriste za binarnu indikaciju, do složenih plazma, LED i LCD panela koji su u stanju da prikazuju pokretne i nepokretne slike

Klasifikacija displeja I

- Prema količini vizuelne informacije koju su u stanju da prikažu, displeji se mogu klasifikovati u sledeće grupe:

- **Diskretni indikatori** – odnose se na diskretne LED diode ili neki drugi tip displeja koji je u stanju da obezbedi **tačkastu indikaciju**.

- Tradicionalne crvene, zelene, plave i žute LED diode obezbeđuju jednostavnu monohromatsku, binarnu indikaciju za prisustvo ili odsustvo (On/Off, Yes/No) nekog događaja.

- Polihromatska indikacija bazirana na RGB LED diodama takođe spada u ovu grupu.

- **Numerički displeji** – tipični predstavnik ove grupe su sedmo-segmentni displeji koji omogućavaju prikaz numeričkih informacija pomoću decimalnih cifara 0 do 9

Klasifikacija displeja II

- **Alfanumerički displeji** – Ovi displeji su u stanju da prikažu numeričke i alfabetske karaktere korišćenjem većeg broja segmenata (14 pa čak i 16) ili korišćenjem matričnih displeja

- Obično su ovi displeji realizovani kao LCD ili LED.

- **Grafički paneli** – ovaj tip displeja sastoji se iz velikog broja tačaka (pixela) koji se mogu adresirati individualno

- Postoji veliki spektar ovih displeja počevši od jednostavnih monohromatskih LCD displeja sa 128x64 ili manje pixela, organskih LED displeja (OLED) i LED matričnih displeja

- Najsloženiji displeji u ovoj grupi su LED, LCD ili plazma ekrani visoke rezolucije koji se koriste u TV prijemnicima ili računarskim monitorima

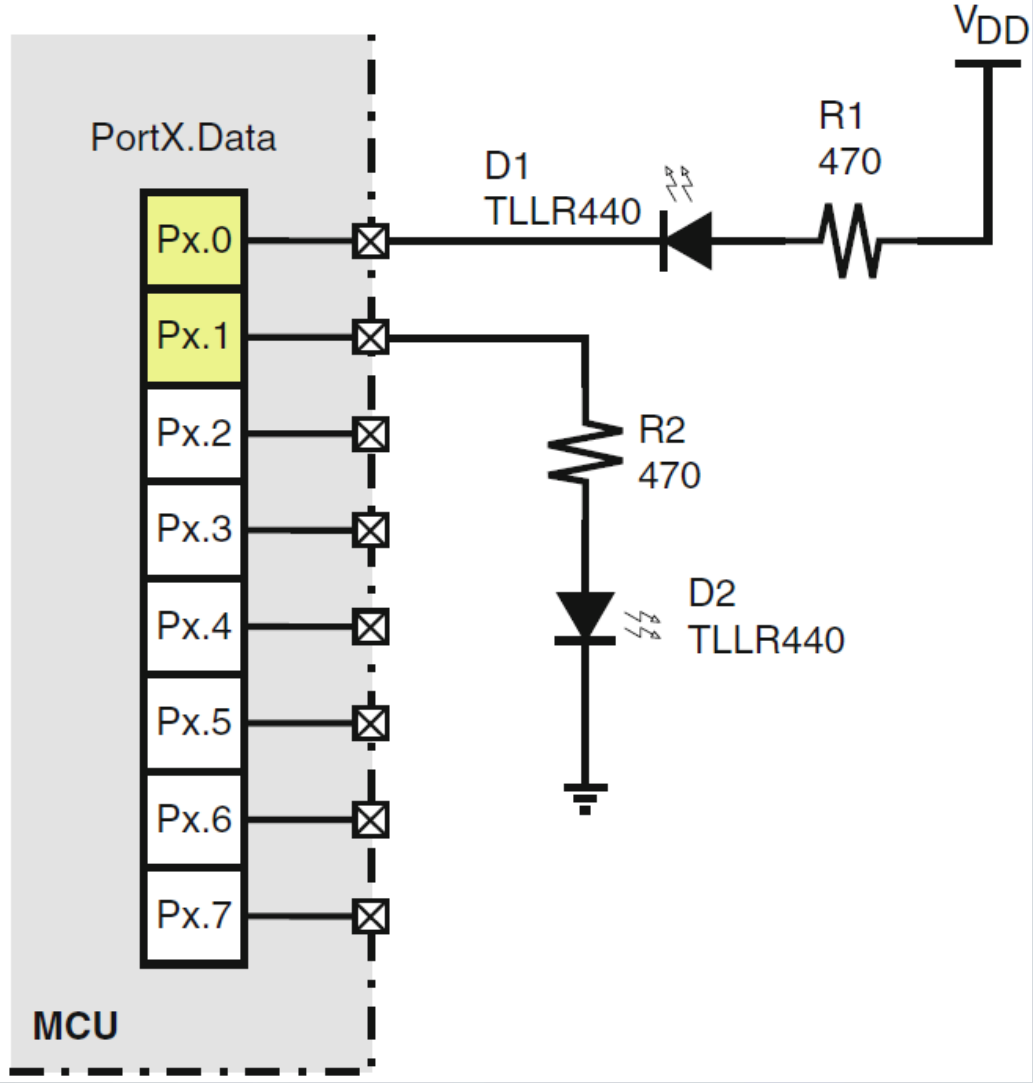
Povezivanje LED dioda

- Najjednostavniji LED interfejs zahteva postojanje naponskog izvora V_{DD} i otpornika R koji će biti povezan na red sa LED diodom
- Vrednost otpornika R bira se na takav način da ograniči jačinu struje koja protiče kroz LED diodu

$$R_1 = \frac{V_{DD} - V_F - V_{LOW}}{I_F}; R_2 = \frac{V_{HIGH} - V_F}{I_F}$$

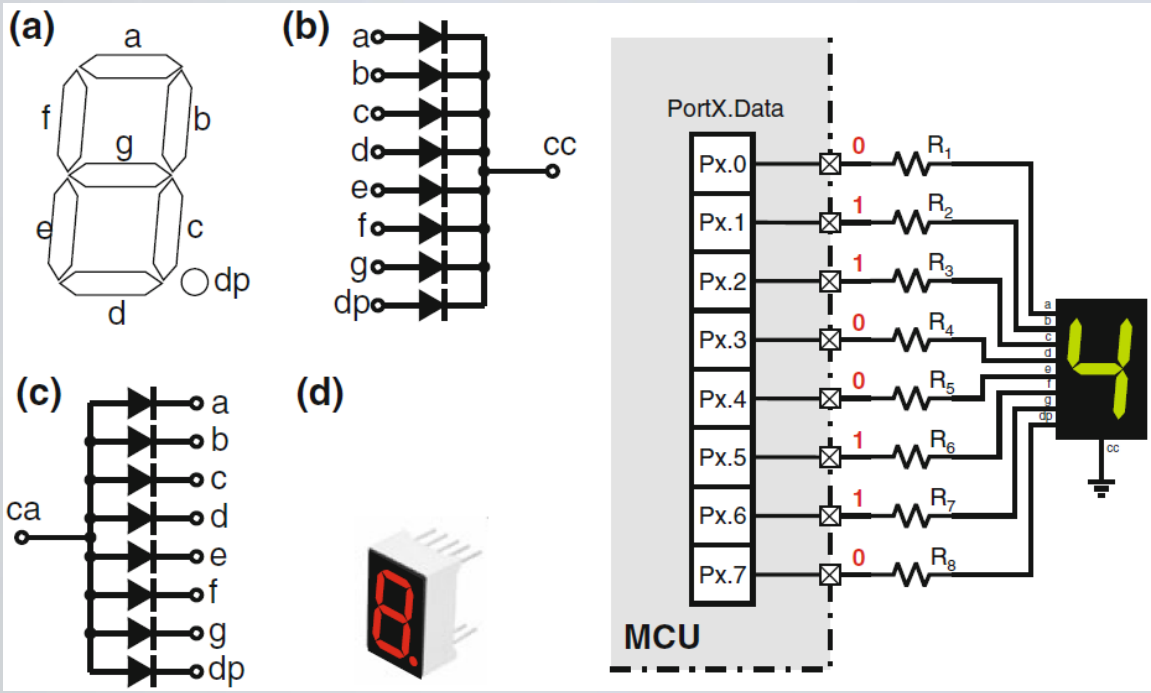
gde je V_{HIGH} i V_{LOW} naponi na I/O nožici kada se nalazi u stanju logičke jedinice, odnosno nule, V_F pad napona na LED diodi, I_F potrebna jačina struja kroz LED diodu

- Prilikom dimenzionisanja otpornika takođe se mora voditi računa da li I/O nožica može da primi, odnosno generiše dovoljnu količinu struje I_F.
- Za većinu LED dioda, V_F se kreće između 1.8-2.2V, dok je I_F između 5 i 20 mA, što je prihvatljivo za većinu GPIO portova



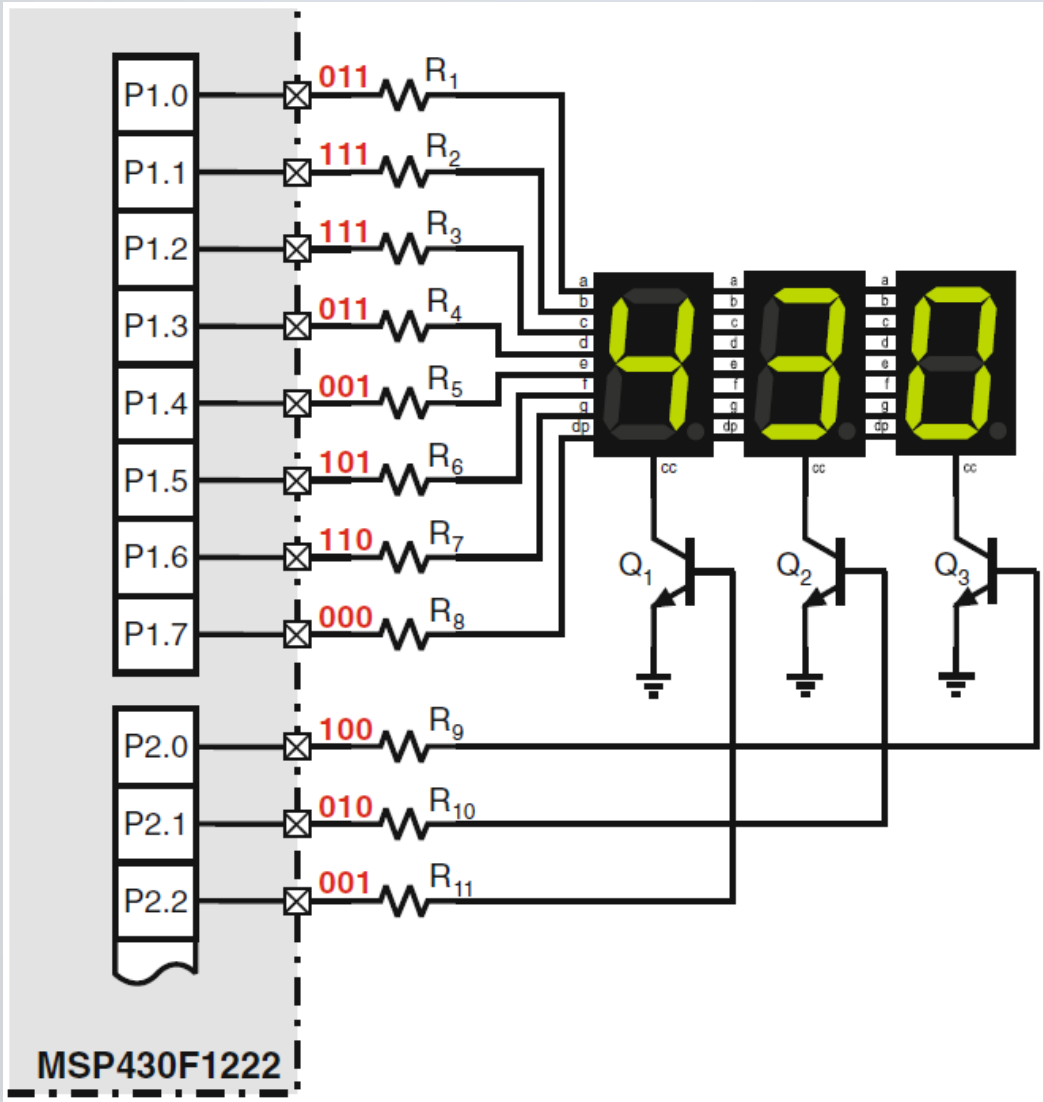
Povezivanje nizova LED dioda

- Sedmo-segmentni LED displej sastoji se iz 7 LED dioda, slika a), u obliku pravougaonika, organizovanih na takav način da se uključivanjem odogvarajućih segmenata mogu prikazati cifre od 0 do 9
- Većina 7-segmentnih displeja ima i osmi segment koji se koristi za ispis decimalne tačke desno od cifre
- Segmenti su interno organizovani tako da imaju zajedničku katodu (b) ili anodu (c)
- 7-segmentnim displejom se upravlja zadavanjem 7-bitnog koda koji uključuje odgovarajuće segmente
- Na primer, na slici desno, prikazan je 7-segmentni kod, “01100110”, koji prikazuje cifru “4” na 7-segmentnom displeju sa **zajedničkom katodom**



Multipleksno povezivanje nizova LED dioda I

- U slučaju da je neophodno povezati veći broj 7-segmentnih displeja obično se koristi multipleksno povezivanje
- Kod multipleksnog povezivanja komunikacija sa individualnim 7-segmentnim displejima je **multipleksirana u vremenu**
- Ovaj pristup, u poređenju sa pristupom kod kojeg bi se svaki 7-segmentni displej povezivao preko posebnog GPIO porta, dovodi do velike uštede u potrebnom broju I/O nožica
- Multipleksovani, višecifarni 7-segmentni displej koristi jedan skup I/O portova za pristup svim segmentima, dok se zajedničkim katodama (ili anodama) pristupa preko odvojenih I/O linija
- U **svakom trenutku** pristupa se **tačno jednom** 7-segmentnom displeju



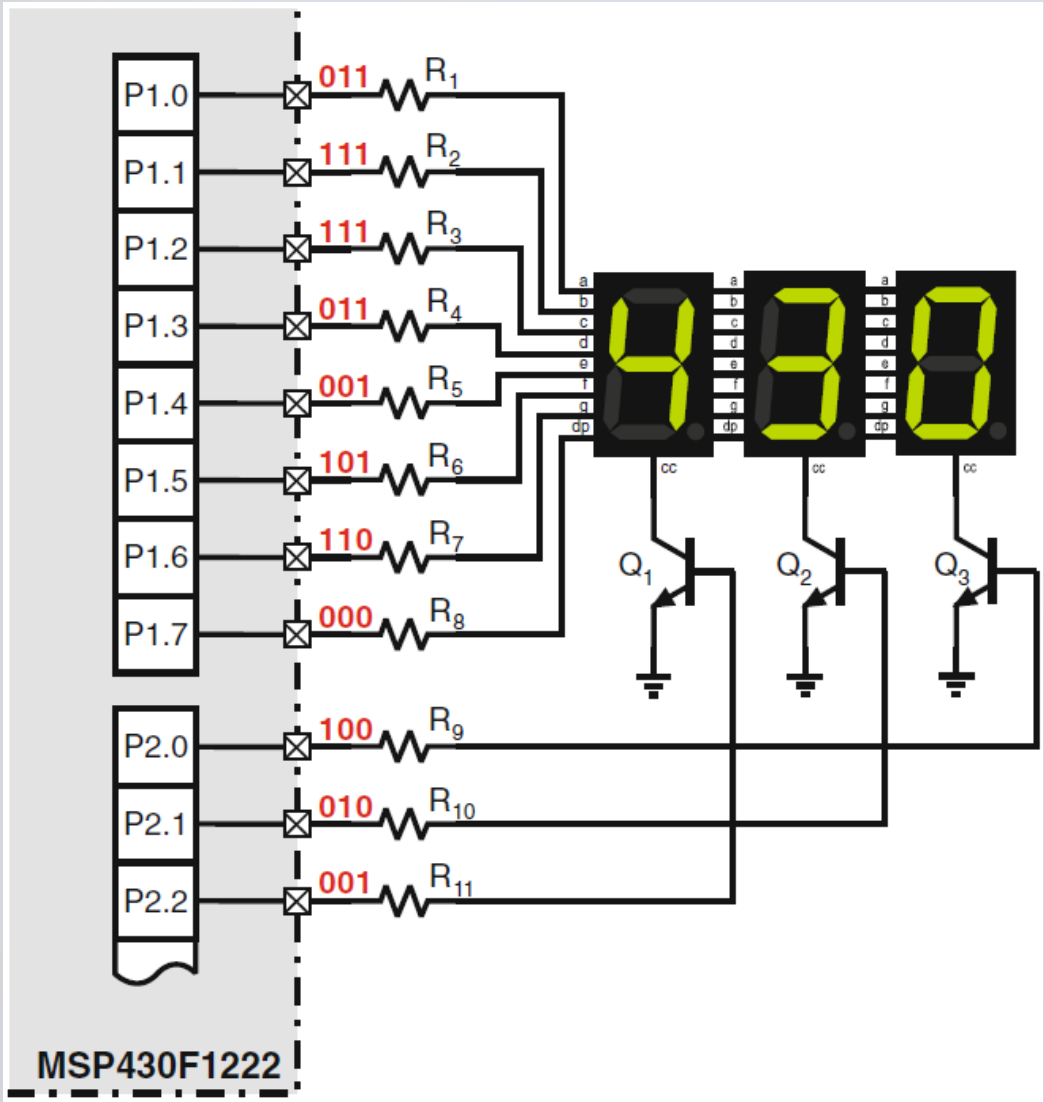
Multipleksno povezivanje nizova LED dioda II

- Da bi se prikazalo *n* cifara displeja, **7-segmentni kodovi za svaku cifru se šalju jedan po jedan**, pri čemu se aktivira po jedan 7-segmentni displej na kome se želi prikazati tekuća cifra

- Ukoliko se ova procedura ponavlja dovoljno brzo, **minimalnu 24 puta u sekundi**, ljudsko oko neće biti u mogućnosti da primeti pojedinačno uključivanje i isključivanje individualnih displeja

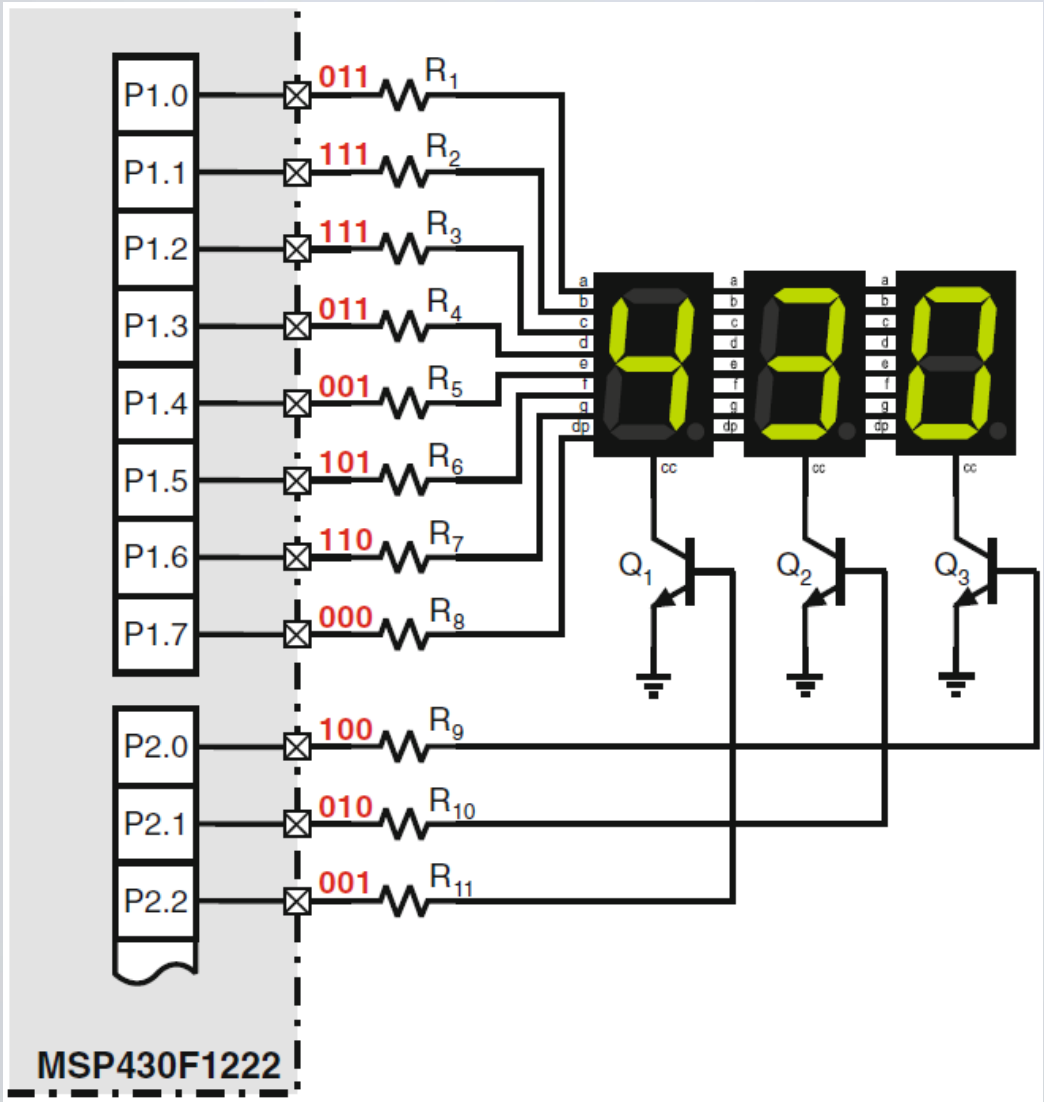
- Da bi se postigla željena **brzina osvežavanja** može se koristiti **tajmer**

- Ukoliko želimo da *n*-tocifreni displej osvežavamo $f_{refresh}$ puta u sekundi, pojedinačne 7-segmentne displeje potrebno je osvežavati $n \times f_{refresh}$ puta u sekundi



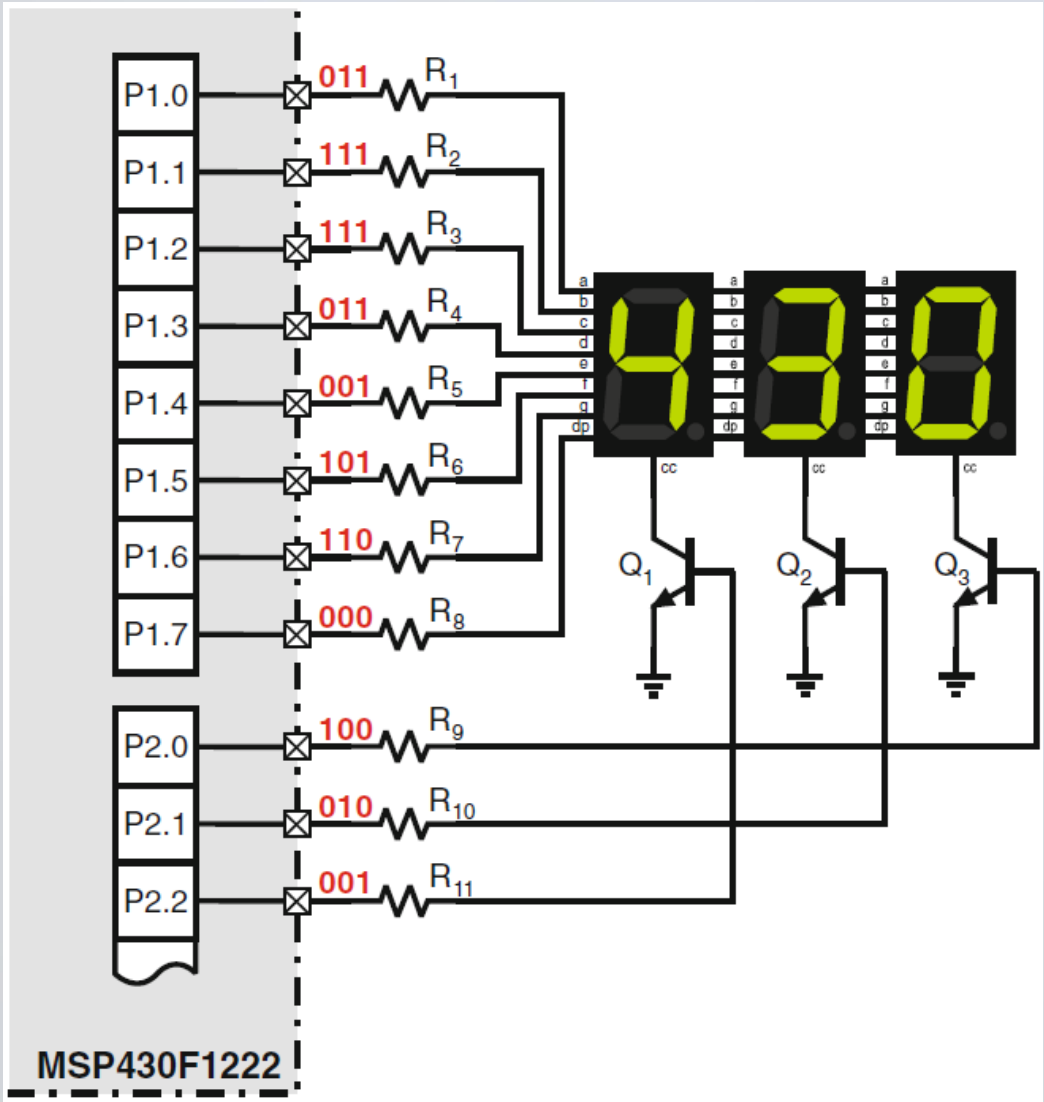
Multipleksno povezivanje nizova LED dioda III

- U isto vreme, tokom jednog ciklusa osvežavanja, svaki pojedinačni 7-segmentni displej biće aktivan samo tokom ***1/n***-tog dela ciklusa
- Ovo će dovesti do smanjenja sjajnosti svake cifre u istom iznosu, što može biti neprihvatljivo
- Jedan od načina da se ovaj problem reši je da se poveća jačina struje koja protiče kroz svaki segment u istoj proporciji, smanjujući vrednosti serijskih otpornika R1-R8
- Prilikom primene ove tehnike mora se voditi računa da se ne prekorači maksimalna jačina struje dozvoljena od strane I/O nožice ili 7-segmentnog displeja



Multipleksno povezivanje nizova LED dioda IV

- Druga bitna stvar o kojoj treba voditi računa jeste jačina struje koja će proticati kroz zajedničku katodu ili anodu 7-segmentnog displeja u slučaju da su svi segmenti uključeni
- Struja zajedničke katode ili anode može dostići vrednost od $8I_F$ u slučaju da su svi segmenti uključeni
- Za većinu GPIO portova ova jačina struje će biti veća od maksimalno dozvoljene
- U tom slučaju potrebno je korišćenje eksternog bafera, u vidu tranzistora Q1, Q2 i Q3, koji će moći da podnesu ovu veliku jačinu struje



Teorijska pitanja P9

Predavanja 9- Periferije embeded sistema

Spisak teorijskih pitanja uz Predavanja 9

1. Struktura ulazno/izlaznog porta opšte namene.

2. Generička struktura ulazno/izlazne nožice i konfigurisnjae GPIO portova.

3. Elektricne karakteristike I/O nožica.

4. Izlazni limiti i vremenske karakteristike I/O nožica.

5. Rad sa prekidačima i tasterima. Pravilno dimenzionisanje RPULL_UP i RPULL_DOWN otpornika.

6. Rad sa realnim prekidačima i tasterima. Vreme smirivanja prekidača i problemi izazvani "odskakivanjem" prekidača.7 Tehnike za elemininaciju efekta "odskakivanja" prekidača.

8. Povezivanje nizova prekidača. DIP prekidači i tastature.

9. Softversko skeniranje tastature.

10. Klasifikacija displeja.

11. Povezivanje LED dioda i nizoa LED dioda.

12. Multipleksno povezivanje nizova LED dioda.