

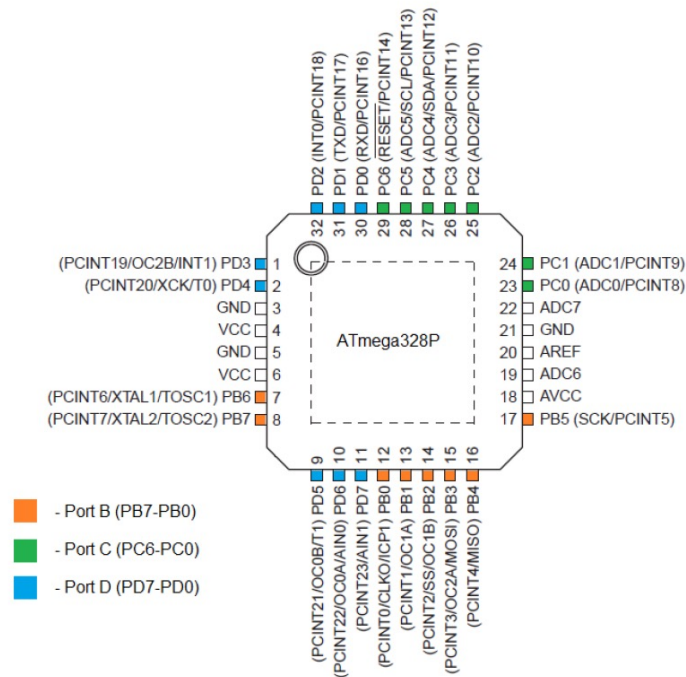


Ulazno-izlazni portovi

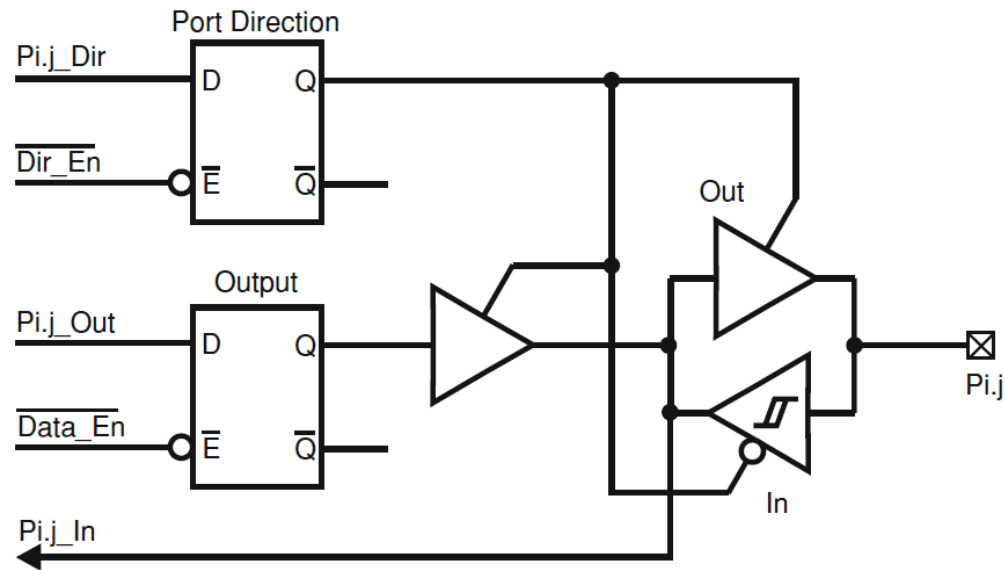
Katedra za elektroniku

Portovi

- Najveći broj nožica (pinova) na kućištu mikrokontrolera predstavlja ulazno-izlazne priključke namenjene povezivanju mikrokontrolera sa periferijskim uređajima.
- Svaki od ovih pinova može da služi kao digitalni ulaz/izlaz opšte namene (engl. GPIO = General Purpose Input/Output). Pored toga, pinovima mogu biti dodeljene i alternativne funkcionalnosti kao što su linije za slanje/prijem serijskog porta, ulazi A/D konvertora, PWM izlazi i sl. Na slici su obeleženi raspored i alternativne funkcije pinova mikrokontrolera ATmega328p.
- Grupe ulazno-izlaznih pinova nazivaju se portovi. U okviru jednog porta obično se nalazi do 8 ulazno-izlaznih priključaka.



Interna struktura GPIO pinova



Uobičajena struktura GPIO pina mikrokontrolera

- U primeru prikazanom na slici je prikazana interna struktura kontrolne logike pina $P_{i,j}$ (pin sa oznakom 'j' u okviru porta sa oznakom 'i')
- Flipflop koji pripada registru za kontrolu smera (*Port Direction* na slici) određuje da li će pin biti korišćen kao ulaz, ili kao izlaz.
- Ukoliko se pin koristi kao izlaz, njegovo stanje određeno je stanjem flipflopa koji pripada izlaznom registru (*Output* na slici)

Konfigurisanje GPIO pinova

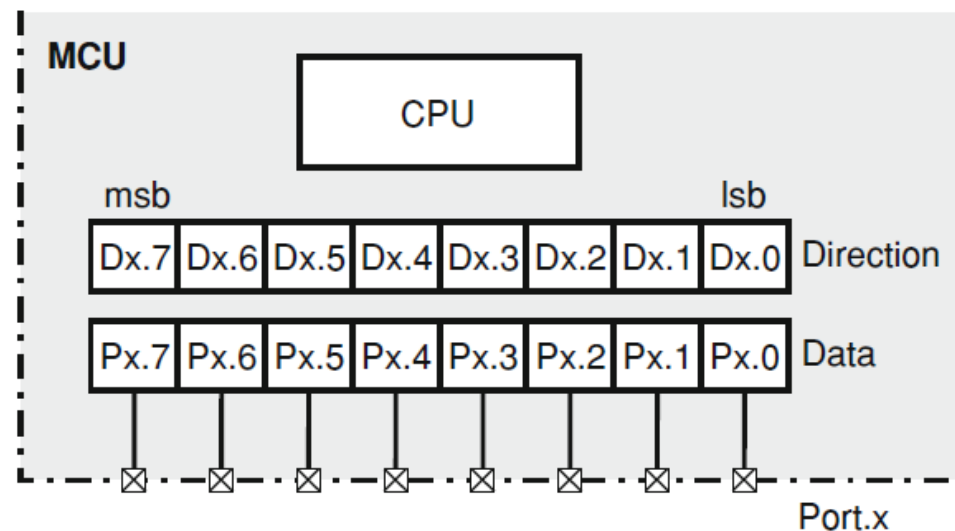
Da bi smo koristili GPIO port potrebno je da:

- a) Konfiguriramo smer toka podataka (kao ulazni ili izlazni)
- b) 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

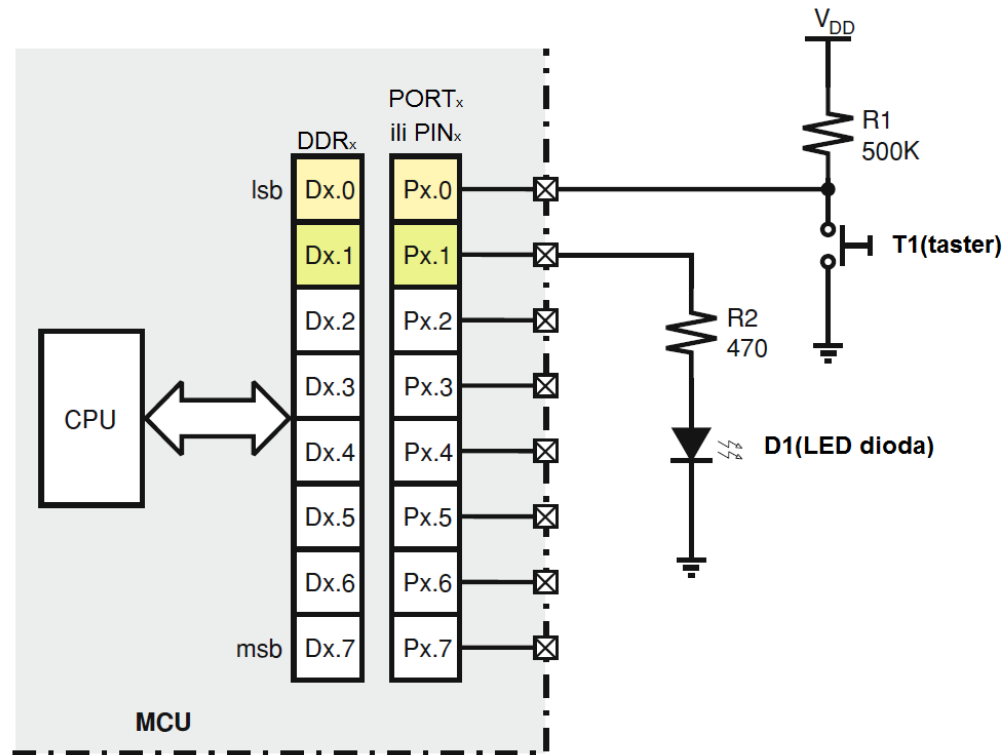
Isto pravilo važi i u slučaju kontrole smera toka podataka kroz individualne I/O linije

Na ovaj način je rad sa GPIO portovima maksimalno olakšan jer se oni tretiraju kao obični registri od strane softvera



Primer: Povezivanje portova mikrokontrolera sa tasterima i LED diodama

- U primeru prikazanom na slici, vidi se način povezivanja jednog ulaznog uređaja (taster) i jednog izlaznog uređaja (LED dioda) na pinove koji pripadaju istom portu (port X)



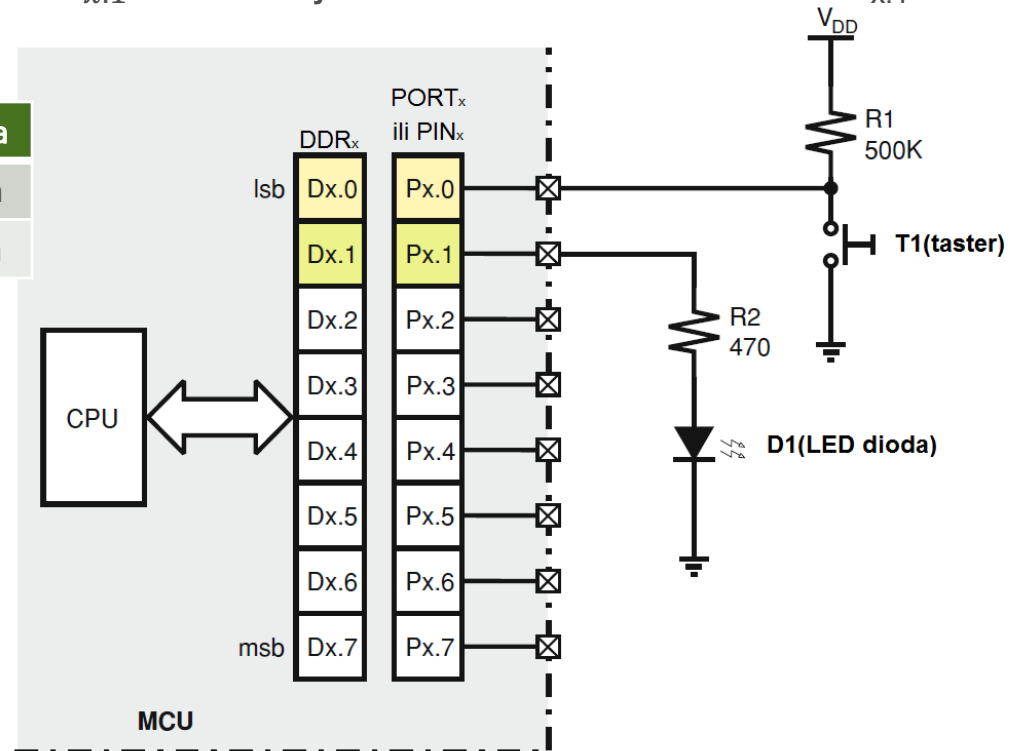
Primer: Povezivanje portova mikrokontrolera sa tasterima i LED diodama

- U primeru prikazanom na slici, vidi se način povezivanja jednog **ulaznog uređaja (taster)** i jednog izlaznog uređaja (LED dioda) na pinove koji pripadaju istom portu (port X)
- Pin $P_{x,0}$ je ulazni $\Rightarrow DDR_{x,0} = 0$. Stanje tastera se čita preko bita $PIN_{x,0}$ na sledeći način:

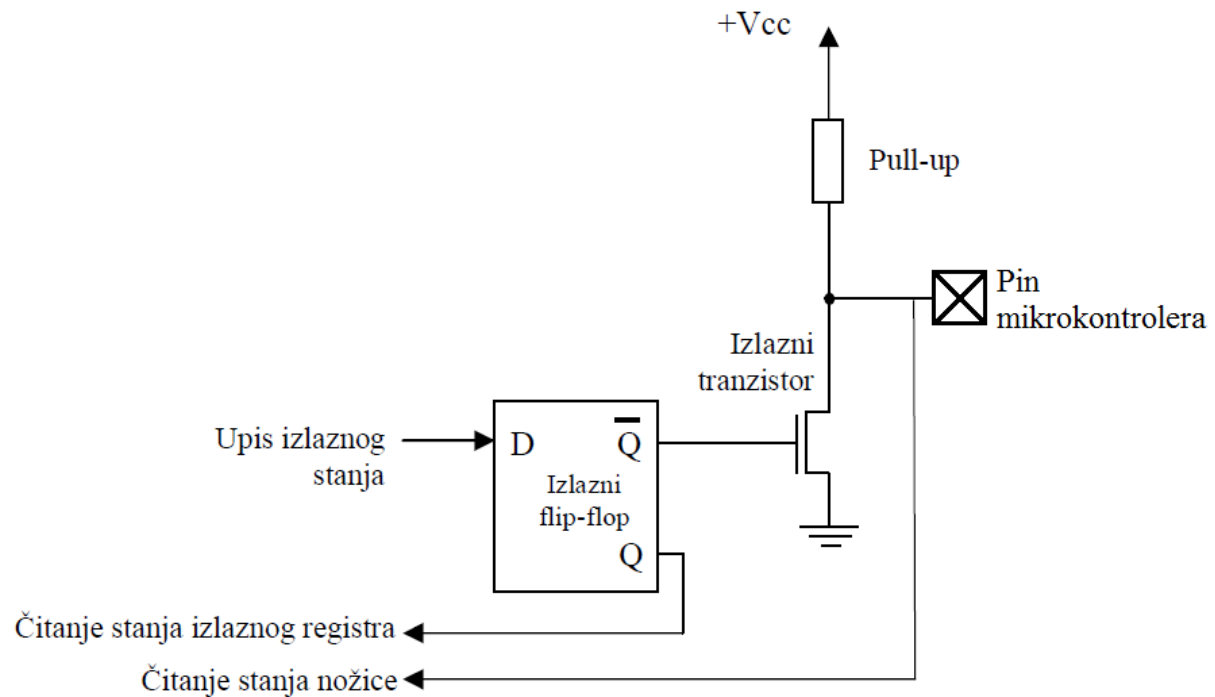
$PIN_{x,0}$	Taster
0	Pritisnut
1	Pušten

- Pin $P_{x,1}$ je izlazni $\Rightarrow DDR_{x,1} = 1$. Stanje LED diode diktira bit $PORT_{x,1}$ na sledeći način:

$PORT_{x,1}$	LED dioda
0	Isključena
1	Uključena



GPIO pinovi sa open-collector (open-drain) logikom



- Pinovi mikrokontrolera sa *open-collector* logikom odlikuju se “jakom” nulom i “slabom” jedinicom: ukoliko je u odgovarajući bit izlaznog registra upisana nula, uključuje se izlazni tranzistor. U suprotnom, izlazni tranzistor je isključen, pa je izlazno stanje određeno pull-up otpornikom.
- Ukoliko se pin koristi kao ulaz, u izlazni flip flop je neophodno upisati logičku jedinicu, da bi izlazni tranzistor bio isključen. Na ovaj način, eksterna logika upravlja stanjem pina i može ga po potrebi oboriti na 0.

Povezivanje uređaja pomoću ulazno/izlaznog porta opšte namene

- Prilikom povezivanja mikrokontrolera sa svojim okruženjem, često se javlja problem da se **električne karakteristike** nožica perifernih 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 perifernih 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

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

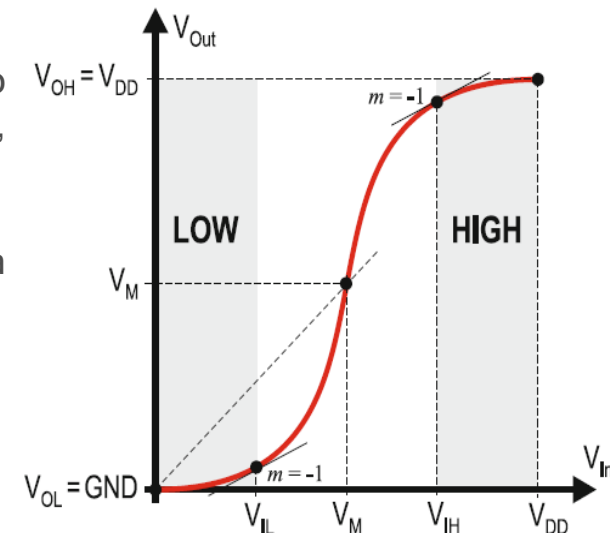
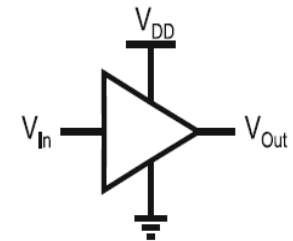
V_{IH} - minimalna vrednost ulaznog napona koja će biti interpretirana kao logička jedinica

V_{OH} - maksimalna vrednost izlaznog napona koja se može pojaviti na izlazu

V_{OL} - minimalna vrednost izlaznog napona koja se može pojaviti na izlazu

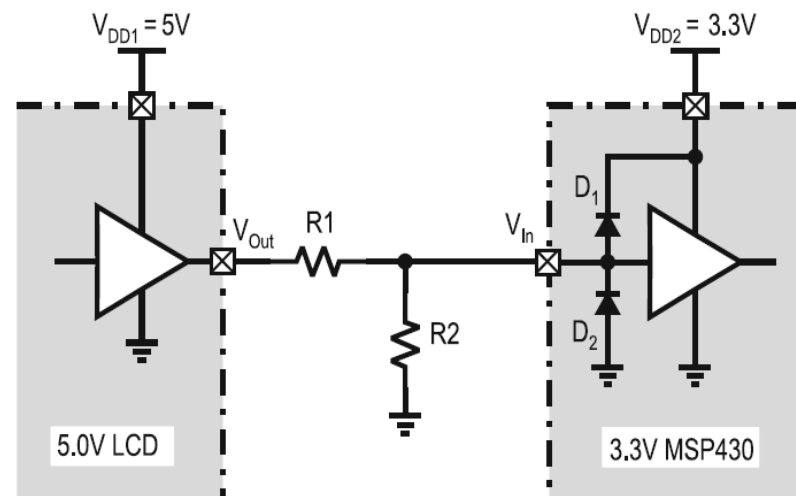
Oblasti vrednosti ulaznog napona koji će pravilno biti prepoznat kao logička nula, odnosno jedinica, označene su sa LOW i HIGH

Oblast koja leži između LOW i HIGH regiona predstavlja **tranzicionu oblast**



Električne karakteristike I/O nožica

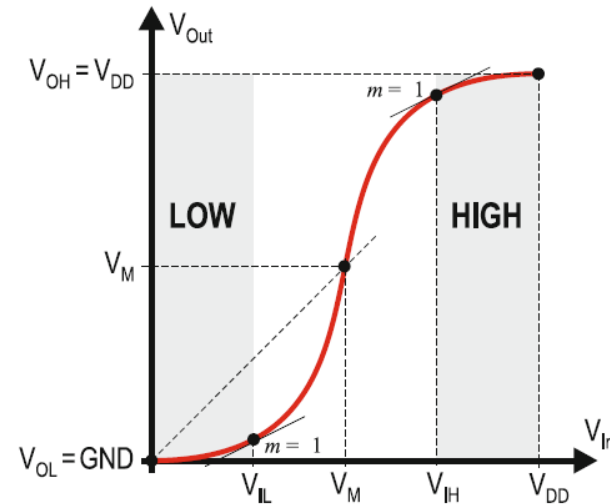
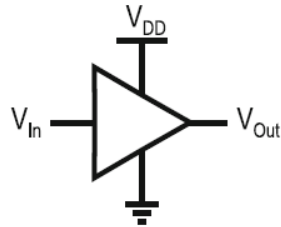
- U normalnom režimu rada vrednosti ulaznog napona nikada ne bi trebalo da napuštaju LOW i HIGH oblasti
- Ukoliko je to slučaj, stabilan rad sistema ne može se garantovati te je neophodno koristiti konvertore naponskih nivoa (level-shifter)
- Ova situacija se često javlja ako se naponi napajanja mikrokontrolera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



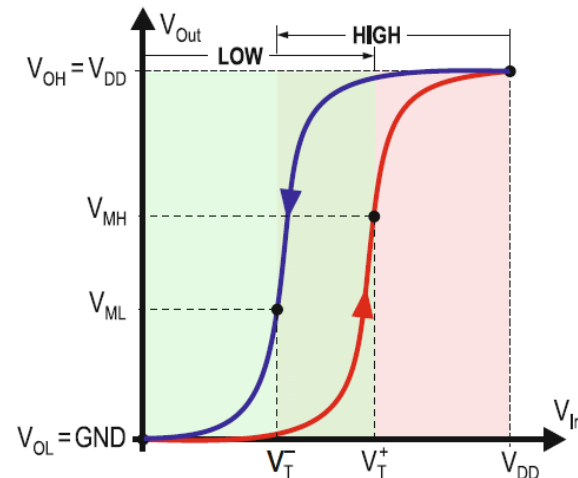
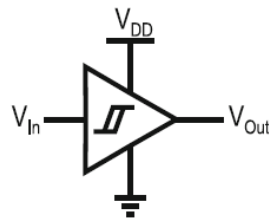
Vremenske karakteristike I/O nožica

- 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

Ulazne karakteristike GPIO pinova

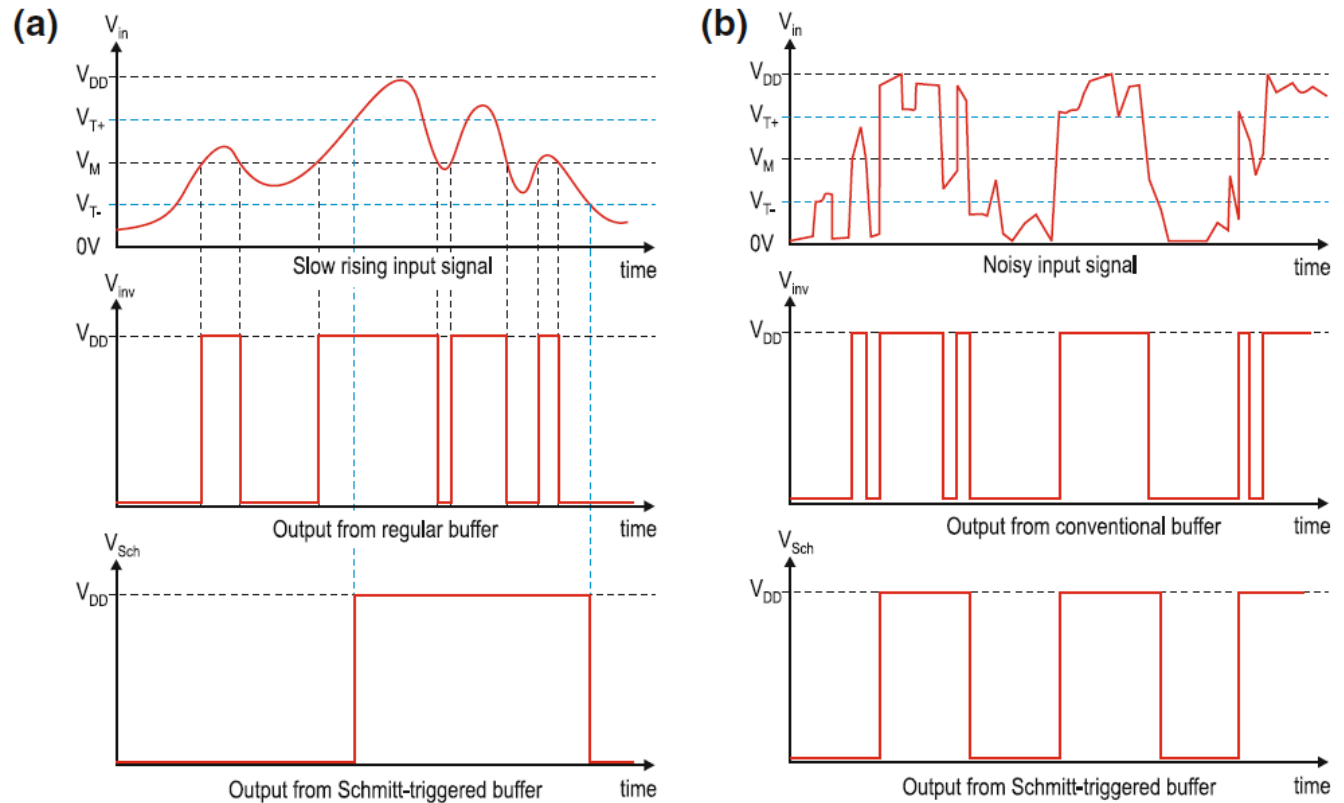


Simbol i prenosna karakteristika konvencionalnog bafera



Simbol i prenosna karakteristika bafera sa histerizisom (Šmit-triger)

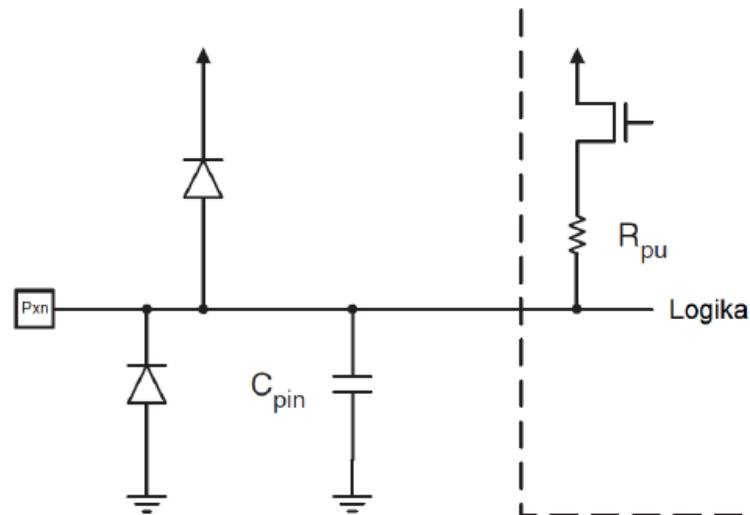
Vremenski odziv običnog i Šmit-triger bafera



- a) Odziv običnog i Šmit-triger bafera na sporo promenljiv ulazni signal
- b) Odziv običnog i Šmit-triger bafera na ulazni signal sa smetnjama

Portovi AVR mikrokontrolera - električne karakteristike

- Svi pinovi AVR mikrokontrolera imaju identične strujne karakteristike. Kada se pin koristi kao izlaz, struja teče od kontrolera ka perifernom uređaju, kada je pin u stanju logičke jedinice, odnosno od perifernog uređaja ka kontroleru, kada je pin u stanju logičke nule.
- Pin može da podnese struju jačine do 40mA, bez obzira na njen smer. To znači da su strujne mogućnosti pinova dovoljne za direktno upravljanje LED diodama.
- Diode koje su postavljene između pina i napona napajanja (V_{cc}), odnosno između pina i mase predstavljaju zaštitu od napona koji izlazi iz opsega ($0-V_d, V_{cc}+V_d$) i koji bi mogao da izazove oštećenje pina.
- Svaki pin ima interni pull-up otpornik R_{pu} , koji je moguće uključiti ili isključiti pomoću kontrolne logike kontrolisane od strane konfiguracionih registara.



Konfigurisanje pinova AVR mikrokontrolera (1)

- Kod AVR mikrokontrolera, svakom portu* su pridružena po 3 kontrolna registra:
 - **DDRx** - Određuje smerove pinova porta x
 - **PORTx** - Određuje stanja onih pinova porta x koji se koriste kao digitalni izlazi
 - **PINx** - Služi za očitavanje stanja onih pinova porta x koji se koriste kao digitalni ulazi

* U nastavku, sve oznake registara i njihovih bita će biti navedene u generalnoj formi, gde "x" označava slovnu oznaku porta (koja može biti B, C, ili D), a "n" označava poziciju bita (od 7 do 0). Npr, PB3 označava pin na poziciji 3 u okviru porta B.

- DDR_{xn} bit u okviru $DDRx$ registra vrši selekciju smera pina P_{xn} , na sledeći način:

$$DDR_{xn} = \begin{cases} 1, & \text{pin } P_{xn} \text{ je konfigurisan kao izlaz} \\ 0, & \text{pin } P_{xn} \text{ je konfigurisan kao ulaz} \end{cases}$$

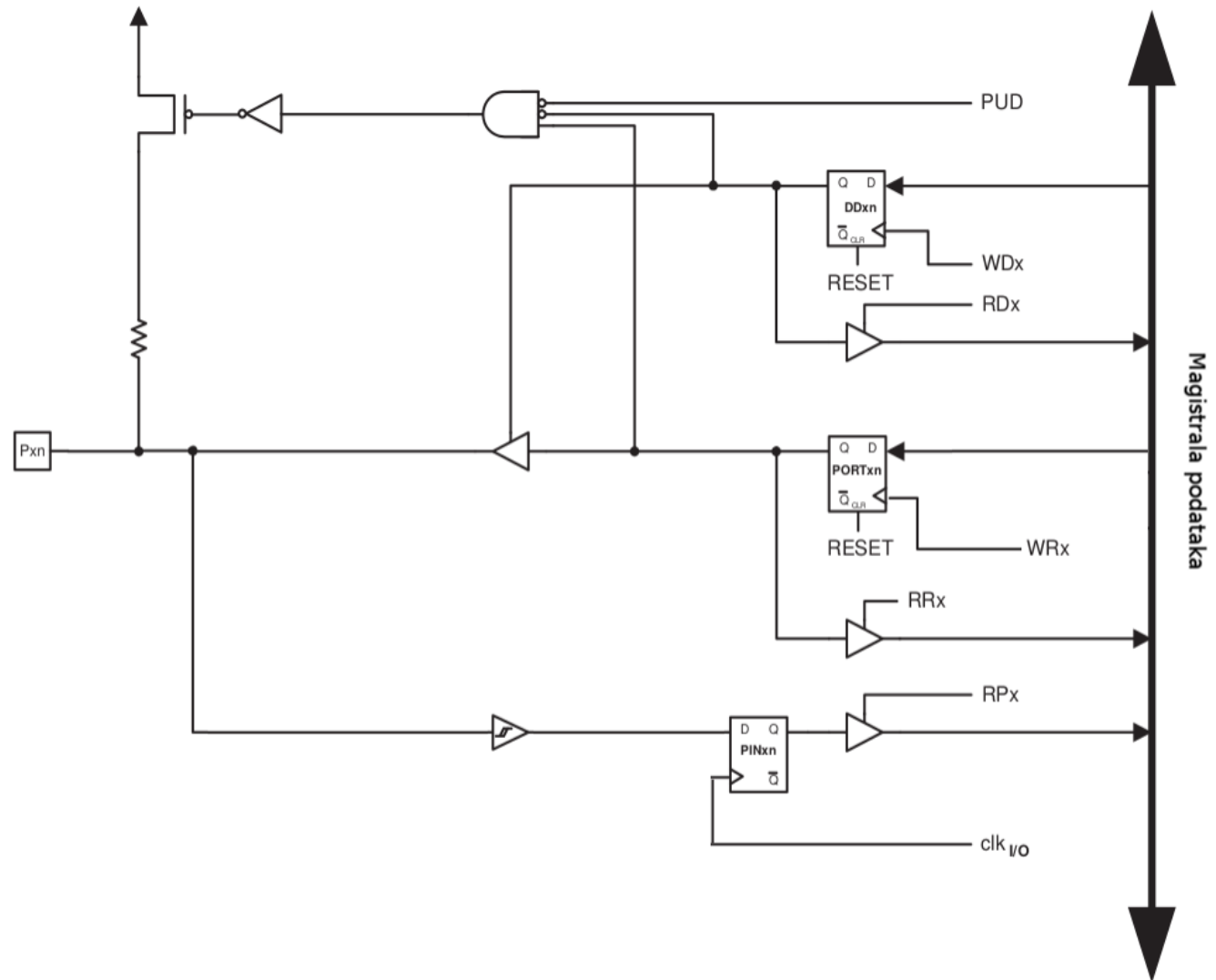
- Kada je $DDR_{xn} = 0$, tj. pin je konfigurisan kao ulaz, bit $PORT_{xn}$ upravlja pull-up otpornikom na sledeći način:

$$DDR_{xn} = 0, PORT_{xn} = \begin{cases} 1, & \text{pull-up otpornik je aktiviran} \\ 0, & \text{pull-up otpornik je isključen} \end{cases}$$

- Kada je $DDR_{xn} = 1$, tj. pin je konfigurisan kao izlaz, bit $PORT_{xn}$ određuje stanje pina P_{xn} .

Kontrolna logika pinova AVR mikrokontrolera

- Na slici je prikazana uprošćena šema kontrolne logike pina P_{xn} :



Kontrolni registri porta B

PORTB – The Port B Data Register

Bit	7	6	5	4	3	2	1	0									
0x05 (0x25)	<table><tr><td>PORTB7</td><td>PORTB6</td><td>PORTB5</td><td>PORTB4</td><td>PORTB3</td><td>PORTB2</td><td>PORTB1</td><td>PORTB0</td></tr></table>								PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	PORTB
PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0										
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W									
Initial Value	0	0	0	0	0	0	0	0									

DDRB – The Port B Data Direction Register

Bit	7	6	5	4	3	2	1	0									
0x04 (0x24)	<table><tr><td>DDB7</td><td>DDB6</td><td>DDB5</td><td>DDB4</td><td>DDB3</td><td>DDB2</td><td>DDB1</td><td>DDB0</td></tr></table>								DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	DDRB
DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0										
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W									
Initial Value	0	0	0	0	0	0	0	0									

PINB – The Port B Input Pins Address⁽¹⁾

Bit	7	6	5	4	3	2	1	0									
0x03 (0x23)	<table><tr><td>PINB7</td><td>PINB6</td><td>PINB5</td><td>PINB4</td><td>PINB3</td><td>PINB2</td><td>PINB1</td><td>PINB0</td></tr></table>								PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	PINB
PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0										
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W									
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A									

Note: 1. Writing to the pin register provides toggle functionality for IO

Kontrolni registri porta C

PORTC – The Port C Data Register

Bit	7	6	5	4	3	2	1	0	
0x08 (0x28)	–	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	PORTC
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

DDRC – The Port C Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x07 (0x27)	–	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	DDRC
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

PINC – The Port C Input Pins Address⁽¹⁾

Bit	7	6	5	4	3	2	1	0	
0x06 (0x26)	–	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	PINC
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

Note: 1. Writing to the pin register provides toggle functionality for IO

Kontrolni registri porta D

PORTD – The Port D Data Register

Bit	7	6	5	4	3	2	1	0	
0x0B (0x2B)	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	PORTD
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

DDRD – The Port D Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x0A (0x2A)	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

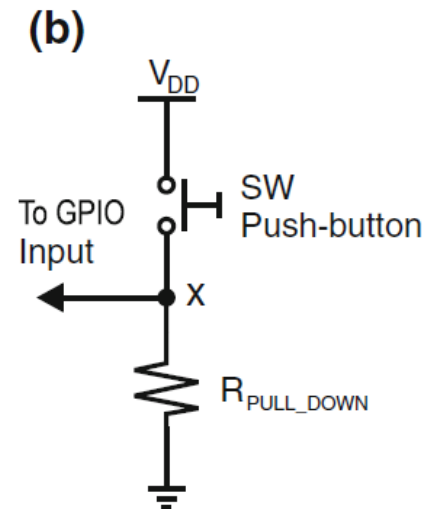
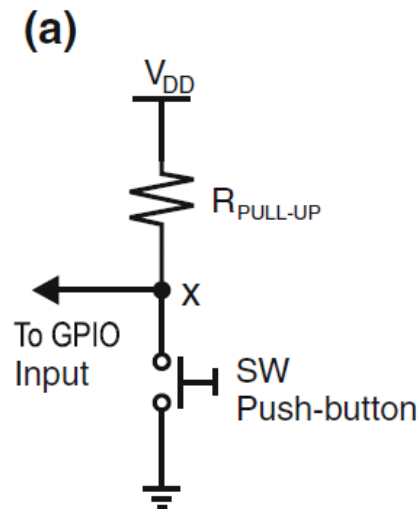
PIND – The Port D Input Pins Address⁽¹⁾

Bit	7	6	5	4	3	2	1	0	
0x09 (0x29)	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	PIND
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

Note: 1. Writing to the pin register provides toggle functionality for IO

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 otpornika

- **RPULL_UP** i **RPULL_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 **RPULL_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 otpornika

- 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 $0.9V_{DD}$, $V_{IH} = 0.9V_{DD}$.
- Dimenzionisati vrednost pull-up otpornika da bi se na GPIO nožicu pouzdano mogao povezati taster.
- Pretpostaviti da je $I_{SWmax} = 20I_{IH}$

Pravilno dimenzionisanje otpornika

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 prethodnog slajda 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 $I_{SWmax} = 20I_{IH} = 20\mu A$ i jednačinu sa prethodnog slajda, dobijamo donju granicu za vrednost pull-up otpornika

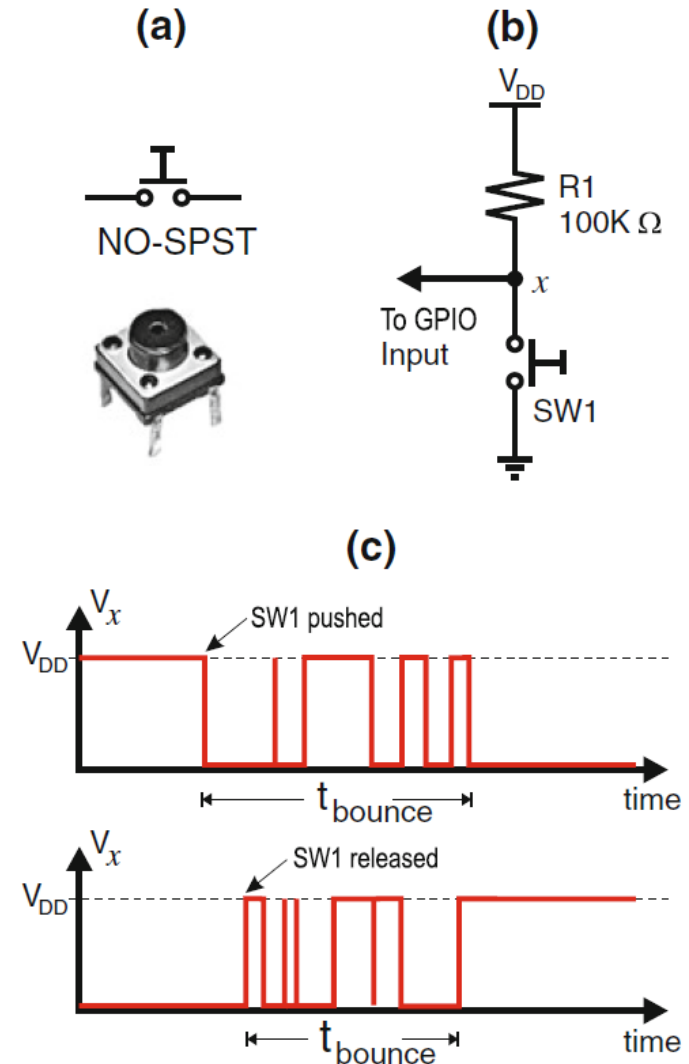
$$R_{PULL_UP} \geq \frac{V_{DD}}{I_{SWmax}} = \frac{5.0V}{20\mu A} = 250K\Omega$$

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 embedded 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 eliminaciju 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



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 očitano vrednost u ostatku programa
- Ukoliko se desi promena sa jedne vrednosti na drugu tokom ovih n sukcesivnih čitanja, proces brojanja se reinicijalizuje i započne 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