

Sistem prekida

dr Predrag Teodorovic

Fakultet Tehničkih Nauka, Novi Sad

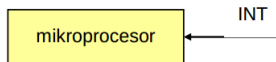
May 17, 2018

Zašto prekidi?

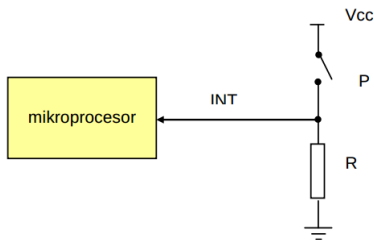
- ▶ U opštem slučaju prekid se koristi u primenama u kojima mikroprocesor obavlja dva zadatka, jedan nižeg prioriteta, koji se obično izvršava kontinualno u vremenu i drugi, višeg prioriteta, koji se izvršava povremeno, na pobudu nekim spoljnim događajem
- ▶ Analogija sa čitanjem knjige i telefonom koji zvonil
 - ▶ čitaj knjigu do kraja
 - ▶ prekini čitanje i zanemari sve pročitano do tada
 - ▶ čeka da telefon zazvoni, pa tek onda krene sa čitanjem knjige
 - ▶

Šta je najbolje?!?

- ▶ Najbolje rezultate daje taktika po kojoj osoba čita knjigu, a kada telefon zazvoni, zapamti mesto do koga je došla u čitanju, odgovori na telefonski poziv, a zatim nastavi sa čitanjem **od zapamćenog mesta u knjizi**
- ▶ Ova taktika daje dobru produktivnost i sinhronizaciju sa pozivima koji se događaju u trenucima koje osoba ne može da predvidi
- ▶ Da bi dobio informaciju o spoljnjem događaju, mikroprocesor ima poseban ulazni signal koji se obično naziva INT, od engleskog '*interrupt*'



Primer sa prekidačem

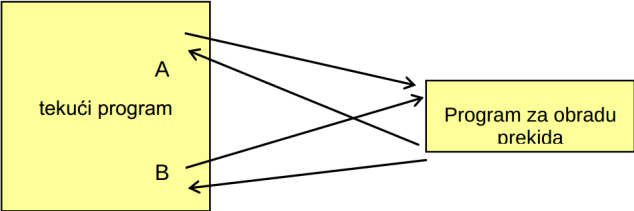
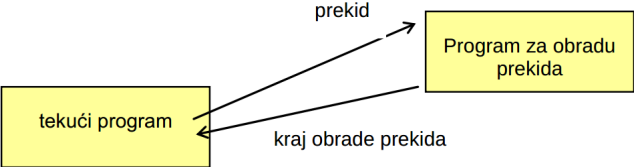


- ▶ Bez mehanizma prekida, mikroprocesor bi morao, sa vremena na vreme, da proverava stanje na ulaznom pinu na kojem je priključen prekidač

Zašto ovo nije dobro?

1. Procesor je u takvoj situaciji znatno manje efikasan jer će puno puta rezultat provere stanja biti takav da se ništa nije desilo, pre nego što se desi promena stanja na ulazu. Na primer, tek posle milion provera naponski nivo na ulazu INT bi bio Vcc, što znači da je procesor mogao nešto pametnije da radi umesto da milion puta proverava napon na datom ulazu
2. Ukoliko se provera stanja na ulazu dešava veoma često, efekat gore naveden je još izraženiji. Ako se, pak, provera vrši ređe (u cilju što ređeg prekidanja rada mikroprocesora), može se desiti da se pritisak tastera ne detektuje (ili da se detektuje se neprihvatljivo velikim kašnjenjem)

Rešenje: Prekid!!!



Terminologija

- ▶ U operativnoj memoriji mikroračunara mora da postoji **program za obradu prekida** ili **prekidna rutina**
- ▶ Program koji mikroprocesor obrađuje kada nema prekida obično se naziva **tekući program** ili **glavni program**
- ▶ Kada se aktivira signal prekida, mikroprocesor sačeka izvršavanje instrukcije tekućeg programa koja je u toku i, nakon toga, pređe na izvršavanje programa za obradu prekida
- ▶ Instrukcija u kojoj je tekući program prekinut naziva se tačka prekida

Problemi?!?

- ▶ Po završetku programa za obradu prekida mikroprocesor mora da se vrati u tekući (prekinuti) program i nastavi izvršavanje od instrukcije u kojoj je prekinuto izvršavanje tekućeg programa
- ▶ Mehanizam prekida mora da reši pitanje povratne adrese, odnosno mora da na neki način omogući mikroprocesoru da nastavi izvršavanje tekućeg programa na mestu u kojem je bio prekinut
- ▶ Kako se ovo rešava?
- ▶ Tu nam ponovo pomaže STEK!

Dodatni problemi

- ▶ U praksi obično postoje višestruki izvori prekida. Na primer, u mikroračunarskom sistemu mogu da postoje prekidi koje generiše tastatura, sat realnog vremena, masovna memorija, komunikaciona linija i tako dalje.
- ▶ U slučaju višestrukih izvora prekida, podsistem za prekid mora da obezbedi:
 - ▶ programe za obradu različitih signala prekida
 - ▶ prepoznavanja izvora prekida
 - ▶ rešavanje prioriteta prekida kada se istovremeno aktivira više od jednog prekidnog signala
 - ▶ prelazak na program za obradu prekida koji odgovara izvoru prekida i
 - ▶ postupak u slučaju da se, u toku obrade jednog prekida, aktivira neki drugi signal prekida

Demonstraciju problema

- ▶ Razmatramo situaciju u kojoj su dozvoljena dva izvora prekida, prekid serijske komunikacije (primljen karakter preko serijskog porta) i eksterni prekid koji se dešava kada se pritisne taster spojen na jedan od pinova mikrokontrolera
- ▶ Recimo, takođe da je eksterni događaj reprezentovan pritiskom tastera visokog prioriteta i da se na ovaj pritisak tastera mora brzo odreagovati: u takvoj situaciji, prekid serijske komunikacije (izazvan na primer prijemom novog karaktera preko serijske veze) je niskog prioriteta, dok je eksterni prekid visokog prioriteta
- ▶ Na sledećem slajdu je prikazan sadržaj memorije mikrokontrolera u kojem su dozvoljena ova dva izvora prekida. Na slajdovima koji slede, posmatramo promene u sistemu, u skladu sa pojavama prekida

Sadržaj memorije na početku



Dešava se prekid dok se izvršava instrukcija sa adrese 0x0100

Pristigao karakter sa serijskog porta

PC=0x0100
SP=0xFFFF

PC →

SP →

adresa	sadržaj
0x0000	
0x0001	
0x0002	
0x0003	
0x0004	
...	
0x0024	
0x0025	
0x0026	
...	
0x0040	
...	
0x0100	
...	
0x0200	
...	
0x1000	
0x1001	
0x1002	
...	
...	
0xFFFF	
0xFFFE	
0xFFFF	

PC →

PC=0x0024
SP=0xFFFE

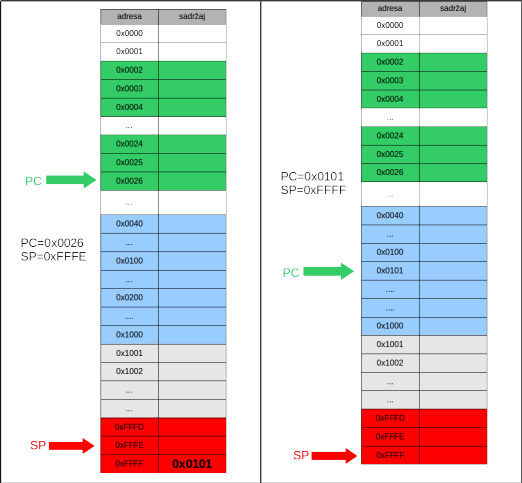
SP →

adresa	sadržaj
0x0000	
0x0001	
0x0002	
0x0003	
0x0004	
...	
0x0024	
0x0025	
0x0026	
...	
0x0040	
...	
0x0100	
...	
0x0200	
...	
0x1000	
0x1001	
0x1002	
...	
...	
0xFFFF	
0xFFFE	
0xFFFF	0x0101

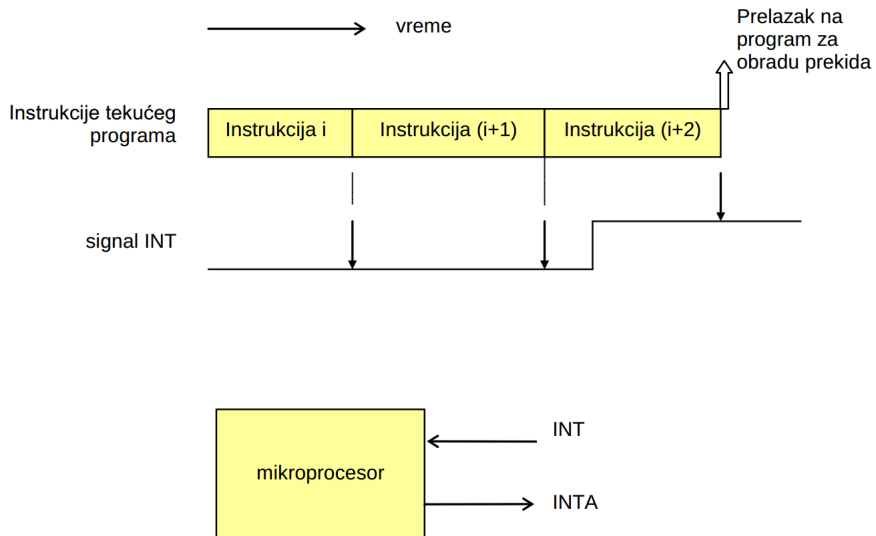
Tokom izvršavanja prekidnog programa, dešava se novi prekid



Nastavak nakon završetka obrade prekida



Odziv mikroprocesora na signal prekida



Koraci prilikom odziva

- ▶ aktivirati signal INTA
- ▶ odrediti početnu adresu programa za obradu prekida (za svaki izvor prekida ove adrese su fiksne i poznate - pogledajte primer gore)
- ▶ staviti na stek povratnu adresu koja se nalazi u programskom brojaču
- ▶ staviti u programski brojač početnu adresu programa za obradu prekida
- ▶ signal INTA prevesti u neaktivno stanje i
- ▶ nastaviti izvršenje instrukcija.

Mikroprogram odziva na signal prekida

- ▶ $INTA \leftarrow 1$; prekid prihvaćen, aktivirati signal INTA
- ▶ $SP \leftarrow SP - 1$; mesto na steku za povratnu adresu
- ▶ $M[SP] \leftarrow PC$; povratna adresa iz PC na stek
- ▶ $PC \leftarrow$ adresa programa za obradu prekida
- ▶ $INTA \leftarrow 0$; signal INTA prevesti u neaktivno stanje

Povratak iz programa za obradu prekida

- ▶ $PC \leftarrow M[SP]$; mesto na steku za povratnu adresu
- ▶ $SP \leftarrow SP+1$; povratna adresa iz PC na stek