

---

# Sistem prekida

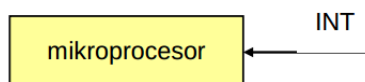
---

Osnovni ciljevi uvođenja mehanizma prekida su povećanje produktivnosti mikroprocesora i sinhronizacija rada mikroprocesora sa događajima u njegovoj okolini. Podsystem prekida omogućava mikroprocesoru da, pod dejstvom nekog spoljnog događaja, prekine izvršavanje tekućeg programa i pređe ne izvršavanje programa koji je pripremljen za obradu tog događaja. Na primer, u mikrorračunarskom sistemu mogu da postoje dva programa: jedan, koji obavlja statističku obradu podataka smeštenih na disku, i drugi, koji prihvata podatke preko komunikacione linije. Ukoliko komunikaciona linija nije aktivna, mikroprocesor obrađuje podatke sa diska. Međutim, u slučaju da su pristigli podaci, mikroprocesor mora da prekine statističku obradu i izvrši program za prijem podataka sa komunikacione linije. Jasno, ukoliko mikroprocesor na vreme ne obavi prijem podataka, oni mogu biti nepovratno izgubljeni. U ovom poglavlju izložene su osnovne ideje i način realizacije podsistema prekida.

## 1 Osnovna ideja i problemi vezani sa sistemom prekida

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. Ovakav scenario može se uporediti sa osobom koja čita knjigu i treba da odgovara na telefonske pozive. Čitanje knjige je zadatak nižeg prioriteta, koji dugo traje i može da se obavlja ako osoba nema nešto drugo da radi. Podrazumeva se da obavljanje ovog zadatka neće biti ugroženo ako osoba odloži čitanje za neki kraći vremenski interval. Pretpostavlja se da je odgovor na telefonski poziv zadatak višeg prioriteta i zahteva da taj zadatak mora da se obavi kada se pojavi potreba za njim. Drugim rečima, zahtev je da osoba mora odgovoriti na poziv pre nego što telefon prestane da zvoni. U slučaju da telefon zazvoni, osoba koja čita knjigu može da primeni različite taktike. Na primer, može pokušati da pročita celu knjigu, pa tek onda da odgovori na poziv, taktika koja skoro sigurno ne daje dobre rezultate. Ili, na primer, osoba može da prekine čitanje knjige, zanemari do tada pročitanji deo knjige tako što nije zapamtila mesto na kome je prekinuta, odgovori na telefonski poziv i onda krene da čita knjigu od početka. U krajnjoj linije, može se zamisliti i taktika u kojoj osoba prvo čeka telefonski poziv, odgovori na njega, i onda krene sa čitanjem knjige. Intuitivno, 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' (prekid), slika 1. Radi jednostavnosti, ostali signali mikroprocesora, kao što su magistrale i upravljački signali, nisu predstavljeni na slici.



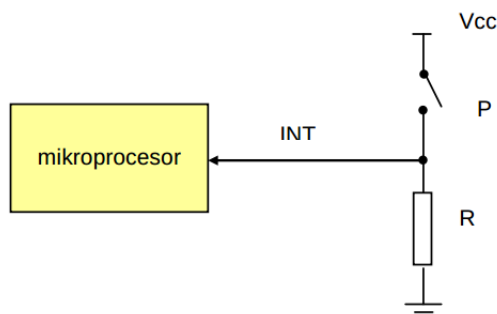
**Slika 1:** Ulazni signal INT za prekid

Naravno, u okviru mikroprocesorskog sistema mora da postoji elektronsko kolo koje, kada detektuje spoljnji događaj, generiše signal prekida.

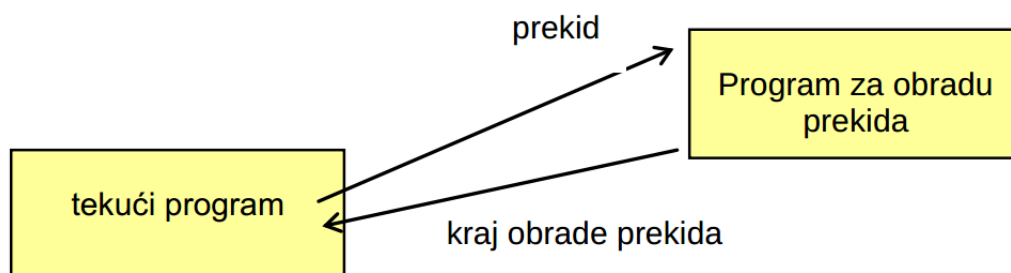
Kao primer uzmimo da mikroprocesor treba da obradi signal prekida koji je nastao kao posledica aktiviranja nekog mehaničkog prekidača. Mehanički prekidač može biti vezan na tipku tastature, tako da pritisak prsta na tipku aktivira prekidač i time signal prekida. Prekidač može aktivirati neki mehaničkih događaj, na primer zatvaranje vrata na liftu ili nailazak dela mašine na prepreku.

Slika 2 prikazuje jednostavno električno kolo koje generiše signal prekida INT u slučaju da je signal prekida aktivan na logičkoj jedinici. Kada je prekidač otvoren, signal INT je preko otpornika R vezan na masu, pa je na niskom naponskom nivou, odnosno na nivou logičke 0. Kada se prekidač zatvori, signal INT je vezan na napon napajanja Vcc i ima vrednost logičke 1. Prema tome, zatvaranjem prekidača aktivira se logička 1 na ulazu INT i time signalizira mikroprocesoru signal prekida. Bez korišćenja mehanizma prekida, mikroprocesor bi morao, sa vremena na vreme, da proverava stanje na ulaznom pinu na kojem je priključen prekidač. Ovo rešenje nije dobro iz dva razloga:

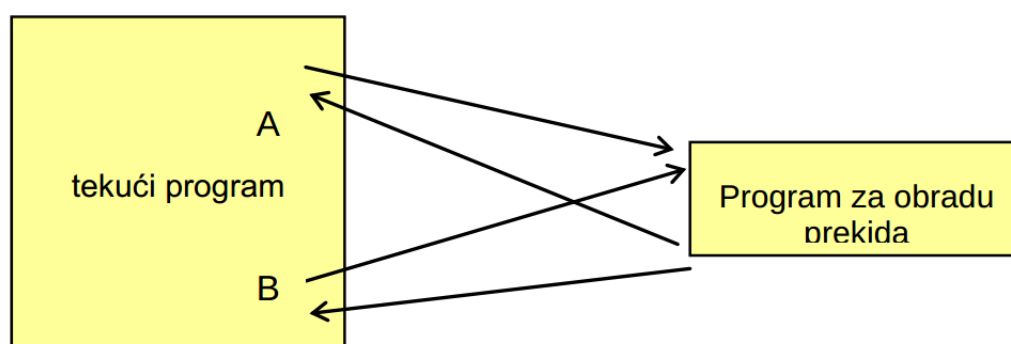
1. Procesor je u takvoj situaciji znatno manje efikasan jer će puno puta rezultat provere stanja biti takav da se ništa nije desilo, pre nego što se desi promena stanja na ulazu. Na primer, tek posle milion provera naponski nivo na ulazu INT bi bio Vcc, što znači da je procesor mogao nešto pametnije 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



**Slika 2:** Jednostavan način generisanja signala prekida mehaničkim prekidačem



Slika 3: Tekući program i program za obradu prekida



Slika 4: Ilustracija problema povratne adrese

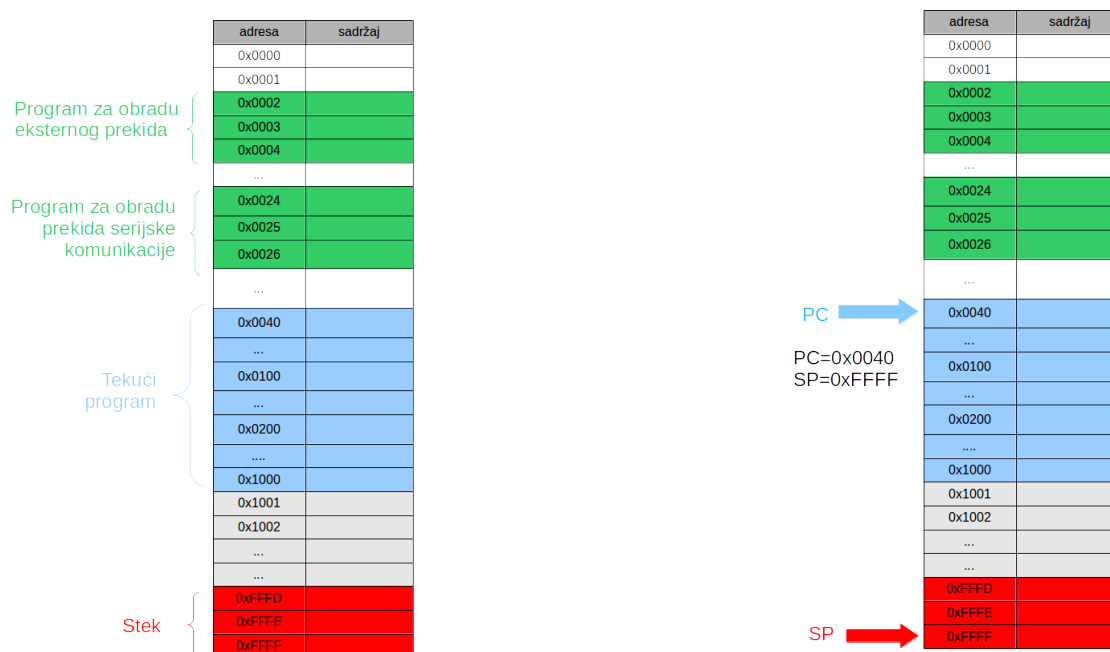
se desiti da se pritisak tastera ne detektuje (ili da se detektuje se neprihvatljivo velikim kašnjenjem).

Pre nego što se koristi mehanizam prekida, u operativnoj memoriji mikror računara mora da postoji program za obradu prekida, slika 3. Program koji mikroprocesor obrađuje kada nema prekida obično se naziva 'tekući program'. Kada se aktivira signal prekida, mikroprocesor sačeka izvršavanje instrukcije koja je u toku i pređe na izvršavanje programa za obradu prekida. Instrukcija u kojoj je tekući program prekinut naziva se tačka prekida.

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. Preciznije, tekući program nastavlja izvršenje od instrukcije koja sledi posle tačke prekida (zato što je instrukcija na tački prekida već izvršena).

Naravno, 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. Slika 4 ilustruje problem povratne adrese kod prekida.

Ako je tekući program prekinut u tački A, onda po završetku programa za obradu prekida, tekući program mora da se nastavi u tački A. Međutim, ako je tekući program prekinut u tački B, onda mora da se nastavi u istoj tački. Drugim rečima, mehanizam prekida mora da omogući tačnu povratnu adresu, koja će posle završetka programa za obradu prekida, obezbediti nastavak tekućeg



Slika 5: Sadržaj memorije sa dva dozvoljena izvora prekida

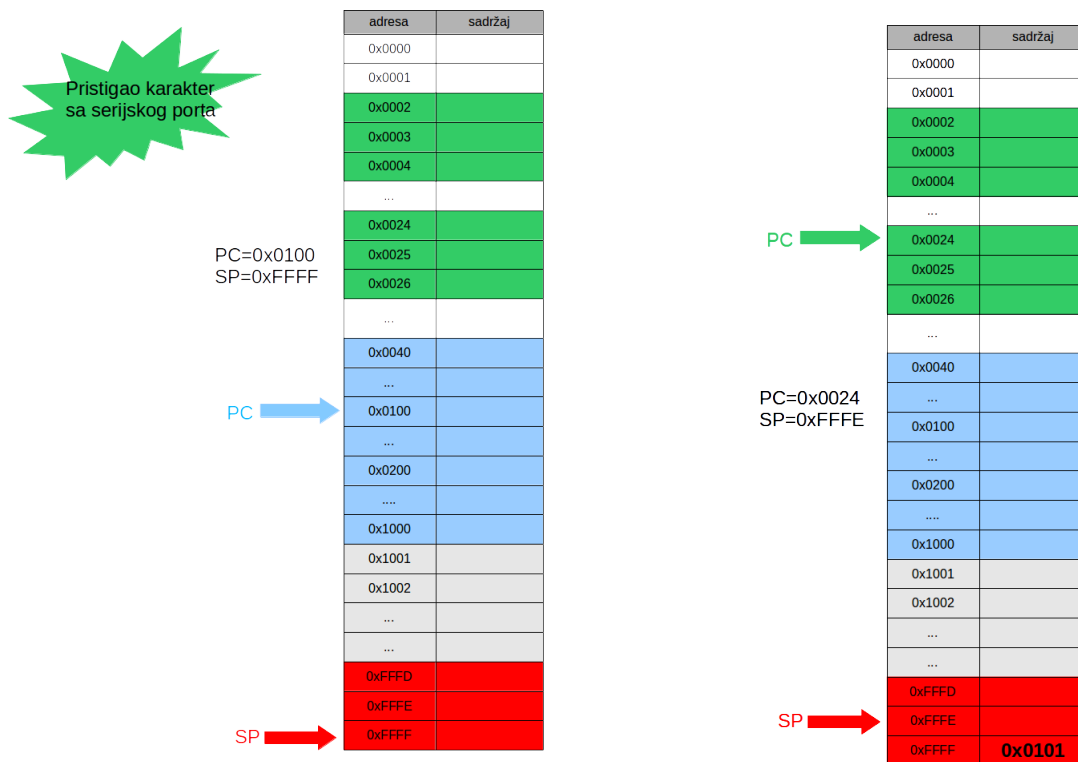
programa u tački prekida. Slično kao kod poziva potprograma, problem povratne adrese efikasno se rešava korišćenjem steka, odnosno stavljanjem povratne adrese na stek. Kod povratka u tekući program mikroprocesor uzme sa steka povratnu adresu na kojoj se nalazi instrukcija od koje se nastavlja izvršavanje tekućeg programa.

U praksi obično postoje višestruki izvori prekida. Na primer, u mikroročunarskom sistemu mogu da postoje prekidi koje generiše tastatura, signal 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.

U daljem tekstu se detaljnije obrađuje odziv mikroprocesora na signal prekida, rešavanje problema vezanih za višestruke prekide i druga pitanja vezana za podsistem prekida.

Da bi se demonstrirao problem o kojem pričamo, razmatramo situaciju u kojoj su dozvoljena dva izvora prekida, prekid serijske komunikacije i eksterni prekid koji se dešava kada se pritisne taster spojen na jedan od pinova mikrokontrolera (slično kao što je prikazano na slici 2). 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



Slika 6: Izvršavanje tekućeg programa i prekid serijske komunikacije

visokog prioriteta. Slika 5 prikazuje primer sadržaja memorije mikrokontrolera, kojem su dozvoljeni prekidi. Programi za prekide se nalaze na fiksni, unapred poznatim memorijskim lokacijama (kao što ćemo videti u narednom odeljku, takva je situacija i kod mikrokontrolera ATmega) i programer unapred zna na koje memorijske lokacije treba da smesti program za obradu prekida (za svaki konkretan izvor prekida koji želi da podrži u svom programu). Neka su u primeru sa slike te adrese 0x0002 za program za obradu eksternog prekida i 0x0024 u slučaju programa za obradu prekida serijske komunikacije. Neka tekući program počinje da se izvršava počevši od adrese 0x0040 na kojoj počinje prva instrukcija tekućeg programa (slika 5):

1) Tekući program se izvršava sve do trenutke kada ne počne sa izvršavanjem instrukcija na adresi 0x0100 (PC=0x0100) kada se generiše prekid serijske komunikacije usled pristiglog karaktera putem serijskog porta (slika 6)

2) Adresa povratka se stavlja na stek (adresa od koje treba nastaviti izvršavanje tekućeg programa nakon što se završi program za obradu prekida serijske komunikacije. Program za obradu prekida serijske komunikacije kreće da se izvršava, nakon što se u PC registar upiše adresa 0x0024, jer se na toj adresi uvek nalazi program za obradu tog prekida (slika 6)

3) Program za obradu prekida serijske komunikacije se izvršava sve dok PC ne dođe do adrese 0x0025, kada se generiše novi prekid, visokog prioriteta, kao posledica tastera koji je pritisnut i na koji treba momentalno odreagovati (slika 7)

4) Na stek se ponovo stavlja adresa povratka, ovaj put ta adresa je 0x0026 (nastaviće se sa izvršavanjem program za obradu prekida serijske komunikacije, čim se završi program za obradu eksternog prekida višeg prioriteta). U PC se upisuje 0x0002 jer je to adresa na kojoj se uvek nalazi



Slika 7: Eksterni prekid tokom izvršavanja programa za obradu prekida serijske komunikacije

program za obradu eksternog prekida (slika 7)

5) Nakon što se do kraja izvrši program za obradu eksternog prekida, sa vrha steka se uzima adresa i upisuje u PC, čime će nastaviti da se izvršava program za obradu prekida serijske komunikacije od adrese 0x0026 (slika 8)

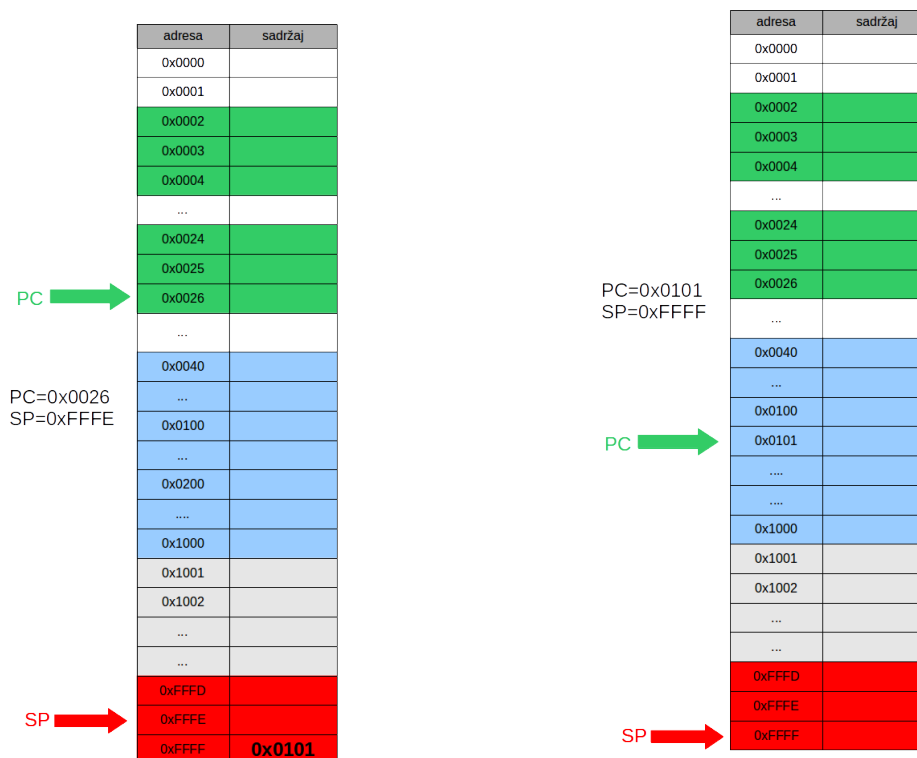
6) Čim se ovaj program završi, na isti način se sa steka uzima adresa povratka i upisuje u PC (PC=0x0101). Na taj način će se nastaviti sa izvršavanjem tekući program koji je prvobitno bio prekinut (slika 8).

## 2 Odziv mikroprocesora na signal prekida

Upravljačka jedinica mikroprocesora nadgleda ulazni signal prekida, neka je taj signal označen sa INT, i proverava stanje ovog signala na kraju izvršenja svake instrukcije. Ako signal INT nije aktivan, mikroprocesor izvršava sledeću instrukciju, a ako je signal INT aktivan mikroprocesor prelazi na obradu prekida.

Slika 9 ilustruje postupak provere stanja signala INT, koji je predstavljen linijom sa dva nivoa. Niži nivo predstavlja logičku 0, a viši nivo logičku 1. Izvršavanje instrukcije u vremenu predstavljeno je pravougaonikom u koji je upisan broj instrukcije koja se izvršava. U primeru na slici počinjemo da posmatramo izvršavanje tekućeg programa od instrukcije sa rednim brojem i, posle koje sledi instrukcija (i+1), zatim (i+2) i tako dalje.

Na završetku svake instrukcije, upravljačka jedinica proverava da li je signal INT aktivan, a trenuci provere (koji se podudaraju sa završetkom instrukcije) na slici su predstavljeni vertikalnim strelicama. Na primer, posle instrukcije sa rednim brojem i signal INT nije aktivan i nastavlja se



**Slika 8:** Nastavak izvršavanja programa za obradu prekida serijske komunikacije (levo) i nastavak izvršavanja tekućeg programa (desno)

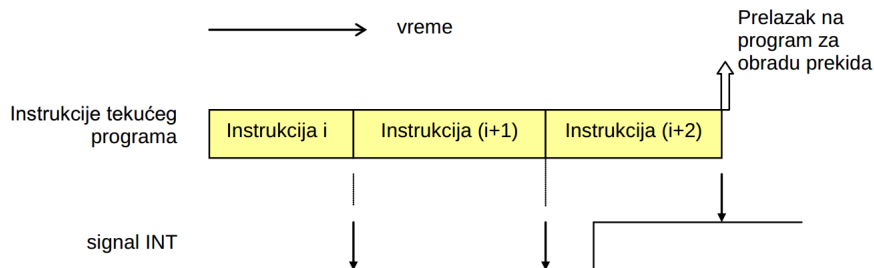
izvršenje instrukcije (i+1). Po završetku instrukcije sa rednim brojem (i+1) signal INT takođe nije aktivan i nastavlja se izvršenje instrukcije (i+2). Međutim, po završetku instrukcije (i+2) signal prekida je aktivan i mikroprocesor prelazi na program za obradu prekida.

Treba primetiti da trenutak aktiviranja signala INT nije od značaja, odnosno signal INT može da se aktivira u bilo kojem trenutku u toku izvršenja instrukcije (i+2). Upravljačka jedinica ignoriše stanje signala INT u toku izvršenja instrukcije, prema tome važno je samo da je u trenutku provere taj signal stabilan i aktivan.

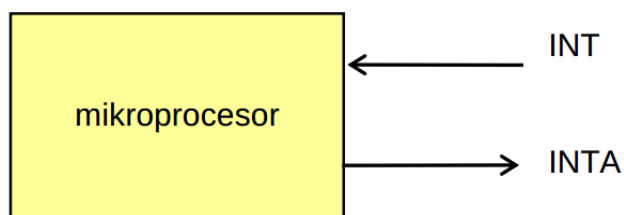
Postoje razlozi da mikroprocesor obavesti perifernu jedinicu, koja je zahtevala prekid, da je prekid prihvaćen i da mikroprocesor prelazi na obradu signala prekida. Obično mikroprocesor ima poseban spoljni izlazni signal koji se aktivira kada mikroprocesor prihvati prekid. Slika 10 prikazuje mikroprocesor sa signalom prekida INT i signalom da je prekid prihvaćen, koji se često označava sa INTA (engleski: Interrupt Acknowledge).

Različiti mikroprocesori na različit način reaguju na signal prekida. U principu, odziv mikroprocesora na aktivan signal prekida obuhvata sledeće korake:

- aktivirati signal INTA,
- odrediti početnu adresu programa za obradu prekida,
- staviti na stek povratnu adresu koja se nalazi u programskom brojaču,
- staviti u programski brojač početnu adresu programa za obradu prekida,



Slika 9: Upravljačka jedinica proverava stanje signala *INT* na kraju svake instrukcije



Slika 10: Mikroprocesor sa signalom za prekid (*INT*) i signalom da je prekid prihvaćen (*INTA*)

- signal *INTA* prevesti u neaktivno stanje i
- nastaviti izvršenje instrukcija.

Uz pretpostavku da povratna adresa staje u jednu memorijsku reč, mikroprogram koji određuje odziv mikroprocesora na signal prekida izgleda ovako:

```

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

```

Naravno, ključni problem u ovom mikroprogramu je određivanje adrese programa za obradu prekida. Ovaj problem obrađen je u narednom odeljku.

Mikroprogram za odziv mikroprocesora na signal prekida sličan je mikroprogramu instrukcije *CALL* poziva potprograma. Ipak, postoje značajne razlike, na primer zato što mikroprogram instrukcije *CALL* ne utiče na signal *INTA*. Povratak iz programa za obradu prekida u prekinuti program može se opisati mikroprogramom:

```

PC ← M[SP] ; povratna adresa sa steka u PC
SP ← SP+1 ; podešavanje pokazivača steka

```

### 3 Vektori prekida kod mikrokontrolera ATmega328

Vektor prekida predstavlja adresu u memoriji na kojoj se nalazi prekidna rutina, tj. podprogram za obradu prekida. Kod mikrokontrolera ATmega 328, postoji 26 izvora prekida, a vektori prekida



su, kao što je već ranije rečeno, fiksni i prikazani su u tabeli.

Br.	Adresa	Izvor prekida	Definicija prekida
1	0x0000	RESET	Eksterni pin za reset, reset prilikom gubitka napajanja,...
2	0x0002	INT0	Spoljašnji prekid 0
3	0x0004	INT1	Spoljašnji prekid 1
4	0x0006	PCINT0	Prekid izazvan promenom stanja na nekom od pinova
5	0x0008	PCINT1	Prekid izazvan promenom stanja na nekom od pinova
6	0x000A	PCINT2	Prekid izazvan promenom stanja na nekom od pinova
7	0x000C	WDT	Watchdog tajmer
8	0x000E	TIMER2 COMPA	Tajmer/brojač 2 poređenje sa registrom A
9	0x0010	TIMER2 COMPB	Tajmer/brojač 2 poređenje sa registrom B
10	0x0012	TIMER2 OVF	Tajmer/brojač 2 prekoračenje
11	0x0014	TIMER1 CAPT	Tajmer/brojač 1 hvatanje
12	0x0016	TIMER1 COMPA	Tajmer/brojač 1 poređenje sa registrom A
13	0x0018	TIMER1 COMPB	Tajmer/brojač 1 poređenje sa registrom B
14	0x001A	TIMER1 OVF	Tajmer/brojač 1 prekoračenje
15	0x001C	TIMER0 COMPA	Tajmer/brojač 0 poređenje sa registrom A
16	0x001E	TIMER0 COMPB	Tajmer/brojač 0 poređenje sa registrom B
17	0x0020	TIMER0 OVF	Tajmer/brojač 0 prekoračenje
18	0x0022	SPI, STC	SPI serijski transfer kompletiran
19	0x0024	USART,RX	USART Rx (prijem) kompletiran
20	0x0026	USART, UDRE	USART Tx prazan registar podataka
21	0x0028	USART, TX	USART Tx (slanje) kompletirano
22	0x002A	ADC	Analogno/digitalna konverzija završena
23	0x002C	EE READY	EEPROM spreman
24	0x002E	ANALOG COMP	Analogni komparator
25	0x0030	TWI	2-Wire Serijski interfejs
26	0x0032	SPM READY	Memorija za smeštanje programa spremna

Više detalja o svakom od gore navedenih izvora prekida može se pronaći u katalogu na adresi <http://www.atmel.com/devices/atmega328p.aspx>