

## Obrada prekida

U slučajevima kada periferija zahteva obradu od mikrokontrolera, postoje dva načina koja se koriste:

- 1) *polling* (prozivka)
- 2) putem obrade prekida

Kod metode prozivke mikrokontroler konstantno proziva periferiju da proveri da li je potrebna obrada. Kada dobije informaciju da je potrebna obrada on je aktivira. Uobičajeno je da se ovo radi u petlji pa je zato mikrokontroler konstantno zauzet i nije u mogućnosti da radi nešto drugo. Ovo je glavna mana ovog metoda.

Primer neefikasnosti ove metode je u slučaju telefonskog operatera koji treba da beleži poruke koje dobija telefonom, a pri tome telefon nema nikakvu indikaciju dolaznog poziva već operater mora stalno da podiže slušalicu i proverava da li je neko pozvao.

Kod metode koja koristi sistem prekida, periferija obaveštava mikrokontroler da je potrebna obrada. Dotle i nakon toga mikrokontroler može da radi druge stvar ili da bude u režimu štednje energije. Ovaj metod je daleko efikasniji od metoda prozivke.

Primeri:

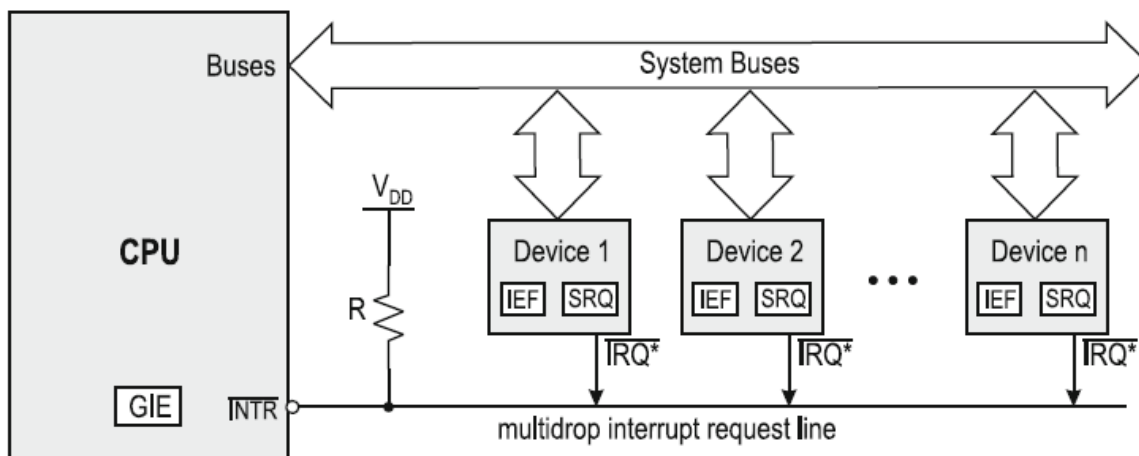
- sistem sa LED diodom i tasterom gde se na svaki pritisak tastera dioda pali/gasi. Paljenje/gašenje se radi u prekidnoj rutini koju aktivira pritisak tastera.
- dolazna poruka koja stiže putem serijskog porta aktivira prekid i u prekidnoj rutini se ona negde smešta ili se preduzima neka akcija.
- monitoring napona baterije gde se prekid generiše kada se dostigne napon koji je ispod nekog definisanog praga.

### Tipovi prekida

Postoje maskirani i nemaskirani prekidi. Maskirane je moguće zabraniti dok su nemaskirani rezervisani za sistemske aktivnosti i nije ih moguće zabraniti.

### Identifikacija izvora prekida

#### 1) Nevektorski sistemi



CPU prima zahtev za prekidom preko jedne linije od svih periferija. Problem nastaje ako više od jedne periferije zahteva obradu prekida. Tada CPU proziva pojedinačno periferije i čita njihove SRQ (*service request*) indikatore. Ovde se prioritet izvršavanja obrada prekida u slučaju više zahteva rešava redosledom prozivke.

## 2) Vektorski prekidi

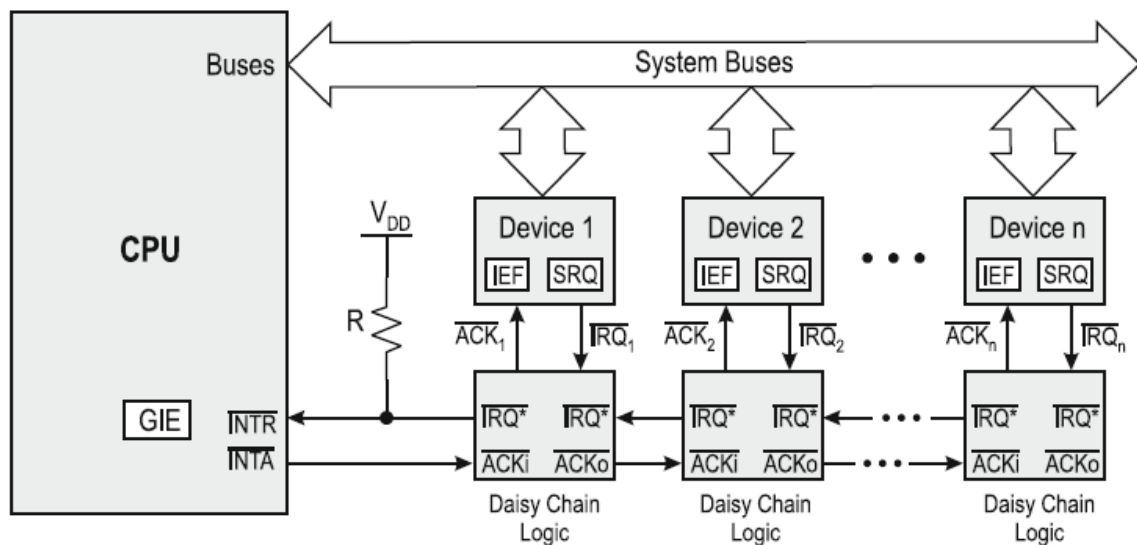
Ovde se dodaje jedan takt za INTA (*Interrupt Acknowledgment*) potvrdu prijema zahteva za prekidom. Svaka periferija ima svoj ID identifikator na osnovu kojeg se računa adresa za početak prekidne rutine. Pod pojmom vektor se misli na adresu prekidne rutine.

## 3) Autovektorski sistemi

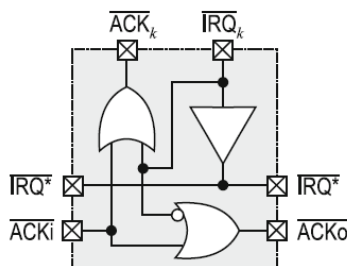
Svaka izvor prekida ili periferija ima fiksnu početnu adresu za obradu prekida i ona se ne može menjati. Nema računanja adrese niti je potreban INTA ciklus. Po pojavi zahteva za obradom prekida CPU automatski učitava početnu adresu za obradu prekida iz tabele vektora prekida. Mikrokontroleri 8051 i MPS430 imaju ovakav način obrade prekida.

### Razrešavanje problema prioriteta pri obradi prekida

#### 1) *Daisy-chain* metod

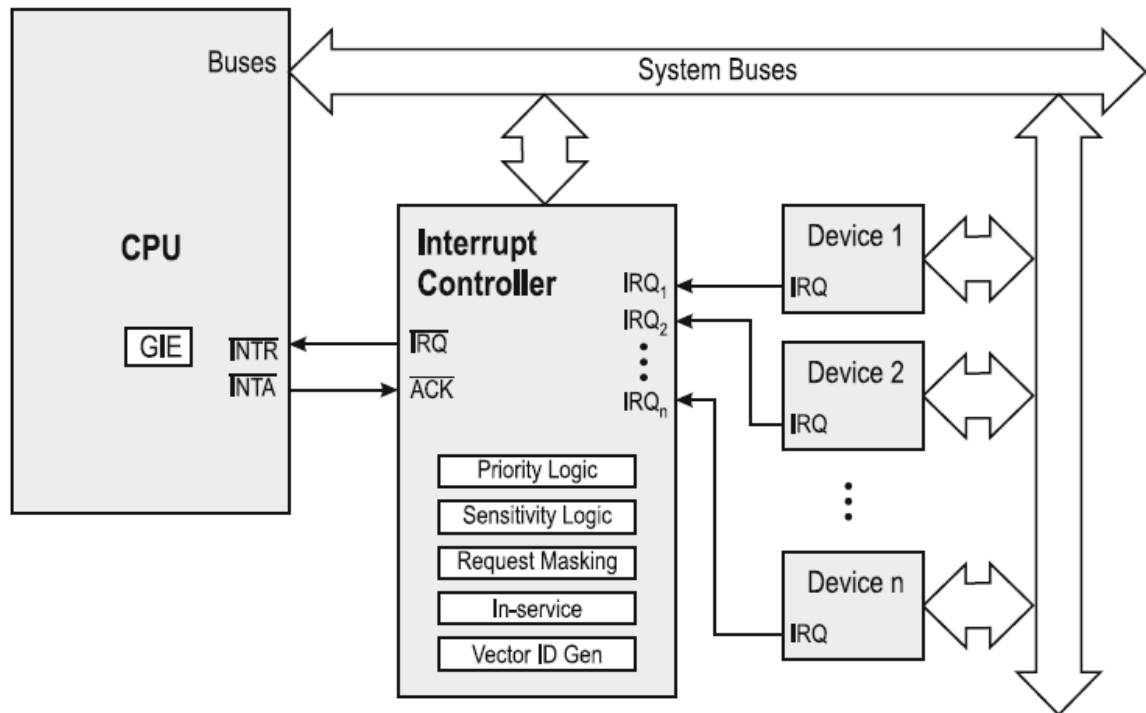


Logička struktura daisy-chain linka je data na narednoj slici:



Ova metoda radi po principu: „što je uređaj bliži CPU ima viši prioritet“ i nije moguće menjati prioritet. Svrha dodatne logike kod ove metode je da blokira *acknowledgment* da ne ide dalje od najvišeg prioriteta onog koje zahtev poslao.

## 2) Upotreba kontrolera prekida



Kontroler prekida vrši arbitražu u vezi prioriteta, ali i druge aktivnosti tipa dozvole/zabrane, podešavanja načina okidanja prekida i slično. Veoma konfigurabilno rešenje ali zahteva dodatnu komponentu u sistemu. Primer kontrolera prekida je Intel 8259A.