Registri mikroprocesora i memorijski modovi

Računarstvo i informatika III_{sm}

doc. dr Miloš Ivanović, mivanovic@kg.ac.rs

Institut za matematiku i informatiku
Prirodno-matematički fakultet, Kragujevac

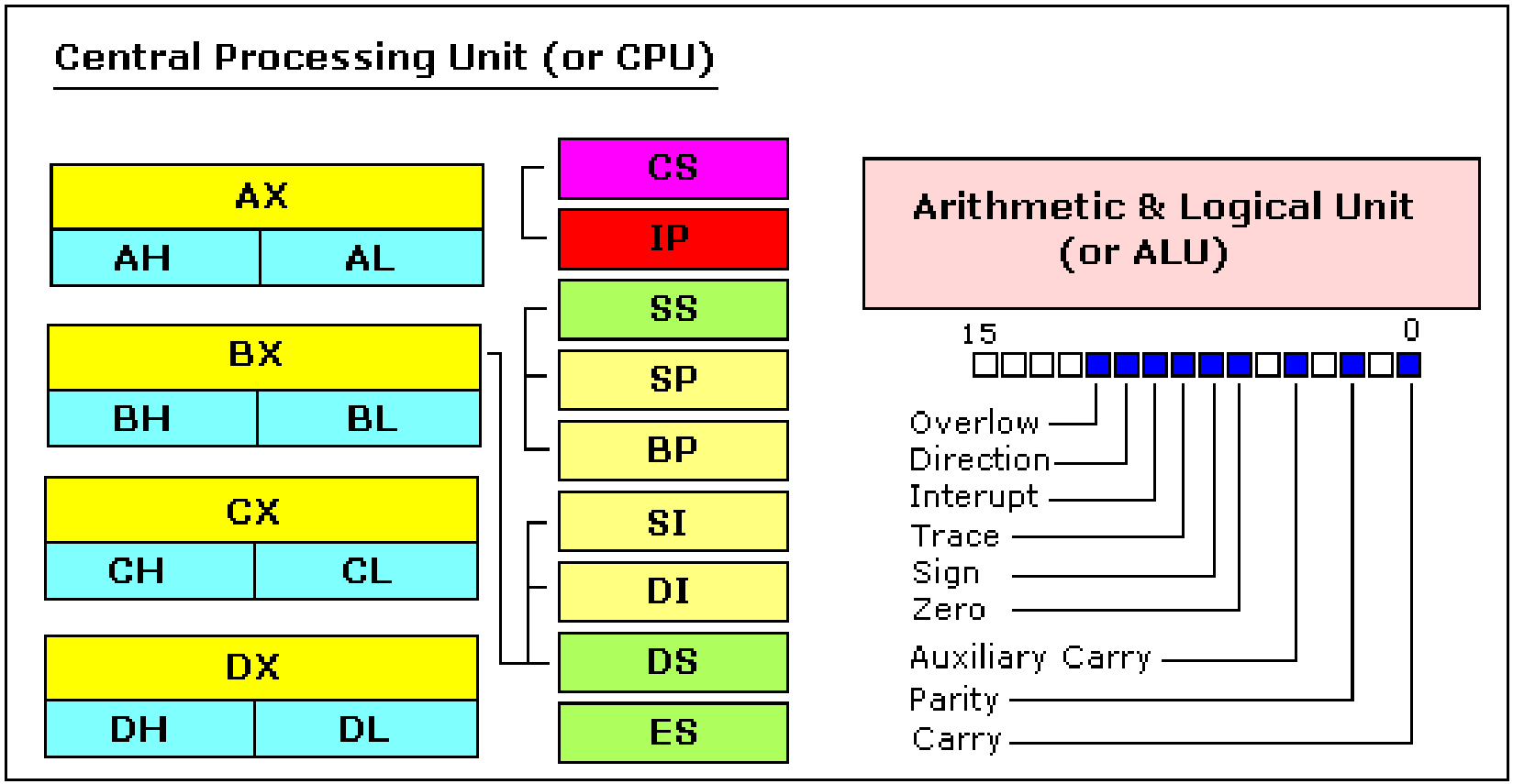
Septembar 2012.

Jedinice memorije

Tabela: Jedinice memorije - terminologija

Naziv	**Vrednost**
word	2 bajta
double word	4 bajta
quad word	8 bajtova
paragraph	16 bajtova

Registri procesora i8086/8088



Registri opšte namene

- AX - akumulator (podeljen na AH/AL)
- BX - registar bazne adrese (podeljen na BH/BL)
- CX - brojački registar (podeljen na CH/CL)
- DX - registar podataka (podeljen na DH/DL)
- SI - *source* indeks registar
- DI - *destination* indeks registar
- BP - *base pointer*
- SP - *stack pointer*

Važno!

Dužina ovih registara je 16 bita. Uprkos nazivima registara, programer je taj koji određuje kako će se registri koristiti!

Registri opšte namene

- 4 registra opšte namene (AX, BX, CX i DX) sastoje se iz 2 odvojena 8-bitna registra, a to su za npr. AX registri AH i AL (H-*High*, L-*Low*)
- Pošto se registri nalaze unutar procesora, oni predstavljaju najbržu memoriju u računaru, ali ih ima svega nekoliko

Primer

Podela registara na viši i niži bajt Ako je npr. AX=**0011000000111001b**, onda je AH=**00110000b**, a AL=**00111001b**. Ukoliko se promeni sadržaj bilo AH, bilo AL, menja se i AX.

Segmentni registri

Formiranje adrese na procesoru i8086

Adresna magistrala procesora i8086 ima 20 bita, što znači da je za smeštanje adrese potrebno dva 16-bitna registra, i to u formi *segment* + *offset*.

- CS (*Code Segment*) - pokazuje na segment u kome se nalazi program koji se izvršava
- DS (*Data Segment*) - pokazuje na segment u kome se nalaze definisane varijable
- ES (*Extra Segment*) - na programeru je da defniše njegovu upotrebu
- ES (*Extra Segment*) - pokazujena segment koji sadrži stek

Segmentni registri

Formiranje efektivne adrese u realnom modu

Primer

Formiranje efektivne adrese iz 2 registra Ako recimo želimo da pristupimo memoriji na fizičkoj adresi **12345h**, onda setujemo **DS=1230h** i **SI=0045h**. CPU množi adresu segmenta sa **10h** i zatim dodaje **45h** da bi dobio **12345h**:

$$\begin{array}{r} 12300h \\ +\ \ 0045h \\ \hline 12345h \end{array}$$

Upozorenje!

Segmentirana adresa nije jedinstvena. Npr. Fizička adresa 04808 se može referencirati sa 047C:0048, 047D:0038, 047E:0028 ili 047B:0058.

Registri specijalne namene

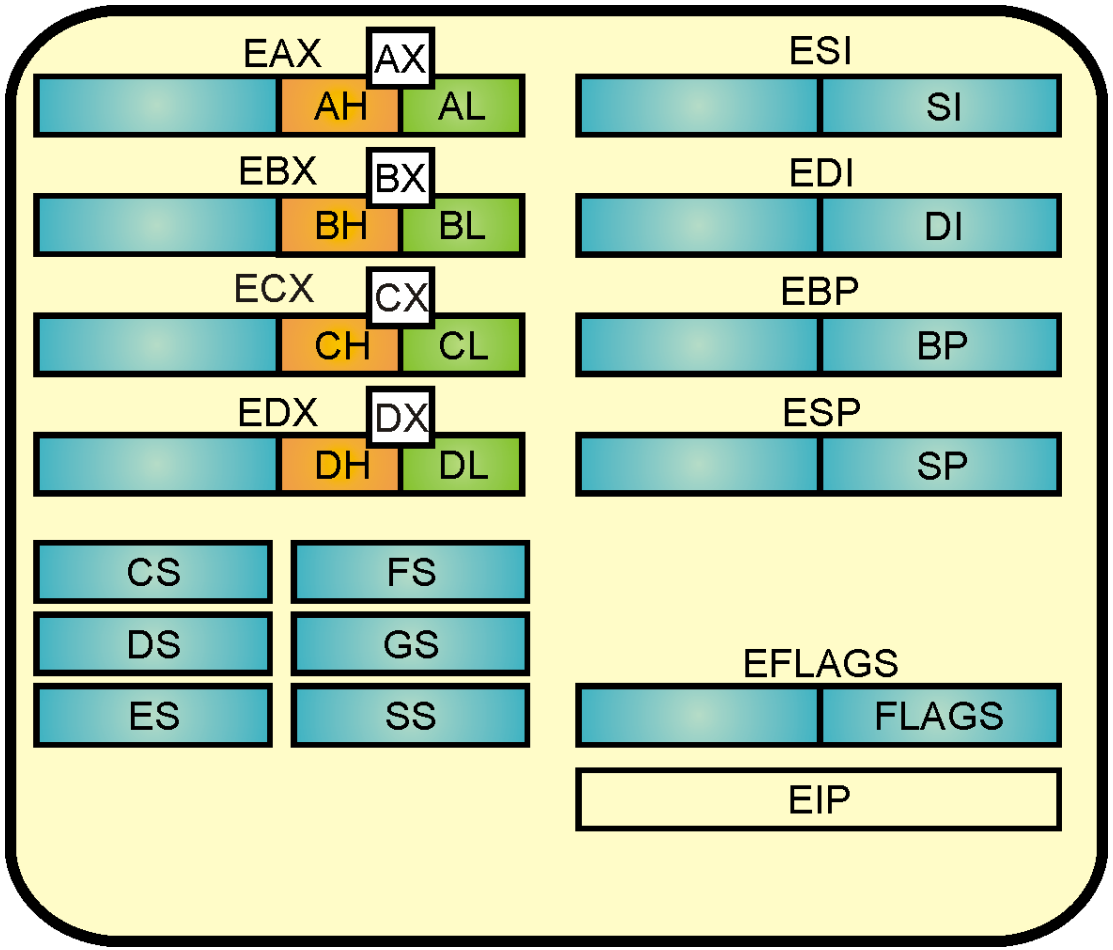
- IP (Instruction Pointer) - Pokazivač na sledeću instrukciju. Radi u sprezi sa **CS** registrom formirajući adresu sledeće instrukcije
- FLAGS (Flags Register) - određuje tekuće stanje procesora. Autmatski se menja nakon svake CPU instrukcije, recimo nakon aritmetičkih operacija

Registri procesora i80386

- i80386 i kasniji procesori imaju prošireni registre, npr. registar AX je proširen na 32 bita
- Novi 32-bitni registar se naziva EAX, dok zbog kompatibilnosti AX i dalje referiše na 16-bitni registar (prvih 16 bita EAX)
- I dalje postoje AL i AH, kao niži i viši bajt AX-a
- Ostali prošireni registri opšte namene su EBX, ECX, EDX, ESI i EDI
- I mnogi drugi registri su prošireni, BP postaje EBP; SP postaje ESP, FLAGS postaje EFLAGS i IP postaje EIP
- Segmentni registri su i dalje 16-bitni i kod procesora 386
- Dodata su dva nova segmentna registra, FS i GS

Registri procesora i80386

Dijagram



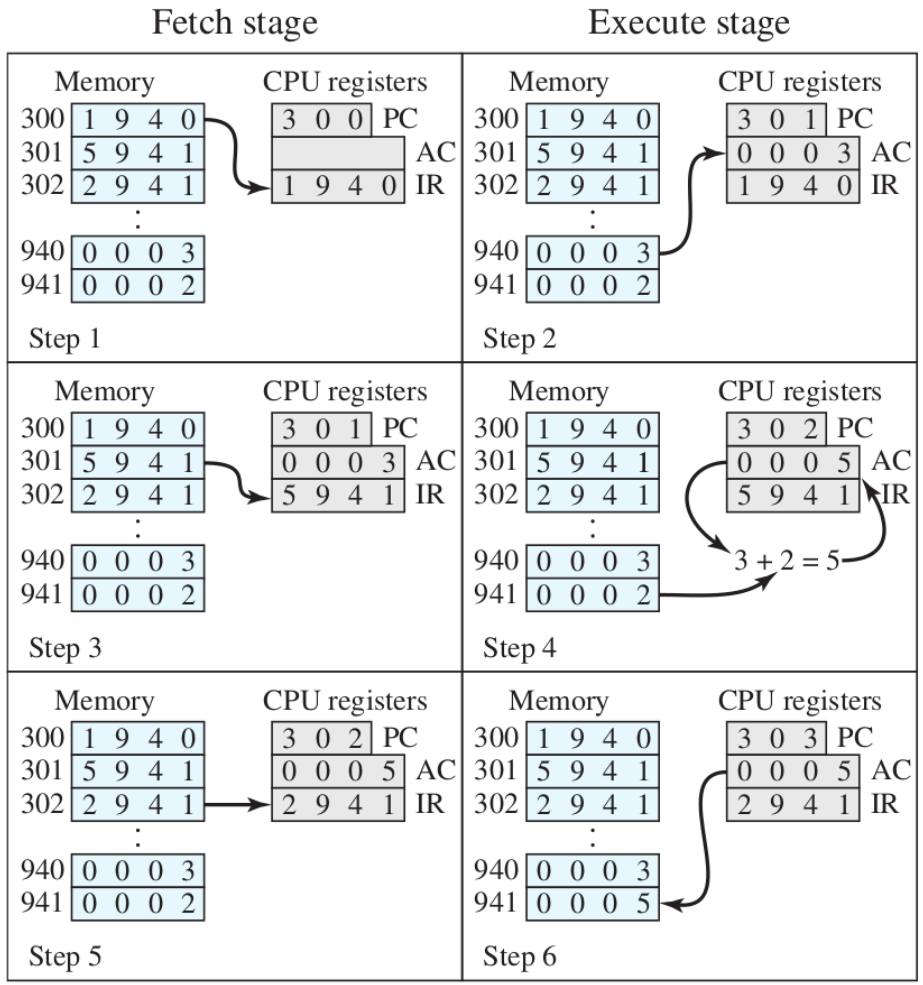
Opšti nazivi registara
Napomena

- IP (*Instruction Pointer*) se ponegde još naziva i PC (*Program Counter*)
- Registar AX (ili njegova prošrena verzija EAX) u opštoj terminologiji se naziva **akumulator** i ima oznaku AC
- Registar u koji se učitava kompletna instrukcija naziva se IR (*Instruction Register*)
- Na većini procesora, **IR nije vidljiv programeru**

Izvršavanje programa na hipotetičkom računaru

Podsetnik

1. Donošenje vrednosti sa adrese 940 u akumulator
2. Sabiranje vrednosti iz akumulatora sa podatkom na adresi 941
3. Prenos rezultata na adresu 941



Prekidi (*Interrupts*)

- Ponekad se normalni tok programa mora prekinuti da bi se obradio neki događaj u sistemu koji zahteva promptnu reakciju
- Sâm harver računara obezbeđuje mehanizam koji rukuje ovim događajima - mehanizam prekida (*interrupts*)
- Primer je kada se glavni program prekida da bi se obradio događaj pomeranja miša (recimo pomeranje kursora)
- Interapti prouzrokuju predaju kontrole rutinama pod nazivom rukovaoci prekida (*interrupt handler*)

Interapt tabela

Svaki *Interrupt handler* ima svoj broj. Na početku fizičke memorije stoji **Interapt tabela** koja sadrži segmentne adrese interapt hendlera. Broj interapta je, u stvari, indeks zapisa u ovoj tabeli.

Prekidi (*Interrupts*)

Interni i eksterni interapti

Eksterni interapti

Eksterni interapti nastaju van procesora. Veliki broj ulazno/izlaznih uređaja podiže ovu vrstu prekida (npr. miš, tastatura, tajmer, disk kontroleri, zvučne kartice itd.

Interni interapti

Interni interapti su prouzrokovani događajem unutar procesora, ili nastankom greške ili interapt instrukcijom. Interapt uzrokovan greškom se naziva i zamka (*trap*). Interapti uzrokovani pozivom interapt instrukcije nazivaju se softverskim interaptima.

Povratak iz prekida

Veliki broj prekidnih rutina vraća kontrolu glavnom programu koji nastavlja kao da se ništa nije dogodilo. Zamke (*traps*) obično prekidaju glavni program.

Namena prekida

Primer

Osnovna namena prekida je da se poboljša iskorišćenost procesora. Preračunati razliku brzine između CPU koji radi na 3 *GHz* i diska koji radi na 7200 *rpm*.

Neki primeri klasa prekida

1. **Program** - prekoračenje, deljenje nulom, segmentacija
2. **Tajmer** - Generiše ga tajmer unutar CPU. OS na taj način može redovno da izvršava neke instrukcije
3. **UI** - Generiše ga UI kontroler
4. **Otkaz hardvera** - npr. greška pariteta memorije

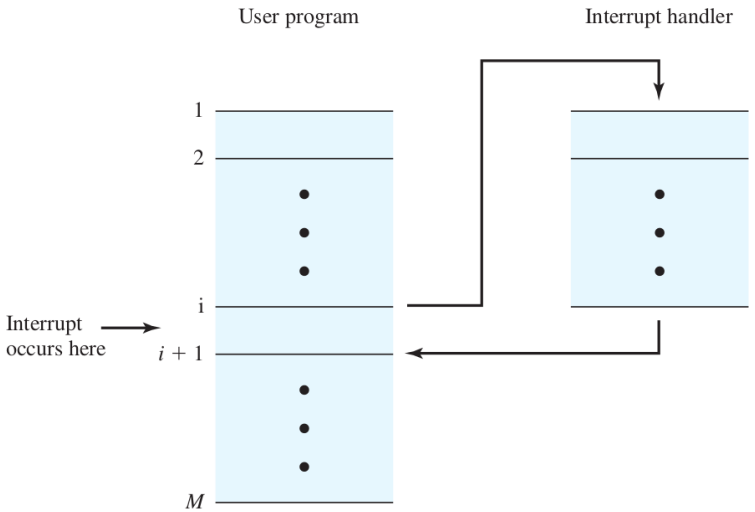
Tabela prekida

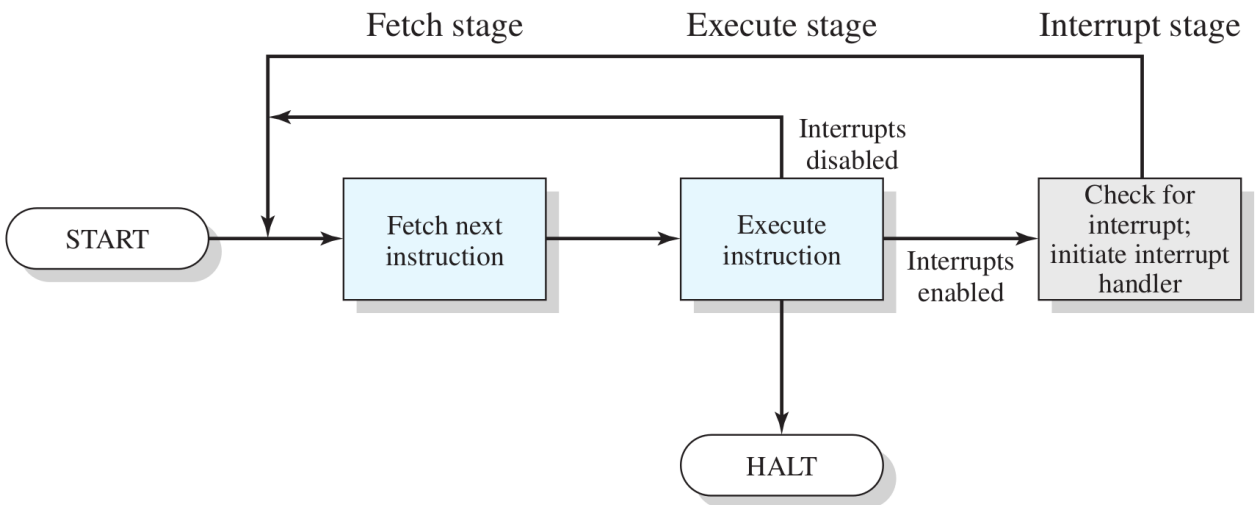
Tabela: Tabela prekida

INT_NUM	Short Description
0x00	Division by zero
0x01	Debugger
0x02	NMI
0x03	Breakpoint
0x04	Overflow
0x05	Bounds
0x06	Invalid Opcode
0x07	Coprocessor not available
0x08	Double fault
0x09	Coprocessor Segment Overrun (386 or earlier only)
0x0A	Invalid Task State Segment
0x0B	Segment not present
0x0C	Stack Fault
0x0D	General protection fault
0x0E	Page fault
0x0F	...

Prekidi (Interrupts)

Obrada prekida i programski ciklus sa prekidom

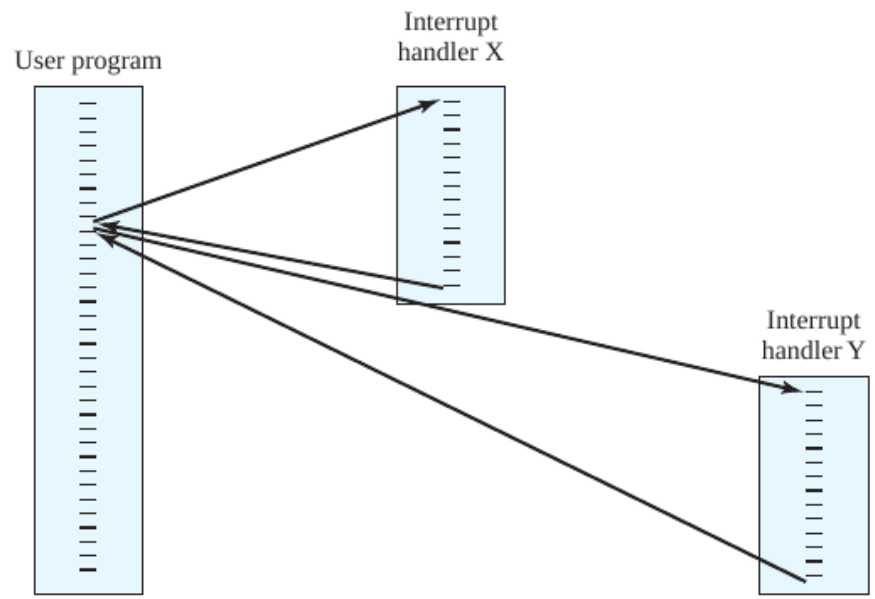




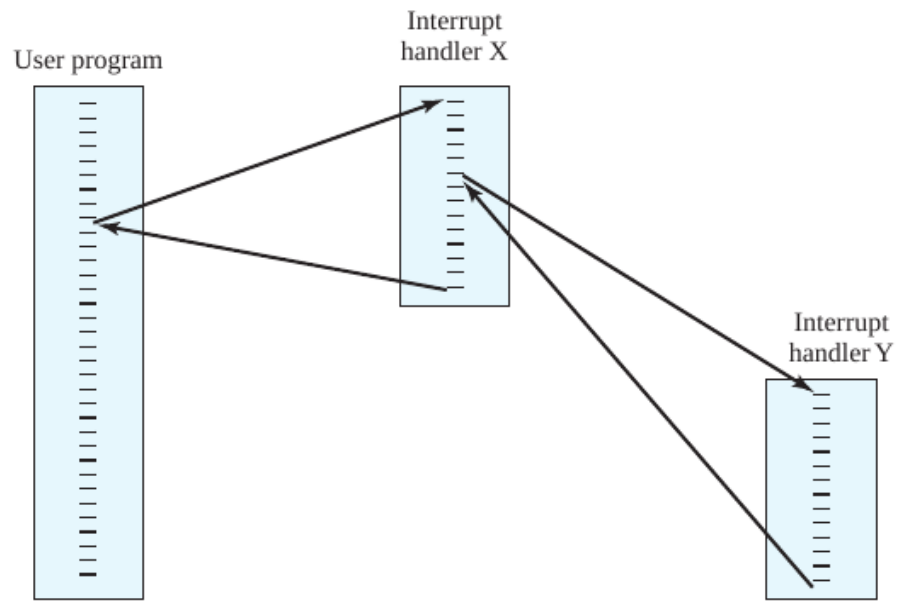
Višestruki prekidi

Vrste višestrukih prekida

Prekid može nastati i tokom izvršavanja neke prekidne rutine. Ovako nastali prekidi se mogu obraditi na **sekvencijalan** ili **ugnežđen** način.



(a) Sequential interrupt processing



(b) Nested interrupt processing

Organizacija prekida po prioritetu

Prioriteti prekida

Prekid višeg prioriteta može da izazove prekid rada prekida sa nižim prioritetom. Oznaka t na slici je vreme.