

Uvod

Pojam arhitekture računara

Program je:

- **algoritam opisan jezikom računara**
- **niz instrukcija koje obavljaju određeni posao**

Računar je mašina koja izvršava zadate instrukcije

Jezik računara (mašine) je **skup** jednostavnih **instrukcija** koje mogu da izvrše elektronska kola računara (procesor)

Dizajn skupa instrukcija i njihove implementacije predstavlja arhitekturu računara

Pojam arhitekture računara

- Posebni problemi **upotrebe (programiranja) i pravljenja (implementacije) računara**
- Način programiranja zavisi od osobina skupa naredbi - **arhitektura naredbi**
- Implementacija arhitekture naredbi obuhvata **organizaciju i realizaciju računara**
- **Arhitektura računara = arhitektura naredbi + implementacija**

Strukturirana organizacija računara



Primer algoritma - NZD

Euklidov algoritam:

$$a = n * \text{NZD}$$

$$b = m * \text{NZD}$$

$$|a - b| = |n - m| * \text{NZD}$$

Programski kod:

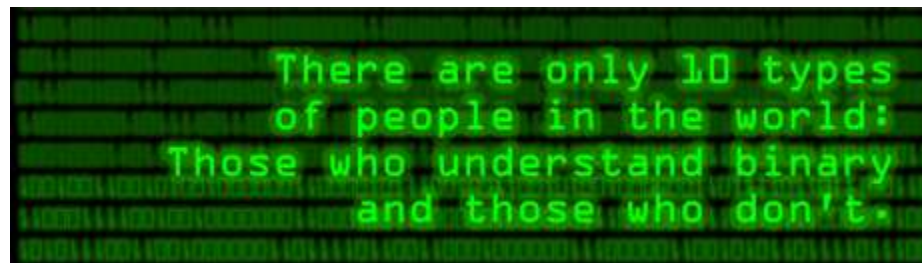
```
int a = 12;  
int b = 10;  
while (a != b) {  
    if (a > b)  
        a = a - b;  
    else  
        b = b - a;  
}
```

Proceduralni programski jezici

- **Prosti tipovi podataka** – celi, realni, znakovni, logički
- **Prosti** (osnovni) **tipovi** – osnova za **složene tipove**
- **Operacije** – opisuju obradu podataka, odnosno rukovanje prostim tipovima
- **Promenljive** – daju opštost obradi podataka, imaju ime, vrednost i tip, operacija dodele
- **Promenljive, vrednosti prostih tipova i operacije** su osnovni elementi imperativnih jezika
- **Uopšteni model računara** – memorija i procesor

Memorija

- Sastavljena od niza **lokacija**
 - **vrednost** – prosti tip
 - **adresa** – numerička oznaka, broj ili labela – simbolička oznaka
- **Univerzalnost**, višeznačna interpretacija
 - kodiranje naredbi i drugih tipova na opseg celobrojnih brojeva



Procesor

- **Logička kola** ili **sklopovi** – izvršavaju operacije, realizuju funkcije
- **Registri** – posebna memorija unutar procesora, označeni su posebnom oznakom
- **Naredba = oznaka operacije + operandi**
- Operandi su ili vrednosti prostih tipova ili adrese lokacija

Adresiranje

- **Omogućava pristup operandima**
- **Ulazni i izlazni operandi** ($c = a + b$, $a = a + b$)
 - **neposredni** (engl. *immediate*) – konstante
 - **direktni** (engl. *direct*) – promenljive prostih tipova
 - **registarski** (engl. *register*) – promenljive prostih tipova
 - **indirektni** (engl. *register indirect*) – pokazivačke prom.
 - **indeksni** (engl. *indexed*) – promenljive složenih tipova

Skup naredbi

- **Aritmetičke, relacione, logičke, upravljačke i naredbe prebacivanja** (između registara i memorijskih lokacija)
- **Aritmetičke:** dovoljno celobrojno + i –
- **Relacione:** $a \text{ ? } b$ može pomoću $a - b$
 - $==$ razlika 0
 - $!=$ razlika različita od 0
 - $<$ razlika negativna
 - $<=$ razlika negativna ili 0
 - $>$ razlika pozitivna
 - $>=$ razlika pozitivna ili 0
- **Logičke:** I, ILI, NE, ekskluzivno ILI (EILI, XOR)
- **Upravljačke:** безусловna i uslovna izmena redosleda izvršavanja – realizacija alternativnog (selekcija) i repetativnog toka (iteracija) izvršavanja naredbi

Mašinske i assemblyerske naredbe

- **Mašinska naredba** = kod naredbe + kodovi operanada
- Mašinski format naredbe
- **Assemblyerska naredba** – simbolička oznaka mašinske naredbe
- **Mašinski i assemblyerski jezik**
- **Assembler**: assemblyerski program u mašinski
- **Prevodilac** (kompajler): izvorni program u assemblyerski (mašinski)

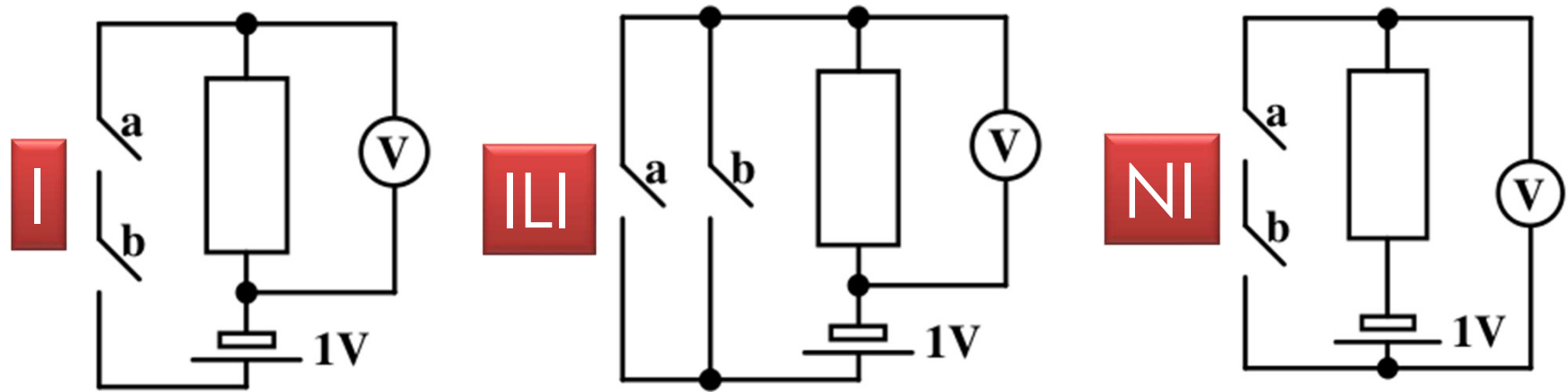
Fizička osnova modela računara

- **Dovoljno je samo predstavljanje celih brojeva**
- Binarni brojni sistem – dva nivoa signala
- Podudarnost cifara binarnog brojnog sistema i logičkih vrednosti omogućava **opisivanje aritmetičkih operacija logičkim funkcijama**

| a | b | a+b | Prenos |
|---|---|-----|--------|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

 $a \oplus b$
 $a \wedge b$

Fizička osnova modela računara



NI (NAND) : $a \circ b \equiv \sim(a \& b)$

Univerzalno logičko kolo:

I: $a \& b == (a \circ b) \circ (a \circ b)$

ILI: $a \mid b == (a \circ a) \circ (b \circ b)$

EILI (XOR): $a \wedge b == ((a \circ a) \circ b) \circ (a \circ (b \circ b))$

ENILI (IFF): $a \equiv b == (a \circ b) \circ ((a \circ a) \circ (b \circ b))$

Fizička osnova modela računara

- **Tranzistor = fizička realizacija prekidača**

- ulaz
- izlaz
- upravljački izvod

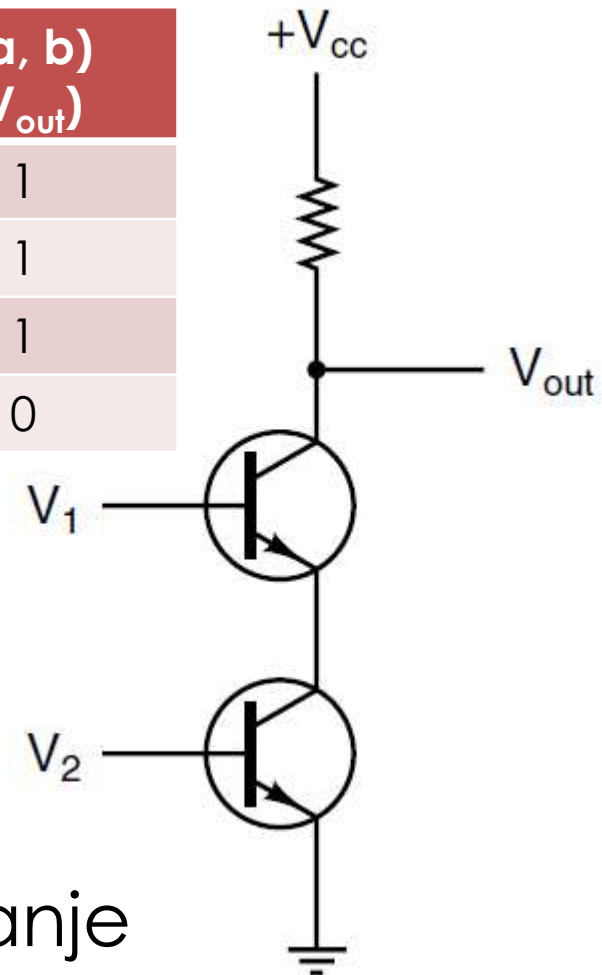
| $a (V_1)$ | $b (V_2)$ | $f(a, b) (V_{out})$ |
|-----------|-----------|---------------------|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

- **Upravljačka logička funkcija**
(prekidačka funkcija)

$$f(a) == a$$

od vrednosti argumenata zavisi stanje

$$f(a, b) == a \circ b$$

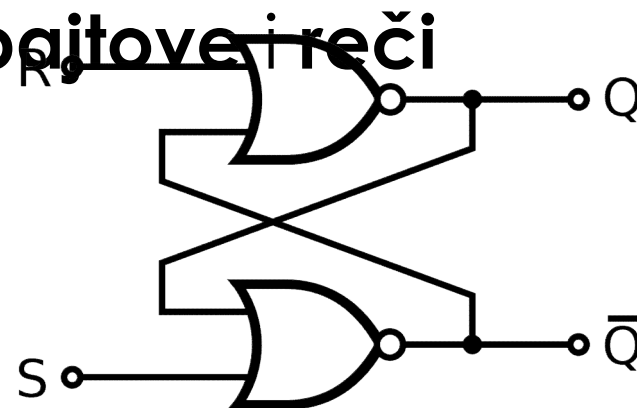


Fizička osnova modela računara

- **Kombinaciona i sekvencijalna kola**
- **Funkcije** od **kombinacionih**, **izlaz** zavisi od **ulaza**
- **Memorija** od **sekvencijalnih**, **izlaz** zavisi od **ulaza** i **stanja**
- **Lokacija = niz bitova** (ćelija)
 - bit = flip-flop kolo, memoriše jednu logičku vrednost

- Lokacija i reči u bajtovima

| S | R | Q_{next} | Akcija |
|---|---|------------|-----------------|
| 0 | 0 | Q | Drži stanje |
| 0 | 1 | 0 | Reset |
| 1 | 0 | 1 | Set |
| 1 | 1 | X | Nije dozvoljeno |



Osnovna logička kola

