

Teorijski uvod u računarstvo

Aleksa Tešić, seminar računarstva

aleksatesicteske@gmail.com

IS Petnica

14.4.2018.

Sadržaj predavanja

- Brojčani sistemi
- Bulova algebra
- Tjuring
- Logička kola
- ALU / CU
- Fon Nojman / Harvard
- Asembler
- Viši jezici
- BIOS
- OS

Brojčani sistemi

...

Gotfrid Vilhelm Lajbnic (1646 – 1716)

- „Izmislilo“ binarni sistem
- Pionir u razvoju mehaničkih kalkulatora
- „Izmislilo“ integralni račun



Kako oni rade?

...

Pozicioni sistemi

- Vrednost cifre je određena njenom pozicijom u broju
- Uvek imamo osnovu sistema (10, 2, 8, 16...)
- Svaki broj se može predstaviti kao zbir osnove koja je podignuta na različite stepene
- $156 = 1 * 10^2 + 5 * 10^1 + 6 * 10^0$
- Osnova sistema će se uvek pisati kao 10

Bitni pozicioni sistemi

- Binarni
- Troični / Trinarni
- Heksadekadni
- Oktalni

Zašto su nam bitni?

...

- **Binarni** – Bulova algebra, jednostavan za implementaciju
- **Troični / Trinarni** – najtačniji teorijski
- **Heksadekadni / Oktalni** – lake transformacije iz binarnog

Nepozicioni sistemi

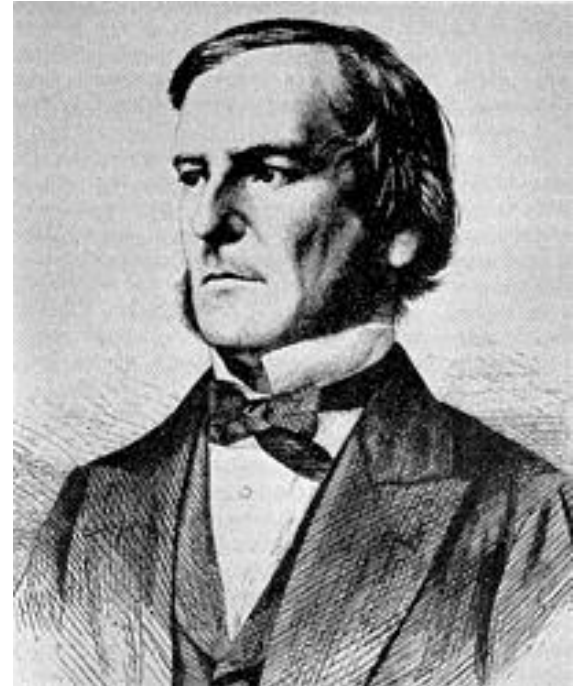
- Vrednost cifre je uvek ista i ne zavisi od pozicije
- Rimski brojevi
- Nepraktični za korišćenje

Bulova algebra

...

Džordž Bul (1815 – 1864)

- Bulova algebra
- Pionir informatike i matematičke logike



Bulova algebra

- Deo univerzalne algebre
- Vrednosti koje postoje su **tačno** i **netačno**
- Osnovne operacije su **i**, **ili**, **negacija**

Zašto nam je bitna?
...

Bulova algebra

- Tačno / Netačno = 1 / 0 (Binarni sistem)
- Sve možemo predstaviti kao logički izraz
- Omogućeno pravljenje logičkih kola
- Laka minimizacija logičkih kola

Šta je algoritam?

...

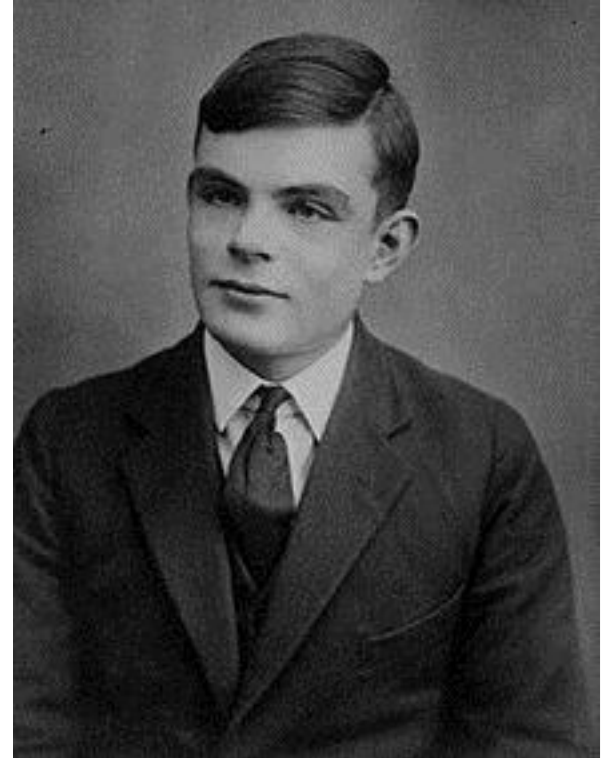
Formalizacija algoritama

- Tjuringova mašina
- λ račun
- URM
- ...
-

Tjuring
...

Alan Tjuring (1912 – 1954)

- Tjuringova mašina
- Teorijski temelji računarstva



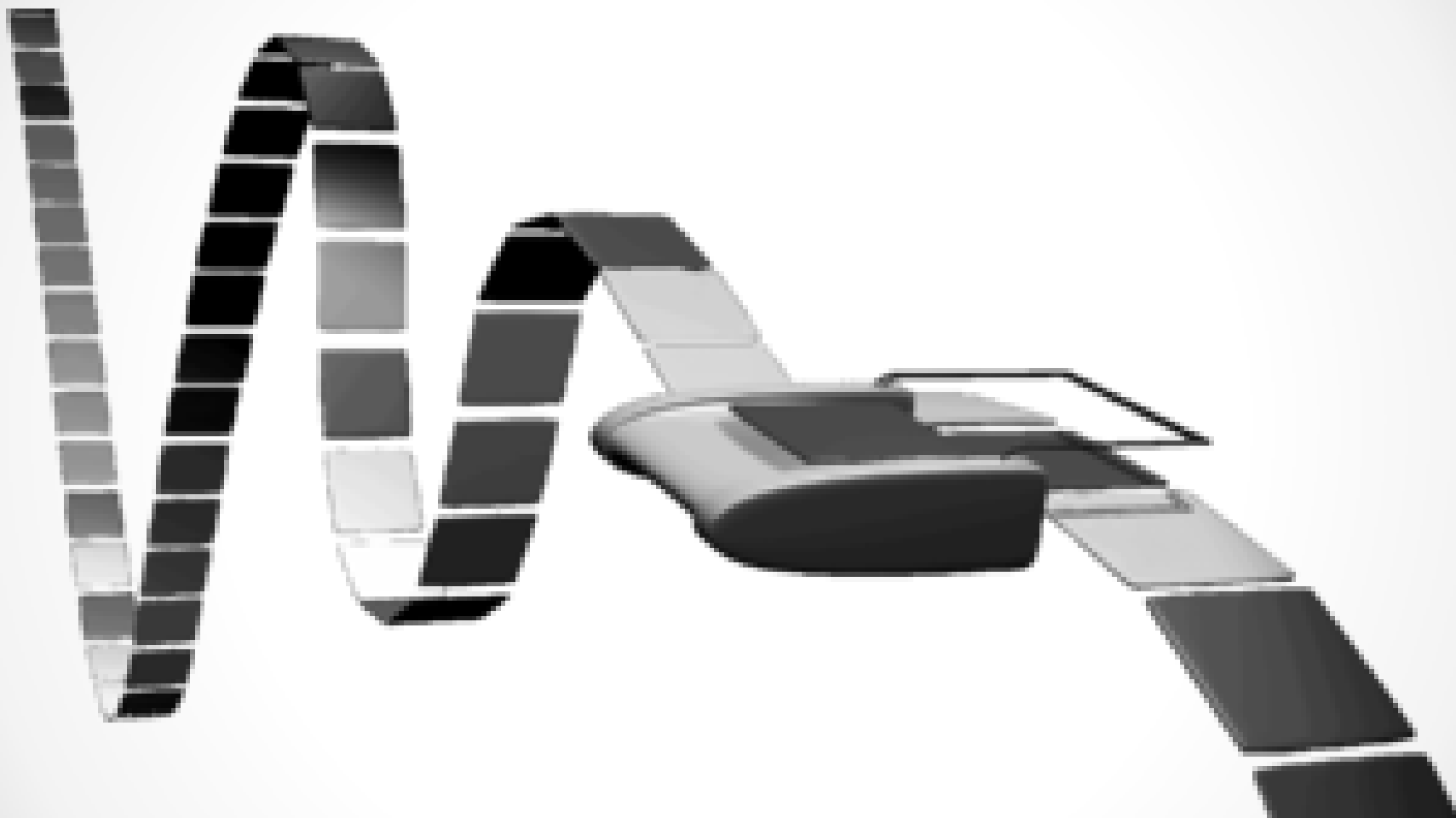
- Turing's Proof
- Tjuringova mašina
- Turing Complete

Turing's Proof

- Pokazano da **ne postoji opšti** algoritam za dokazivanje nekih problema
- Ne možete očekivati samo 1 / 0 kao odgovor u svakom slučaju
- Postoje problemi koji su neodlučivi
- Primer: Halting problem

Tjuringova mašina

- Konačno mnogo stanja
- Levo, Desno, Pročitaj, Upiši
- Beskonačno memorije (traka)
- Podaci i instrukcije su zajedno



Turing complete

- Svaki programski jezik koji može da simulira TM
- Jedino ograničenje je memorija
- Brainfuck, Minecraft, Power Point(možda linkovi)

Logická kola

...

Logička kola

- AND / OR/ NOT – osnovna kola
- NAND / NOR / XOR...
- Pomoću njih se prave **sva** složenija kola
- Logisim

Minimizacija kola

...

Minimizacija kola

- Uklanjam nepotrebne elemente
- Manje elemenata = Manje prostora
- Manje elemenata = Manja cena
- Primer: $AB + AB' = A(B + B') = A$

Kombinatorne mreže

...

Kombinatorne mreže

- Vrše selekciju signala na neki način
- Omogućuju dalje usložnjavanje logičkih kola
- Primer: Multiplexer, binarni sabirač

Binarni sabirač

A	B	C	P
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Sekvencijalne mreže

- Mogu da čuvaju vrednost
- Koriste se pri izradi memorija
- Moraju biti sinhronizovane – koriste clock
- Primer: D flip-flop, J-K flop-flop, **Brojači**

Brojači

- Služe kao normalni brojači (FOR petlja)
- Mogu se koristiti za prelazak iz stanja u stanje

Fon Nojmanov model

...

Džon fon Nojman (1903 -1957)

- Fon Nojmanov ili Prinstonski model



Fon Nojmanov model

- Dizajn za računar **opšte namene**
- Zajednička memorija za instrukcije i podatke
- Paralela sa ljudskim nervnim sistemom

Šta je računar opšte
namene?

...

Fon Nojmanov model

- Potreban nam je nekakav deo koji računa - CA
- Potreban nam je deo koji upravlja i kontroliše sve ostale delove - CC
- Potreban nam je deo koji će pamtit i rezultate, i u kom će stajati podaci – M

Fon Nojmanov model

- CA , CC i M bi bili centralni nervni sistem
- Organi koji vrše I / O operacije i komuniciraju sa njima bi bili periferni nervni sistem

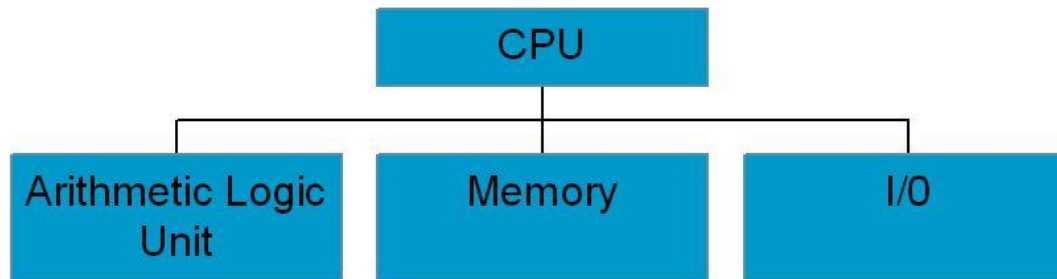
Zašto je dobar?

...

Fon Nojmanov model

- Omogućava nam da programiramo
- Čita instrukcije kao podatke => možemo konstruisati linkere, asemblere, kompajler...
- Bottleneck: Magistrala (rešeno keš memorijom)

The Von-Neumann Architecture



Harvardski model

...

Harvardski model

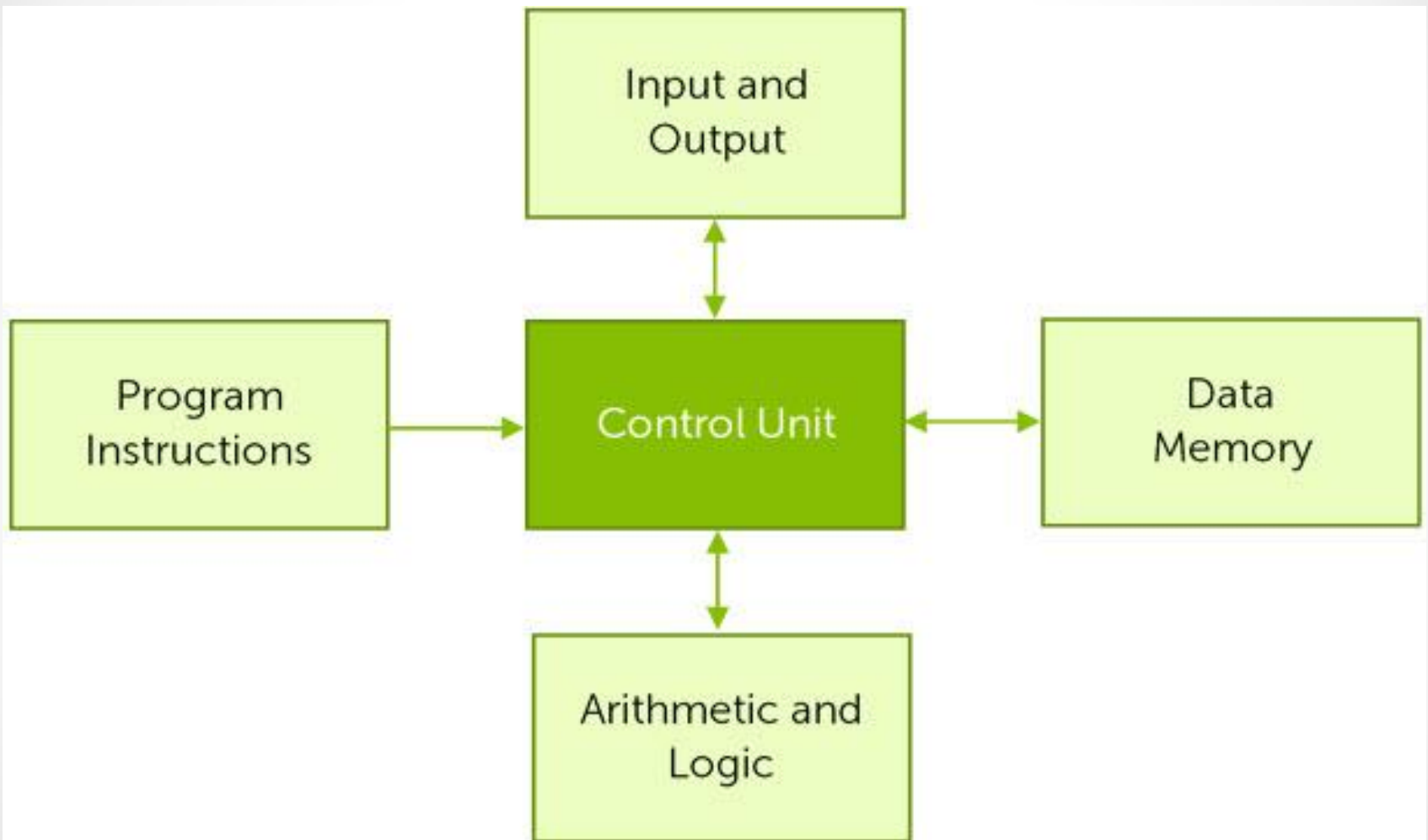
- Odvojena memorija za instrukcije i podatke
- Danas preovlađuje modifikovana harvardaska arhitektura
- Koristi se u mikrokontrolerima

Zašto je dobar?

...

Harvadski model

- **Ispis podatka i pristup instrukciji su mogući u isto vreme**
- Operacije iz keš memorije (modifikovan H. model)
- Koristi se u mikrokontrolerima



ALU i CU

...

ALU

- **A**rithmetic **L**ogic **U**nit
- Obavlja (ko bi rekao) aritmetičko-logičke operacije na zahtev CU-a
- Povezana samo sa CU
- +, -, =, <, > ...
- Može se (a i ne mora) implementirati samo kombinatornim kolima

CU

- **C**ontrol **U**nity
- Upravlja svim procesima u računaru
- Povezana sa svime
- Sinhronizuje procese

(ne baš) Savremeni računar

...

(ne baš) Savremeni računar

- Imamo CPU
- Imamo memoriju
- Znamo kako da ih povežemo
- Kako programiramo?

Assembler

...

Šta je assembler, i čemu on
služi?

...

Šta je assembler i čemu on služi?

- Šta je mašinski kod?
- Assembler pokreće asemblerški jezik
- Asemblerški jezik? – Niz oznaka koje (skoro) 1 na 1 kodiraju instrukcije procesora
- Zamena za mašinski kod

```
// I = 15;  
MOV R3, #15  
STR R3, [R11, #-8]  
  
// J = 25;  
MOV R3, #25  
STR R3, [R11, #-12]  
  
// I = J;  
LDR R2, [R11, #-8]  
LDR R3, [R11, #-12]  
ADD R3, R2, R3  
STR R3, [R11, #-8]
```

ASSEMBLY LANGUAGE

ASSEMBLER

```
1100 1010 1011 0011  
1100 1010 1011 0011  
1100 1010 1011 0011  
1100 1010 1011 0011  
1100 1010 1011 0011  
1100 1010 1011 0011
```

MACHINE CODE

Gde se danas koristi?

- Kod mikrokontrolera
- Programi koji moraju da budu veoma brzi
- Kompajleri
- Drajveri
- Specijalni procesori (nema kompajlera)

Viši programski jezici

...

Šta je problem sa
asemblerom?

...

Šta je problem sa assemblerom?

- Premalo instrukcija
- Gomila stvari mora da se radi „peške“
- **Nizak nivo apstrakcije**

Viši programski jezici

- „Nadograđen“ assembler
- Programer ne mora da zna šta se nalazi „ispod haube“
- Potrebno manje koraka da bi se napisao program
- Prevode se u asemblerški kod

Nizak nivo apstrakcije

- Programer mora voditi računa o memoriji
- Programer mora voditi računa o tipovima
- Dosta brzi, ali „teži“ za upotrebu
- C, Pascal, Cobol, Fortran...

Visok nivo apstrakcije

- Programer nema dodira sa registrima, memorijom, adresama...
- Koriste se apstraktni tipovi
- Manje koda, ali sporiji
- Python, MATLAB, C#, ...

A hardver?

...

BIOS

...

BIOS

- **B**asic **I**nput **O**utput **S**ystem
- Proverava ispravnost hardvera pri pokretanju
- Učitava OS sa diska
- Nalazi se u ROM-u

OS

...

OS

- **O**perating **S**ystem
- Upravlja resursima računara (memorija, procesor, registri...)
- Omogućava nam da koristimo hardver računara
- Windows, UNIX arhitecture

Pitanja?

...

Sadržaj predavanja

- Brojčani sistemi
- Bulova algebra
- Tjuring
- Logička kola
- ALU / CU
- Fon Nojman / Harvard
- Asembler
- Viši jezici
- BIOS
- OS

Teorijski uvod u računarstvo

Aleksa Tešić, seminar računarstva

aleksatesicteske@gmail.com

IS Petnica

14.4.2018.