

# Uvod u mikroprocesore

Milan Banković

3. 10. 2007.

# Sadržaj I

- 1 Osnovne informacije o kursu
  - Tema kursa
  - Predavanja vs. vežbe
- 2 Organizacija mikroprocesora
  - Logička kola
  - Struktura mikroprocesora
  - Ostale komponente mikroračunarskog sistema
  - VLSI dizajn
- 3 Arhitektura mikroprocesora
  - Osnovne karakteristike
  - Mašinski jezik
  - FonNojmanovi računari
  - Struktura mašinske instrukcije

## Sadržaj II

- Način adresiranja operanada

### 4 Asemblerski jezici

- Pojam i svojstva asemblerskih jezika
- C kompilator vs. assembler
- Povezivanje

### 5 GNU Alati

- GNU C kompilator
- GNU assembler
- GNU linker

### 6 Literatura

# Tema kursa

- Mikroprocesorski sistemi
- Dizajn i organizacija mikroprocesora
- Arhitektura mikroprocesora
- Asemblersko programiranje

## Predavanja vs. vežbe

- Na predavanjima se proučava dizajn i princip rada logičkih kola koja učestvuju u izgradnji mikroprocesora.
- Takođe na predavanjima se izučavaju osnove VLSI dizajna.
- Na vežbama se proučava arhitektura IA-32, kao i programiranje na asemblerskom jeziku ove arhitekture.

# Logička kola

- Osnovna logička kola (AND, OR, NAND, NOR, NOT)
- Kombinatorna kola – logičke funkcije
- Sekvencijalna kola – memorije

# Struktura mikroprocesora

- Aritmetičko logička jedinica
- Jedinica za rad sa pokretnim zarezom
- Kontrolna jedinica
- Registri
- Unutrašnje magistrale

## Ostale komponente mikroračunarskog sistema

- Unutrašnja memorija
- Memorija sa stalnim sadržajem
- Keš memorija
- Spoljašnje magistrale
- Ulazno-izlazni uređaji



# VLSI dizajn

- VLSI dizajn – onako kako stvari zaista stoje
- Konkretni problemi implementacije kola
- Mikroelektronika

## Osnovne karakteristike

- Skup registara i njihova upotreba
- Skup instrukcija
- Širina registara i magistrala
- Adresiranje operanada instrukcija

# Mašinski jezik

- Mašinski jezik je jedini jezik koji računar razume.
- Instrukcija mašinskog jezika je niz bitova koji se dovodi na ulaze mikroprocesora, u cilju da procesor izvrši neku operaciju
- Mašinski jezik je samim tim određen arhitekturom mikroprocesora

# FonNojmanovi računari

- Program je niz instrukcija koje se nalaze u memoriji, i koje se izvršavaju jedna za drugom, osim u slučaju instrukcije skoka.
- Podaci se čuvaju u istoj memoriji u kojoj i program.
- Svaka instrukcija se izvršava u dve faze: faza dohvatanja i faza izvršenja.

## Struktura mašinske instrukcije

- Mašinska instrukcija se sastoji iz operacionog koda i nula ili više operanada.
- Operacioni kod definiše šta treba uraditi, kao i kako su dati operandi.
- Operandi mogu biti vrednosti registara, podaci iz memorije ili konstante.
- U zavisnosti od maksimalnog broja memorijskih operanada, procesori se dele na jednoadresne, dvoadresne, i troadresne računare.
- Postoje i nuloadresni procesori.

# Način adresiranja operanada

- Neposredno adresiranje
- Direktno adresiranje
- Indirektno adresiranje
- Indeksno adresiranje

## Pojam i svojstva asemblerskih jezika

- Asemblerski (simbolički) jezik je simbolički ekvivalent mašinskom jeziku procesora.
- Umesto operacionog koda navodi se simbolička oznaka instrukcije.
- Umesto adresa operanada navode se simboli (labele)
- Registri se predstavljaju svojim imenima
- Konstante se zapisuju na uobičajen način
- Relativno vs. apsolutno adresiranje
- Assembler je program koji prevodi asemblerski program u mašinski.

## C kompilator vs. assembler

- Prevođenje viših programskih jezika je znatno komplikovanije.
- C kompilator najpre proizvodi asemblerski kod, koji se zatim assemblerom prevodi na mašinski jezik.
- Fajl dobijen asembliranjem naziva se objektni fajl.
- Prevođenjem C funkcije proizvodi se istoimeni simbol koji označava adresu početka prevedenog koda.
- Gornje svojstvo se koristi za povezivanje C koda sa asemblerskim.



## Povezivanje

- Objektni fajlovi u sebi sadrže prevedeni kod pojedinih funkcija. Simboli koji odgovaraju ovim funkcijama predstavljaju globalne simbole datog objektnog fajla.
- Unutar jednog objektnog fajla se mogu referencirati simboli iz drugih objektnih fajlova. Zato je neophodno da takav simbol postoji u nekom od ostalih objektnih fajlova, i da je jedinstven.
- Spajanje svih objektnih fajlova u jedan rezultujući fajl, pri čemu se povezuju sve reference koje nisu interno razrešene sa simbolima u drugim objektnim fajlovima naziva se povezivanje (linkovanje).
- Prilikom linkovanja se vrši povezivanje i sa bibliotečkim funkcijama
- Povezivanje može biti statičko, deljeno i dinamičko.
- Rezultat povezivanja je izvršni fajl.

## GNU C kompilator – gcc

- Prevodilac za programski jezik C.
- Pored C-a podržava i druge jezike (C++, Ada, F77).
- Prevođenje se izvodi u nekoliko faza: pretprocesiranje, kompajliranje, asembliranje i linkovanje.
- U opštem slučaju obavlja sve ove faze, ali se uz pomoć opcija komandne linije može prekinuti nakon svake od ovih faza.
- Za treću fazu koristi GNU assembler – as.
- Za četvrtu fazu koristi GNU linker – ld. Podrazumevano linkuje standardnu biblioteku jezika C.

## GNU asembler – as

- Prevodi kod napisan u asemblerskom jeziku na mašinski jezik.
- Razume dve vrste sintakse: AT&T sintaksu, i Intel-ovu sintaksu.
- Ulazni fajl je tekstualni fajl sa ekstenzijom .s. Izlazni fajl je binarni fajl, sa ekstenzijom .o (i istim imenom kao i polazni)
- Podržava mnogo različitih arhitektura.

## GNU linker – ld

- Povezuje date objektna fajlove kao i navedene biblioteke u jedinstven izvršni fajl.
- Podrazumevano vrši dinamičko povezivanje.
- Za povezivanje se može koristiti i gcc, što je pogodno zbog automatskog povezivanja sa standardnom bibliotekom.

# Literatura

- Mikroprocesorski sistemi – Aleksandar Samardžić
- GNU programerski alati – Aleksandar Samardžić
- IA-32 Intel Architecture Software Developer's Manual (Vol 1,2,3)
- Vežbe  
([http://www.matf.bg.ac.yu/~milan/?content=mikro\\_papers](http://www.matf.bg.ac.yu/~milan/?content=mikro_papers))