

Sažetak predavanja: Arhitektura računala 1 (Ugradbeni sustavi)

1. Taksonomija računalne arhitekture

- **Von Neumann vs. Harvard arhitektura**
 - Von Neumann: dijele istu memoriju za instrukcije i podatke.
 - Harvard: koristi odvojene memorijske sabirnice (bolje za DSP i ugradbene sustave).
 - Harvard arhitektura često se koristi za **obradu signala (DSP)**, gdje su važni **brzina i paralelna obrada podataka**.
 - **RISC vs. CISC arhitektura**
 - **CISC** (Complex Instruction Set Computing)
 - Starija arhitektura s kompleksnim instrukcijama (više mikroinstrukcija u jednoj).
 - Manje instrukcija u memoriji, ali sporije izvođenje.
 - Prikladna za aplikacije gdje je optimizacija memorije važna.
 - **RISC** (Reduced Instruction Set Computing)
 - Jednostavnije i brže instrukcije.
 - Load-store arhitektura – više memorijskog prostora, ali brža obrada.
 - Omogućuje **pipeline** obradu (više instrukcija u isto vrijeme).
 - **Instrukcijski skup i procesorska svojstva**
 - Definira kako softver komunicira s hardverom.
 - Ključne razlike:
 - Fiksna vs. varijabilna dužina instrukcija
 - Način adresiranja instrukcija
 - Broj operanada
 - Brzina takta, cache memorija, sabirnica i sustav prekida utječu na izbor arhitekture.
-

2. Mikroprocesor i njegove komponente

- **Mikroprocesor** = "mozak" računala, upravlja izvršavanjem instrukcija.
 - **Glavni dijelovi:**
 1. **Registri**
 - **Programsko brojilo (PC)** – pokazuje na sljedeću instrukciju.
 - **Kazaljka stoga (SP)** – upravlja memorijom poziva funkcija.
 - **Registar stanja (SR)** – sadrži zastavice (carry, zero, overflow).
 - **Registri opće namjene (R0 - RN)** – privremeno pohranjuju podatke.
 2. **Aritmetičko-logička jedinica (ALU)** – izvršava matematičke i logičke operacije.
 3. **Upravljačka jedinica** – dekodira instrukcije i generira upravljačke signale.
 4. **Međuspremnici:**
 - **Adresni međuspremnik** – služi za adresiranje memorije.
 - **Podatkovni međuspremnik** – posreduje u prijenosu podataka.
 - **Instrukcijski međuspremnik** – privremeno pohranjuje dekodirane instrukcije.
-

3. Izvršavanje instrukcija (Simulacija rada mikroprocesora)

- **Primjer: Izvršavanje instrukcije INC \$05FF**
 1. **Dohvaćanje instrukcije iz memorije** pomoću **programskog brojila (PC)**.
 2. **Dekodiranje instrukcije** u upravljačkoj jedinici.
 3. **Dohvaćanje operanada** i njihovo spremanje u privremeni registar.
 4. **Izvršavanje operacije** (inkrementacija vrijednosti u memoriji).
 5. **Pohrana rezultata** na memorijsku lokaciju.
-

4. Ulazno/izlazni (I/O) uređaji

- **Mikrokontroler** = mikroprocesor + I/O uređaji na istom čipu.
 - **I/O uređaji koriste registre:**
 - **Podatkovni registri** – za prijenos podataka između CPU-a i uređaja.
 - **Statusni registri** – sadrže informacije o stanju uređaja.
 - **Načini komunikacije CPU-a i I/O uređaja:**
 1. **Isolated I/O (izolirani I/O)**
 - Odvojen adresni prostor za memoriju i I/O uređaje.
 - Efikasnija implementacija, ali zahtijeva dodatne instrukcije.
 2. **Memory-Mapped I/O (memorijski mapiran I/O)**
 - I/O uređaji koriste isti adresni prostor kao memorija.
 - Brže izvođenje, ali smanjuje kapacitet dostupne memorije.
-

5. Usporedba Isolated I/O vs. Memory-Mapped I/O

| Karakteristika | Isolated I/O | Memory-Mapped I/O |
|-----------------|--|--|
| Adresni prostor | Odvojen za memoriju i I/O | Dijele isti adresni prostor |
| Performanse | Sporije izvođenje I/O operacija | Brže I/O operacije |
| Programiranje | Složenije (posebne instrukcije) | Jednostavnije (isti set instrukcija za memoriju i I/O) |
| Fleksibilnost | Veća – može se dodavati I/O uređaje bez utjecaja na memoriju | Manja – adresni prostor memorije se smanjuje |
| Primjena | Ugradbeni sustavi, mikrokontroleri, real-time sustavi | Grafičke kartice, mrežni uređaji, DMA |

- **Primjena Memory-Mapped I/O:**
 - **Grafičke kartice** – CPU može brzo pristupati podacima kao da su u memoriji.
 - **Mrežni uređaji** – koristi se za efikasan prijenos podataka preko mreže.
 - **DMA (Direct Memory Access)** – omogućuje direktan prijenos podataka bez CPU intervencije.
- **Primjena Isolated I/O:**
 - **Mikrokontroleri** – svaki uređaj ima poseban port.
 - **Real-time sustavi** – precizno upravljanje uređajima bez utjecaja na memoriju.

Ključni koncepti za učenje

- ☒ Razumijevanje Harvard vs. Von Neumann arhitekture
- ☒ Razlike između RISC i CISC arhitekture
- ☒ Uloga mikroprocesora i njegovih registara
- ☒ Koraci izvršavanja instrukcija
- ☒ Razlike između Isolated I/O i Memory-Mapped I/O
- ☒ Kako CPU komunicira s I/O uređajima
- ☒ Primjene različitih arhitektura u realnim sustavima