

3

OSNOVNI PRINCIPI DELOVANJA RAČUNALNIKOV

BRANKO ŠTER

PO KNJIGI - DUŠAN KODEK: ARHITEKTURA IN ORGANIZACIJA RAČUNALNIŠKIH
SISTEMOV

Von Neumannov računalniški model

Von Neumann-ov računalnik:

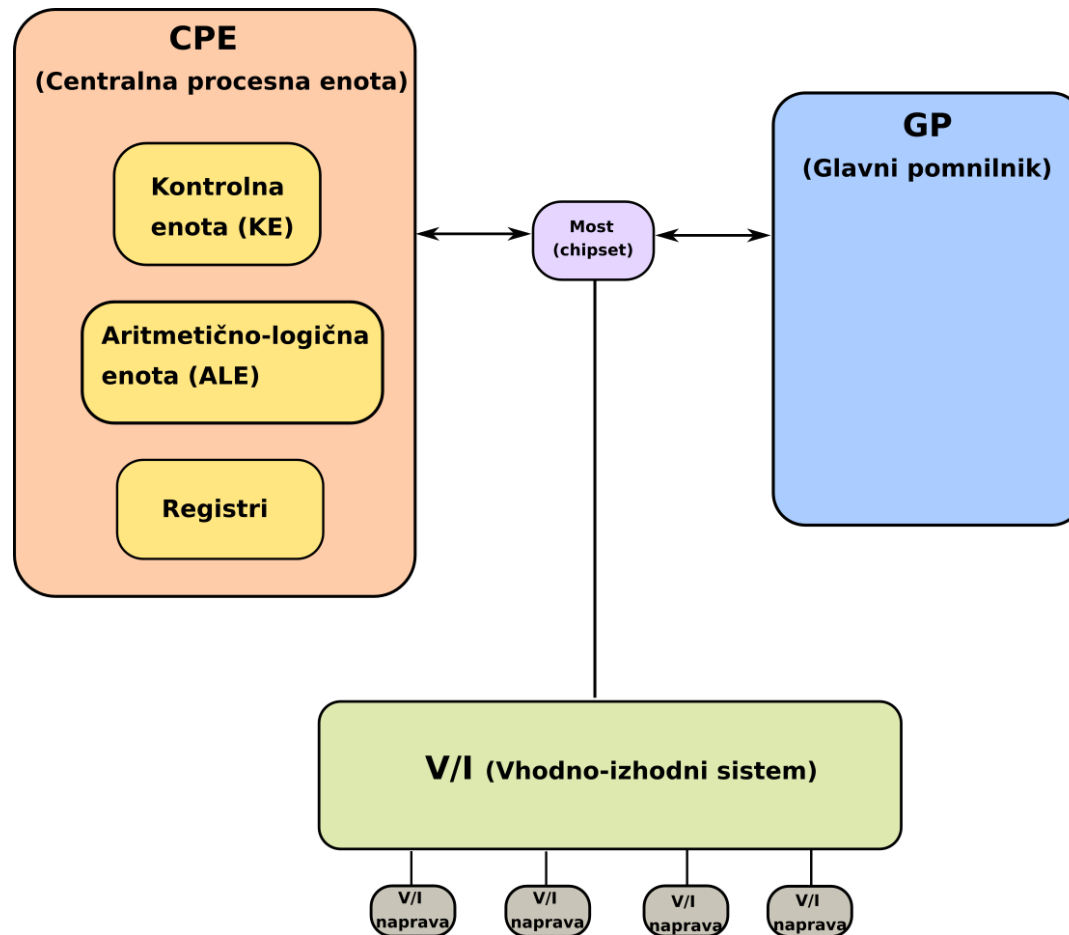
1. Sestavljajo ga

- centralna procesna enota (CPE)
- glavni pomnilnik (GP)
- vhodno/izhodni (V/I) sistem

2. Ima program shranjen v GP

3. CPE jemlje ukaze programa iz GP in jih zaporedoma izvršuje

Zgradba von Neumannovega računalnika



Glavni deli von Neumannovega računalnika

1. CPE oz. procesor

- mikroprocesor
- vodi dogajanje v računalniku
- osnovna naloga CPE je jemanje ukazov iz pomnilnika in njihovo izvrševanje
- CPE delimo na tri dele:
 1. **kontrolna enota** nadzoruje aktivnosti
 - prevzem ukazov in operandov
 - aktiviranje operacij
 2. **aritmetično-logična enota (ALE)** izvršuje večino ukazov
 3. **registri** začasno shranjujejo podatke

2. Glavni pomnilnik

- v njem so shranjeni ukazi in operandi
- GP sestavljajo pomnilniške besede (vsaka ima svoj naslov)
- tehnologija DRAM

3. Vhodno/izhodni (V/I, ang. I/O) sistem

- prenos informacije iz in v zunanji svet
- vhodno/izhodne oz. periferne naprave so fizično najvidnejši del računalnika
 - tipkovnica, miška, monitor, modem, disk, tiskalnik, ...
 - pretvarjajo informacijo iz CPE v obliko, primerno za človeka ali druge naprave
 - nekatere služijo kot pomožni pomnilnik

Ukaz

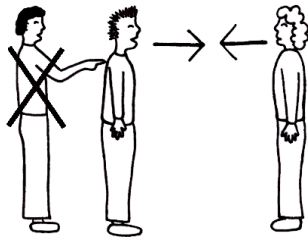
- Ukaz je shranjen v eni ali več (sosednih) pomnilniških besedah
- Vsak ukaz vsebuje
 - operacijsko kodo (katera operacija naj se izvrši)
 - informacijo o operandih, nad katerimi naj se izvrši operacija
- Format ukaza pove, kako so biti ukaza razdeljeni na operacijsko kodo in operande



- Naslov prvega ukaza (po vklopu računalnika) je vnaprej določen
- Pri vsakem ukazu sta 2 koraka:
 1. **Prevzem ukaza iz pomnilnika (fetch)**
 - to so **ukazi strojnega jezika** ali **strojni ukazi** (zaporedje ukazov je **program**)
 - strojni ukaz se bere iz tiste besede v pomnilniku, na katero kaže **programski števec** (PC, Program Counter)
 2. **Izvrševanje ukaza (execute)**
 - ukaz vsebuje operacijo in operande
 - CPE (običajno ALE) ukaz izvrši
 - PC nato vsebuje naslov naslednjega ukaza
 - običajno $PC \leftarrow PC + 1$ (razen pri **skočnih ukazih**)

Prekinitve

- Zaporedje teh 2 korakov se ponavlja ves čas delovanja računalnika
 - izjema so **prekinitve** (interrupt) in **pasti** (trap)

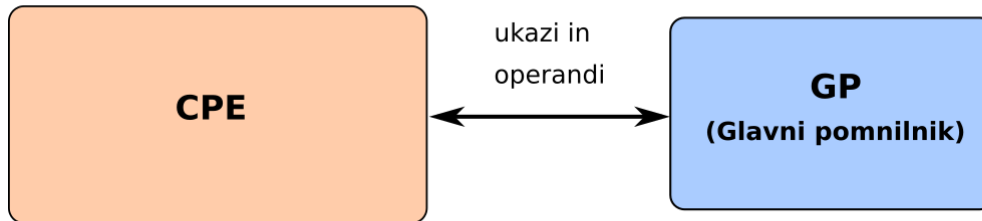


- takrat se izvrši skok na prvi ukaz **prekinitvenega servisnega programa** (PSP)
 - pred tem se shrani vrednost PC

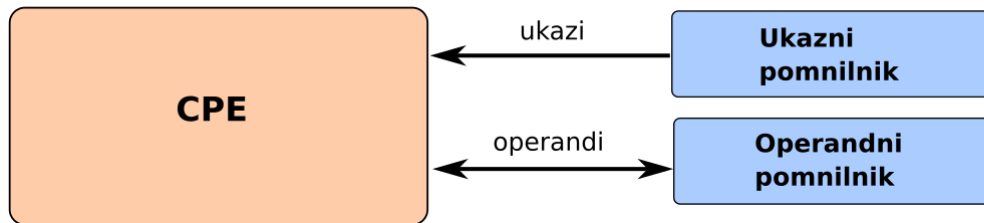
Glavni pomnilnik

- V glavni pomnilnik (GP) se shranjujejo ukazi in operandi
- GP je pasiven
- Za zmogljivost računalnika je pomembno, da se med CPE in GP lahko prenese dovolj informacije
 - ozko grlo von Neumann-ovega računalnika
 - ena od rešitev je Harvardska arhitektura (po Harvard Mark I-IV)
 - ima pomnilnik za ukaze in pomnilnik za operande
 - običajna arhitektura se imenuje Princetonska (zaradi IAS)

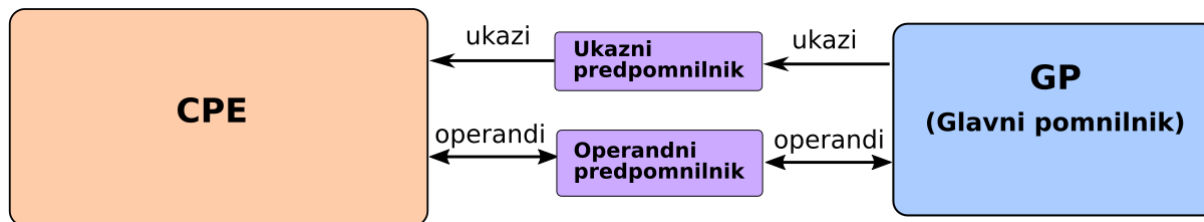
Princetonska arhitektura



Harvardska arhitektura



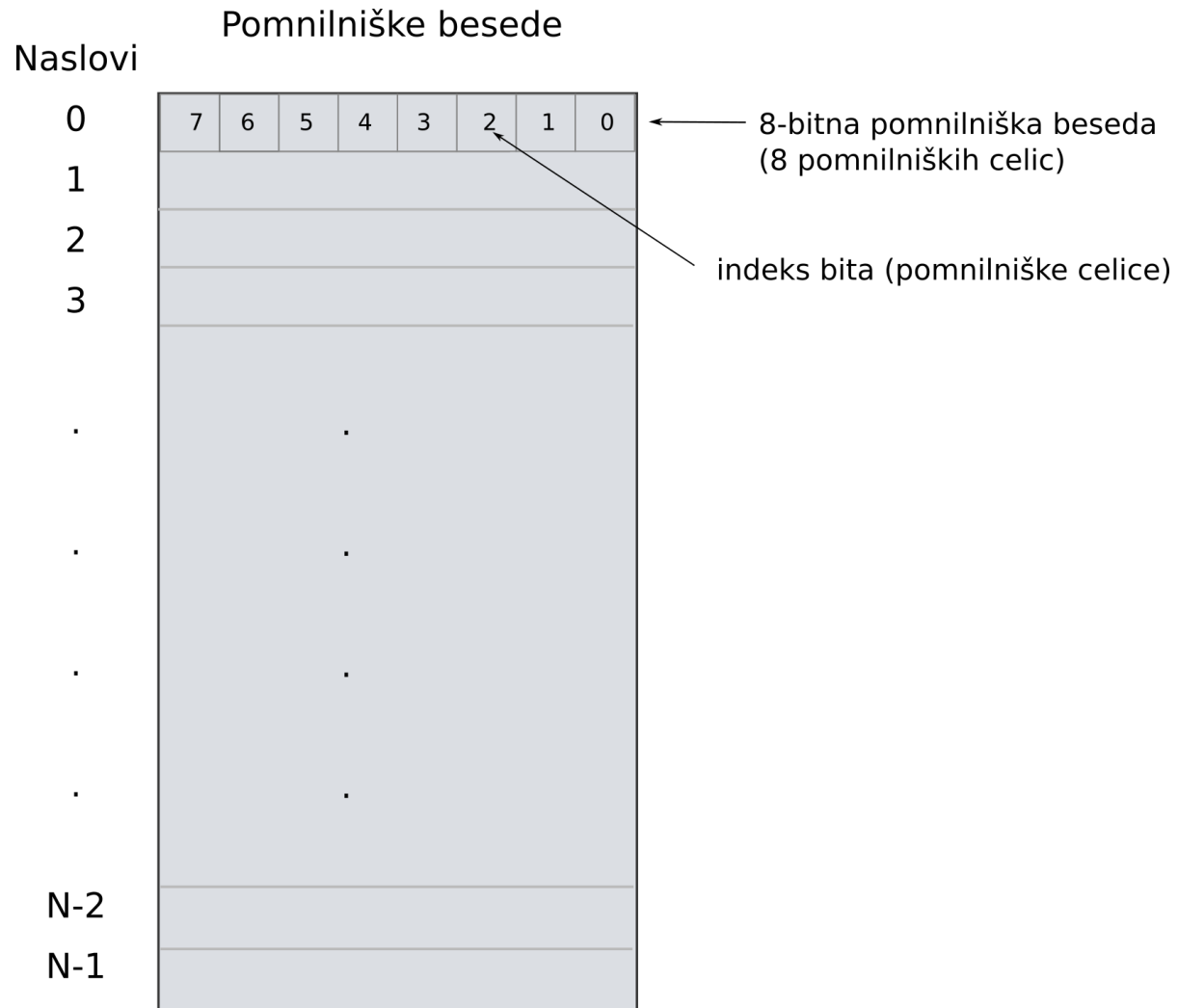
Danes prevladuje Princetonska arhitektura, vendar z ločenima *predpomnilnikoma* za ukaze in operande:



Pomnilniške besede

- GP je zaporedje **pomnilniških besed** oz. **pomnilniških lokacij**
 - **dolžina pomnilniške besede** je število pomnilnih celic v njej (vsaka hrani 1 bit informacije)
 - dolžina pomnilniške besede je najpogostejše 8 bitov (1 **byte** oz. **bajt**, 1B)
 - vsaka lokacija ima svoj naslov
 - pomnilniška beseda je definirana kot najmanjše število bitov s svojim naslovom
 - iz pomnilnika ni možno prebrati (ali vanj vpisati) manj kot eno besedo

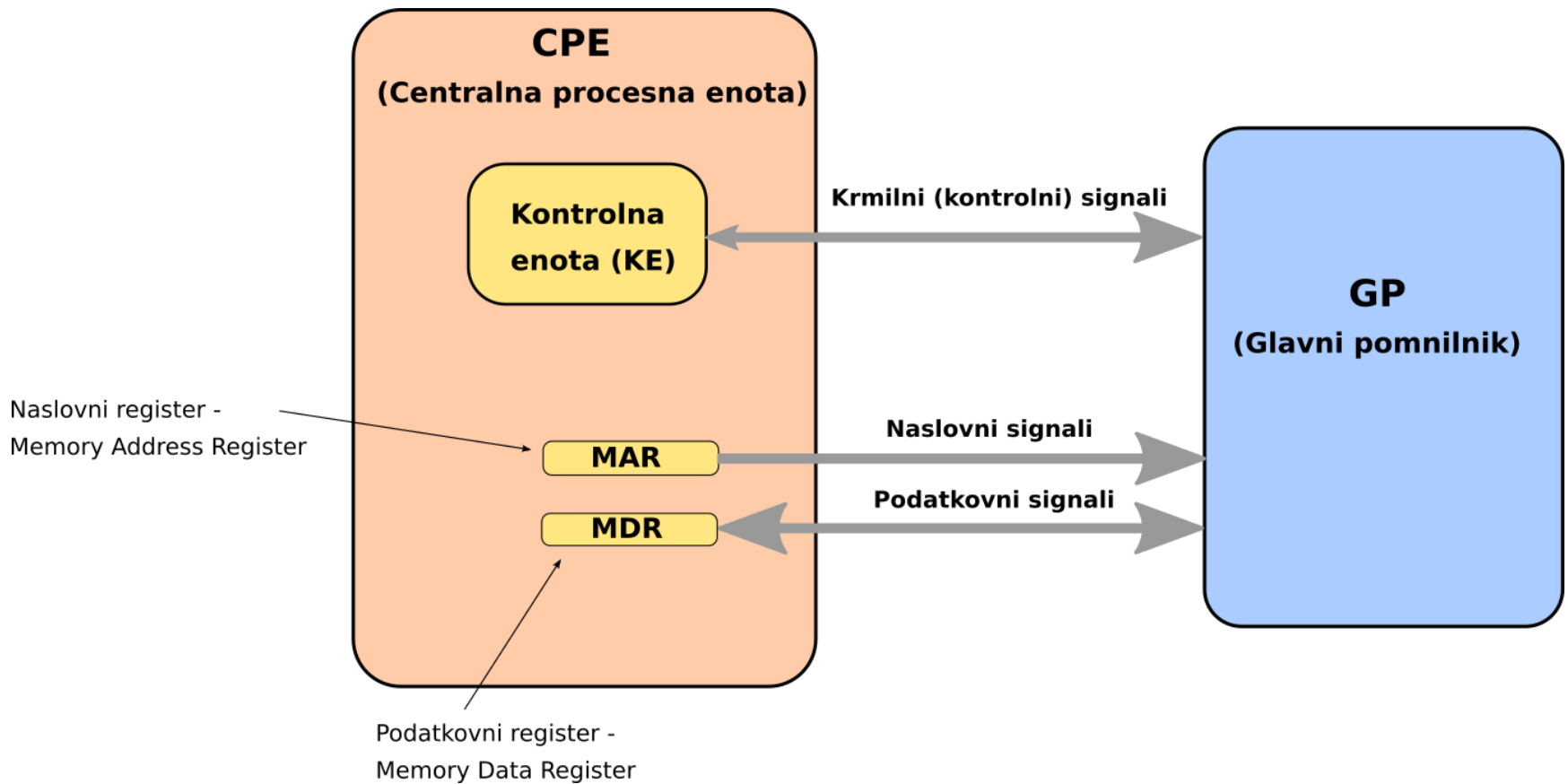
GP z dolžino besede 8 bitov:



Naslovni prostor

- Velikost **naslovnega prostora** = $2^{\text{dolžina naslova}}$ (v bitih)
 - npr. pri 12-bitnem naslovu je naslovni prostor velikosti $2^{12} = 4096$ pomnilniških besed oz. 4K
 - $2^{10} = 1024 = 1\text{K}$ (kilo),
 - $2^{20} = 1\,048\,576 = 1\text{M}$ (mega),
 - $2^{30} = 1\,073\,741\,824 = 1\text{G}$ (giga)
 - $2^{40} = 1\text{T}$ (tera)
- Vsebina pom. besede se lahko spreminja
 - v 8-bitno besedo lahko shranimo 2^8 različnih vsebin
- Če so registri večji kot pomnilniška beseda, je možen dostop tudi do več besed naenkrat (vsaj pri večini računalnikov)
 - npr. 32-bitni registri in 8-bitna beseda: dostop do 4 zaporednih besed hkrati (GP v obliki 4 pom.)

- CPE poda naslov besede in smer prenosa (lahko pa tudi št. besed)
- **Dostop** do pomnilnika (glede na smer prenosa):
 - **branje** iz pomnilnika (5x bolj pogosto)
 - **pisanje** v pomnilnik
- Informacije potujejo po ***vodilih***
- CPE da naslov *na naslovno vodilo* in s *krmilnimi (kontrolnimi) signali* pove pomnilniku, da želi dostopiti do pomnilniške besede s tem naslovom
 - Pri branju pričakuje, da bo pomnilnik dal podatek na ***podatkovno vodilo***
 - Pri pisanju da CPE na ***podatkovno vodilo*** podatek, ki se zapiše v pomnilnik



MAR in MDR

- CPE običajno vsebuje tudi
 - **naslovni register** oz. **MAR** (memory address register)
 - vsebuje naslov pomnilniške besede, do katere želimo dostopiti
 - **podatkovni register** oz. **MDR** (memory data register)
 - sem se pri branju zapiše iz pomnilnika prebrana vrednost
 - pri pisanju je v njem vrednost, ki naj se zapiše v pomnilnik
- MAR in MDR sta povezana s pomnilnikom preko naslovnih oz. podatkovnih signalov (vodil)
 - poleg teh obstajajo tudi kontrolni signali (smer prenosa (branje/pisanje), število besed, časovni parametri, ...)

- Dolžina MAR je enaka dolžini naslova
 - isto dolžina PC
 - če naslovni prostor postane premajhen, je to lahko velik problem
 - naslovi nastopajo tudi kot operandi
 - povečanje naslova pomeni drugačno zgradbo ukazov in s tem nekompatibilnost za nazaj (kar kažejo tudi ☹ izkušnje proizvajalcev)

- Dolžina MDR določa število bitov, ki se lahko naenkrat prenesejo med CPE in GP
 - enaka večkratniku dolžine pom. besede
 - njeno povečanje ni tako težavno
 - dolžina MDR vpliva na število dostopov za operand določene velikosti (npr. $64=2*32$)
 - programer tega ne vidi

Povzetek

- CPE da naslov na naslovno vodilo in s kontrolnimi signali pove pomnilniku, da želi dostopiti do pom. besede s tem naslovom
 - Pri branju pričakuje, da bo pomnilnik dal podatek na podatkovno vodilo
 - Pri pisanju da CPE na podatkovno vodilo podatek, ki se zapiše v pomnilnik

Vhod in izhod

- Osnovna naloga V/I sistema je pretvorba informacije
 - izjema so naprave za shranjevanje informacije
 - pomožni pomnilniki (npr. magnetni disk, optični disk, magnetni trak)
 - cena, obstojnost informacije
- Osnovni način delovanja V/I sistema je prenos podatkov
 - med GP in V/I napravami ali
 - med CPE in napravami
- Razlike med računalniki glede izvedbe V/I so velike
 - pri znanstvenem računanju malo V/I prenosov
 - pri poslovnem veliko

2 skupini izvedb V/I sistema :

1. Programski vhod/izhod (Programmed I/O)

- z V/I napravo komunicira CPE
- vsak podatek se prenese iz GP v CPE in nato v napravo ali obratno
- prenos z zaporedjem ukazov
- počasnost in zasedenost CPE

2. Neposreden dostop do pomnilnika (Direct memory access - DMA)

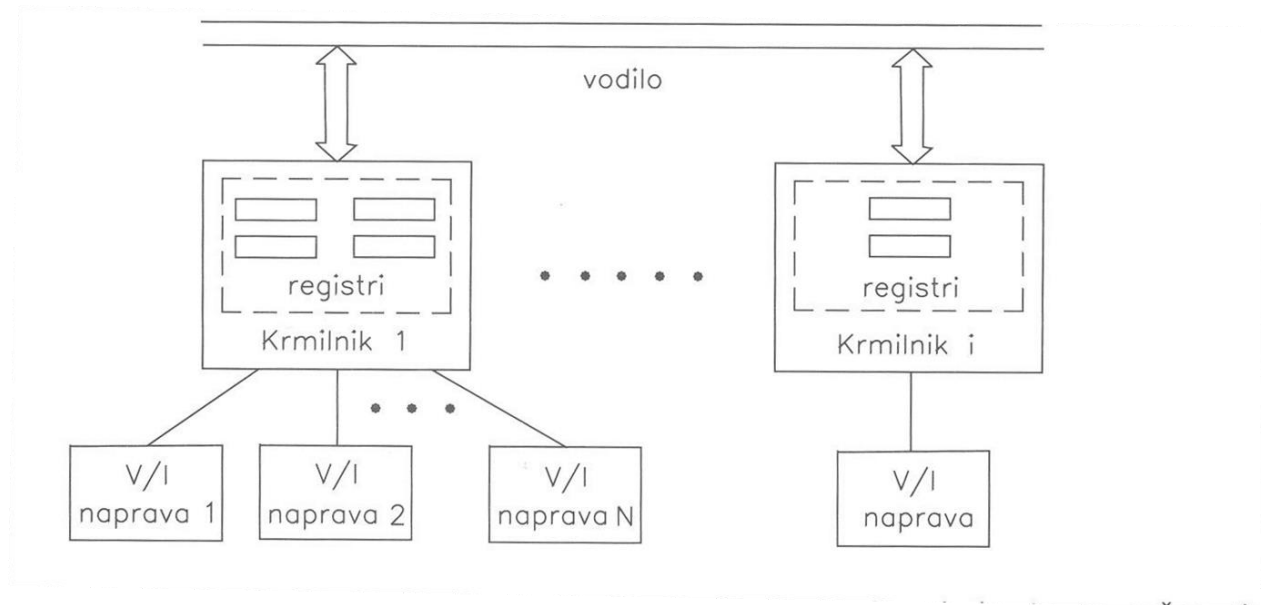
- naprava komunicira neposredno z GP
- **DMA krmilnik** nadomesti CPE
- posebna izvedba DMA krmilnikov so **vhodno/izhodni procesorji**

➤ Pri mnogih računalnikih srečamo oba načina dostopa

- za počasne naprave je primeren programski V/I
- za hitre oz. podatkovno zahtevne nujen DMA

Vsaka V/I naprava je priključena preko **krmilnika naprave** (device controller)

- vezje, ki omogoča prenos podatkov v napravo in iz nje
- lahko tudi več naprav na krmilnik
- s krmilnikom komuniciramo preko njegovih registrov



Registri krmilnikov so lahko v istem naslovnem prostoru kot GP, lahko pa v posebnem

Ločimo 3 izvedbe:

1. Pomnilniško preslikan vhod/izhod (Memory-mapped I/O, MMIO)

- registri krmilnikov so v pomnilniškem naslovnem prostoru
- ni posebnih V/I ukazov

2. Ločen vhodno/izhodni prostor

- registri krmilnikov so v posebnem naslovnem prostoru
- za dostop do registrov so potrebni *posebni V/I ukazi*

3. Posredno preko vhodno/izhodnih procesorjev

- tudi tu so registri krmilnikov v posebnem naslovnem prostoru, ki pa iz CPE ni neposredno dostopen
- vmes so še vhodno/izhodni procesorji (razbremenijo CPE)
- pri velikih računalnikih

Računalnik kot zaporedje navideznih računalnikov

- Večine uporabnikov arhitektura računalnika (pravzaprav) posebno ne zanima
 - programske jezike lahko implementiramo na različnih računalnikih
- Tanenbaum, 1984:
 - Računalnik kot zaporedje navideznih računalnikov
 - Vsak nivo si lahko predstavljamo kot navidezni računalnik, ki ima za “strojni” jezik kar jezik tega nivoja
 - večina uporabnikov se spodnjih nivojev niti ne zaveda

2 mehanizma za prehod med nivojema:

- Prevajanje (prevajalnik)
 - izvorni program v enem jeziku
 - ciljni program (object program) v drugem (nižjem) jeziku
 - izvornega načelno ne rabimo več
- Interpretacija (interpreter)
 - izvorni program se prevaja sproti
 - ukaz se prevede in izvrši
 - izvorni program rabimo ves čas
 - bolj fleksibilno, prenosljivo
 - manjša hitrost
- delno prevajanje
 - prevajanje v vmesno kodo, ki se jo interpretira
 - npr. Java

6 nivojev:

Nivo 5: Višji prog. jezik

- prevajanje ali interpretiranje

Nivo 4: Zbirni jezik

- prevajanje

Nivo 3: Operacijski sistem

- interpretiranje

Nivo 2: Strojni jezik

- interpretiranje

Nivo 1: Mikroprogramski jezik

- interpretiranje

Nivo 0: Digitalna logika

Strojna in programska oprema računalnika

➤ Delitev

- hardware
- software
- firmware
 - program, ki je vgrajen v HW napravo (kot ROM ali bliskovni pomnilnik) in skrbi za njeno osnovno funkcionalnost

➤ Strojna in programska oprema sta funkcionalno ekvivalentni

- poljuben računalnik bi se načeloma dalo realizirati samo z elektroniko (dovolj kompleksno)