ZAPIS INFORMACIJE

Informacija v racunalniku -ukazi

-operandi

ASCII (American standart code for information interchange)

7-bit(128 znakov)

Razširjen ASCII

8-bit

256 znakov

Koda BCD(Binary coded decimal)

4-bit binarna predstavitet desetikih cifer



1. Predzank-velicinski

-PV zapis ni primeren za seštevanje/odštevanje

-Primeren za množenje/deljenje (ki pa sta manj pogosti operaciji)

2. Zapis (predstavitev) z odmikom (PO)

Hibe:

• pri seštevanju je treba odmik odšteti

• pri odštevanju je treba odmik prišteti

3.Eniški komplement (1’K)

4. 2’K je najpogosteje uporabljan zapis

• primeren za seštevanje/odštevanje

◦ nima EAC

• le ena predstavitev za ničlo

• predznaka ni treba obravnavati posebej

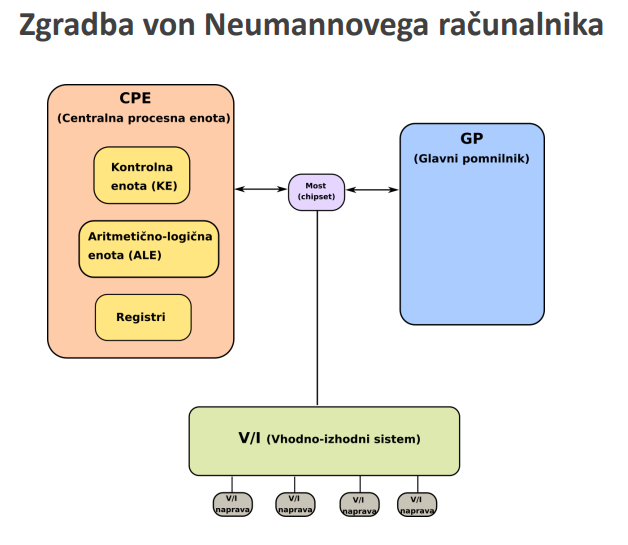
Do previla pride kdaj

-števili imata enaki predznak

-vsota ima drugačen predznak kot števili

OSNOVNI PRINCIPI

DELOVANJA RAČUNALNIKOV



1. CPE(procesor)

-mikroprocesor

-vodi dogajanje v racunalniku

-osnovna naloga CPE je jemanje ukazov iz pomnilnika in njihovo izvrsevanje

-CPE delimo na tri dela:-ALE(aritmeticko-logicka enota)

-registri

-kontrolna enota

1. Glavni pomnilnik

-v njemu su shranjeni operandi in ukazi

-GP sestavljajo pomnilniske besede

-tehnologija DRAM(Dynamic random access memory)

1. Vhodno/Izhodni sistem

-namenjem prenosu informacija v zunajni svet

UKAZ

Vsak ukaz vsebuje

▪ operacijsko kodo (katera operacija naj se izvrši)

▪ informacijo o operandih, nad katerimi naj se izvrši operacija



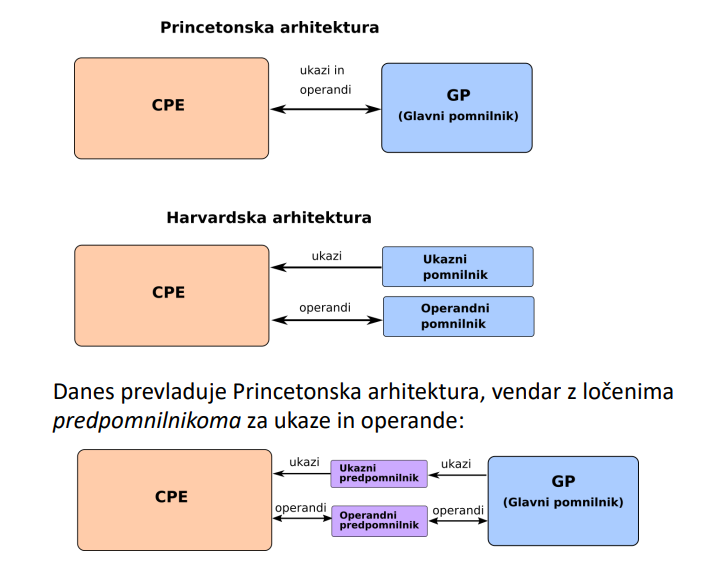
Pri vsakem ukazu sta 2 koraka: 1. Prevzem ukaza iz pomnilnika(fetch)

1. Izvrševanje ukaza(execute)

Prekinitve

prekinitvenega servisnega programa (PSP)

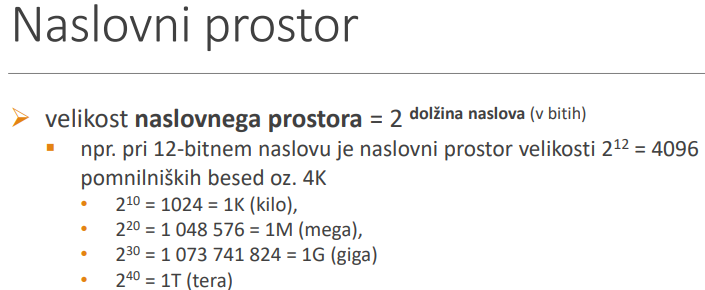
GLAVNI POMNILNIK



Harvardska arhitektura (po Harvard Mark I-IV)

• ima pomnilnik za ukaze in pomnilnik za operande

• običajna arhitektura se imenuje Princetonska (zaradi IAS)



2 mehanizma za prehod med nivojema:

PREVAJANJE

◦ izvorni program v enem jeziku

◦ ciljni program (object program) v drugem (nižjem) jeziku

◦ izvornega načelno ne rabimo več

INTERPRETACIJA

◦ izvorni program se prevaja sproti

◦ ukaz se prevede in izvrši

◦ rabimo ga ves čas

◦ bolj fleksibilno

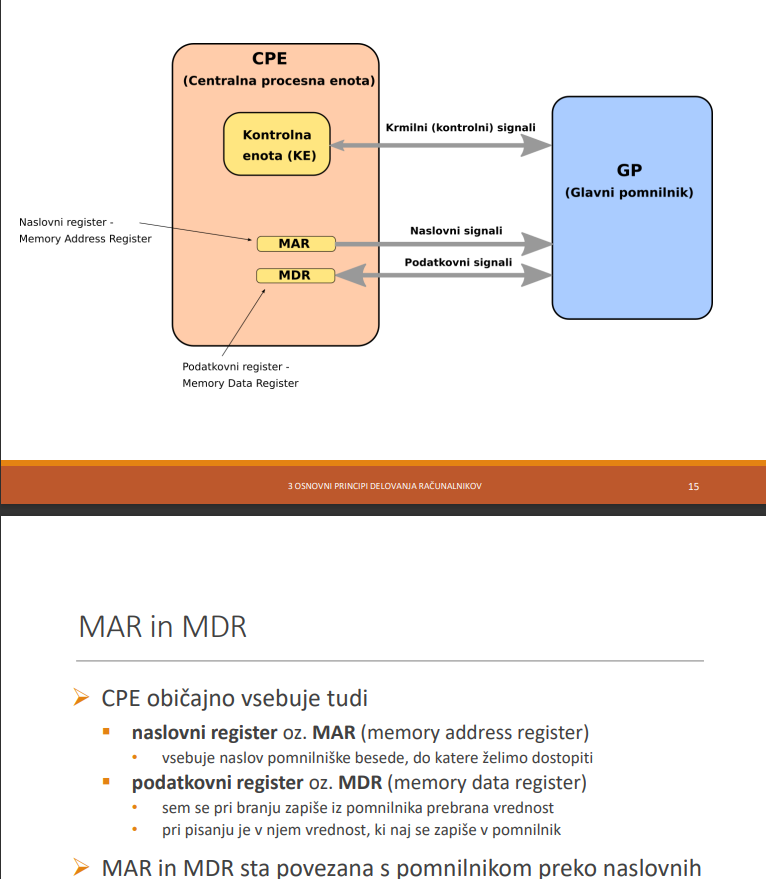
◦ večja prenosljivost

◦ manjša hitrost

DELNO PREVAJANJE

◦ prevajanje v vmesno kodo, ki se jo interpretira

◦ npr. Java



RISC-V in ukazi load/store

Ukazna arhitektura (ISA, Instruction set architecture)

-natancno definira vse ukaze

Lastnosti RISC-V

➢ 8-bitna pomnilniška beseda

➢ 32-bitni pomnilniški naslov

➢ Način shranjevanja operandov v CPE

▪ 32 32-bitnih splošnonamenskih registrov x0, x1, …, x31

• Vsebina x0 je vedno 0 (pri pisanju vanj se ne zgodi nič)

➢ Število eksplicitnih operandov v ukazu

▪ vsi ALE ukazi imajo 3 eksplicitne operande

• Tip R ima dva izvorna (source) registra rs1 in rs2 ter en ciljni

destination register rd

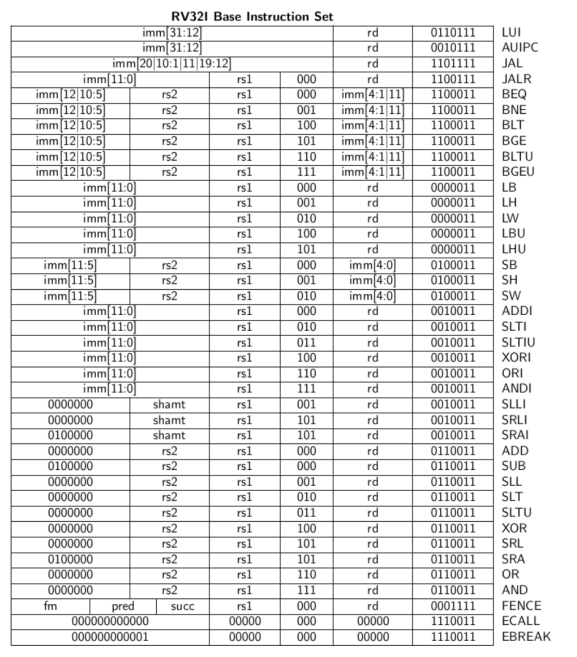
• Tip I ima en izvorni (source) register rs1, 12-bitni takojšnji

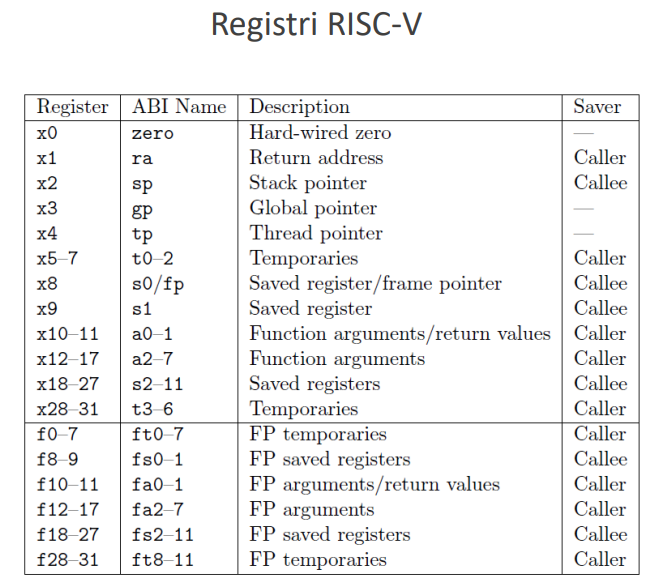
(immediate) operand imm ter en ciljni destination register rd

Vrste ukazov

1. Ukazi za prijenos podatkov(load, store)
2. ALE ukazi (aritmeticko logicne operacije)
3. Kontrolni ukazi(skoki)
4. Sistemski ukazi

Aritmetični ukazi





ABI(aplication binary interface)

-Doloca nacin zapisa podatkovnih strukturi n nacin klicanja podprogramov na nizkem nivoju

ABI doloca:1.Strukturo registrov, organizacijo sklada, vrste pomnilniskih dostopov

2.Podatkovne tipi(velikost,poravnatost,endian)

3.nancin klica sistemskih klicev

4.dogovor o klicih podprogramov

ALE ukazi

1. aritmetične operacije (+, −)

• ADD, ADDI,

• SUB

• LUI, AUIPC

2. logične bitne operacije (&, V, )

• AND, ANDI

• OR, ORI

• XOR, XORI

3. pomiki (shift) (levi, desni; logični, aritmetični)

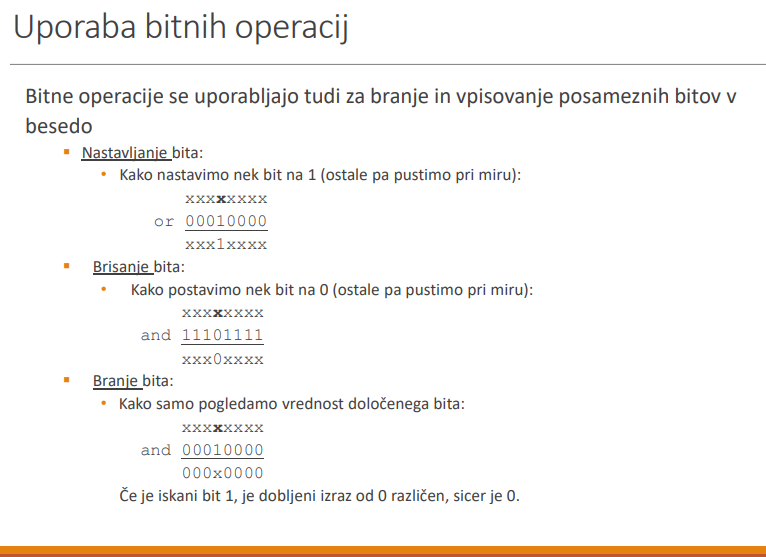
• SLL, SLLI

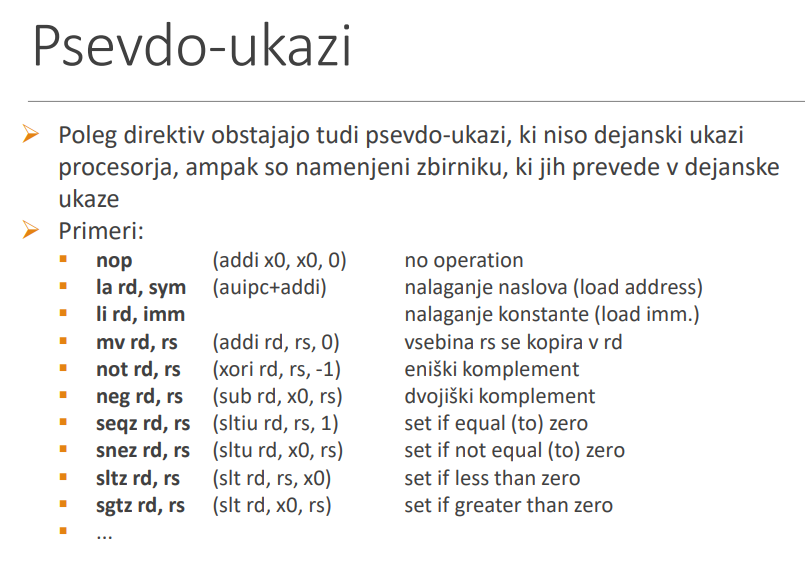
• SRL, SRLI

• SRA, SRAI

4. ukazi za primerjavo oz. set operacije (pogoj: < )

• SLT, SLTI, SLTU, SLTIU





Splošne lastnosti ukazov

Dimenzija

1. Način shranjevanja operandov v CPE

2. Število eksplicitnih operandov v ukazu

3. Lokacija operandov in načini naslavljanja

4. Operacije

5. Vrsta in dolžina operandov

Nacini shranjevanja operandov v CPE

1. Akumulator
2. Sklad
3. Mnozica registrov

2- in 3-operandni računalniki se delijo še na:

▪ registrsko-registrske računalnike

• najbolj razširjeni

• vsi operandi v registrih CPE

• reče se tudi load/store računalniki (ker rabimo load in store)

▪ registrsko-pomnilniške

• en operand v registru, drugi lahko v pomnilniku

▪ pomnilniško-pomnilniške

• vsak operand lahko v pomnilniku

• zapleteni ukazi, CISC (npr. VAX)

Nacini naslavljanja

1.Takojsnje naslavljanje(immidiate addressing)

-operand je v ukazu podan z vrednostjo

2.Neposredno naslavljanje(direct addresing)

-operand je v ukazu odan z naslovom

3.Posredno naslavnja (indirect addressing)

-v ukazu je naslov lokacije, na kateri je shranjen naslov

Operanda

Slupine operacij

1. Prijenos podav(dana transfer)

◦ LOAD: GP → R

◦ STORE: R → GP

◦ MOVE: R → R ali GP → GP

◦ PUSH: GP ali R → Sklad

◦ POP (PULL): Sklad → GP ali R

1. Aritmeticne in logicne operacije

-Izvajajo se v ALE

-seštevanje, odštevanje, množenje,

deljenje, aritm. negacija, absolutna vrednost, inkrement,

dekrement

-za vsako je več ukazov (različne dolžine operandov)

- Logične operacije: AND, OR, NOT, XOR, pomiki

1. Kontrolne operacije

-pogojni in brezpogojni skoki

1. Operacije v plavajoci vejici

-izvaja ih posebna enota(FPU-floating point unit, ni del ALE)

1. Sistemske operacije

-vplivajo na nacin delovanja racunalnika

1. Vhodno/Izhodne operacije

-obstajajo na nekaterih računalnikih

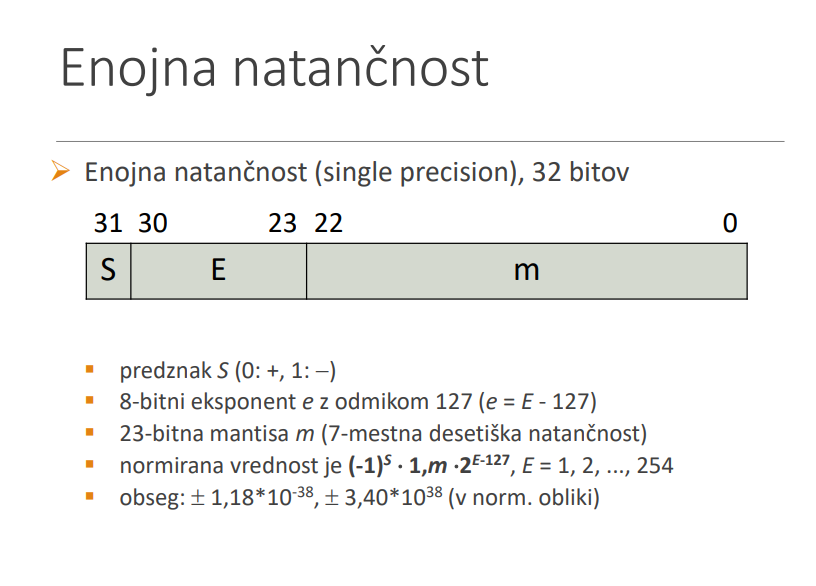
-na drugih se uporabljajo običajni ukazi za prenos podatkov

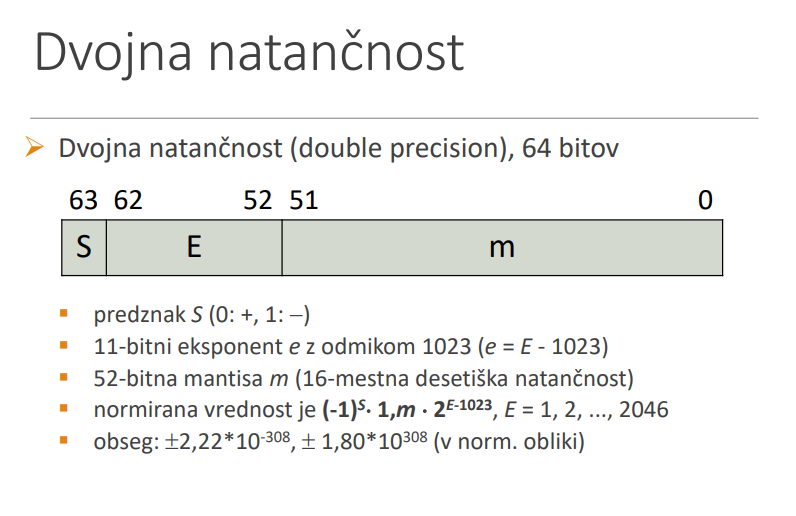
Ortogonalnost ukazov(medsebojna neodvisnost parametrov ukaza)

1. Informacija o operaciji neodvisna od info o operandih
2. Informacija o enem operandu neodvisna od info o ostalih operrandih



ARITMETIKA

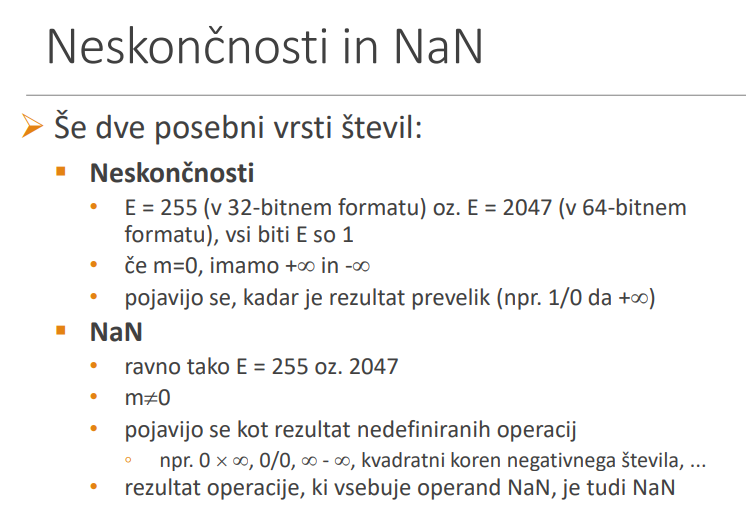




Denormirana stevila(zelo majhna stevila)

-E=0

-implicitni normalni obliki je enak 0



Dodatni biti

-Varovalni bit je potreben ker je vsota lahko za eno mesto daljsa od operandov

-Zaokrozevalni bit omogoca bolj natancno zaokrozevanje

-Ljepljiv bit se uporablja zato da se iz izpadlih bitov vidi ali je bil kak razlicen od 0(zaradi zaokrozevanja k sodem stevilu, v tem primeru treba zaokroziti navzgor)

CENTRALNA PROCESNA ENOTA

Splosno

-CPE je digitalen sistem

-Delovanje CPE je odvisno od: 1.trenutneg stanja

2renutnih vhodov

1. Dobava ukaza iz pomnilnika(fetch)
2. Izvrsenje ukaza: 1. dekodiranje

2.izvedba ALE operacije

3.prenos operandov v CPE iz pomnilnika

4.shranjevanje rezultata – pisanje v register

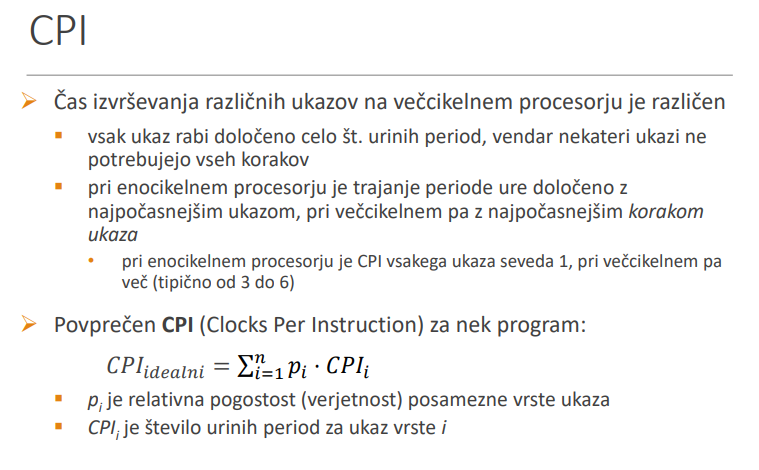
5.PC <- PC+1

Nacini implementacije CPE

1.Celoten ukaz v eni periodi ure(single cycle)

2.Ukaz v več periodah ure(multi cycle)

3.Ukaz v več periodah ure, a se ukazi casovno delno prekrivajo(pipeline-cevovod)



PARALELIZEM NA NIVOJU UKAZOV

Cevovodne nevarnosti

1. Strukturne nevarnosti

-kadar vec stopenj cevovoda v neki uri potrebuje isto enoto

1. Podatkovne nevarnosti

-Kadar ukaz potrebuje kot vhodni operand rezultat prejesnjega, se ni dokocanega ukaza

1. Kontrolne nevarnosti

-mozne pri skokih, klicih in drugih kontrolnih ukazih

5 korakov izvrševanja ukazov: 5 stopenj cevovoda

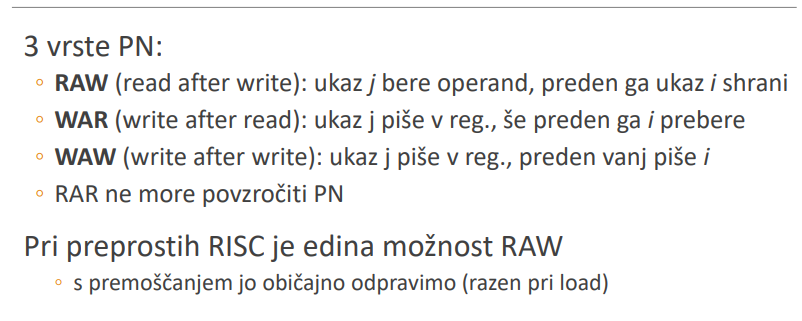
1. IF: prevzem ukaza in sprememba PC

2. ID: dekodiranje ukaza in dostop do registrov

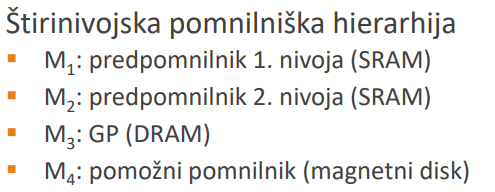
3. EX: izvrševanje operacije

4. MEM: dostop do pomnilnika

5. WB: shranjevanje rezultata



PREDPOMNILNIKI

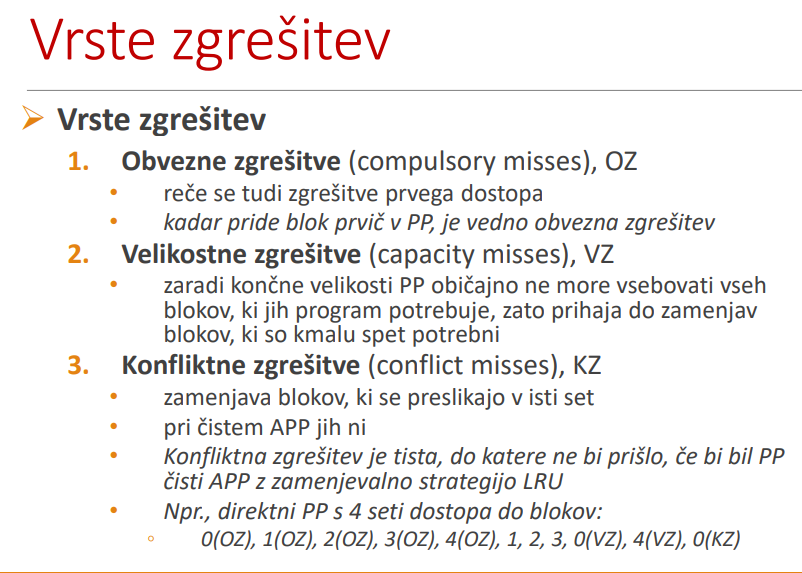


DRAM VS SRAM

In summary, SRAM is faster, more expensive, and uses more transistors per cell, making it suitable for applications requiring high speed and reliability. DRAM, on the other hand, is slower but offers higher density and lower cost per bit, making it ideal for applications where large amounts of memory are needed.

Predpomnilnik

PP hrani dolocene podatke kis o tudi v glavnemu pomnilniku(PP je podmnozica vsebine GP)



PODPROGRAMI

sklad (angleško "stack") je podatkovna struktura, ki deluje po načelu LIFO (Last In, First Out), kar pomeni, da je zadnji element, ki je bil dodan na sklad, tudi prvi, ki je odstranjen. Sklad se pogosto uporablja za začasno shranjevanje podatkov in ima pomembno vlogo pri upravljanju funkcijskih klicev in rekurzije v programskih jezikih.

POMNILNIKI

Lastnosti pooilnikov

1. Cena
2. Hitrost dostopa
3. Nacin dostopa

-nakljucni dostop (RAM- DRAM, SRAM)

-zaporedni dostop(cas dostopa je odvisen od prejesnjeg naslova)

-krozni dostop(npr. magnetni disk s fiksnimi glavami)

-kombinacija zaporednog in krozega dostopa

Asociativnpomnilniki

-pomn. z dostopom preko vsebine(CAM, Content addressable memory)

1. Spremenljivost

-Bralni pomnilniki(ROM-Read only memory)

-Programirljivi bralni pomnilniki(Programmable ROM-PROM)

-Bralno-pisalni pomnilniki(Random access memory)

1. Obstojnost
2. Zanesljivost

Organizacija glavnega pomnilnika

Osnovna parametra pomnilnika sta:1.Pomnilniska beseda

2.Pomnilniski naslov