p.s. snov ki manjka: lokalnost pomnilniških dostopov, število eksplicitnih operandov v ukazu, načini naslavljanja!

Von Neumannov računalniški model

Velika večina sodobnih računalnikov spada v skupino tako imenovanih von Neumannovih računalnikov. Model je 1945 predlagal ameriški matematik madžarskega rodu John von Neumann. Povsem nova je bila ideja o računalniku s shranjenim programom, ki je bistveno drugačen od takrat že delujočega prvega elektronskega računalnika ENIAC.

Tipičen von Neumannov model sestavljajo trije deli:

- **CENTRALNO PROCESNA ENOTA (PROCESOR)**
  V von Neumannovem računalniku se večina dogajanja odvija v CPE ali pod njeno kontrolo. Glavna naloga CPE je da iz pomnilnika jemlje ukaze in jih izvršuje. Običajno jo delimo na 3 dele:
  - KONTROLNA ENOTA-vodi delovanje računalnika, skrbi za prevzem ukazov in operandov ter za aktiviranje ustreznih operacij.
  - ARITMETIČNO LOGIČNA ENOTA-izvršilna enota, ki izvaja operacije
  - REGISTRI-ena ali več med seboj povezanih pomnilniških celic. V eno celico lahko shranimo en bit. Služijo za shranjevanje

    Registre delimo na: Programsko dostopne (branje, pisanje)
    -dostopne v strojnem jeziku
    Programsko nedostopne
    -ni ukaza s katerim bi prišli do registra

- **GLAVNI POMNILNIK**
  Je sestavljen iz pomnilniških besed, od katerih ima vsaka svoj enoumno določen naslov. V skupnem pomnilniku so shranjeni ukazi in operandi. Glavni pomnilnik je pasiven, naredi samo tisto kar od njega zahteva CPE in V/I naprava.
  Pomnilniška beseda:
  -ima svoj pomnilniški naslov(se ne spreminja)
  -ima vsebino

- **VHODNO/IZHODNI SISTEM**
  V CPE in v glavnem pomnilniku je informacija shranjena v obliki, ki je zunanjemu svetu nedostopna. Vsak računalnik ima zato del, ki ga imenujemo vhodno/izhodni sistem in je namenjen prenosu informacije v in iz zunanjega sveta. Vhodno/izhodni sistem je sestavljeni iz: -vhodno/izhodnih vmesnikov
  -vhodno/izhodnih naprav (tipkovnica, tiskalnik, miška)

DELOVANJE VON NEUMANNOVEGA RAČUNALNIKA
(CPE mora poznati naslov prvega ukaza)

2 KORAKA

- (fetch cycle-ukazno prevzemni cikel) Jemanje (branje) ukaza iz pomnilnika. Naslov strojnega ukaza ki naj se prebere iz pomnilnika se nahaja v posebnem programsko nedostopnem registru, ki mu pravimo programski števec ali PC. Ta prebere pomnilniški naslov, iz katerega naj se prebere naslednji ukaz.

- (execute cycle-izvršilni cikel) Vsak strojni ukaz mora vsebovati dve vrsti informacij:
  -informacija o operaciji, ki naj se izvede
  -informacija o operandih, s katerimi naj se izvede operacija

Vsebina PC se poveča, tako da vsebuje pomnilniški naslov naslednjega ukaza.PC←PC+1.
Ta dva koraka se ciklično ponavljata, dokler se ne pojavita izjemi.

1. **IZJEMA**: če pride v izvajanje ukaz, ki spremeni vsebino PC-SKOČNI UKAZ (jump).
2. **IZJEMA**: signal, s katerim neka naprava zahteva, da je CPE obravnavan-PREKINITEV.

GLAVNI POMNILNIK V VON NEUMANNOVEM RAČUNALNIKU
Tu so shranjeni operandi in ukazi. Že prej smo povedali, da je ta naprava pasivna, sestavljajo ga več različnih vrst pomnilnikov, ki so gledano iz CPE kot en sam glavni pomnilnik.
Povezava med glavnim pomnilnikom in CPE mora biti hitra, da lahko CPE sprejema ukaze tako hitro, kakor jih izvršuje, zato da ne čaka. Temu prenosu pravimo PROMET.
To pot med glavnim pomnilnikom in CPE omejuje hitrost, zato temu rečemo von Neumannovo ozko grlo.

PRINCETONSKA ARH.
CPE je povezan z glavnim pomnilnikom, kjer se nahajajo in ukazi in operandi, in je homogen.

HARVARDSKA ARH.
CPE je povezan posebej z ukaznim pomnilnikom in z operandnim pomnilnikom.

Še nekaj malega o gl. pomnilniku:

Pomnilniška beseda ima naslov (lahko več, če je sestavljen operand), ki je enovit, nespremenjen. Sestavljena je iz 1-bitne pomnilniške celice. Dolžina pomnilniške besede se meri v bitih, ki pove koliko 1-bitnih pomnilniških celic sestavlja pomnilniško besedo.
Danes je pomnilniška beseda običajno dolga 8 bitov ali 1 bajt.

Prenos iz gl. pomnilnika v CPE→bralni dostop (vsebina ostane nespremenjena)
Prenos iz CPE v gl. pomnilnik→pisalni dostop (vsebina pomn. besede se običajo spremeni)
Branj je veliko več kot pisanj!

LOČITI MORAMO:
Naslov pomnilniške besede: je nespremenljiv, dolžina naslov (število bitov, ki sestavljajo pomnilniški naslov) določa velikost naslovnega prostora.
Naslovni prostor so vsi možni različni naslovi, ki jih generira CPE.

Dolžina pomnilniškega naslova: Če je dolžina pomnilniškega naslova 16-bitna, to pomeni da je velikost naslovnega prostora 2^{16} .Na vsak naslov lahko damo eno pomnilniško besedo.

Lastnosti von Neumannovega pomnilnika:

1. pomnilnik je enodimenzionalen in organiziran v besede. Vsaka beseda ima svoj enoličen naslov. Pravimo mu, da ima obliko linearnega vektorja.

2. ni razlike med ukazi in operandi. Ukaze lahko obravnavamo kot operande in obratno. Pri harvardski arhitekturi prevzema CPE ukaze iz drugega pomnilnika kot operande, vendar je vsebino enega pomnilnika vedno mogoče prenesti v drugega in obratno.
3. pomen ni sestavni del operandov, Iz vsebine neke pomnilniške besede ni mogoče razbrati, ali je v njej shranjen operand števila s fiksno vejico, plavajočo, ASCII abecedi,…

VHOD/IZHOD

Tak sistem je potreben zato, ker je informacije v CPE in gl. pomnilniku shranjena v obliki, ki zunanjemu svetu ni dostopna. Torej je osnovna naloga V/I pretvarjanje informacije iz ene oblike v drugo.
Osnovni način delovanje V/I sistema je prenos podatkov med glavnim pomnilnikom in V/I napravami.

VRSTI PRENOSA:

1. **Programski vhod/izhod**
   Pri tem načinu komunicira z V/I napravo CPE. Vsak podatek se prebere iz pomnilnika v CPE in nato prenese v napravo ali obratno. Prenos je realiziran z ustreznim zaporedjem ukazov (programom), ki poskrbi za vse, pri prenosu potrebne podrobnosti. Pri tem se lahko, vendar ni nujno uporabljajo prekinitve.

   Slabosti: počasnost, ker je za prenos vsakega podatka treba izvršiti več ukazov, poleg tega se z njim zasede CPE, ki jo je koristneje uporabiti za reševanje problema.

2. **Neposreden dostop do pomnilnika**
   Pri tem načinu komunicira V/I naprava neposredno z glavnim pomnilnikom. Vsak podatek se piše iz pomnilnika v napravo ali bere iz naprave v pomnilnik. Prenos je realiziran s posebno enoto, ki ji pravimo DMA krmilnik (direct memory access). Ta je sposoben sam komunicirati z glavnim pomnilnikom, in med prenosom sodelovanje CPE ni potrebno. Ker za prenos ni potrebno prevzemati ukazov, je hitrejši, vendar zaradi DMA krmilnika tudi dražji.
   Programski V/I in neposreden dostop se med seboj ne izključujeta, pri mnogih računalnikih srečamo oba.

KRMILNIK NAPRAVE

Vsaka V/I naprava je priključena preko tako imenovanega krmilnika naprave. Pri preprostih napravah je krmilnik lahko npr.: flip flop ali register. Pri zapletenih napravah kot je magnetni disk, pa si krmilnik lahko predstavljamo kot specializiran računalnik s fiksnim programom, ki je prilagojen lastnostim naprave.

Osnovna naloga krmilnika je, da omogoča prenos podatkov v napravo in iz nje. Zaradi večje ekonomičnosti so zapleteni krmilniki pogosto narejeni tako, da je nanje mogoče priključiti več naprav iste ali podobne vrste.

Registri krmilnikov so lahko v istem naslovnem prostoru kot glavni pomnilnik, lahko pa imamo zanje poseben naslovni prostor. V drugem primeru ima računalnik dva ali celo več ločenih naslovnih prostorov.

Pri današnjih računalnikih uporabljene rešitve lahko razdelimo v naslednje tri skupine:

1. **Pomnilniško preslikan V/I**
   Pri tem načinu so registri krmilnikov v pomnilniškem naslovnem prostoru. Gledano iz CPE so videti enako kot pomnilniške besede. Za branje in pisanje lahko uporabimo vse ukaze za dostop do pomnilnika, posebni V/I ukazi niso potrebni.

2. **Ločen V/I prostor**
   Pri tem načinu so registri krmilnikov v posebnem naslovnem prostoru, ki je ločen od pomnilniškega. Za dostop do registrov so potrebni posebni V/I ukazi. Med izvajanjem teh ukazov CPE aktivira signal, ki pove da se naslavlja V/I naslovni prostor.

3. **Posredno preko V/I procesorjev**
   Tudi pri tej rešitvi so registri krmilnikov v posebnem naslovnem prostoru, vendar ta prostor iz CPE ni neposredno dostopen. Do njega imajo dostop V/I procesorji. Delo z V/I napravami poteka tako, da CPE sporoči svoje zahtave V/I procesorjem, ki poskrbijo za podrobnosti pri izvrševanju prenosa podatkov.

CASE/AMDAHNLOVI EMPIRIČNI PRAVILI (1967)

1. Velikost glavnega pomnilnika v bajtih, mora biti najmanj enaka številu ukazov, ki jih CPE izvede v eni sekundi;
2. Zmogljivost V/I sistema v bitih na sekundo mora biti najmanj enaka številu ukazov, ki jih CPE izvede v eni sekundi.

   Smisel obeh pravil je v tem, da podajata pogoje pri katerih je računalnik uravnotežen. Neuravnotežen računalnik je manj ekonomičen. Pravili veljata predvsem za splošne računalnike, in ne za računalnike, ki so namenjeni za veliko število numeričnih izračunov. Pravili nista absolutni, vendar predstavljata dobro začetno izhodišče za načrtovanje uravnoteženega računalnika. Če upoštevamo da sta pravili nastali pred več kot 30 leti, je njuno današnje upoštevanje največji dokaz za pravilnost.

PRENOSNE POTI / POVEZOVALNE POTI

V vsakem računalniku je CPE, glavni pomnilnik in V/I sistem treba na nek način povezati. Povezave tvorijo tako imenovane prenosne poti, po katerih se prenašajo podatki.

Količino informacije, ki jo je mogoče prenesti po prenosni poti, lahko v principu povečujemo na dva načina:
1. s krajšanjem časa, ki je potreben za en prenos (več prenosov/s)
2. s povečanjem števila hkrati prenesenih bitov v enem prenosu.

POVEZOVALNE STRUKTURE

Pri manjših računalnikih imamo največkrat eno samo prenosno pot. Označujemo jo z besedo **VODILO** (omnibus-za vse). Tu je mišljeno, da uporabljajo vodilo vse nanj priključene naprave. Tam, kjer so priključene enote, imajo linije odcepe. Ti odcepi so bistvena značilnost vsakega vodila.
Če prenosna pot povezuje samo dve enoti, odcepov ni. V tem primeru govorimo o povezavi **TOČKA V TOČKO**, ki povezuje samo dve enoti, in ne o vodilu.

VRSTE SIGNALOV, KI SE PRENAŠAJO PO POVEZOVALNIH POTEH

PODATKOVNI: Po njih se prenašajo podatki. Njihovo število je enako največjemu številu bitov, ki se prenašajo naenkrat. To število imenujemo tudi širina prenosne poti. Ni treba, da se pri vsakem prenosu uporabljajo vsi podatkovni signali-pri 64 bitni širini so možni tudi 8,16 ali 32 bitni prenosi.

NASLOVNI: Z njimi je določen naslov pomnilniške besede ali naslov V/I naprave, na katero se nanaša prenos podatkov. Število naslovnih signalov določa velikost naslovnega prostora.

KONTROLNI: Ti signali določajo smer prenosa (branje/pisanje), število prenesenih bitov, ter začetek in konec prenosa.

Pri nekaterih sistemih se podatkovni in naslovni signali prenašajo po istih linijah, tako da se v enem trenutku prenašajo podatkovni, v drugem pa naslovni signali. Temu pravimo MULTIPLEKSIRANJE.

V vsakem trenutku lahko poteka po prenosni poti največ en prenos. Prenos vedno zahteva in vodi ena enota, ki ji pravimo **gospodar** (master). Gospodar generira naslovne in kontrolne signale, pri pisanju pa tudi podatkovne.
Pri vodilih je lahko gospodarjev več, pri povezavi točka v točko pa samo eden. Če je gospodarjev več, je v danem trenutku aktiven samo eden, vsi ostali so sužnji. Nekatere enote so vedno sužnji (glavni pomnilnik), druge pa so lahko oboje (DMA krmilnik).
Pri večih gospodarjih je potreben nek mehanizem, s pomočjo katerega se gospodarji dogovorijo, kdo bo imel nadzor nad vodilom. Temu mehanizmu pravimo arbitraža.

ZAPOREDJE DOGODKOV PRI PRENOSU

1.) Gospodar generira naslovne signale;
2.) S kontrolnimi signali poda smer prenosa, ter število bitov, ki naj se prenesejo.
Pri branju gospodar sprejme podatkovne signale, pri pisanju jih generira,
3.) Pri vsakem prenosu je treba povedati, kdaj se prične in kdaj zaključi. To se lahko naredi s trajanjem določenih kontrolnih signalov, ali pa s posebnimi dodatnimi kontrolnimi signali. Vse enote, ki so priključene na prenosno pot, opazujejo signal, ki določa začetek prenosa, ko ugotovijo, da se je prenos začel, pričnejo primerjati svoje naslove z naslovom na naslovnih signalih. Enota, ki ugotovi enakost je naslovljena in naredi prenos, ki se od nje zahteva

Glede na način določanja začetka in konca prenosa razlikujemo dve vrsti prenosa:

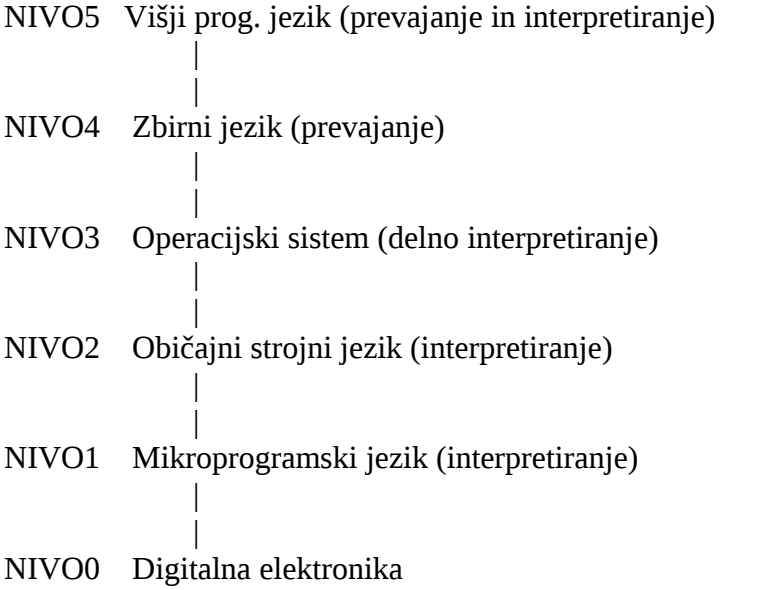
SINHRONSKI: Čas prenosa je vedno enak T, ki predstavlja celo število urinih period. Če imamo na prenosno pot priključene različno hitre enote, mora biti čas T izbran tako da ustreza tudi najpočasnejši enoti. To pomeni, da vse enote delajo s hitrostjo najpočasnejše in v tem je slabost sinhronskega prenosa.

ASINHRONSKI: Čas prenosa ni v naprej določen. Začetek prenosa je enak kot pri sinhronskem, zaključek pa je drugačen. Gospodar ne umakne ali spremeni naslovnih in kontrolnih signalov po preteku vnaprej določenega časa T. To stori šele ko od naslovljene enote dobi potrditveni signal. Trajanje prenosa je torej odvisno od tega, kako hitro bo naslovljena enota aktivirala ta signal. Hitre enota aktivirajo potrditveni signal hitreje in počasnejše počasneje. S tem je odpravljena slabost sinhronskega načina prenosa.

Pri obeh vrstah prenosa mora biti izpolnjenih več časovnih parametrov. Med najpomembnejšimi so naslednji trije:

1) **Vzpostavitveni čas:** enota, ki sprejema informacijo, mora imeti na svojih podatkovnih signalih prisoten stabilen podatek nek minimalen čas pred zaključkom prenosa. Ta minimalen čas imenujemo vzpostavitveni čas in je podan k specifikaciji vsake enote.
2) **Čas dostopa:** pri BRANJU je čas dostopa definiran kot čas, ki preteče od trenutka vzpostavitve naslovnih in kontrolnih signalov, do trenutka da podatek na naslovne signale.
   Pri PISANJU: je čas dostopa definiran kot čas, ki preteče od trenutka vzpostavitve naslovnih, kontrolnih in podatkovnih signalov, do trenutka, ko so ti zapisani v enoto in niso več potrebni.
3) **Držalni čas:** pri nekaterih napravah je zahtevano, da so podatki prisotni še nek minimalen čas pred zaključkom prenosa.

OPIS RAČUNALNIKA KOT ZAPOREDJE NAVIDEZNIH RAČUNALNIKOV

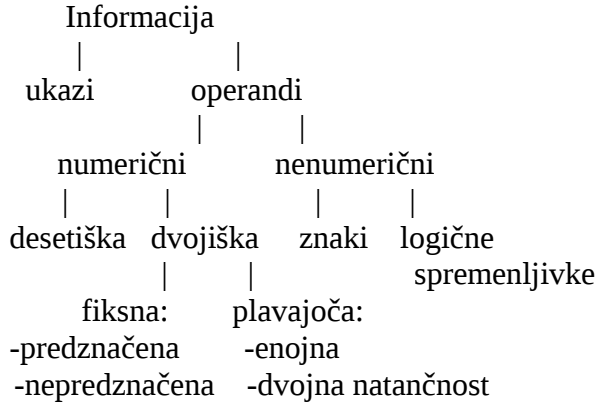


program v jeziku L1 → **prevajalnik** → program v jeziku L2 → izvrševanje

program v
jeziku L1 → **interpreter**

PREVAJANJE: tukaj dobimo kot rezultat prevedeni program v jeziku L1, ki mu pravimo ciljni program. Za program, ki ga prevajamo, uporabljamo izraz izvorni program. Obstoj ciljnega programa, ki nadomesti izvorni program je osnovna značilnost prevajanja.
INTERPRETIRANJE: tukaj ciljni program ne obstaja. Namesto tega se vsak ukaz izvornega programa sproti prevede v L1 in takoj izvrši. Prevod se ne shrani in izvorni program potrebujemo ves čas izvrševanja. Kadar se isti ukazi večkrat izvršujejo, se prevedeni ukazi spet prevedejo in izvršijo.

PREDSTAVITEV INFORMACIJ V RAČUNALNIKU
Informacija, ki je v Neumannovem računalniku neposredno dostopna za delo je shranjena v glavnem pomnilniku in v registrih v CPE. Osnovna enota za shranjevanje informacij je pomnilniška beseda.



Nenumerični operandi:
Skupno ime za črke, številke, vejice, pike, presledke,…
Znaki so predstavljeni z abecedo. Abeceda je predpis, ki določa preslikavo elementov ene množice v elemente druge množice.

Do leta 1964 so proizvajalci uporabljali 6-bitno BCD abecedo (26 velikih črk angleške abecede, 10 številk in 28 posebnih znakov). Ker to ni bilo dovolj, se je začela uporabljati 7-bitna ASCII abeceda, potem je nastala še 8-bitna ASCII abeceda, ki je dala 128 dodatnih znakov. To je takoimenovana razširjena ASCII abeceda.
Najbolj znana je UNICODE 16-bitna abeceda (2^{16}=65.536), dovolj za vse pisave na svetu.

PREDSTAVITEV NUMERIČNIH OPERANDOV

1). FIKSNA VEJICA:
Števila lahko opišemo kot zaporedje številk, ki so ločene z vejico. Levo od vejice je celo Število, desno pa število, ki predstavlja ulomek. Lahko ga posplošimo na tak način:

$193{,}73 = 1 * 10^2 + 9 * 10^1 + 3 * 10^0 + 7 * 10^{-1} + 3 * 10^{-2}$

utež je oblike r^i, kjer je r baza ali radiks številskega sistema. V računalniku se je največ uporabljala baza 2 ali 10. 2 je dvojiška prestavitev in dvojiška aritmetika;
10 je desetiška predstavitev in desetiška aritmetika.

UPORABLJAMO NASLEDNJE 4 NAČINE ZA PREDSTAVITEV ŠTEVIL

1. **PREDZNAK IN VELIKOST:** pri tem načinu najbolj levi bit predstavlja predznak števila (0= pozitivno, 1= negativno) Slabost je, da ima ta predstavitev pozitivno in negativno ničlo, je sicer dobro za množenje in deljenje, vendar sta ti operaciji manj pogosti od seštevanja in odštevanja. Ta predstavitev se danes redko uporablja.
2. **ODMIK:** pri tej predstavitvi se k številu najprej prišteje konstanta, ki zagotovi da je vrednost pozitivna. Pri seštevanju je treba odmik odšteti (ker se oba odmika seštejeta imamo dvojni odmik), pri odštevanju pa prišteti (ker se oba odmika odštejeta, odmika ni več).
3. **ENIŠKI KOMPLEMENT:** tudi tukaj najbolj levi bit predstavlja predznak. Pozitivna števila so predstavljena enako kot pri predznak in velikost. Negativna števila dobimo tako, da ga najprej zapišemo kot pozitivno število in nato invertiramo vse bite vključno s predznakom.
4. **DVOJIŠKI KOMPLEMENT:** ta predstavitev je zelo podobna eniškemu komplementu. Razlika je v tem, da neg. število invertiramo, temu pa prištejemo enko.

2.) PLAVAJOČA VEJICA

Obseg števil, ki jih lahko predstavimo s števili v fiksni vejici je za veliko problemov premajhen. Taka števila običajno pišemo v tako imenovani znanstveni notaciji, ki omogoča njihovo predstavitev z razmeroma majhnim številom številk.

Npr.: 1 000 000 000 000 000 000 lahko pišemo kot

$1*10^{18} = m*r^e$

kjer je m-mantisa; e-eksponent in r-baza ali radiks.

Osnovne lastnosti predstavitve števil v IEEE 754 standardu:
-uporablja bazo r=2
-eksponent je predstavljen v načinu odmik
-mantisa je predstavljena v načinu predznak in velikost
-uporabljena je implicitna predstavitev normalnega bita.
-dva formata : enojna (32-bitni format) in dvojna (64-bitni format) natančnost

UKAZI

Vsak ukaz more vsebovati informacijo o operaciji, ki naj se izvede in informacijo o operandih nad katerimi se bo izvršila operacija.
Ukaz je vedno shranjen v pomnilniški besedi ali v več sosednjih pomnilniških besedah.
Obe vrsti operacij sta običajno opisani z biti v poljih, na katere je ukaz razdeljen.
Polje, ki vsebuje informacijo o operaciji imenujemo OPERACIJSKA KODA. Polja, ki vsebujejo informacijo o operandih, lahko vsebujejo kar operande, še pogosteje pa je v njih informacija o naslovu, na katerem so shranjeni operandi. Pri nekaterih ukazih je informacija o operandih vsebovana kar v operacijski kodi.

Razdelitev ukaza na polja z obema vrstama informacije ter pomen posameznih bitov v teh

poljih imenujemo zgradba ali FORMAT ukaza.

Kakšne formate bo uporabljal nek računalnik je odvisno od:
-dolžine pomnilniške besede (dolžina ukazov v bitih=mnogokratnik dolžine pomn. besede)
-dolžine pomnilniškega naslova
-število operacij (ukazov)
-število registrov v CPE.

OSNOVNE LASTNOSTI UKAZOV (po katerih se procesorji razlikujejo)

1. **NAČINI SHRANJEVANJA OPERANDOV V CPE**

a) **AKUMULATOR:** Pri tem načinu je pomnilnik v CPE narejen iz enega samega registra, ki mu pravimo akumulator. Vanj lahko shranimo en operand. Prednost je preprostost ukazov.
Ker je akumulator en sam, ga v ukazih ni treba eksplicitno navajati. Ukazi so zato kratki. Preprosti so tudi prevajalniki, ker ni potrebno da se odločajo med velikim številom možnosti za shranjevanje operandov.
Slabosti: ker je v CPE prostor za en sam operand, je treba vmesne rezultate pogosto prenašati v glavni pomnilnik. Posledica je počasnejše delovanje in veliko število prenosov med pomnilnikom in CPE.

b) **SKLAD:** Preprosto pot za povečanje pomnilnika v CPE dobimo, če ga naredimo v obliki sklada. Pri skladu je v vsakem trenutku dostopna samo najvišja lokacija, ki jo imenujemo vrh sklada. Če jo beremo, se vse besede pomnilnika pomaknejo navzgor, če v sklad pišemo, se nova vrednost shrani v najvišjo lokacijo, prejšnje pa se pomaknejo za eno mesto navzdol. Ta način delovanja se označuje kot zadnji noter prvi ven.
Ukaz za prenos operanda iz glavnega pomnilnika v sklad je PUSH, za prenos iz sklada pa POP ALI PULL.
Vse prednosti, ki veljajo za akumulator, veljajo tudi za sklad, ker pa je v skladu prostora za več operandov, je ta boljši.

c) **MNOŽICA REGISTROV:** Najbolj splošno rešitev za pomnilnik v CPE dobimo, če ga naredimo kot množico registrov, do katerih je možen dostop brez omejitev. Vsak register ima svoj naslov. Pri današnjih računalnikih lahko govorimo o množici registrov, ko se njihovo število giblje od 8 do nekaj 100.

Glede na svobodo pri uporabi registrov v množici, razlikujemo danes dve rešitvi:
-vsi registri so ekvivalentni

-množica registrov je razdeljena na dve skupini:
1. ena se uporablja za aritmetično-logične operande
2. druga za računanje z naslovi

3. **LOKACIJE OPERANDOV IN NAČINI NASLAVLJANJA**

REGISTRSKI OPERAND
V majhnem hitrem pomnilniku, ki se nahaja v CPE. Največkrat je to množica programsko dostopnih registrov, lahko pa tudi kot sklad ali kot akumulator.

POMNILNIŠKI OPERAND
V eni ali več sosednjih pomnilniških besedah glavnega pomnilnika. Ker je prostor glavnega pomnilnika velik, obstaja veliko načinov za podajanje lokacije operandov v njem.

V/I OPERAND
V enem od registrov krmilnika V/I naprave ali v V/I procesorju.

Poznamo 3 vrste računalnikov glede na lokacijo operandov:

1. **REGISTRSKO-REGISTRSKI (load/store računalniki)**
   Pri njih so vsi operandi ALE ukazov v registrih CPE.

   Prednosti: preprosti in kratki ukazi, ki jih je enostavno uporabljati, kratek čas dostopanja, čas, v katerem se izvršijo ALE ukazi je vedno enak.

   Slabosti: za rešitev istega problema je potrebnih več ukazov, kot pri računalnikih, ki imajo v ALE ukazih lahko tudi operande v pomnilniku.

   Sem spadajo vsi RISC računalniki in vsi 3-operandni računalniki.

2. **REGISTRSKO-POMNILNIŠKI RAČUNALNIKI**
   Pri njih je eden od operandov lahko v pomnilniku ali v registru, drugi pa so vedno v registru. Pri reg.-pomn. računalniku je mogoče v ALE ukazih uporabljati pomnilniške operande, ne da bi jih bilo pred tem treba z ukazom LOAD prenesti v register.
   Slabosti: ukazi so daljši, zapleteni (različna naslavljanja), čas izvajanja ALE ukaza je odvisen od tega, kje se ukaz nahaja v pomnilniku.
   Večina 2- operandnih računalnikov spada v to skupino.

3. **POMNILNIŠKO-POMNILNIŠKI RAČUNALNIKI**
   Pri njih je lahko vsak operand v pomnilniku ali v registru. Taki računalniki so bolj splošni in omogočajo veliko število rešitev pri programiranju istega problema.
   Slabosti: ukazi so zapleteni, čas izvajanja programa je odvisen od lokacije operandov.
   V to skupino spadajo CISC računalniki.

CISC/RISC PROCESORJI

CISC-complex instruction set computer
RISC-reduced instruction set computer

Razlogi za povečevanje števila ukazov → CISC

1. Semantični prepad: z njim označujemo razliko med računalnikom kot ga vidi

programer v višjem programskem jeziku in med tistim, kar vidi programer v strojnem jeziku. Proizvajalci so se na te problem odzvali s povečevanjem števila operacij in načinov naslavljanja, ter z vključevanjem nekaterih ukazov iz programskega jezika. Posledica tez poskusov je povečevanje števila ukazov.

2. Mikroprogramiranje: poskusi zmanjševanja semantičnega prepada s povečevanjem števila ukazov so se časovno ujeli z razširjenostjo mikroprogramiranja. Na mikroprogramskih računalnikih je dodajanje novih ukazov preprosto in pogosto zadošča, da se v CPE poveča kontrolni pomnilnik, v katerem so shranjeni mikroprogrami. Če je povečanje veliko, ima to za posledico nekoliko počasnejše izvajanje vseh ukazov.

3. Razmerje med hitrostjo CPE-glavni pomnilnik. Hitrost dostopa do informacije v CPE je bila v 1960 in 1970 tipično več kot 10-krat višja kot pri dostopu do glavnega pomnilnika. Jemanje ukazov iz glavnega pomnilnika je bilo počasno. Dodajanje kompleksnih ukazov je bilo smiselno in ni imelo negativnih posledic za hitrost delovanja.

Razlogi za zmanjševanje števila ukazov→RISC

1. Težave pri uporabi kompleksnih ukazov. W. Wulf je leta 1981 analiziral vzroke za Težave pri uporabi kompleksnih ukazov in ugotovil, da prevelika semantična vsebina ukaza povzroči, da je ukaz uporaben samo v majhnem številu primerov. Ta pojav je imenoval semantični trk. Taki ukazi so pogosto premočni-naredijo več kot je običajno potrebno, posledica je nepotrebno delo in počasen ukaz.
2. Razmerje hitrosti med hitrostjo glavnega pomnilnika in CPE. Pri CISC so v kontrolnem pomnilniku zamrznjena zaporedja mikroukazov, za katere se je pri razvoju odločil proizvajalec. Pri RISC računalnikih pa to zaporedje ukazov vsakič sproti tvori prevajalnik, ki se prilagaja programu.
3. Uvajanje paralelizma v CPE: Razvoj mikroelektronskih vezij omogoča, da se v CPE vse več dela izvaja paralelno. Glavni način, kako to dosežemo je cevovodno procesiranje.

DEFINICIJA RISC PROCESORJA (arhitekture)

1. Večina ukazov se izvrši v eni urini periodi. Izjema so množenje, deljenje, operacije v plavajoči vejici.
2. Registrsko-registrski računalnik (load/store).Ta je nujna, če želimo izpolniti zahtevo po Izvrševanju v eni urini periodi.
3. Ukazi so realizirani s trdo ožičeno logiko in ne mikroprogramsko. Tudi ta zahteva je nujna za izvrševanje ukazov v eni urini periodi.
4. Malo ukazov, malo načinov naslavljanja. To omogoča, da je njihovo dekodiranje in izvrševanje hitro in preprosto.
5. Vsi ukazi imajo isto dolžino. Tudi to omogoča preprosto dekodiranje.
6. Dobri prevajalniki. Prevajalniki morajo biti narejeni tako, da upoštevajo zgradbo CPE in po potrebi zamenjujejo vrstni red operacij oziroma izločujejo nepotrebne.