Osnove računalniških sistemov

Knjiga: D.Kodek - Arhitektura Računalniških sistemov

Opis, zgradba in delovanje računalnika

Izraz arhitektura izhaja od IBM (1964) – računalnik kot ga vidi programer na nivoju strojnega jezika – je jezik, ki ga procesor izvaja

- register
- pomnilnik
- povezave med regisri
- registri, vhodi, izhodi enot
- logična enota

Narava računanja

- Zgodovina računalnikov
- Teoretični del računalnika
- Sestavni deli, lastnosti
- Kako je predstavljena informacija
- Ukazi

RISC – nabor ukazov je reduciran na minimum (na procesorju)

1.) **Razlogi za strojno računanje**
Umsko računanje: - velike napake
- počasnost
2.) **Povezava med ročnim in strojnim računanjem**
Ročno: papir služi za shranjevanje informacij

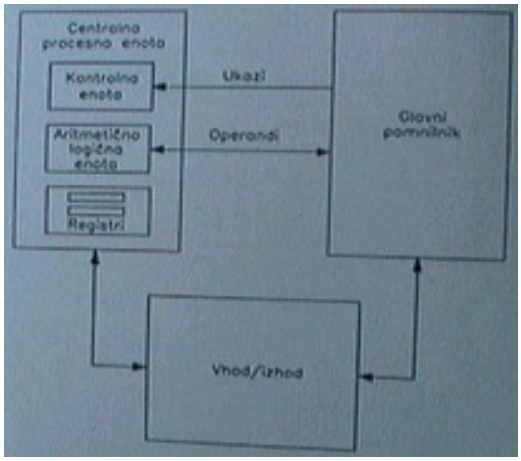
- navodilo za računanje (= ukazi)
- števila, simboli (= operandi)

Računalniki : zaporedje ukazov sestavlja program
podatki : operandi

Ročno računanje: možgani

- kontrolna funkcija (prevzem navodil ali ukazov)
- izvršilna funkcija (izvajanje ukazov ali navodil)

Zgradba današnjega računalnika



CPE – centralno procesna enota
ALE – aritmetično logična enota

Računanje in izračunljivost

Rezultat
⇓
z = f (x) – vhodni podatek
⇑
predpis
Algoritem (obstaja) ⇒ funkcija je izračunljiva
⇒ do 1931 določljivi problemi
⇒ funkcija neizračunljiva
nedoločljivi problemi

Pomnilnik
M = 2^{20}
K = 2^{10} = 1024

M = 10^6
K = 10^3

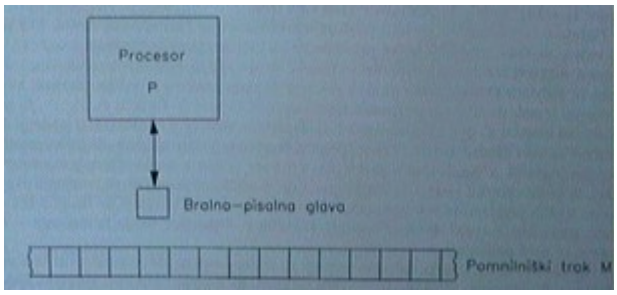
Teoretični način računanja

- Lambda način (Church)

- Produkcijski sistem (Post)
- Turingov stroj (A. Turing)

CHURCH-TURINGOVA HIPOTEZA : nek problem je izračunljiv, če ga je mogoče izračunati na Turingovem stroju na končnem številu korakov.

TURINGOV STROJ : matematični model, po katerem se da izračunati, kar se izračunati da.



Teoretični model računanja, Von Newmannov rač. Model je ekvivalenten Turingovemu stroju.

Omejitve računalnikov

- neizračunljivi problemi (problem ustavljanja)
- neobvladljivi problemi (končni pomnilnik, čas računanja)
- CPE 10^8 ukazov /s
- Program 10^{18} ukazov

Funkcija	n = 10	n = 1000
5 x n	50	5000
n!	10! = 3.6 x 10^6	nepredstavljivo veliko

Fizikalne omejitve
Noben računalnik ne more v 1s obdelati več kot 2 x 10^{50} bitov informacije/kg mase < 10^{12}

Pri šahu je možnih potez 10^{120}

m = 6 x 10^{24} kg (masa zemlje)
čas od nastanka vesolja (1.5 x 10^{10} let)

Dosedanji razvoj strojev za računanje

DIGITALNI : (prst, število) pri računanju s števili mora biti točno določeno število stanj (0.1 : 0,1,2,3, … , 9 (danes)
ANALOGNI : računanje s posebnimi fiz. veličinami (do leta 1970)

Obdobje mehanike

- **Prvi kalkulatorji** +,-,*,/ (17.stol.)
- **Stroji C. BABBAGEA** (1792 – 1871)
- diferenčni stroj
- diferenčni stroj št. 2

- analitični stroj (računanje poljubnega problema)

- **Elektomehanski stroji** luknjaste kartice Hollerith je ustanovil podjetje Tabulating Machine Co 1896 ⇒ 1924 Internacional Business Machine Co IBM
- K. Zuse (1910 – 1996) prvi delujoči stroj po Babbageju 1934 Z1, Z2, Z3, Z23
- **Elektronski računalniki** ENIAC programiranje je ročno s preklapljanjem stikal
- ENIAC (30m x 3m x 1m, 18000 elektronk, 1500 relejev, 140KW, ni bilo mogoče shraniti programa) ⇒ ideja o shranjenem programu
- **Elektronski računalnik s shranjenim programom**
- John von Neumann
- EDVAC program shranjen v pomnilniku, lahko spreminjanje programa (kot vhodni podatek), rač. spreminja ukaze programa, ki ga izvaja. Program vzame program 1 in rač. program in naredi program 2.
- EDVAC 1951 ima 1024 16 bitnih besed, 20k pomožni pomnilnik, Ukazi : A1 A2 A3 A4 OP; A1, A2 naslova obeh operandov; A3 naslov za shranitev rezultata; A4 naslov naslednje operacije; OP vrsta operacije.
- IAS 1951 Ukaz : OP A; OP vrsta operacije; A naslov 1. operanda; 2. operand v akumulatorju; rezultat v akumulatorju.

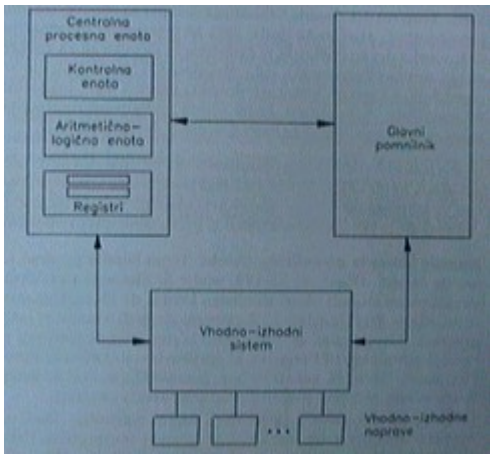
Dogovor : Ukazi v pomnilniku so v zaporedju po naraščajočih naslovih.

Osnovni principi delovanja računalnikov

Von Neumannov računalniški model
- teoretični model
- resnični stroj

Von Neumannov stroj
- sestavljen je iz treh delov : CPE, glavni pomnilnik, V/I enota
- program shranjen v glavnem pomnilniku
- ukazi tega programa se izvajajo eden za drugim

Primer tipičnegaVon Neumanovega računalnika:

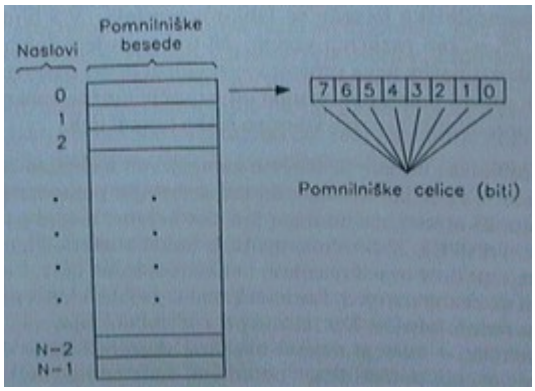


1.) **CPE** (iz glavnega pomnilnika jemlje ukaze injih izvaja)

- kontrolna enota (skrbi za prevzemanje ukaza iz glavnega pomnilnika in začne z operacijo)
- ALE – aritmetično logična enota (izvršuje operacije)
- Registri (ena ali več med seboj povezanih celic)
- Programsko dostopni registri
- Programsko nedostopni registri

2.) Glavni pomnilnik

Služi za shranjevanje ukazov in operandov



3.) Vhodno izhodni sistem

Delovanje Von Neumannovega računalnika

Ukaze jemlje enega za drugim

Ukaz ima dva koraka (strojni ukazi)

- jemlje iz pomnilnika (fetch) – prevzem ukaza, branje ukaza

Programski števec (PC – program counter) je register v katerem se vedno nahaja nabor naslednjega ukaza.

PC ⇐ PC +1

- izvrševanje v 1. koraku preteklega ukaza (execute)

Prekinitev (interrupt)

CPE ne prevzame naslednjega programa ampak ukaz nekega drugega programa, ko se ta program izvrši do konca, nadaljuje tam kjer se je prej ustavilo (prekinitveni servisni program).

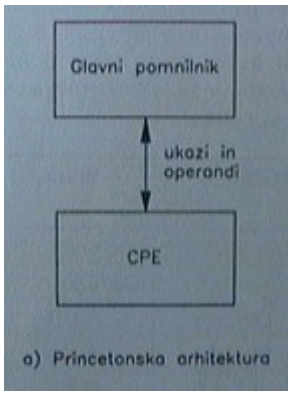
GLAVNI POMNILNIK v von Neumannovem računalniku

Pomnilniška hierarhija

- predpomnilnik
- glavni pomnilnik
- pomožni pomnilnik (navidezni)

Promet – prenosi informacij iz CPE ⇔ glavni pomnilnik

Von Neumannovo ozko grlo



Dolžina naslova: število bitov s katerimi je naslov podan

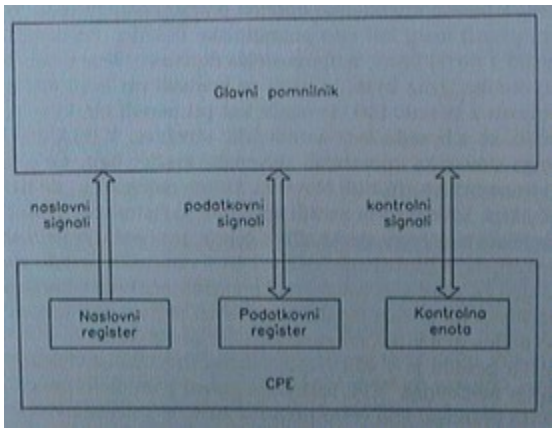
Dolžino določa velikost naslovnega prostora.

Pentium
32- bitni naslov ⇒ naslovni prostor 2^{32} različnih naslovov $2^2 \times 2^{30} = 46$ naslovov

Dolžina pomnilniške besede : danes 8 bitov = 1 byte = 1 bajt
1 Mbajt = 2^{20} bajtov

Dolžina naslova	Naslovni prostor
10 bitov	$2^{10} = 1024 = 1K$ (kilo) $\neq 10^3$
20 bitov	$2^{20} = 1048576 = 1M$ (mega) $\neq 10^6$
30 bitov	$2^{30} = 1073741824 = 1G$ (giga) $\neq 10^9$

Prenos med glavnim pomnilnikom in CPE

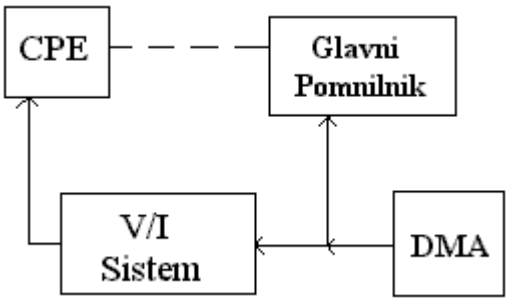


Procesor ne zazna razlike med ukazi in operandi

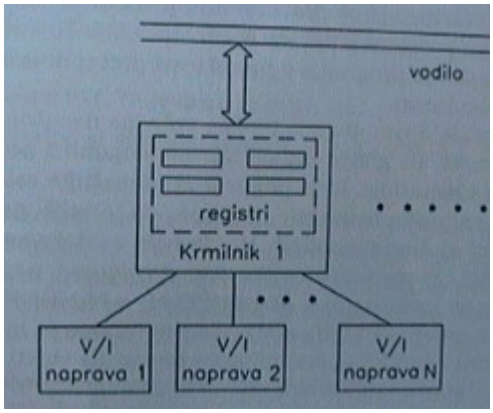
VHOD / IZHOD v von Neumannovem računalniku

Prenos V/I naprava ⇔ pomnilnik

- programiran vhod/izhod (CPE izvaja program za prenos)
- neposreden dostop do pomnilnika (DMA) – direct memory access



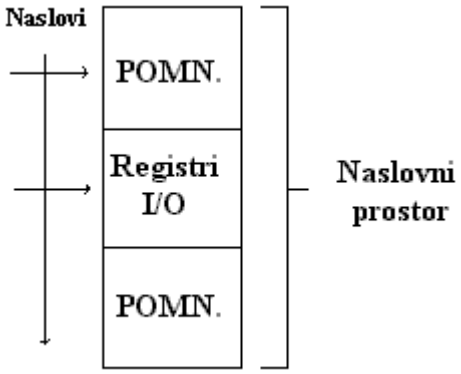
Prenos z V/I procesorji



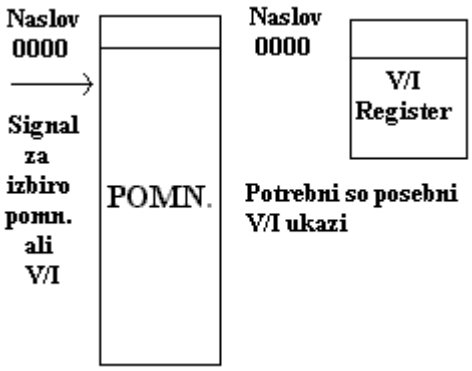
- Registri V/I krmilnikov vsebujejo isti naslovni prostor kot pomnilnik
- V/I registri uporabljajo ločen registerski prostor

Definicija registerskih naslovov (tri variante):

- Pomnilniško preslikan vhod/izhod (memory mapped I/O)



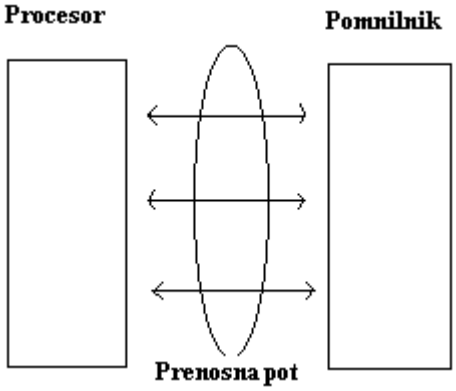
- Ločen V/I naslovni prostor



- Posredno naslavljanje preko V/I procesorjev

Dostop do V/I prostora imajo le V/I procesorji

Prenosne poti



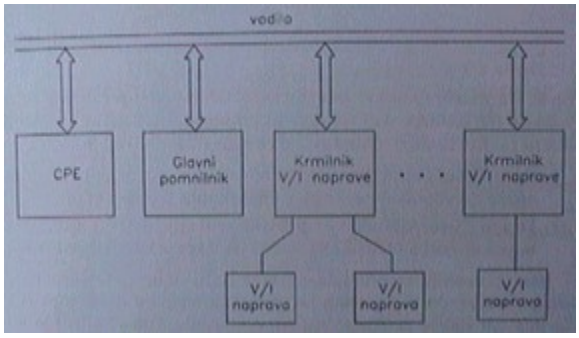
Povečevanje količine prenesenih informacij:

- hiterjši prenos – hitrejša elektronika
- večja širina prenosnih poti
- povezovalne strukture – več povezovalnih poti

Ena sama povezovalna pot (pot povezuje vse naprave CPE, pomnilnik)

Vodilo (bus, omnibus):

- VNIBUS
- AT BUS
- PCI



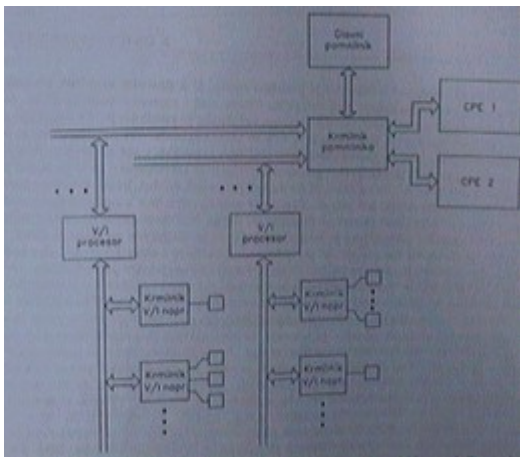
Vrste signalov

IBM 360/370

1.) Podatkovni signali
    64 povezovalnih linij – širina prevozne poti je 64 bitov, 8 bitov, 16 bitov, 32 bitov, 64 bitov
2.) Naslovni signali
    Določijo naslov V/I naprave
    Število naslovnih signalov ⇒ velikost naslovnega prostora
3.) Kontrolni signali
    Določijo smer prenosa, število prenesenih bitov ter začetek in konec prenosa

Multipleksiranje

Gospodar prenosa (tisti, ki vodi prenos pri enem vodilu). Glavni pomnilnik to ne more biti, lahko pa sta to krmilnik V/I naprave ali DMA krmilnik, ki sta lahko tudi sužnja. Obstajati mora nek mehanizem, ki razsodi , kdo bo imel nadzor nad vodilom (gospodar).



Prenos:
- branje
- pisanje

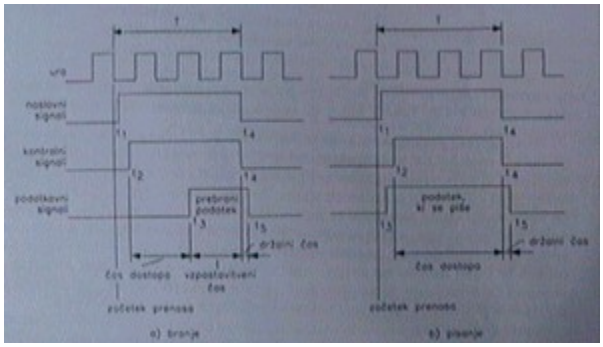
Zaporedje dogodkov

1.) Gospodar
    - ⇨ naslovni signali
    - ⇨ izbira naslovnega prostora (če ima pomnilnik preslikan V/I)

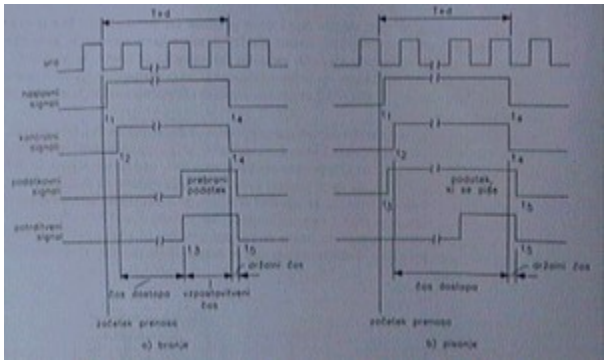
2.) Gospodar
    - ⇨ smeri prenosa s kontrolnimi signali
    - ⇨ branje / pisanje

3.) Gospodar – določi začetek in konec prenosa (kontrolni signal)

Sinhroni prenos



Asinhroni prenos



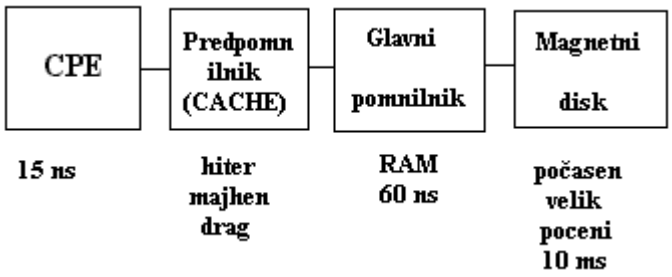
Kapaciteta prenosne poti

B (bitov/s)
Kapaciteta B = 32 (bitov) x 1 / 50 x 10^{-9}s = 640 x 10^{6} bitov/s = 640 Mbitov/s = 80 Mbajtov/s (1 bajt = 8 bitov) = 80 MB/s

Čas prenosa T = 50 ns – sinhroni prenos

Lokalnost pomnilniških dostopov

Značilnost von Neumannovih računalnikov
(pomnilniška hierarhija)



CPE pomnilnik je velik in hiter za zmerno ceno

Navidezni pomnilnik

Prostorska lokalnost
A(i)
A(i + n)
A(i + n) – A(i) = majhno
Verjetnost

Časovna lokalnost
T – v tem času tvori program neke naslove v pomnilniku. Zelo verjetno je, da jih je že ali jih še bo. Vzrok je v zankah.

AHMDALOV ZAKON
Če za faktor v pohitrimo vse operacije lahko s pomočjo formule izračunamo, koliko hitreje se bodo operacije hitreje izvajale.

N – faktor pohitritve
F – N del operacij ostane enako hiter
(1 – f) ti pohitri N krat
S(N) povečuje hitrost celotnega sistema

$S(N) = 1 / f + (1 - f) / N = N / 1 + (N – 1) \times f$

Primer:

F = 10 % ostane enako hiter
N = ∞

$S(N) = 1 / 0.1 + 0.9 / \infty = 1 / 0.1 = 10$ krat

Pohitritev je odvisna od tistega dela, ki ga ne moremo pohitriti.

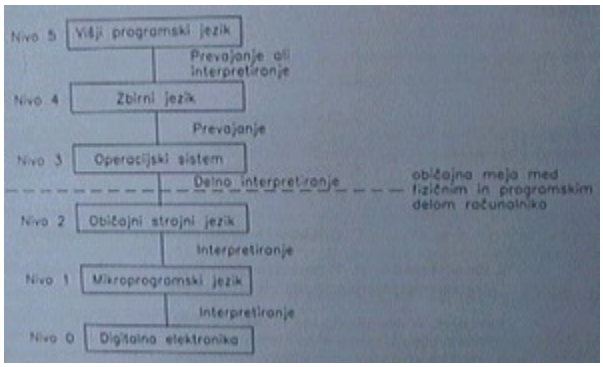
CASE AHMDALOVI PRAVILI
1.) Velikost glavnega pomnilnika v bajtih mora biti najmanj taka, kot število ukazov na sekundo.

Velikost glavnega pomnilnika (bajti) ≥ številu ukazov / s

2.) Velikost procesiranih bitov mora biti najmanj enako številu,ukazov, ki jih CPE izvede v 1 s.
Zmogljivost V/I (bitov / s) ≥ številu ukazov / s

Računalnik kot zaporedje navideznih računalnikov

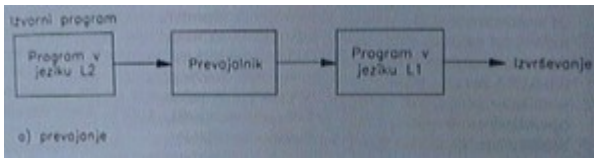
Porazdelitev računalnikov glede na to kateri jezik uporabljajo:



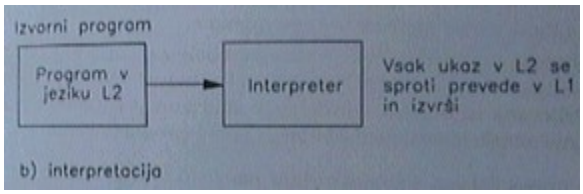
PREHOD NA NIŽJI NIVO

- prevajanje
- interpretiranje

1.) Prevajanje : opravlja nek program



2.) Interpretiranje



Fizična in programska oprema

⇓ ⇓

Hardware Software

Predstavitev informacije in aritmetika

Informacija je v računalniku shranjena v pomnilniku ali registru z 0 ali 1. Osnovna enota za shranjevanje informacije je pomnilniška beseda – najmanjše število bitov, ki imajo svoj naslov.
Porazdelitev informacije v osnovne tipe:

INFORMACIJA
- UKAZI
- OPERANDI (numerični, nenumerični)
  - ŠTEVILA
    - desetiška
    - fiksna vejica
    - plavajoča vejica
  - NENUMERIČNI OPERANDI (znaki, logične spremenljivke)

1.) Predstavitev numeričnih operandov:
Znak (character)
Abeceda
Do 1964 : se je uporabljala 6-bitna abeceda 64 znakov
26 črk angleške abecede (samo velike)
10 številk
28 posebnih znakov
64 zankov = 2^6

Danes se uporablja 8-bitna abeceda.
ASCII (american standard code for information interchange)
7-bitna abeceda (bit 7 = 0) 2^7 = 128 kombinacij
Izvedenka 8-btna LATIN 2

EBCDIC – Extended binary coded decimal interchange code

Dolžina pomnilniške besede je danes 8-bitov = 1 bajt

ASCII Znak	Desetiško število	EBCDIC
0	0110000	11110000
1	0110001	11110001
2	0110010	11110010
.		
.		
.		
9	0111001	11111001

ASCII
A 01000001
B 01000010
.
.
.
Z 01011010

2.) Predstavitev numeričnih operandov
- fiksna vejica (fixed point)
- plavajoča vejica (floating point)

<u>Predstavitev s fiksno vejico</u>:

6 1 2 , 3 5 - ulomek 35 / 100 = $6x10^2+1x10^1+2x10^0+3x10^{-1}+5x10^{-2}$

10^2 10^1 10^0 10^{-1} 10^{-2}

Decimalna vejica je že v naprej določena (fiksno) POZICIJSKA NOTACIJA

Posplošitev

r^i ⇐ utež
⇑
baza

V računalnikih r = 2 (binarni, dvojiški sistem)
r = 10 (desetiški 0-9 BCD predstavitev)

$10100111 = 1 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$

Celo število (integer)
Zapis s fiksno vejico pri čemer je decimalna vejica desno od zadnjega bita

Za decimalna števila uporabljamo drugačno obliko

$1101{,}01 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} = 13 + ¼ = 13{,}25$

$$V = \sum_{I=-m}^{n-1} bi \times 2^i;\ bi = 0 \text{ ali } 1$$

$$V = \sum_{I=-2}^{3} bi \times 2^i$$

Fiksna vejica ⇒ celo število (vejica desno od najlažjega bita)

<u>Uvedba predznaka</u>
- predznak in velikost
- predstavitev z odmikom
- eniški komplement
- dvojiški komplement

n – bitov dolgo zaporedje

$b = b_{n-1}, b_{n-2}, \ldots, b_2, b_1, b_0$

Primer : 3 bitno zaporedje
$b = b_2\ b_1\ b_0$

1.) Predznak in velikost

$b_{n-1}, b_{n-2}, \ldots, b_2, b_1, b_0$

predznak velikost

to zaporedje predstavlja $V = (-1)^{b_{n-1}} \sum_{i=0} b_i 2^i$

$b_{n-1} = 0\ (-1)^0 = 1$
$b_{n-1} = 1\ (-1)^1 = -1$
$b_{n-1} = 0 \Rightarrow$ + (pozitivno število)

$b_{n-1} = 1 \Rightarrow$ - (negativno število)

Primer za n = 3
011 +3
010 +2
001 +1
000 +0
100 -0
101 -1
110 -2
111 -3

2.) Predstavitev z odmikom
K številu (poz./neg.) prištejemo konstanto $\Rightarrow$ rezultat.
Samo pozitivno število.
Konstanta: Odmik (BIAS)
Pri n-bitnem zaporedju je odmik 2^{n-1}

$$V = \sum_{I=0}^{n-1} bi \times 2^i - 2^{n-1}$$
$\Downarrow$
nova vrednost

Primer n = 3

+3 011 najbolj pozitivno
+2
+1
0
-1
-2
-3
-4 100

Število $+2^{n-1}$ število + 4
Poz. / neg.

Odmik			Dvojiško
+3	+4	7	111
+2	+4	6	110
+1	+4	5	101
0	+4	4	100
-1	+4	3	011
-2	+4	2	010
-3	+4	1	001
-4	+4	0	000

3.) Eniški komplement

Pozitivno število enako kot pri Predznak in velikost.
Negativno število ⇒ predstavitev s komplementom

Negativno število ⇐ pozitivno število in invertiramo vse bite

Primer (n = 3)

+3	011
+2	010
+1	001
0	000
0	111
-1	110
-2	101
-3	100

$$V = \sum_{I=0}^{n-1} b_i 2^i - b_{n-1}(2^{n-1})$$

4.) Dvojiški komplement (število odštejemo od 2^n) aritmetično

dvojiški komplement – invertiramo vse bite in prištejemo 1
Število 01011010 ⇒ 10100101 + 1 ⇒ 10100110

1. začnemo skrajno desno
2. prepišemo vse bite vključni do prve enice

Primer (n = 3)

+3	011
+2	010
+1	001
0	000
-1	111
-2	110
-3	101
-4	100

-4 ≤ obseg ≤ +3
n = 3

$$V = \sum_{I=0}^{n-1} b_i 2^i - b_{n-1}(2^n)$$

Primer 1
+3 -1
+3 + (-1)
⇑
(+1s komponentom)

011
+111 + (-1)

Primer 2
+1 -2

+1 + (-2)
⇑
komp.
001 +2
+110

010 = +2 111 = -1

---

n = 8 – 8 bitov
$-2^{n-1} \leq$ število $\leq 2^{n-1} - 1$
$-128 \leq$ število $\leq +127$

poz. Stran $2^{3-1} - 1 = 3$
neg. stran $2^{3-1} = -4$

Prekoračitev (overflow) je pojav, če rezultat pri računanju pade izven dvojiškega komplementa.

Pri n = 8 > 127

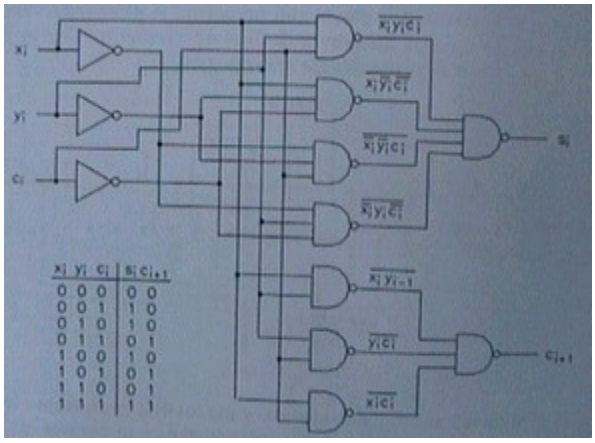
Prenos (carry) velja za računanje s pozitivnim številom.

N = 8, če je rezultat ⇒ 255 (1111 1111)
$0 \leq$ poz. Števila $\leq 2^{n-1}$

Binarni seštevalnik

Binarno seštevanje
0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0

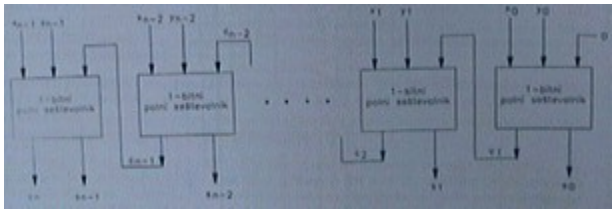
Polni 1-bitni seštevalnik – vhod za prenos s predhodnjega mesta



x_i	y_i	C_{i-1}	S_i	C_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0

1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Polni 1-bitni seštevalnik – vhod za prenos s predhodnjega mesta



Vključitev ALE v računalnik

Problemi: 1. prekoračitev (overflow)

- ALE postavi bit in signalizira prekoračitev
- ALE sproži past (prekinitev)

2. dolžina produkta (kvocienta)
množenje dveh n-bitnih števil ⇒ rezultat dolžine 2n

Znanstvena notacija

$3.200.000 = 3.2 \times 10^6$
$= 0.32 \times 10^7$
$= 32 \times 10^5$
$= 3200 \times 10^3$

Položaj vejic se spreminja, odvisen je od eksponenta
Predstavitev v plavajoči vejici (floating point)

Splošno : $m \times r^e$
m = mantisa
r = osnova (baza)
e = eksponent

3.2×10^6
m = 3.2
r = 10
e = 6

V računalniku :
$m \times r^e$
r = 2
m, e – celi števili s predznakom (fiksna vejica s predznakom)

1. natančnost ⇒ odvisna od mantise
2. obseg(največje, najmanjše) ⇒ eksponent

n = 8 dvoj. $-128 \leq x \leq +127$

Število 3.215.300

Normalizirana mantisa
Mantisa na levi strani nima ničel.

M = 4 mestno Prva številka mantise različna od 0.
3.215×10^6
0.321×10^7
0.003×10^9
0.3000

Položaj vejice:
- vejica je tik pred prvo številko mantise 0.32×10^7
- vejica je tik za prvo številko mantise 3.2×10^6
- vejica je desno od zadnje številke 32×10^5

Mantisa = predznak in velikost

Predstavitev ničle

m = 0
$V = m \times r^e = 0 \times 2^e$

Zaželjeno je, da je eksponent e najbolj negativno število.
e = 0 – najboljnegativno število pri predstavitvi z odmikom (se najbolj uporablja) m = 0, e = 0

1. Mantisa – predznak in velikost
2. Eksponent – predstavitev z odmikom

Standard za število v plavajoči vejici je TEEE 75A

Osnovne lastnosti standarda
Baza r = 2
Eksponent (e) predstavitev z odmikom
Mantisa (m) predznak in velikost
Implicitna predstavitev normalnega bita
1.0010×2^e

m = 0010 $\Rightarrow$ vrednost m = 1.0010

Vejica desno od normalnega bita
Dva formata:
- 32 bitni enojna natančnost
- 64 bitni dvojna natančnost

32 bitni format

$V = (-1) \times (1,m) \times 2^{e - 127}$
E = 0 dejanski eksponent –127
E = 127 0
E = 255 128

$\pm 1.18 * 10^{-38}$ do $\pm 3.4 * 10^{+38}$

64 bitni format
$\pm 2.22 * 10^{-308}$ do $\pm 1.80 * 10^{+308}$
Predstavitev števil v standardu

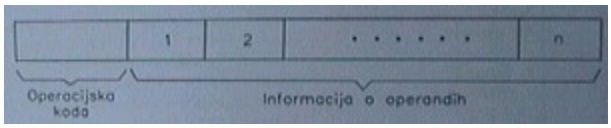
- normalizirano število
- ničla (+0, -0) m = 0, E = 0
- ∞ (±∞ E = 1, m = 0)
- NaN (not a number) E = 1, m ≠ 0)

FPU (floating point unit)
ALE za računanje
V plavajočo vejico

Ukazi

Pri pogovoru o ukazih mislimo na strojne ukaze.
Primer ukaza :
Seštej vsebini dveh registrov rezultat pa shrani v tretji register.
ADD R1, R2, R3

Ukaz vsebuje dve vrsti informacije:
Informacijo v informaciji (ADD)
Informacijo o operandih



Operacija: seštevanje
Ukaz: seštej dve celi števili
seštej dve števili v plavajoči vejici

Informacija o operandih je podana:

- explicitno (naslov registra, naslov plavajoče vejice)
- implicitno (skrito ⇒ ukaz deluje po naprej znanem pravilu)

Osnovne lastnosti ukazov:

- načini shranjevanja operandov v CPE
- število eksplicitnih ukazov v operandu
- lokacija operandov
- vrste operacij in operacije
- vrste in dolžine operandov

Načini shranjevanja operandov v CPE

Programsko dostopni registri (pomnilnik v samem CPE)
Prednosti če imamo podatke v CPE:

- večja hitrost
- krajši ukazi

Načini shranjevanja podatkov v CPE:

- en sam register v CPE (akumulator) najstarejši računalniki
- sklad v CPE (stack)
- množica registrov v CPE (register set) = 8 do 100 registrov, vsak ima svoj naslov
  a.) vsi registri so enakovredni (ekvivalentni) – splošno namenski registri
  b.) registri razdeljeni v dva dela:
  ALE operacije
  Za računanje z naslovi

Število eksplicitnih operandov

Današnji računalniki imajo največ 3 eksplicitne operande v ukazu.

ADDP6, OP1, OP2, OP3, OP4, OP5, OP6

V ukazu m operandov
⇓
m – operandi
m – naslovi računalnika

Porazdelitev računalnikov glede na število eksplicitnih operandov v ukazu
-3 +1 operandni računalniki (naslovni) – EDVAC (se ne uporablja več)
Simbolično: OP3 ⇐ OP2 + OP1
PC ⇐ PC + 1 vrstni red ukazov je v naprej določen

3 operandni računalniki (današnji INTEL)
OP3 ⇐ OP2 + OP1
PC ⇐ PC + 1 vrstni red ukazov je naprej določen
(implicitno s pravilom PC ⇐ PC + 1)

2 opreandni računalniki (so trenutno še v uporabi)
OP2 ⇐ OP2 + OP1
PC ⇐ PC + 1

Enooperandni računalniki – v CPE en sam register (akumulator)
AC ⇐ AC + OP
PC ⇐ PC + 1

Brezoperandni računalniki (skladovni računalniki)
$SKLAD_{VRH} \Leftarrow SKLAD_{VRH} + SKLAD_{VRH-1}$
PC ⇐ PC + 1

Lokacije operandov in načini naslavljanja

- kje so operandi
- kako do njih (načini naslavljanja)

Kje? – v CPE : - akumulator
- sklad
- množica registrov
- v eni ali več besed glavnega pomnilnika
- v registru krmilnika V/I naprave (ali V/I procesorja)

Delitev rač. glede na lokacijo operandov (velja za ALE ukaze):
Registersko – registerski računalniki (LOAD/STORE računalniki) – vsi 3 operandni računalniki in vsi RISC računalniki
Registersko – pomnilniški računalniki – večinoma 2 operandni računalniki
Pomnilniško – pomnilniški računalniki

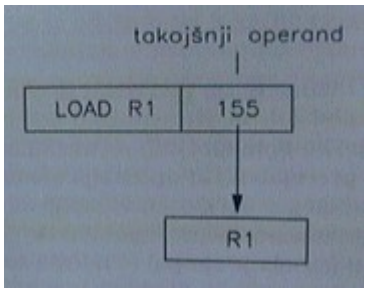
Število operandov v pomnilniku (tipični ALE ukaz)	Število eksplicitnih operacij v ukazu (tipični ALE ukaz)	Tipični predstavniki
0 – (reg. –reg.)	2	IBM PC, RT
	3	SPARC
		MIPS (SG)
		POWER PC
1 – (reg. – pom.)	2	IBM 370
	3	MOTOROLA 68040
2 – (reg. – pom.)	2	IBM 370 (del ukazov)
	3	CDC, PDP - 11
3 – (pom. – pom.)	3	VAX

Kako je podana informacija (v ukazu) – kje se nahaja?

- v registrih (registrski operandi)
- v pomnilniku (pomnilniški operandi)

Načini naslavljanja:
- takojšnje naslavljanje (immediate adressing)
- neposredno naslavljanje (direct adressing)
- posredno naslavljanje (indirect adressing)

Takojšnje naslavljanje:
Operand je podan v ukazu (vrednost operanda)
⇒ takojšnji operand – literal
Primer:
LOAD : ukaz za prenos enega podatka iz enega mesta na drugo(običajno pomnilnik ⇒ register)
Izvor : pomnilnik
Ponor : register

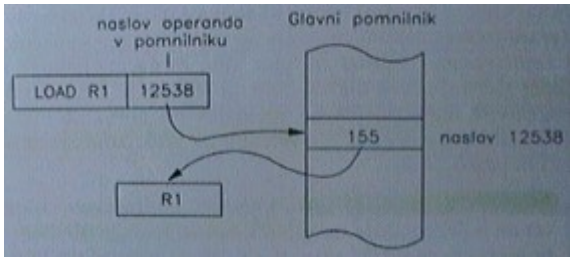


LOAD R1, #155

⇑

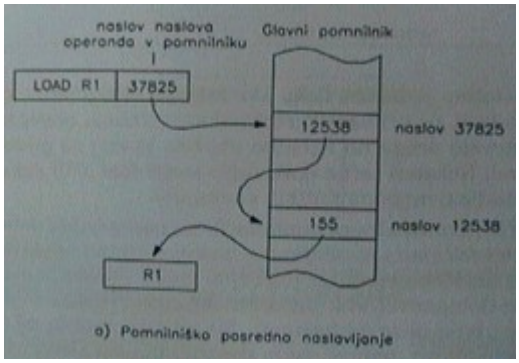
oznaka za takojšnji operand

Neposredno naslavljanje (slab način naslavlanja)
Operand je v ukazu podan z naslovom.



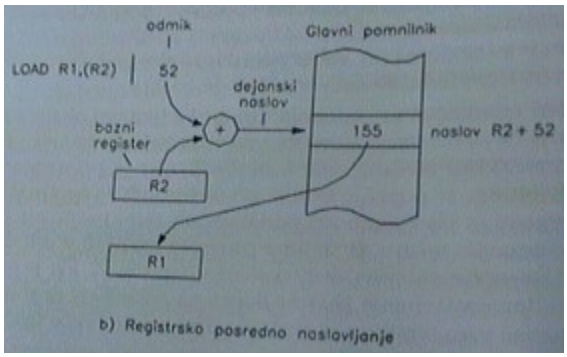
Posredno naslavljanje – v ukazu je posredni naslov operanda
V ukazu je naslov podan nekje drugje v neki drugi vrednosti.

Pomnilniško posredno naslavljanje



Registersko posredno naslavljanje (največ naslavljanja)
V ukazu je naslov registra v katerem je naslov operanda.

LOAD R2, (R3)
LOAD R2, @R3



Naslov op: (R3) + D
⇓ ⇓
vsebina R3 odmik

Variante registerskega posrednega naslavljanja:
- bazno naslavljanje
- indeksno naslavljanje
- pre – dekrementno
- po – inkrementno

Bazno naslavljanje: (najbolj uporabljano)
A = R1 + D
Max do + 255 d(D) = 8
A = pomnilniški naslov
D = odmik (displacement)
R1 = bazni register
d = dolžina
d(D) < d(A)

Indeksno naslavljanje:
A = R1 + R2 +D = R1 + D_1
Odmik izvedemo z baznim naslavljanjem D = R2 + D

Avtomatsko indeksiranje (autoindex reg.)
- pred dekrementno naslavljanje

1. Rx ⇐ Rx - Δ
2. Naslov : A = (Rx) + D

- po inkrementno naslavljanje

1. Naslov A = (Rx) +D
2. Rx ⇐ Rx + Δ

- pozicijsko neodvisno naslavljanje
- PC relativno naslavljanje
  1. Nasov A = (PC) + D

Povezava naslavljanj:

- pomnilniško
- registersko – bazno d(0) < d(A)
  - indeksno d(0) = d(A)
    - pred dekrementno
    - po inkrementno

- PC relativno

Operacije

LOAD (CD, L, L1 …)
STORE – iz registra prenese ukaz v pomnilnik
ADD
SUB
.
.
.

- ALE operacije
- Prenosi podatkov
- Centralne operacije
- Operacije v plavajoči vejici
- Sistemske operacije
- V/I operacije

ALE operacije
Aritmetične – operacije v plavajoči vejici +, -, *, / (ADD, SUB, MUL, DIV)
Logične - $\cup$, $\cap$, $\neg$, $\oplus$
- Pomik : - aritmetični
  - logični
  - krožni

Prenos podatkov

IZVOR $\Rightarrow$ PONOR

LOAD PUSH
STORE PULL (POP)

Kontrolne operacije
Operacije, ki spreminjajo vrstni red izvajanja ukazov.
V ukazu je ciljni naslov (target adress):
- direktni naslov
- PC relativno naslavljanje
- posredno naslavljanje

1. **Pogojni skoki**

Ali je pogoj izponjen? $\Rightarrow$ DA $\Rightarrow$

⇓ ⇓

NE Ukaz na ciljnem naslovu

⇓

Naslednji ukaz

2. **Brezpogojni skoki**

Skok se vedno izvrši!

Terminologija : JUMP brezpogojni skok

BRANCH pogojni skok

3. **Klici in povratki iz procedur** (podprogramov)

KLIC PROCEDURE ⇒ ZAČETEK PODPROGRAMA

.

.

.

NADALJEVANJE ⇐ RETURN

GLAVNEGA PROGRAMA

PROC2 CALL PROC1

Operacije v plavajoči vejici (FPU operacije – floating point unit)

Sistemske operacije – z njimi spreminjamo delovanje računalnika

- omogočen prehod
- onemogočen preh

V/I Operacije

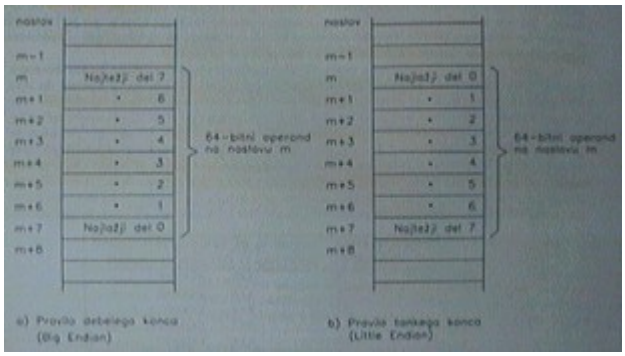
Vrste in dolžine operandov

1. 1 bit
2. Znak (character) dolžina 8 bitov ASCII, EBCDIC
3. Celo število (integer) dolžina 16, 32 bitov
4. Realno število (real) 32, 64 bitni format IEEE 754
5. Desetiško število (BDC) 8 bitni znak (pokrajšano 2 številki v 8 bitov)

Kako je podana vrsta operasndov:

- v operacijski kodi
- operandom so dodani biti (metabiti) in določijo vrsto operanda ⇒ označeni operandi

Problem sestavljenih operandov



Sestavljen pomnilniški operand
Naslov m – pomnilniški naslov sestavljenega operanda

1. na naslovu m težji del – pravilo debelejšega konca (big ending) - Motorola
2. na naslovu n lažji del – pravilo tanjšega konca (little ending) – Intel

Problem poravnanosti sestavljenih operandov

Dolžina operanda (v bitih)	Naslov A, na katerem je operand poravnan (v dvojiškem zapisu)
8	XXX...XXXXXX
16	XXX...XXXXX0
32	XXX...XXXX00
64	XXX...XXX000
128	XXX...XX0000

Naravni naslov
A mod S = 0
⇑ ⇑
Naslov dolžina op. v besedah

Zgradba ukazov
Format ukaza – število polj v ukazu in njihova zgradba
Operacijska koda – polje, ki definira operacijo

- fiksna dolžina operacijske kode
- razširljiva operacijska koda

Število ukazov: vse možne operacijske kode in vsi možni načini naslavljanja.

CISC – Računalniki z velikimi števili ukazov (Complex Instruction Set Computer)
RISC – Računalniki z majhnim številom ukazov (Reduced Instruction Set Computer)

Vzroki za povečevanje ukazov:

- semantični prepad
- mikroprogramiranje
- razmerje med hitrostjo CPE in glavnim pomnilnikom (10:1)

Zakaj RISC?
Prvi RISC računalnik leta 1975 (IBM 801)

- prevajalniki težko uporabljajo kompleksnejše ukaze

- razmerje hitrosti CPE in glavnega pomnilnika (2:1)
- paralelizem v CPE (lažje: kratki in enostavni ukazi)

1980 – RISC I; RISC II — RISC računalniki porabijo mnogo manj časa za izvršitev ukaza.
MIPS — Pentium in 486 nista RISC računalnika

Definicija RISC arhitekture

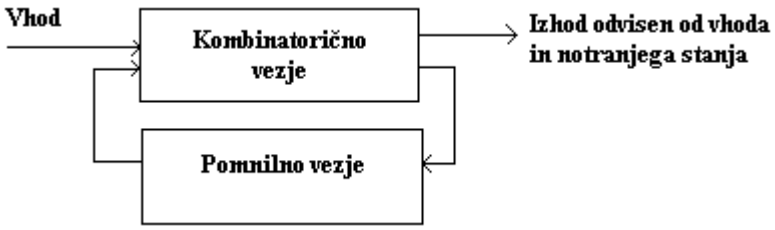
1. Večina ukazov se izvrši v eni periodi CPE
2. Registersko – registerski računalniki (LOAD/STORE)
3. Ukazi niso mikroprogramirani – realizirani s trdo ožičeno logiko
4. Malo ukazov in malo načinov naslavljanja
5. Vsi ukazi imajo isto dolžino
6. Dobri prevajalnik

Centralna procesna enota

Zgradba in delovanje CPE
CPE – digitalni sistem
Digitalna vezja: kombinatorična vezja

VHOD ⇒ KOMBINATORIČNO VEZJE ⇒ IZHOD

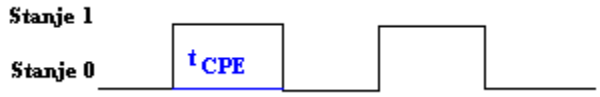
Sekvenčno vezje : sposobnost pomnenja



Delovanje CPE:
1. Jemanje ukaza iz pomnilnika
- ukazno prevzemni cikel
- naslov s katerega se ukaz prevzame je v PC
2. Izvrševanje ukaza (execute)
- izvršilni del vsebuje:
  - analiza ukaza (dekodiranje)
  - prenos operandov v CPE (če že niso v CPE)
  - uvedba operacije
  - shranitev rezultata
  - PC ⇐ PC + 1 (naslov naslednjega ukaza)

Izjema : - prekinitev
- past

CPE – ura



sprememba 0 ⇒ 1 (pozitivna fronta)
sprememba 1 ⇒ 0 (negativna fronta)
t_{CPE} – perioda CPE [ns]
f_{CPE} – frekvenca CPE [Hz], [MHz]
$f_{CPE} = 1 / t_{CPE}$

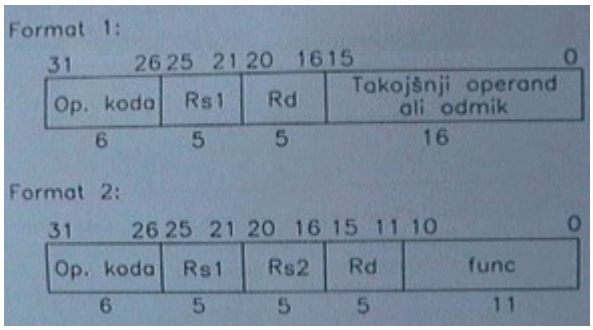
CPE
- kontrolna enota
- podatkovna enota (ALE, registri)

Zgled delovanja: - hipotetični rčunalnik
- RISC računalnik
- registersko - registerski
- 32 splošno namenskih registrov

Lastnosti hipotetičnega računalnika:
- registersko – registerski
- 3 operandi (2 vhodna + rezultat)
- dolžina pomnilniškega naslova = 32 bitov (MAR, PC)
- dolžina pomnilniške besede
- pomnilniški operandi: 8, 16 ali 32 bitni

Število predstavljivo v dvojiškem komplementu
Naslavljanje: (pomnilniški operand)
1. Takojšnje naslavljanje – 16 bitni takojšnji operand
2. Bazno naslavljanje s 16 bitnim odmikom

Vsi ukazi so 32 – bitni (2 formata)



Pomnilniški op.: LOAD, STORE
Izvrševanje ukaza v 5 korakih:
- prevzem ukaza
- dekodiranje ukaza (dostop do registrov)
- izvrševanje(LOAD/STORE, ALE)
- dostop do pomnilnika
- shranjevanje rezultatov

Vrsta i	Vrsta ukaza	Največje št. urinih kazalcev	Število pomnilniških dostopov	Povprečno število urinih period
1	LOAD	7	2	9
2	STORE	7	2	9
3	ALE	5	1	6
4	SKOKI	4	1	5
5	KLICI	6	1	7

LOAD

1. korak	2 periodi	- zadetek v pred. (informacija(ukaz, podatek)
2. korak	1 perioda	- zgrešitev (le 10%) – (informacija ni v pred.
3. korak	1 perioda	temveč je v glavnem pomnilniku)
4. korak	1 perioda	
5. korak	2 periodi	

Pogostost v %

PROG C	SPICE
36	28
46	65
16.2	6.09
1.8	0.9
Pi (C)	Pi (Spice)

CPI – Cycles Per Instruction ⇒ Povprečno število urinih period na ukaz. N – število nasl. vrst ukazov
CPI_i – povprečno št. urinih period za i-to vrsto ukazov.
p_i – pogostost (dinamično) izvajanje i-te vrste ukazov v nekem programu.

$$CPI = \sum_{i=1}^{n} CPI_i * p_i$$

CPI (spice) = 9 * 0.28 + 6 * 0.65 + 5 * 0.0609 + 7 * 0.0091 = 6.78

⇓

Toliko urinih period potrebuje za en ukaz

MIPS – Million Instructions Per Second (miljon ukazov na sekundo)
Frekvenca $f_{CPE} = 1 / t_{CPE}$ [Hz]
Perioda t_{CPE} [s]
Povprečen čas trajanja enega ukaza:
CPI * t_{CPE} [s/ukaz]
Število ukazov, ki se izvršijo v eni sekundi
1/CPI * t_{CPE} [ukazov/s]

$$MIPS = 1 / CPI * t_{CPE} * 10^6$$

f_{CPE} = 166 [MHz]
CPI = 6.78 (spice)

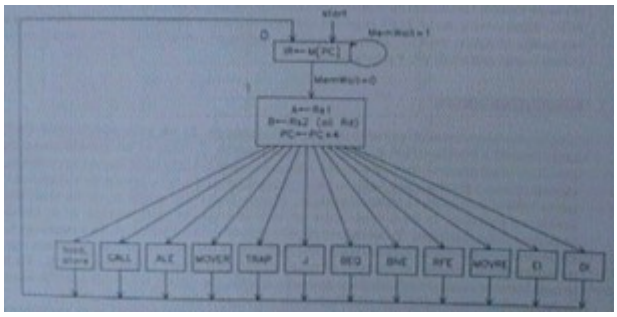
MIPS = f_{CPE} / CPI * 10^6
$t_{CPE} = 1/f_{CPE}$
CPE = Št. Ukazov / MIPS *10^6

MIPS = 166 *10^6 / 6.78 * 10^6 = 24.48 Miljonov ukazov na sekundo

Kontrolna enota

- trdo ožičeno (vsi RISC računalniki)
- mikroprogramirano (skoraj vsi računalniki)
- kontrolna enota
- podatkovna enota
- diagram stanj: -določijo končno število stanj, povezavo med njimi, ter prehode med njimi
- stanje: ⇒ eno urino periodo – opis vseh operacij v tej periodi
  ⇒ določa vse kontrolne signale, aktivne v tej periodi

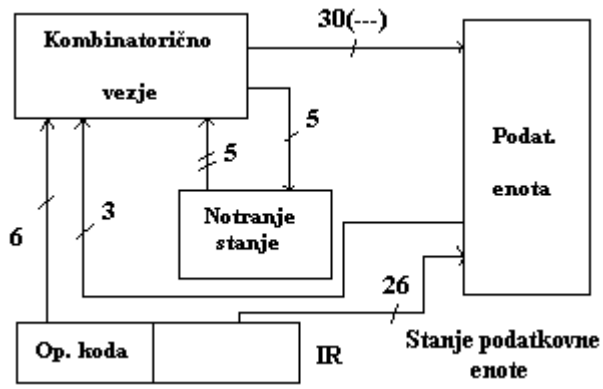
- začetno stanje : prvo stanje, običajno prevzem enakega števila stanj za našo kontrolno enoto ~ 20.



Neskončni avtomat z 20 stanji.
Realizacija s trdo ožičeno logiko.
Operacijska koda 6 bitov.

Vhodi v kontrolno enoto:
- 6 bitov operacijska koda ukaza
- 5 bitov za informacijo o notranjem stanju
- 3 biti informacije stanju podatkovne enote

Izhodi:
- 30 kontrolnih signalov
- 5 bitov stanje kontrolne enote



Mikroprogramirana kontrolna enota

Diagram prehajanja stanj ⇒ mikroprogram sestavljajo mikroukazi
mikroprogramski pomnilnik – kontrolni pomnilnik

Simulacija na mikroprogramskem nivoju = EMULACIJA

Prekinitve in pasti

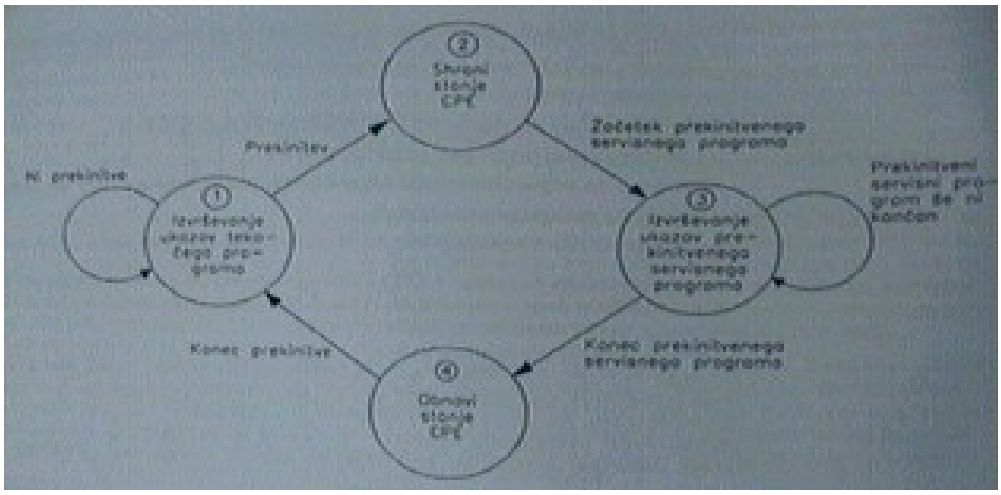
Prekinitev (pride od zunaj): je dogodek, ki povzroči, da CPE konča z izvajanjem programa in začne izvajati nek drug program. – Prekinitveni servisni program (PSP) je običajno kratek.
Prekinitev: je signal, ki od zunaj pride v CPE

Past (se zgodi v CPE): je posebne vrste prekinitev, ki jo zahteva sama procesna enota.
Vir: - program

- neobičajen dogodek v računalniku (deljenje z ničlo, deljenje neskončnosti z neskončnostjo)

IBM ⇒ Interrupts
Motorola ⇒ Exeptions
Intel ⇒ Interrupts, Exeptions, Faults

Dogajane pri prekinitvi



Stanje CPE: Obvezna vsebina PC
Stanje registrov programsko dostopnih registrov

Kdaj CPE reagira na prekinitveno zahtevo?
Po koncu ukaza (zahteva je običajno preprosta)

Prostor kamor se shrani stanje:
Sklad (LIFO – last in first out, dajemo na vrh in vzamemo iz vrha)
Vgnezdene prekinitve
Registri
CPE vedno ob sprejemu prekinitve onemogoči naslednje prekinitve, ki bi lahko prišle v CPE.

CPE po vsaki prekinitveni zahtevi (pred začetkom PSP) onemogoči naslednje prekinitve (pri vseh računalnikih).

Od kje dobi CPE naslov PSP?
- past
- prekinitev

- PRIORITETA prekinitvenih zahtev
- Prepoznavanje naprave, ki je zahtevala prekinitev
- Potrjevanje prekinitvene zahteve

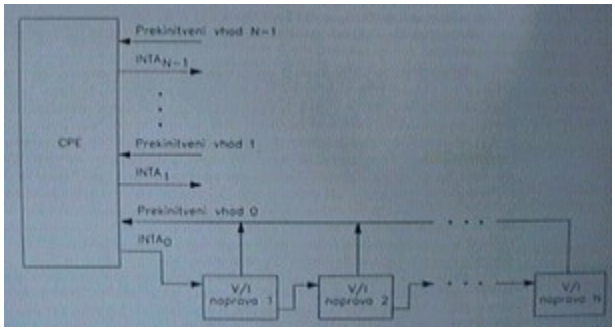
<u>Prioriteta prekinitvenih zahtev:</u>

En sam prekinitveni vhod
Če je na en vhod priključenih več naprav, ki lahko generirajo prekinitvene zahteve.

Programsko
Prioriteta je vrstni red »gledanja« (ugotavljanja), katera naprava je generirala prekinitveno zahtevo (pri tem primeru ima prioriteto N1, ki je prva na vrsti, vendar če program ugotovi da ta ne zahteva prekinitve gre »gledati« do naslednjega …
Programsko izpraševanje (polling) je počasen način in pa tudi najbolj enostaven (Motorola 6802)

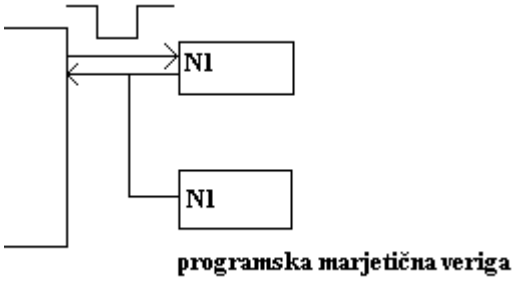
Strojno (Marjetična veriga)

Več prekinitvenih ukazov



Prekinitvena zahteva je sprejeta, če je nivo zahteve višji od trenutne prioritete CPE.

Prepoznavanje naprave
Ko naprava zazna, da je signal CPE prišel do nje vzpostavi visoko stanje.



programska marjetična veriga

Velikanski prekinitveni sistem
Naprava, ki je zahtevala prekinitev pošlje CPE prekinitveni vektor ⇒ naslov ali del naslova, kjer se v pomnilniku nahaja naslov začetnega ukaza PSP za to napravo.