

Veselko Guštin

## Računalniška organizacija

(zapiski za predavanja)

Kazalo.	Str.
1. Računalniški vidiki (datoteka: aPog01)	1
2. Načrtovanje procesorjev	
2.1 Primer: 4-bitni procesor (aPog02)	17
2.2. Procesor MOVE	
3. Super tekoči trak, superskalarnost, predpomnilnik in vodila (aPog03)	32
3.1. Super tekoči trak in superskalarnost	
3.2. P6 družina Intelovih mikrorprocesorjev	
3.3. Predpomnilnik in asociativnost	
3.4. Primer: organizacija predpomnilnika pri Alpha 21264	
3.5. Organizacija pomnilnikov	
3.6. Vodila v CPE	
3.7. PCI vodilo	
4. RISC arhitekture (aPog04)	82
4.1. Uvod in kratka zgodovina	
4.2. Analitične osnove za snovanje RISC arhitektur	
4.3. Skupne značilnosti RISC arhitektur	
4.4. Zgledi posameznih arhitektur	
4.5. Primer: mikroprocesor Alpha	
4.6. Mikroračunalniki, oziroma mikrokontrolerji	
Literatura	134

## 1. Sedem vidikov računalniškega sistema.

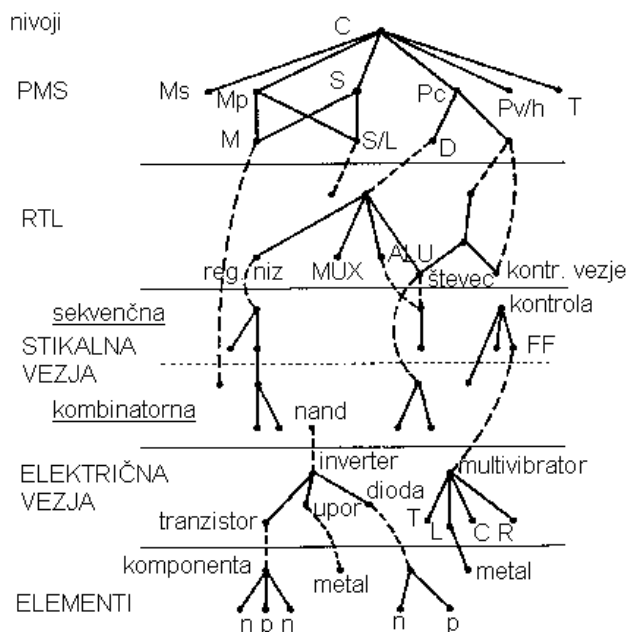
Računalnik določa več dejavnikov: arhitektura, strukturne lastnosti, tehnične lastnosti, človeški dejavniki in ne nazadnje prostor, kjer je nastajal, bil narejen in načrtovan. V tem poglavju bomo obravnavali sedem vidikov, za katere menimo, da jih računalnikarji morajo poznati. Vidiki so med seboj neodvisni, vsak pa ponuja svoje videnje računalniškega sistema. Računalniški strokovnjak ali matematik bo videl računalnik kot nivo interpreterjev. Inženir bo računalnik videl predvsem kot strukturo, seveda pretežno kot logično zasnovano strukturo. S strani uporabnika je računalnik predvsem prodajni artikel. Medtem ko vsi ti ljudje vsak po svoje utemeljujejo svoje poglede, delno spoznavajo tudi druge vidike. Namen tega poglavja je ravno ta, da povečamo razumevanje tudi drugačnih vidikov računalnika ali računalniškega sistema. Poglavje nam bo osnova za lažje razumevanje rač. sistemov v preteklosti, sedanjosti in prihodnosti.

### 1.1. Vidik 1: Strukturni nivoji računalniškega sistema.

Že v [1] so bila predstavljena načela za opisovanje, razumevanje, analiziranje, načrtovanje in uporabo rač. sistemov. Model je sicer doživljal nekatere spremembe, na primer izdelavo rač. v enem integr. vezju ali pa spremembe v arhitekturi (vektorski rač., mreže računalnikov ipd). Kot vidimo na Sl. 1. sistem sestavlja 5 nivojev, s katerimi lahko opišemo nek računalniški sistem. Za vsak nivo je značilen poseben opisni jezik za predstavitev komponent ali modulov nivoja, načinov kombiniranja le-teh in seveda njihovih pravil obnašanja. Znotraj vsakega nivoja obstoja celotna hierarhija sistemov in podsistemov, ker pa so vsi ti opisani z istim opisnim jezikom, ne tvorijo novega nivoja, zato lahko spoznavamo in delujemo na posameznem nivoju, začevši na najnižjem.

**Najnižji** nivo je na Sl. 1 nivo elementov (surovin). Tukaj so sestavni deli (ali komponente) p-tip in n-tip polprevodnikov, dialektrični materiali, metali, ki so oblikovani na različne načine (kot metalizacija na silikonski ploščici, ali konektor, ali vez na tiskanem vezju ipd). **Naslednji** (drugi) nivo je nivo vezij. Na tem nivoju so komponente upori, kondenzatorji, tuljave, napetostni viri in nelinearni elementi (tranzistorji, diode, ...). Obnašanje sistema se na tem nivoju meri z napetostjo, tokom ali magnetnim fluksom. Te fizikalne veličine se v nekem časovnem obdobju stalno spremenjajo, njihovo obnašanje (fizikalnih spremenljivk) bi lahko opisali z enačbami (tudi diferencialnimi). Komponente imajo običajno končno število izvodov in jih zato lahko povezujemo z drugimi komponentami (uporaba jezika VHDL).

Nad nivojem vezij je nivo **preklopnih vezij** (logičnih vezij). Čeprav je nivo vezij v digitalni tehniki zelo podoben onemu v elektrotehniki, je nivo logičnih vezij tisti, ki ju povsem ločuje. Obnašanje sistema je sedaj opisano z diskretnimi spremenljivkami, ki zavzamejo lahko le vrednosti 0 in 1 (ali + in -, H in L, res in ni res, ...).



Slika 1.1. Pet strukturnih nivojev rač. sistema.

Komponente izvajajo logične funkcije, ki jih poznamo kot AND, OR, NAND, NOR ali NOT. Sistem je prav tako zgrajen kot na nivoju vezij (2. nivoja), s povezovanjem izvodov komponent med seboj. Na novo povezane komponente dajo tudi nove vrednosti. Ko je sistem zgrajen, lahko z upoštevanjem izrekov in postulatov Boolove algebre izračunamo obnašanje sistema.

Poleg odločitvenih (decizijskih) preklonnih vezij, katerih izhodi so neposredno odzivajo ob spremembi vhodov, imamo še časovno odvisna (sekvenčna) preklonna vezja, ki lahko hranijo (poljubno dolgo) neko informacijo ali stanje.

Problem, ki ga rešujemo na sekvenčnem nivoju je v tem, da dobimo niz izhodov v času  $t$  kot funkcijo poljubnega števila vhodov v času  $t$ . Obnašanje sekvenčnih vezij je podobno decizijskim, le da moramo dodati pomnilne elemente (v jeziku ABEL to lepo vidimo, če napišemo:

$$Q = Q + 1; \text{ decizijska funkcija } Q, \text{ oziroma}$$

$$Q := Q + 1; \text{ sekvenčna funkcija } Q.$$

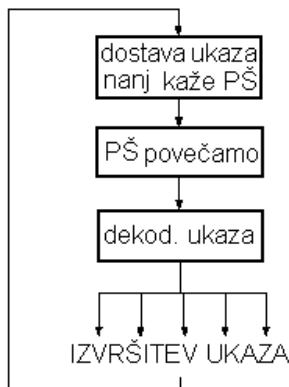
Enačbe, ki opisujejo sekvenčna vezja so običajno taka, da v njih uporabljamo čas, na primer  $Q(t)$ ,  $Q(t+1)$ , in niso preproste decizijske (od časa neodvisne) funkcije.

Nivo nad preklonnimi vezji je **nivo RTL** (register transfer language), opisni jezik prenosov vsebin med registri. Komponente RTL so registri in prenosi med njimi. Funkcijski prenosi se vršijo kot diskretne operacije, torej vrednosti posameznih registrov se med seboj kombinirajo po nekih pravilih in se shranjujejo v druge registre. Pravilo za prenos je lahko od preprostega prenosa ( $A \leftarrow B$ ) do logične ( $A \leftarrow A \& B$ ) ali aritmetične funkcijske povezave ( $A \leftarrow A + B$ ). Opis RTL velja za Boolove enačbe na nivoju sekvenčnih vezij, pa do diferencialnih enačb na nivoju vezij, in ne opisuje nič drugega, kot obnašanje sistema.

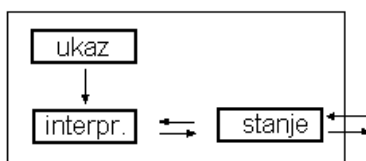
Zadnji ali **5. nivo** na Sl. 1.1. je PMS (procesor - memory - switch) nivo. Le-ta nivo opisuje najvišji nivo računalniškega sistema, ki zajema centralni procesor (ali mikroprocesor), pomnilnik, trak, disk, v/i procesorje, komunikacijske linije, tiskalnik, krmilnike tračnih in (trdih ali mehkih) diskovnih enot, vodila ipd. Računalniški sistem je videti kot stroj, ki procesira informacijo, ki jo - kot vemo - merimo v bitih (oziroma digitih, znakih, besedah, ipd). Komponente sistema imajo svoje zmogljivosti hranjenja, procesiranja in prenosa informacij, torej imajo svoje operativne lastnosti.

## 1.2. Vidik 2: Nivoji interpretiranja.

V nasprotju s strukturnim nivojem, je to funkcionalni nivo. Po [1] je računalniški sistem sestavljen iz nivojev interpretiranja. Kaj je interpreter? To je procesni sistem, ki ga vodijo ukazi in deluje glede na stanje, v katerem je. Osnovna zanka interpretiranja je na Sl. 1.2. in je najbolj poznana na strojnem nivoju računalnika, možno pa jo je najti tudi drugje. Formalizirano zanko vidimo na Sl. 1.3.



Slika 1.2. Osnovna zanka interpretiranja



Slika 1.3. Procesni sistem.

Stanje, ki vpliva na interpreter, je ali zunanje ali notranje. To bomo lažje razumeli, če si ogledamo naslednjih 5 nivojev procesiranja sistema s Tab. 1., ki tvorijo osnovo na primer za sistem rezervacij letalskih kart (ali pa kalkulatorskega programa ipd).

Nivo 0 sistema je logika, na osnovi katere deluje mikroprogramski avtomat nivoja 1. Nivo 1 je torej realni mikroprogramski avtomat. To je stroj, ki ga vidi načrtovalec logičnih vezij. Sistemski nivo 2 je centralna procesna enota (CPE). To je stroj, ki ga vidi programer v zbirniku. Sistemski nivo 3 je prevajalnik visokega programskega jezika, na primer fortran, pascal, C ipd.

Sistemski nivo 4 je aplikativni nivo, v našem primeru je to programski paket za rezervacije let. kart. Lahko pa je to kakršna koli druga aplikacija: bančni sistem, skladiščni sistem, ali pa - kot smo že omenili - kalkulatorski program ipd.

Tabela 1. Pet nivojev interpretiranja program za rez. let. kart.

nivo 4	ukaz: interpreter: notranje stanje: zunanje stanje:	zahteva po sedežu letalski programski paket za rez. kart število trenutnih zahtev, štev. že prijaljenih potnikov, štev. letov številka rezerviranega sedeža, ime let. družbe, datum
--------	--	---

nivo 3	ukaz: interpreter: notranje stanje:  zunanje stanje:	stavki progr. jezika pascal, C, fortran izvedba stavkov progr. jezika ime uporabnika, virtualno naslavljanje, obseg glavnega pomnilnika, prostor na diskih, prekinitve, velikost polj imena podprogramov, vrednosti podatkov, stavki, velikost programa, vrednosti izrazov, programske zanke, iztiskani znaki na tiskalniku
nivo 2	ukaz: interpreter: notranje stanje: zunanje stanje:	ukazi v strojnem jeziku procesor ali mikroprocesor progr. registri, PC, reg. stanj podatki v pomnilniku, reg. v v/i kontrolerjih
nivo 1	ukaz: interpreter: notranje stanje: zunanje stanje:	mikroprogramski ukazi mikroprogramski avtomat ali krmilnik ukazni reg., notranji pomožni reg., skad programski reg., reg. stanj, PC
nivo 0	ukaz: interpreter: notranje stanje: zunanje stanje:	logična vezja mikroprogramski krmilni avtomat ura, števniki, mikroprogramski takti mikroprogramski avtomat, konzolna stikala

Štirje od teh sistemskih nivojev tvorijo hierarhijo, ki jo vidimo na Sl. 1.4. Vsak od sistemov je interpreter, ki z izvajanjem ene ali več operacij naredi korak naprej za sistem na višjem nivoju. Najvišji nivo sistema za rez. let. kart je interpreter ukazov ali sporočil, ki jih dobi zunaj sistema (stranka, omrežje). Sistem deluje iz stanja v stanje, sprejema (testira) sporočila in pošilja nova sporočila v zunanost sistema, ki torej opravlja - intepretira - ukaze zunanjega sveta.

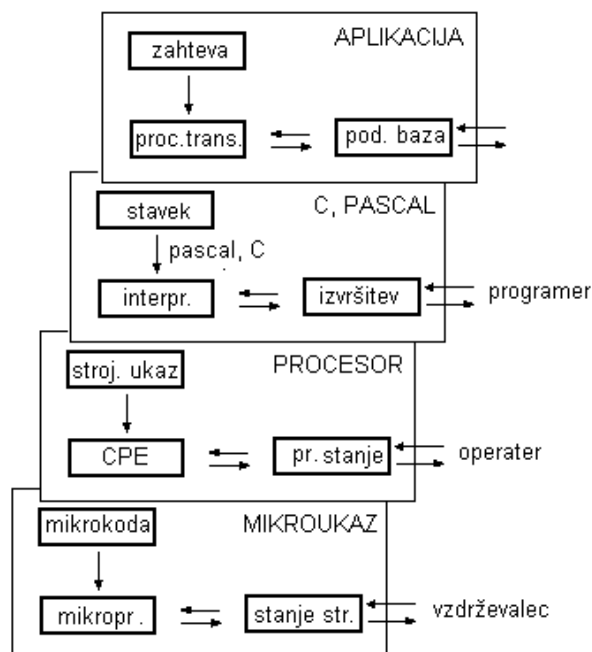
V praksi je le nekaj sistemov, kjer so nivoji čisti interpreterji. Odstopanje od tega modela je tako zaradi hardvera, kot softvera. Hardversko odstopanje se kaže v tem, da nivo 1 (mikroprograski kontroler) ni vselej prisoten, saj je nivo 2 narejen kar kot (mikroprogramski) avtomat. K združitvi nivoja 0 in 1 je največ prispeval prav razvoj bralnih pomnilnikov (ROM).

Obstajajo dve možnosti odstopanja od čistega interpretiranja:

a. visoki programki jeziki se običajno bolj prevajajo (fortran, pascal, C tudi basic), kot pa interpretirajo (basic), in

b. nekatere nivoje lahko obidemo, če se le nahajajo boljši primitivi na nižjih nivojih (na primer seljenje - vertikalna migracija - ukazov DOS na nivo ukazov v zbirniku pri nekaterih računalnikih).

V končni analizi število nivojev visokega programskega jezika ni pomembno. Izboljšanje sistema nas vodi v brisanje nivojev, poslabšanje zmogljivosti sistema



(obsežnost sistema) pa nas vodi v dodajanje novih nivojev (tako kot se dodajajo lupine na primer pri čebuli).

Slika 1.4. Hierarhija interpretiranja.

Osnovni hardver si lahko zamišljamo kot osnovni ali bazni nivo interpretiranja. Običajno je ta nivo razširjen še z operacijskim sistemom. Seveda je tudi nivojev operacijskega sistema več (DOS kliči na nižjem nivoju, DOS ukazi na višjem) in stroj lahko zgradimo v več nivojih. Osrednji del stroja (kernel, jedro znotraj lupine) lahko upravlja in diagnosticira hardverske komponente (disk, terminali) in poskrbi za sinhronizacijo, tako da več programov, ki uporabljajo isti hardver, lahko teče hkrati. Še bolj obsežne operacije kot so delo z datotekami ali zaslone lahko dodamo, če vpeljemo pravila upravljanja in dodeljevanja resursov. Če torej napravo pogledamo skozi operacijski sistem lahko vidimo, da je stroj veliko bolj kompleksen kot je bil na nivoju ISP. Dejansko stroj najbolj spoznamo z uporabo simboličnega zbirnika. Le-ta običajno vključuje osnovne ukaze stroja, ima sposobne vhodno/izhodne operacije in si ga lahko deli več uporabnikov (oziroma programov).

Načrtovalci operacijskega sistema menijo, da so vsi ti nivoji potrebni, če želimo vpeljati nivo interpretiranja visokega programskega jezika. Nivo prevajalnikov ali interpreterjev nam torej omogoča, da prevedemo programske stavke v jezikih basic, fortran, cobol, pascal, C v strojni jezik.

### 1.3. Vidik 3: Nivoji integracije in oblike pakiranja.

Opisali bomo strukturne nivoje, ki vključujejo razne komponente (pakiranja) tako softverske kot hardverske:

1. Integrirana vezja.
2. Moduli (tiskanega vezja).
3. Ohišje posameznih delov (gibki disk, tračna enota, ...).

4. Ohišje ali kabinet, ki hrani celoten hardver.
5. Operacijski sistem (VMS, UNIX, DOS, ...)
6. Standardni jeziki (pascal, C, ...)
7. Posebni jeziki (Abel, VHDL, Verilog, ...).
8. Aplikativne komponente.
9. Aplikacije.

Ta vidik je najpomembnejši, saj kaže kako je računalniški sistem strukturiran, pa tudi kako je njegova cena sestavljena (strukturirana). S strukturnega vidika za objekt, ki ga prodajamo, je cena funkcija tehnologije, organizacije proizvodnje in tržišča.

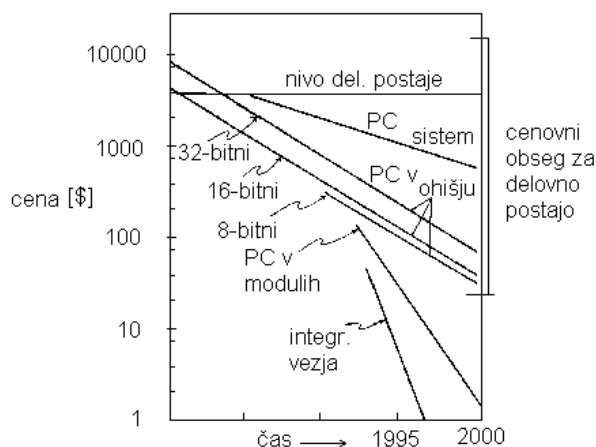
Sistem je podvržen trojim večjim spremembam:

- a. Spremembam na nivoju hardvera, kjer ima zmanjševanje fizikalnih dimenzij (funkcij sistema) troje posledic:
  - nižji nivoji zajemajo višje nivoje,
  - proizvajalci polprevodniških vezij prevzemajo večjo in večjo odgovornost za načrtovano vezje,
  - nivoji izginjajo.
- b. Tudi spreminjanje softverskih nivojev ima troje posledic:
  - vsak nivo zase s časom naraste, ker se pač povečuje njegova funkcionalnost,
  - dodaja se več nivojev, medtem ko se širi uporabnost mikroprocesorjev,
  - funkcije se selijo iz višjega nivoja v nižji (migrirajo navzdol), na primer nekateri ukazi za grafični izris, govor ali sliko, omrežja ipd (DOS ukazi, MMX /multimedijski dodatek ukazov/ procesor, ISSE /internetni dodatek ukazov/ ipd).
- c. Spremembe v vmesniku med hardverom in softverom, ko se veliko funkcij seli iz softvera v hardver za povečanje zmogljivosti stroja (hardversko množenje, pomikanje, računanje s plavajočo vejico - aritmetični koprocessorji ipd).

Za nekatere spremembe hardverskih nivojev je zanimivo, da prav povezovanje in pakiranje najbolj vplivata na načrtovanje sistema. Ovira, ki jih predstavljajo tako povezave kot pakiranja, nastajajo predvsem zato, ker proizvajalcem pomenijo precejšnje stroške ravno fizikalne strukture. Ker so povezave (na primer kabli) potrebni, če gradimo obsežne in velike sisteme, povzročajo precej stranskih učinkov. Električne povezave zahtevajo kable, ki potrebujejo prostor in zmanjšujejo vpliv hlajenja. Dolge povezave (kabli) vplivajo na hitrosti prenosa, ta pa zmanjšuje zmogljivost sistema. Ne nazadnje na prenos signalov vplivajo elektromagnetne sile (motnje), pa tudi sami (kabli) pomenijo izvor elektromagnetnega sevanja, ki ga je potrebno kontrolirati. Na Sl. 1.5. vidimo ceno za razne nivoje integracije. Na ceno vpliva predvsem dolžina besede (mikro)procesorja. Če je beseda majhna, pridobimo nekaj na ceni. Tako na primer za (cenene) ročne kalkulatorje uporabljajo 4(8) bitne računalnike s fiksnim programom, vse skupaj pa je v enem integriranem vezju.



Na Sl. 1.5. vidimo, da polprevodniki, ki so na najnižjem nivoju tehnologije, imajo največji naklon prodajne cene. Cena modulov bo po predvidevanjih padala



počasneje, saj je mešanica integriranih vezij, tiskanih vezij in (veliko) ročnega (ali robotskega) dela: nabiranja, vstavljanja, spajkanja in testiranja. Še počasneje bo padala cena celotnega sistema v ohišju, ki je ali kovinsko ali plastično. Ker ohišja sestavljajo in preizkušajo ljudje, je tu naklon padanja cene najmanjši, skoraj izgine.

Slika 1.5. cena v odvisnosti od stopnje integracije.

Na velik del cene vpliva tudi nova tehnologija. Namen le-te je uporabiti čim manj energije (iz TTL smo prešli na MOS ali CMOS) in prostora (iz DIP na MSD), kar pomeni preprostejša in manj zahtevna pakiranja (sestavljanja) sistemov (iz več omar opreme smo prišli na namizno ohišje).

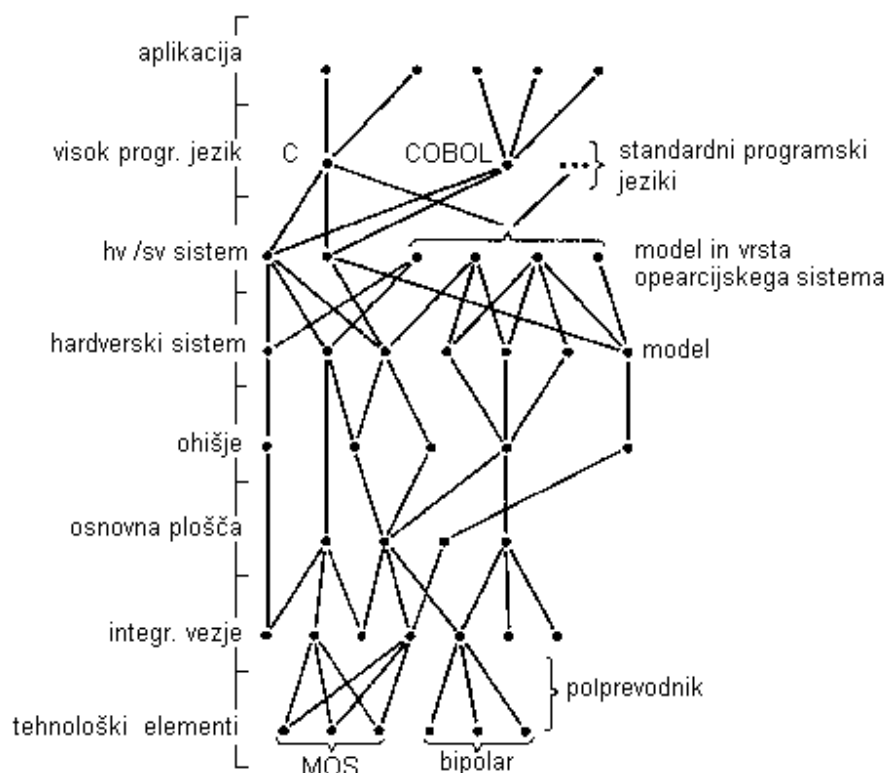
Tržno usmerjena oseba bi lahko pomislila: "Kako z vso razpoložljivo tehnologijo narediti nekaj izjemnega, tako da bi povečali dobiček (doprinos) s strani tehnoloških novosti, a da ne bi bilo potrebno trošiti za zahtevno - sicer banalno - sestavljanje sistemov?" Odgovor bi bil lahko sledeč: "Znižajte ceno tako, da ne nabavite na primer napajalnega vira in hardvera za sestavljanje (ohišja, konektorjev,..). Uporabniku dobavite vse posamezne dele (add-on) in naj sam sestavi računalnik, kakor želi. Tako bo cena kupljenih komponent nižja, čeprav za uporabnika to ne pomeni tudi cenejšega sistema (ali izdelka)!"

Spremembe v nivojih integracije so vplivale na proizvajalce polprevodnikov. V zgodnjih 70. letih so se začela pojavljati vezja večje integracije, proizvajalci so šli še celo tako daleč, da so ponudili že cel procesor v enem int. vezju. Ta pristop pa je preprečil vsakršno poseganje v logični nivo snovanja. Uporabniki so imeli možnost povečevati zmogljivost naprave le z enotami, ki so obsegale 1000 ali 10k ali 100k tranzistorjev in več. Nivoje integracije lahko prikažemo kot komponente enega nivoja, ki jih povezujemo v sistem na višjem nivoju. Nivo torej pomeni enovito načrtovalsko področje (disciplino), ki določa funkcijo, strukturo, zmogljivost in ceno določenega nivoja. Čeprav govorimo o nivojih, vidimo, da sistem sestavlja mreža med seboj povezanih gradnikov.

Na Sliki 1.6. vidimo razne nivoje programskih jezikov, ki uporabljajo le nekatere hardversko/ softverske sisteme, ki pa lahko tečejo na različnih (mikro)procesorjih. Nek (mikro)procesor pa je moč najti v različnih inačicah (in ohišjih).

#### 1.4. Vidik 4: Tržni vidik raznih vrst računalnikov.

Proizvodnja računalnikov prav gotovo sledi potrebam tržišča. Le-to narekuje ceno, zmogljivost in sprejemljivost (kako se bo računalnik "prijel").



Slika 1.6. Računalniški sistem je mreža nivojev.

Ker lahko različni računalniki delujejo z različnimi zmogljivostmi ob različnih cenah, lahko uporabnost računalnikov merimo na različne načine (cena /zmogljivost). Tako na primer lahko uporabimo velik računalnik s centralnim obdelovanjem (paketnim procesiranjem) za celotno organizacijo, ali vsaki organizacijski enoti damo svoj računalnik, ki ga deli tudi z drugimi enotami (shared), ali nek posameznik lahko dela s svojim namiznim računalnikom (PC, delovna postaja), ali pa ima samo svoj žepni kalkulator. Odnos cena/zmogljivost ni edini faktor, ki vpliva na tržno vrednost. Tudi programska združljivost med raznimi stroji je zelo pomembna (na primer osebni rač. z mikroprocesorji družine 80X86).

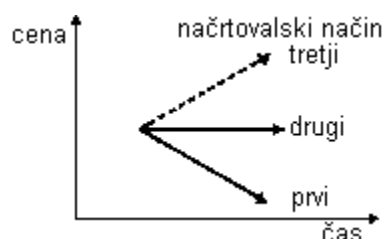
Združljivost z drugimi sistemi je povsem posledica ekonomskih potreb, da se uporabljajo skupne podatkovne baze. Porabnikove investicije v softver običajno zasenčijo tiste, ki jih imajo proizvaljci rač. opreme. Na primer če za vsak računalnik, ki jih je 50000, vložimo samo 1 človek/leto dela (20000\$), kar z drugimi besedami pomeni 5000 programskih ukazov, to v celoti pomeni investicijo 2 mlrd\$, oziroma 250 milj. novih ukaznih vrstic programov (v novih) računalnikih. Ta investicija pa je približno tolikšna, kot je investicija v aparaturno opremo ( $50000 * 20000\$ = 2 \text{ mlrd}\$$ ). Medtem ko želimo z vpeljevanjem nove tehnologije dati novim načrtovanim programom boljši odnos med ceno in zmogljivostjo, moramo pri tem upoštevati združljivost s predhodnim sistemom in podatkovnimi bazami. Uporabnik mora uporabljati nespremenjene programe, če želi izkoristiti prednosti, ki mu jih daje nova tehnologija, sicer se dogodi, da moramo za vsak nakup novega računalnika vedno na novo tudi izdelati programsko opremo.

Na podoben način združljivost družine računalnikov v nekem času omogoča uporabniku izbrati stroj, ki najbolj pokriva njegove zahteve, istočasno pa ima možnost izbrati velik ali majhen računalnik, ki bo reševal njegove nove programe.

Zaradi tega so skoraj vsi novi računalniki iste družine med seboj združljivi (kompatibilni), kar se pozna tudi v ceni. Tehnologija poskrbi za nove proizvode približno vsakih 5-6 let (na primer ZX80, PC XT, PC AT, PC strežnik, RISC postaje), vsak nov proizvod pa pomeni povečano zmogljivost (vsaj za nekaj krat) za približno isto ceno.

Načrtovalcem bi torej lahko dali naslednje napotke. Ko se izkaže, da jim je neka nova tehnologija dosegljiva, se le-ti morajo držati štirih navodil (napotkov), ko vpeljujejo novo tehnologijo:

- a. uporabijo naj novo tehnologijo zato, da zgradijo cenejši sistem, z istimi zmogljivostmi;
- b. držijo naj ceno približno konstantno, uporabijo pa naj dosežke nove tehnologije, da izboljšajo zmogljivost sistema;
- c. pri načrtovanju naj izkoristijo najnovejše dosežke tehnologije in načrtovalske prijeme, tako da kar najbolj povečajo zmogljivost in ceno;
- d. poiščejo naj nove strukture in uporabljajo računalnik le kot "kalkulator", tako da načrtovani projekt odstopa od omenjenih razvojnih poti.



Slika 1.7. Načrtovalske strategije ali načini: 1. konstantna zmogljivost, 2. boljša zmogljivost za konstantno ceno in 3. boljši sistem za večjo ceno.

Na Sl. 1.7. vidimo trajektorije prvih 3 načinov načrtovanja. V splošnem pa je načrtovanje lahko kombinacija različnih razvojnih poti (navzdol). V prvem načrtovalskem stilu držimo zmogljivost na istem nivoju (konstantno) in dosežke v tehnologiji uporabimo le za to, da znižamo ceno sistemu. To načrtovanje sledi načelu minimalnega računalnika. Le-ta je tudi način vpeljevanja nove aplikacije, saj je minimalni računalnik možno vselej zgraditi ob vsakršni tehnologiji. Vsako leto se cena min. računalnika zmanjša, zato cene novih aplikacij postajajo spremenljivejše. Drugi način drži konstantno ceno, kar pomeni, da uporablja boljšo in boljšo tehnologijo, da izboljšuje zmogljivost sistema.

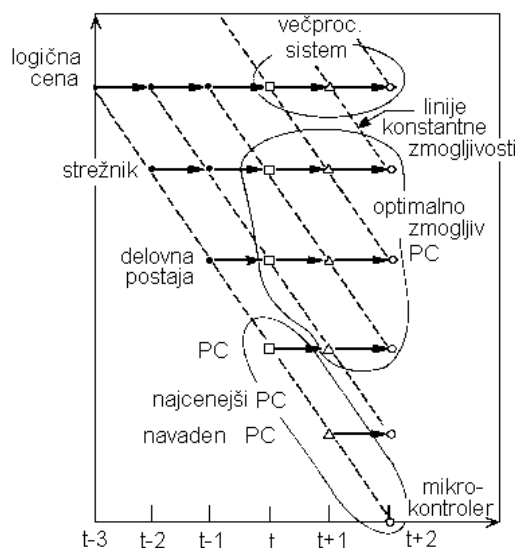
Tretja alternativa uporablja novo tehnologijo zato, da zgradi najboljši sistem. Nov sistem običajno odpravi vse pomankljivosti prejšnjih sistemov in kar se tega tiče predstavlja izjemen dosežek (state - of - the - art). Seveda moramo biti pri teh rečeh zelo previdni, saj nas pretiravanje v ceni ali zmogljivosti (na robu tehničnih zmožnosti) vodi v sistem z majhno zmogljivostjo in veliko ceno (na primer: rač., ki je sposoben sprejemati barvno sliko je gotovo želja, vendar pa se lahko zgodi, da nam tehnologija **trenutno** tega še ne omogoča realizirati zanesljivo in dobili bomo sistem, ki bo spuščal slike - torej bo neuporaben!) Predvsem sta dva razloga, ki nas silita, da delamo na robu

razvojnih možnosti: pionirsko razvojno-raziskovalno delo z uporabo najnovejše tehnologije (smo prvi, ki se s tem ukvarjamo) in nacionalni interes, kjer je običajno na razpolago precej sredstev (to velja predvsem za razvite in bogate države), saj se obeta skoraj neomejen profit (nova tehnologije, novi programi: Windows 95, office97, prevajalniki za pascal, C, basic, VHDL, ALTRA, XILINX, ABEL ipd).

### Razredi računalnikov:

Če upoštevamo tri načela načrtovanja, tedaj nam različne generacije računalnikov dajo graf, ki ga vidimo na Sl. 1.8. To je graf, ki ponazarja obnašanje tržišča računalnikov. Le-ti se ločijo po ceni in tvorijo družino, ki ji pravimo:

- mikroračunalniki, mikrokontrolerji,
- osebni računalniki (PC ali podobni),
- delovne postaje (RISC),
- strežniki (PC ali RISC),
- večprocesorski sistemi,
- super računalniki, veliki rač. sistemi (vektorski, povezovanlni stroji, ...).



Slika 1.8. Odnos med ceno in velikostjo računalniškega sistema. Vodoravne linije pomenijo konstantno ceno, medtem ko poševne smeri pomenijo konstantno zmogljivost.

Načelo uvajanja novih razredov je **cena**, medtem ko za nek že definiran razred zmogljivost običajno določa nov razred. Ko se nek nov razred uveljavi na tržišču, se uporabniki hitro navadijo nanj in skušajo ovrednotiti razred na nivoju zmogljivosti, čeprav gre za novo tehnologijo. Označevanje razredov na nivoju zmogljivosti strojev je za naše razmišljanje pomembno, ker vsaka nova tehnologija zmogljivost poveča za en razred. Tako postanejo zmogljivosti delovnih postaj (v času t-1) dostopne že na PC-jih (v času t). Postavi pa se vprašanje, koliko se lahko cena zmanjša, da pridemo na nov razred. Upoštevamo načelo načrtovanja, da je razmerje med konstantno zmogljivostjo (= konst) in padajočo ceno naraščajoče. To s strani vrednotenja drži, vendar nov razred računalnikov običajno ne začne proizvajati ista organizacija (na primer PC je nastal na osnovi mikroračunalnikov proizvajalca Intel, ki se je "odcepil" od IBM-a), ki izboljšuje svoje proizvode v istem razredu (novi modeli PC, ali del. postaj Silicon graphics, DEC, ...). Nova družba ali organizacija običajno potrebuje nove ideje, da se

loti novega razreda računalnikov. Medtem ko je želja načrtovalcev računalnikov iz razreda v razred pogojena predvsem z (večjo) ceno in zmogljivostmi, je le-ta delno upočasnjena s softversko kompatibilnostjo. Isti računalniški razred ne pomeni tudi isti, a hitrejši nabor ISP. Običajno se lastnosti višjih nivojev računalnikov prenašajo v nižji nivo. Na primer: pri večjih računalnikih je poznanih več podatkovnih tipov, večji je naslovni prostor (fizični in virtualni), sčasoma pa se te lastnosti selijo tudi na nižje nivoje.

### **1.5. Vidik 5: Aplikativni in funkcionalni vidik razredov računalnikov.**

Zaradi splošnega značaja računalnikov, se vse funkcionalne specialnosti zgodijo v času programiranja, kot pa načrtovanja le-tega. Zato je tudi malo specialnih hardverskih računalniških rešitev, ki bi zadostile funkcijskim potrebam. Tako poznamo 4 osnovne tehnike prilagajanja:

a. PMS nivo konfiguracije rač. sistema: Konfiguracijo izberemo tako, da kar najbolj pokriva funkcije zahteve. Uporabnik (načrtovalec) izbere velikost pomnilnika, število in velikost sekundarnega pomnilnika, vrsto stikal in vhodnih/izhodnih enot, ki bi najbolj pokrivala njihovo aplikacijo.

b. Izbira ohišja (pakiranja): Posebne zahteve okolja uporabljamo, ko želimo računalnik uporabiti na primer v tovarniški delovni dvorani, podmornici, letalu ipd.

c. Podatkovni tipi: Računalniki so običajno narejeni za enega (ali več) podatkovnih tipov (8, 16, 32 bitov), ki so najbolj primerni za neko opravilo (task). Nekateri bodo potrebovali aritmetiko s plavajočo vejico, drugi spet procesiranje z nizi (stringi), ali vektorji. Posebni procesorji (na primer DSP) so tudi del te kategorije.

d. Operacijski sistem: Splošen rač. sistem lahko programiramo (s pomočjo operacijskega sistema) tako, da deluje kot PC, več uporabniški sistem, sistem v realnem času, sistem za procesiranje transakcij, terminal ipd.

Možnost uporabe računalnikov smo v letih pred 1970 videli v dveh smereh: znanstveni in komercialni (vojaške uporabe niti ne omenjamo!). Po tem letu pa ločimo naslednje možnosti uporabe: znanstveni, poslovni, krmilni, komunikacijski, obdelovanje datotek, terminalske (tudi osebni računalniki) in večprogramski ter večuporabniški sistemi (dodeljevanje čas. rezin, na primer pri delovnih postajah). Funkcionalne kategorije kot so poslovni, krmilni, komunikacijski in sistemi za obdelovanje datotek lahko nekoliko bolj razdelamo, tako kot kaže Tab. 2.

Tabela 2. Uporabnost računalnikov.

---

Komercialno okolje:

- finančne obdelave v industriji, prodaji in distribuciji
- izstavljanje računov, izračun plač, inventar ipd,
- shranjevanje podatkov,
- običajno paketno (terminalsko) procesiranje
- analiza gospodarjenja \*

Znanstveno okolje, inženirstvo in načrtovanje:

- števila, algoritmi, simboli, tekst, grafika, hranjenje in procesiranje le-teh,
- običajno paketno in terminalsko procesiranje \*,
- zajemanje (analognih in digitalnih) podatkov,
- interaktivno procesiranje \*,
- procesiranje v realnem času (kalkulator, editiranje, ...),
- podatkovne baze (INSPEC, COMPENDEX, ...),

#### Proizvodnja:

- zapisovanje, hranjenje in procesiranje podatkovnih zapisov,
- paketno ali terminalsko procesiranje,
- shranjevanje podatkov in preverjanje alarmov,
- neprekinjeno real-time upravljanje,
- diskretno real-time upravljanje,
- strojno podprta proizvodnja,
- beleženje prisotnosti,

#### Komunikacije in publicistika:

- posredovanje sporočil,
- v/i procesiranje, predprocesorji,
- omrežja,
- govorno krmiljenje,
- terminali in sistem,
- procesiranje teksta, tudi rač. konference in publicistika,

#### Prenosni sistemi:

- omrežja in problemi prenosa (transporta),
- krmiljenje na mestu samem,

#### Vzgoja:

- rač. podprto učenje,
- algoritmi, simboli, hranjenje teksta in procesiranje,
- vaje in pridobivanje izkušenj,
- knjižnični podatki.

#### Uporaba doma:

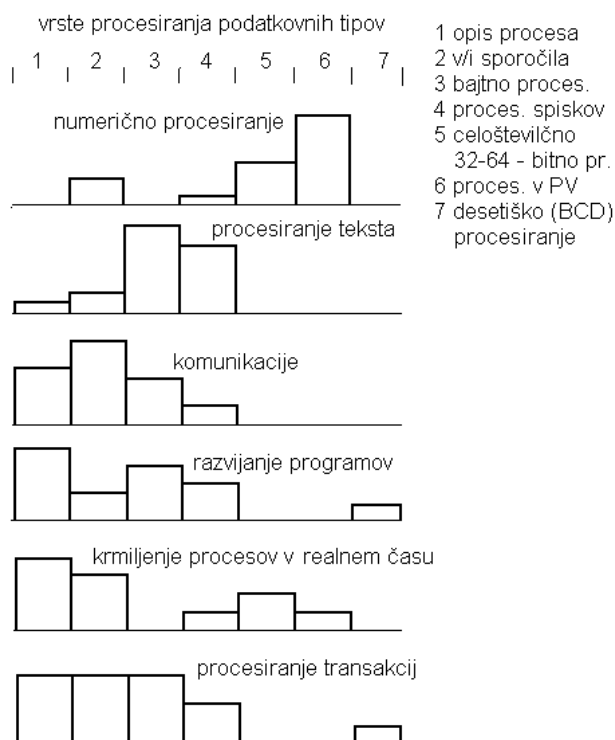
- zabava, hranjenje podatkov, navodil, podatkovnih baz ipd.  
(Z \* so označeni programi, ki zahtevajo neprestani razvoj.)

Transportne aplikacije za letala, vlake ali tovornjake so primer kontrole v realnem času, ki uporabljajo diskretno in/ali zvezno kontrolo. Kontrolni del realiziramo v dveh delih: v samih vozilih in omrežju (letalskem ali železniškem). Transportne kontrolne funkcije narekujejo tri značilnosti računalnika:

- a. Biti mora zanesljiv. Družbe običajno postavijo visoko ceno za človeško življenje, zato noben od rač. v sistemu ne sme povzročiti nesreče,
- b. Biti mora majhen, tudi z majhno uporabo el. energije.

c. Delati mora v ekstremnih pogojih okolja (na primer na lokomotivi so temperature zelo visoke).

Komunikacijski rač. se uporabljajo v telefonskih centralah, pri posredovanju sporočil (message switching), predprocesorjih v večje rač. sisteme ipd. Z uporabo takih sistemov je prenos podatkov iz analogne oblike prenosa počasi prešel v digitalno. Letak način tudi omogoča prenašanje tako zvoka kot slike.



Slika 1.9. Uporabnost posameznih podatkovnih tipov.

Uporabnost računalnikov doma postaja čedalje bolj razširjena. Pravladujoča oblika uporabe računalnikov doma je zabava, predvsem za razne rač. igre, ki pričarajo skoraj realno okolje (virtual reality). Žal je veliko manj uporaben rač. na primer v kuhinji ali kot pomoč pri drugih delih v hiši.

Stroji, kot vidimo, bodo vsebovali več in več funkcij. Ker je osnovni (prvi) podatek ravno podatkovni tip, na Sl. 1.9. vidimo uporabnost posameznih podatkovnih tipov v odvisnosti raznih aplikacij. Seveda so podatki na Sl. 1.9. zelo približni (grobi).

## 1.6. Vidik 6: Izkušnje pri načrtovanju.

Če so prejšnji vidiki opisovali načrtovani objekt, tedaj ta vidik opisuje **načrtovanje** objekta. Predstavljena bosta dva modela (Asimov, Simon), in sicer načrtovanje na osnovi razvoja (in izkušenj) in na osnovi inovativnosti (ali iznajdb). Čeprav srečamo oba načina, prevladuje prvi. Avtor [1] predlaga sledeča načela načrtovanja:

a. **Potreba:** Načrtovanje mora biti posledica posameznikovih ali družbenih potreb, ki jih je možno zadovoljiti z določeno tehnološko rešitvijo.

b. **Možnost realizacije:** Objekt realizacije je v materialni obliki, ali v obliki dela, (oziroma storitve), ki ga je možno fizikalno realizirati.

- c. **Ekonomski faktor:** Materialna oblika ali delo (usluga), ki zajema naš objekt načrtovanja ne sme po ceni presegati možnosti uporabnika.
- d. **Finančne zmožnosti:** načrtovanje proizvodnje in distribucija morata biti finančno podprti.
- e. **Optimalnost:** Metoda načrtovanja mora biti optimalna izbira med več alternativami. Izbira načrtovalnih načel mora biti optimalna tudi glede vseh dovoljenih javnih nastopanj.
- f. **Kriteriji načrtovanja:** Optimalnost mora veljati med načeli načrtovanja, ki predstavljajo kompromis načrtovalca med posameznimi konfliktnimi situacijami: potrošnika, proizvajalca, distributerja in njega samega.
- g. **Morfologija:** Načrtovanje je napredovanje od ideje k realizaciji.
- h. **Proces načrtovanja:** Načrtovanje je iterativni način reševanja problema.
- i. **Manjši problemi:** Ko rešujemo naš osnovni problem, pri tem rešujemo vrsto malih "problemčkov". Na končno rešitev vpliva reševanje vsakega od malih problemov.
- j. **Zmanjšanje dvomov:** Načrtovanje je procesiranje podatkov, ki gredo od dvomljivosti v uspeh h končnemu cilju.
- k. **Ekonomska uspešnost:** Informacija in njeno procesiranje ima neko ceno, ki jo moramo upoštevati, ko načrtovani izdelek uspe, oziroma propade.
- l. **Najmanjše odločitve:** Reševanje problema v vsakem trenutku izvajanja, odločitve pa, ki so potrebne za izvajanje projekta, naj bodo le tolikšne, kot je nujno potrebno za izvedbo načrtovanega dela. To nam bo dalo najboljše možnosti svobodnega odločanja za rešitve na (naj)nižjem nivoju načrtovanja.
- m. **Komunikacije:** Načrtovanje predstavlja opis tako objekta, ki ga nameravamo realizirati, kot njegove proizvodnje, zato je tudi nujno, da je predstavljen z dostopnimi načini komunikacij.

To so bila osnovna načela (filozofije) načrtovanja, ki jih je postavil Asimov [1]. Nadaljeval pa je s fazami, ki v celoti definirajo projekt.

- a. **Študij izvedljivosti (feasibility):** Ugotoviti moramo koristne rešitve za načrtovani problem. Dopuščamo možnost, da problem v celoti definiramo in testiramo, s čimer bomo ugotovili, ali osnovno zamisel, ki je sprožila proces načrtovanja rešitve problema, lahko realiziramo. Tukaj torej osnovne zamisli rešitve formuliramo in testiramo.
- b. **Predhodno načrtovanje (design).** Tu gre za pretehtavanje (ocenjevanje) možnih alternativnih rešitev, zato da najdemo uporabno inačico, ki jo bomo potlej razdelali v podrobnosti.
- c. **Podrobno (detaljno) načrtovanje.** V tej fazi izdelamo inženirski opis tako testiranja, kot proizvodnje načrtovanega objekta. Medtem ko so zgornji opisi predhodne faze načrtovanja, sledijo še štiri faze, ki so poledica proizvodnje in sprejemanje proizvoda na tržišču.
- d. **Planiranje proizvodnega procesa.** To je dejansko nov proces načrtovanja, katerega cilj je načrtati in izdelati sistem, ki bo proizvedel (naš) objekt.
- e. **Planiranje distribucije.** Ta aktivnost zajema vse vidike prodaje, odprave, skladiščenja, promocije in razstavljanja produkta (v smislu reklamiranja, predstavitve).
- f. **Planiranje obrabe (ali izrabe).** Tu moramo upoštevati vzdrževanje, zanesljivost delovanja, varnost, uporabnost, estetski vidik, ekonomiko operativnosti in osnovo za izboljšanje in podaljšanje življenske dobe produkta.
- g. **Ukinitev produkta.** (Produkt nehamo proizvajati.)



Prav gotovo se vse te aktivnosti med seboj prepletajo skozi čas realizacije načrovanega objekta.

### 1.7. Vidik 7: Značilnosti načrtovanja rač sistema (Blaauw).

Podobno kot prej, si tudi tu oprimemo dela Blaauwa [1], ki predvsem ločuje med arhitekturo (zamisljivo), izvedbo (implementacijo) in realizacijo, kot treh neodvisnih nivojev izdelave česar koli, seveda tudi rač. sistema (Tab. 3).

Arhitektura rač. sistema določa funkcionalnost sistema, kot jo vidi programer (na nivoju strojnega jezika) in jo lahko zapišemo z ISP. Arhitektura se ne spreminja na dolgi rok.

Izvedba ISP (implementacija ISP) rač. sistema je (realni) hardver, s katerim razpolagamo in ga opišemo z RTL ali diagramom poteka. Tu so zajeti tudi nekateri algoritmi nadzora stroja.

Realizacija zajema trenutno dosegljivo tehnologijo in vrsto logike ter oblike pakiranja in povezovanja le-te. Realizacija zajema vse podrobnosti, ki so potrebne za fizično izvedbo stroja.

Tabela 3. Značilnosti načrtovanja.

	arhitektura	implementacija	realizacija
cilj	funkcija	cena in zmogljivost	izdelava in vzdrževanje
proizvod	načela delovanja	log. vezja	prehod v proizvodnjo
jezik	napisani algoritmi	diagrami prehajanja	liste in diagrami
meritev kvalitete	preverjanje smiselnosti	analizatorji, osciloskopi	zanesljivost delovanja.
orodje	ISP	RTL (mikropr. kontr., avtomat)	realizacija in implementacija vezij