

1

UVOD

Digitalni računar je mašina koja ljudima pomaže tako što izvršava zadate instrukcije. Niz instrukcija čijim se izvršavanjem obavlja određeni posao naziva se **program**. Elektronska kola računara mogu prepoznati i direktno izvršiti samo ograničen skup jednostavnih instrukcija, pa svaki program, da bi bio izvršen, treba prevesti u takav skup instrukcija. Te osnovne instrukcije retko su složenije od sledećih:

Saberi dva broja.

Proveri da li je zadati broj nula.

Kopiraj skup podataka iz jednog dela memorije računara u drugi.

Sve osnovne instrukcije koje računar razume, obrazuju jezik pomoću koga ljudi komuniciraju s računarom. To je **mašinski jezik**. Konstruktori novog računara uvek moraju brinuti o tome koje će instrukcije uključiti u njegov mašinski jezik. Oni obično pokušavaju da osnovne instrukcije učine što jednostavnijim i usaglase ih s namenom konkretnog računara i njegovim performansama, kako bi smanjili složenost i cenu elektronskih komponenata potrebnih za njegovu izgradnju. Pošto je mašinski jezik računara izuzetno jednostavan, ljudi se njime služe teško (naročito ako računaru treba da zadaju iole složeniji zadatak).

Navedeno zapažanje je s vremenom dovelo do strukturiranja računara u niz (hijerarhijski uređenih) apstrakcija koje se nadgrađuju jedna na drugu. Na taj način se ovladava složenošću računara da bi se računarski sistemi mogli projektovati sistematski, na organizovan način. Pomenuti pristup nazivamo **strukturirana organizacija računara** (engl. *structured computer organization*). U sledećem odeljku objasnićemo šta taj naziv podrazumeva, a posle toga ćemo se pozabaviti razvojem računara, sadašnjim stanjem u toj oblasti i ponuditi neke važne primere.

1.1 STRUKTURIRANA ORGANIZACIJA RAČUNARA

Kao što je pomenuto, velika je razlika između onoga što odgovara ljudima i onoga što odgovara računaru. Ljudi bi želeli da urade neko *X*, ali računari mogu da urade samo *Y*. To otvara probleme. Svrha ove knjige je da objasni kako se takvi problemi mogu prevazići.

1.1.1 Jezici, nivoi i virtuelne mašine

Problemu se može prići na dva načina, a oba zahtevaju da se stvori nov skup instrukcija koji ljudima više odgovara od skupa ugrađenih mašinskih instrukcija. Taj skup novih instrukcija takođe formira jedan jezik koji ćemo zvati L1, za razliku od mašinskog jezika L0 koji razume računar. Dva prilaza problemu razlikuju se u načinu na koji računar izvršava programe napisane na jeziku L1, iako već znamo da računar može da izvršava samo programe napisane na njegovom mašinskom jeziku L0.

Jedan način izvršavanja programa pisanih na jeziku L1 jeste da se najpre svaka njegova instrukcija zameni ekvivalentnim nizom instrukcija na jeziku L0. Tako će se rezultujući program sastojati isključivo od instrukcija na jeziku L0. Računar tada izvršava nov program na jeziku L0 umesto starog programa na jeziku L1. Ova tehnika se naziva **prevođenje** (engl. *translation*).

Drugim načinom se program pisan na jeziku L1 učitava instrukciju po instrukciju, svaka se instrukcija redom ispituje i odmah izvršava njoj ekvivalentan skup instrukcija na jeziku L0. Na taj način se izbegava prethodno generisanje čitavog novog programa na jeziku L0. Tehnika se zove simultano prevođenje, direktno tumačenje ili **interpretiranje** (engl. *interpretation*), a program koji je koristi naziva se **interpreter** (engl. *interpreter*).

Prevođenje i interpretiranje su slične tehnike. U oba slučaja računar izvršava instrukcije na jeziku L1 tako što izvršava ekvivalentne nizove instrukcija na jeziku L0. Razlika je u tome što se pri prevođenju ceo program pisan na jeziku L1 najpre prevede u program na jeziku L0, odbaci se prvi program (L1), a novi program (L0) učitava se u memoriju računara i izvrši. Tokom izvršavanja, aktivan je novi program (L0) i on upravlja resursima računara.

Pri interpretiranju, svaka instrukcija na jeziku L1 analizira se i prevodi, a zatim odmah izvršava. Ne generiše se nov program. Ovde interpreter upravlja računarom.

Program napisan na jeziku L1 za njega predstavlja samo ulazne podatke. U praksi se koriste oba načina, a sve češće i njihove kombinacije.

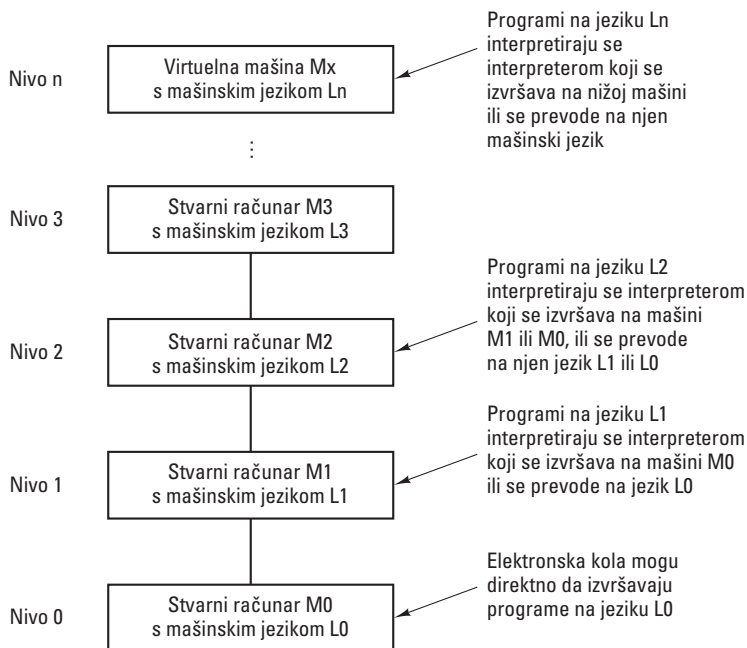
Umesto da razmišljate o prevođenju i interpretiranju, često je jednostavnije zamisliti hipotetički računar – **virtuelnu mašinu** (engl. *virtual machine*) koja razume mašinski jezik L1. Nazovimo tu virtuelnu mašinu M1 (a neku drugu koja razume jezik L0, mašinom M0). Kada bi se takva mašina mogla konstruisati uz niske troškove, uopšte ne bi bilo potrebe za jezikom L0 ili mašinom koja izvršava programe pisane na jeziku L0. Ljudi bi pisali svoje programe na jeziku L1 i računar bi ih direktno izvršavao. Čak i kada bi konstruisanje mašine koja razume jezik L1 bilo previše skupo ili složeno, ljudi bi i dalje mogli pisati programe za nju. Te programe bi mogao interpretirati ili prevesti program napisan na jeziku L0, koji se i sam direktno može izvršavati na datom računaru. Drugim rečima, ljudi mogu pisati programe za virtuelne mašine, baš kao da one stvarno postoje.

Da bi prevođenje i interpretiranje u praksi bilo efikasno, jezici L0 i L1 ne smeju se međusobno previše razlikovati. Ovo ograničenje često znači da će jezik L1, iako bolji od jezika L0, i dalje biti daleko od idealnog za većinu primena. Ta činjenica možda obeshrabruje kada se uzme u obzir prvobitna svrha pravljenja jezika L1 – oslobađanje programera muke da algoritam iskaže jezikom koji mnogo više odgovara mašini nego ljudskoj prirodi. Međutim, situacija ipak nije beznačajna.

Prvo bi nam palo na um da treba napraviti još jedan skup instrukcija koji je, za razliku od jezika L1, više prilagođen ljudima, a manje računarima. Taj treći skup takođe obrazuje jezik koji ćemo nazvati L2 (i zamisliti virtuelnu mašinu M2 koja ga razume). Ljudi sada mogu da pišu programe na jeziku L2 baš kao da stvarno postoji virtuelna mašina čiji je mašinski jezik L2. Takvi programi se mogu prevoditi na jezik L1 ili ih direktno može izvršavati interpreter pisan na jeziku L1.

S pravljenjem čitavog niza jezika, pri čemu je svaki sledeći pogodniji za ljude od njegovog prethodnika, može se ići u nedogled ili sve dok se ne oformi jedan jezik koji je dovoljno pogodan za ljudsku upotrebu. Svaki od jezika u ovom nizu koristi prethodni jezik kao osnovu, tako da računar koji primenjuje ovu tehniku možemo posmatrati kao niz **slojeva** (engl. *layers*) ili **nivoa** (engl. *levels*), poređanih jedan na drugi, kao na slici 1-1. Najniži jezik i nivo su najjednostavniji, a najviši jezik i nivo su, naravno, naj-složeniji.

Postoji važan odnos između jezika i virtuelne mašine. Svaka mašina ima svoj mašinski jezik, sastavljen od svih instrukcija koje može da izvršava. Mašina u stvari definiše jezik. I obrnuto, jezik definiše mašinu tj. mašinu koja može da izvršava sve programe pisane na tom jeziku. Naravno, mašina koju definiše određeni jezik može da bude izuzetno složena a njeno fizičko konstruisanje od elektronskih komponenata može skupo koštati, ali je bez obzira na to ipak možemo zamisliti. Mašina s jezikom C ili C++ ili s jezikom Java kao svojim mašinskim jezikom bila bi zaista složena, ali bi se lako mogla napraviti pomoću savremene tehnologije. Ipak, postoji dobar razlog da se takav računar ne napravi: on cenom ne bi mogao konkurisati drugim računarima. To što se nešto može, nije dovoljno ukoliko u praksi nije i isplativo.



Slika 1-1. Mašina s više nivoa.

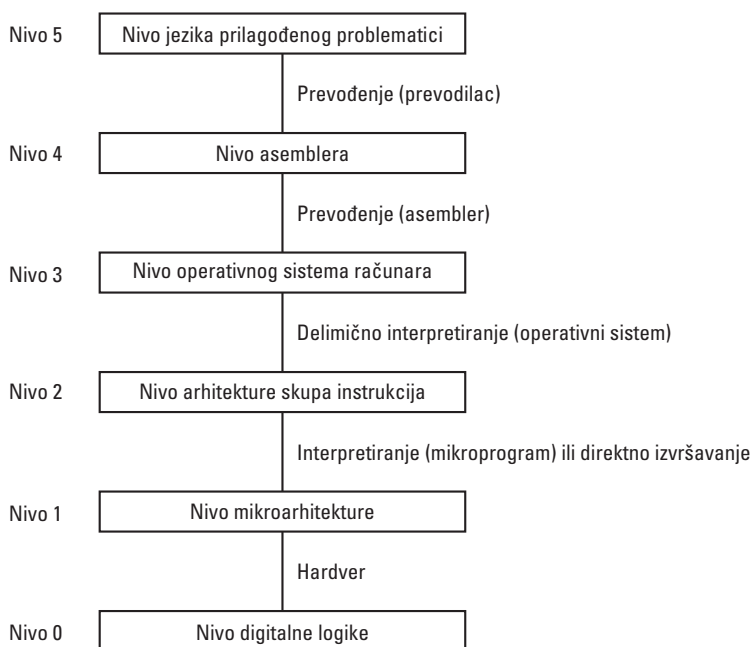
Računar sa n nivoa može se u izvesnom smislu posmatrati kao niz od n različitih virtuelnih mašina, od kojih svaka „govori“ drugačijim mašinskim jezikom. Smatraćemo da izrazi „nivo“ i „virtuelna mašina“ znače isto. Elektronska kola mogu direktno da izvršavaju samo programe pisane na jeziku L_0 – nema potrebe za prevođenjem ili interpretiranjem. Programe pisane na jezicima L_1 , L_2 , ..., L_n mora interpretirati interpreter koji se izvršava na nižem nivou ili se moraju prevesti na jezik koji odgovara nižem nivou.

Osoba koja piše programe za virtuelnu mašinu nivoa n ne mora da brine o skrivenim interpreterima i programskim prevodiocima. Sama struktura mašine obezbeđuje da se ovi programi izvršavaju na određeni način. Programera ne treba da zanima da li će njegove programe izvršavati (korak po korak) interpreter koga izvršava drugi interpreter ili će ih direktno izvršavati elektronska kola. Rezultati su u oba slučaja isti: programi se izvršavaju.

Većinu programera koji koriste mašine sa n nivoa, zanima samo najviši sloj, onaj koji se najviše razlikuje od mašinskog jezika u najnižem nivou. Međutim, ako želite da razumete kako računar funkcioniše, morate se udubiti u sve nivoe. Ljudi koji konstruišu nove računare ili nove računarske slojeve (tj. nove virtuelne mašine) takođe moraju dobro poznavati sve nivoe. Konceptije računara i tehnike njihovog konstruisanja iz niza uzastopnih nivoa, kao i detalji samih nivoa, glavne su teme ove knjige.

1.1.2 Savremeni računari s više nivoa

Većina savremenih računara ima dva ili više nivoa. Postoje računari i sa šest nivoa, kao na slici 1-2. Nivo 0, u samom dnu, predstavlja hardver računara. Njegova elektronska kola izvršavaju programe na mašinskom jeziku nivoa 1. Istine radi, treba reći da postoji i nivo ispod nivoa 0. Taj nivo, koji nije prikazan na slici 1-2 zato što spada u elektrotehniku (i zato izlazi iz okvira ove knjige), zove se **nivo uređaja** (engl. *device level*). Na tom nivou, konstruktor vidi pojedinačne tranzistore koji za njega predstavljaju najosnovnije komponente računara. Ukoliko nekoga zanima kako tranzistor radi iznutra, moraće da pređe na polje fizike čvrstog stanja.



Slika 1-2. Računar sa šest nivoa. Ispod svakog nivoa naznačen je podržani način izvršavanja (zajedno sa imenom programa)

Na najnižem nivou o kome ćemo govoriti, na **nivou digitalne logike**, zanimaće nas tzv. **logička kola** (engl. *gates*). Iako su izgrađena od analognih komponentata, kao što su tranzistori, logička kola se mogu precizno modelovati kao digitalni uređaji. Svako logičko kolo ima jedan ili više digitalnih ulaza (na koje se dovode signali koji predstavljaju 0 ili 1) i izlaz na kome se pojavljuje neka jednostavna funkcija ulaznih podataka, npr. AND (konjunkcija) ili OR (disjunkcija). Svako logičko kolo sastoji se od najviše nekoliko tranzistora. Kombinovanjem nekoliko logičkih kola može se napraviti jednobitna memorija, u koju se može smestiti jedna nula ili jedna jedinica. Jednobitne memorije se mogu kombinovati u grupe od npr. 16, 32 ili 64 da bi se dobili

registri. Svaki **registar** može da čuva jedan binarni broj ograničene veličine. Kombinovanjem logičkih kola može se napraviti i mikroprocesor računara. Logička kola i nivo digitalne logike detaljno ćemo doraditi u poglavlju 3.

Naredni viši nivo je **nivo mikroarhitekture** (engl. *microarchitecture level*). On po pravilu sadrži skupove od 8 do 32 registra koji čine lokalnu memoriju i elektronsko kolo zvano **ALU (aritmetičko-logička jedinica)**, (engl. *Arithmetic Logic Unit*) koje može da obavlja jednostavne aritmetičke operacije. Registri su povezani sa ALU jedinicom i obrazuju **putanju podataka** (engl. *data path*) kojom teku podaci. Osnovni zadatak putanje podataka je da izabere jedan ili dva registra s kojima će ALU jedinica raditi (na primer, da sabere njihov sadržaj) i da rezultat smesti u neki registar.

Na nekim računarima, radom putanje podataka upravlja tzv. **mikroprogram**. Na drugim računarima, putanjom podataka upravlja direktno hardver. U prva tri izdanja ove knjige, ovaj nivo smo zvali „nivo mikroprogramiranja“ jer je do sada to uvek bio softverski interpreter. Pošto putanjom podataka sada često (delimično) upravlja direktno hardver, već u četvrtom izdanju smo na odgovarajući način promenili ime ovog nivoa.

U računarima u kojima se putanjom podataka upravlja softverski, mikroprogram je interpreter instrukcija na nivou 2. On preuzima, ispituje i izvršava jednu po jednu instrukciju, koristeći za to putanju podataka. Na primer, instrukcija ADD se najpre preuzima, pronalaze se njeni operandi i prenose u registre, izračunava se zbir u ALU jedinici i na kraju se rezultat upućuje na određeno mesto. Na računaru s hardverskom kontrolom putanje podataka izvršavaju se isti koraci, ali ne postoji određen (zaseban) program za interpretiranje instrukcija na nivou 2.

Na nivou 2 imamo tzv. **nivo arhitekture skupa instrukcija** (engl. *Instruction Set Architecture level, ISA level*). Svaki proizvođač za svoje računare štampa priručnik koji se, na primer, zove „Referentni priručnik mašinskog jezika“ ili „Principi rada računara Western Wombat Model 100X“. U priručnicima se ne govori o nižim nivoima, već upravo o nivou ISA. Kada se u njima opisuje skup mašinskih instrukcija, u stvari se opisuju instrukcije koje interpretira mikroprogram ili koje izvršavaju hardverska elektronska kola. Ako bi proizvođač neki od svojih modela računara opremio s dva interpretera, koji interpretiraju dva različita nivoa ISA, morao bi da obezbedi i dva referentna priručnika za „mašinski jezik“, po jedan za svaki interpreter.

Naredni nivo je obično hibridan. Većina instrukcija na njegovom jeziku istovremeno pripada i nivou ISA. (Nema razloga da instrukcija koja se pojavljuje na jednom nivou ne postoji i na drugim nivoima.) Osim toga, tu postoji i skup novih instrukcija, drugačija organizacija memorije, mogućnost istovremenog izvršavanja dva ili više programa i još štošta. Definicija nivoa 3 varira mnogo više od definicija nivoa 1 ili 2.

Nove mogućnosti nivoa 3 izvršava interpreter nivoa 2, koji se iz istorijskih razloga zove operativni sistem. Instrukcije nivoa 3 koje su identične instrukcijama nivoa 2 direktno izvršava mikroprogram (ili hardver), dok nove instrukcije izvršava operativni sistem. Drugim rečima, neke od instrukcija nivoa 3 izvršava operativni sistem, a neke direktno mikroprogram. Zato kažemo da je ovaj nivo „hibridan“. Do kraja knjige, ovaj nivo ćemo nazivati **nivo operativnog sistema računara** (engl. *operating system machine level*).

Između nivoa 3 i 4 postoji fundamentalna razlika. Najniža tri nivoa nisu namenjena prosečnom programeru. Oni su prvenstveno mesto za izvršavanje interpretera i programskog prevodioca u cilju podrške višim nivoima. Te interpretere i programske prevodioce pišu **sistemske programeri**, ljudi koji su se specijalizovali za projektovanje i ugrađivanje novih virtuelnih mašina. Nivo 4 i nivoi iznad njega namenjeni su programerima aplikacija koje treba da rešavaju različite probleme.

Druga novost na nivou 4 jeste način podržavanja viših nivoa. Nivoi 2 i 3 uvek se interpretiraju. Nivoi 4, 5 i viši obično se, mada ne uvek, prevode.

Još jednu razliku između nivoa 1, 2 i 3, s jedne strane, i nivoa 4, 5 i viših, s druge, čini priroda korišćenog jezika. Mašinski jezici nivoa 1, 2 i 3 u osnovi su numerički. Programi pisani na tim jezicima sastoje se od dugih nizova brojeva koji se dopadaju računarima, ali ne i ljudima. Počev od nivoa 4, jezik se sastoji od reči i tekstualnih skraćenica koje za ljude imaju određeno značenje.

Nivo 4 (nivo asemblera) sadrži mašinski jezik iz nižih nivoa preveden u skup tekstualnih simbola. Ovaj nivo pruža ljudima mogućnost da pišu programe za nivoe 1, 2 i 3 u obliku koji nije tako neprijatan kao jezici virtuelnih mašina. Programi napisani u assembleru najpre se prevode na jezik nivoa 1, 2 ili 3, a zatim ih interpretira odgovarajuća virtuelna mašina ili stvarni računar. Program koji obavlja prevođenje zove se **assembler** (engl. *assembler*).

Nivo 5 obično sadrži jezike namenjene programerima aplikacija, koje treba da rešavaju konkretne probleme. Takvi jezici se obično zovu **jezici visokog nivoa** (engl. *high-level languages*) i postoji ih bukvalno na stotine. Nekoliko poznatijih su C, C++, Java, LISP i Prolog. Programe pisane na ovim jezicima, na jezike nivoa 3 ili 4 obično prevode programski prevodioci poznati kao **kompajleri** (engl. *compilers*), mada se ponekad i interpretiraju. Primera radi, programi pisani na Javi po pravilu se prvo prevode na jezik sličan jeziku nivoa ISA (tzv. Javin bajtkôd) koji se zatim interpretira.

U nekim slučajevima, nivo 5 se sastoji od interpretera za specifičnu oblast prime- ne, kao što je simbolička matematika. On obezbeđuje podatke i operacije za rešavanje problema iz te oblasti na način koji stručnjaci iz te oblasti lako razumeju.

Sve u svemu, osnovno je zapamtiti da su računari projektovani kao nizovi uzastop- nih nivoa. Svaki nivo predstavlja jasno odvojenu apstrakciju, s drugačijim objektima i operacijama. Projektujući i analizirajući računare na ovaj način, u stanju smo da pri- vremeno zaboravimo na nebitne detalje i da tako složen problem svedemo na nešto smisljeno.

Skup tipova podataka, operacija i mogućnosti svakog nivoa naziva se njegova **arhi- tektura**. Arhitektura se bavi aspektima koji su vidljivi korisniku do tog nivoa. Osobine nivoa koje vidi programer (npr. količina raspoložive memorije) delovi su arhitekture. Aspekti implementacije (npr. vrsta tehnologije koja se koristi za ugradnju memorije) nisu deo arhitekture. Skup znanja potrebnih za projektovanje delova računarskog siste- ma koji su dostupni (vidljivi) programeru, zove se **arhitektura računara** (engl. *com- puter architecture*). U praksi, međutim, arhitektura računara i organizacija računara imaju isto značenje.

1.1.3 Evolucija računara s više nivoa

Da bismo računarima s više nivoa dali „treću“ dimenziju, ukratko ćemo se pozabaviti njihovim istorijskim razvojem i opisati kako je s vremenom rastao broj nivoa i kako se uporedo menjala njihova priroda. Programe pisane na pravom mašinskom jeziku računara (nivo 1) mogu direktno da izvršavaju njegova elektronska kola (nivo 0) bez ikakvog interpretiranja ili prevođenja. Elektronska kola, zajedno s memorijom i ulazno-izlaznim uređajima čine **hardver** računara. Hardver se sastoji od opipljivih objekata: integrisanih kola, štampanih ploča (kartica), kablova, uređaja za napajanje, memorijskih čipova i štampača. U hardver ne spadaju apstraktne ideje, algoritmi ili instrukcije.

Nasuprot tome, **softver** se sastoji od **algoritama** (detaljnih uputstava kako da se nešto uradi) i njihovih računarskih realizacija – programa. Programi se mogu skladištiti na čvrstom disku, na disketi, na kompaktnom disku i drugim medijumima, ali suština je da je softver skup instrukcija koje sačinjavaju programe, a ne fizički medijum na kome se programi beleže.

Na prvim računarima, granica između hardvera i softvera bila je kristalno jasna. Ipak ona se s vremenom prilično izgubila, prvenstveno zbog dodavanja, uklanjanja i međusobnog prekrivanja nivoa. Danas je često teško reći šta je hardver, a šta softver (Vahid, 2003). Centralna tema ove knjige je sledeća:

Hardver i softver su logički ekvivalentni.

Svaka operacija koja se izvršava softverski može se i direktno ugraditi u hardver, naročito pošto se detaljno shvati svaki njen korak. Kao što kaže Karen Panetta Lentz: „Hardver nije ništa drugo do okamenjeni softver“. Naravno, važi i obrnuto: svaka instrukcija koju izvršava hardver može se i softverski simulirati. Odluka da se određene funkcije realizuju hardverski, a neke druge softverski, zavisi od činilaca kao što su cena, brzina, pouzdanost i učestalost očekivanih promena. Postoji samo nekoliko čvrstih pravila o tome šta mora da ide u hardver, a šta izričito mora da se programira. Takve odluke se menjaju s trendovima u ekonomiji proizvodnje, s potražnjom i načinom korišćenja računara.

Nastanak mikroprogramiranja

Prvi digitalni računari iz četrdesetih godina prošlog veka imali su samo dva nivoa: nivo ISA, rezervisan za programiranje i nivo digitalne logike koji je izvršavao programe. Elektronska kola na nivou digitalne logike bila su složena, teško ih je bilo razumeti i proizvesti i bivala su nepouzdana.

Maurice Wilkes, istraživač na Kembridžu, 1951. godine predložio je projektovanje računara s tri nivoa kako bi se drastično pojednostavio hardver (Wilkes, 1951). Takva mašina je trebalo da ima ugrađen, nepromenljiv interpreter (mikroprogram), čija bi funkcija bila da interpretiranjem izvršava programe na nivou ISA. Pošto bi hardver, umesto da izvršava programe na nivou ISA, morao da izvršava samo mikroprograme sa užim skupom instrukcija, bilo bi potrebno manje elektronskih kola. U to

vreme su elektronska kola pravljena od elektronskih cevi, pa bi predloženo pojednostavljenje značilo manji broj elektronskih cevi i veću pouzdanost (manji broj dnevnih otkaza sistema).

Tokom pedesetih godina napravljeno je nekoliko računara s tri nivoa, a znatno više tokom šezdesetih. Ideja da se nivo ISA interpretira mikroprogramski umesto direktno elektronikom, dominirala je tokom sedamdesetih. Svi glavni računari tog doba koristili su takav sistem.

Nastanak operativnog sistema

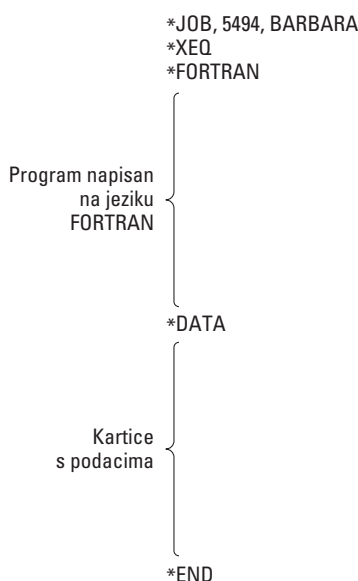
Tih ranih dana računarsko vreme se uglavnom iznajmljivalo, što znači da je svaki programer morao lično da radi s računarom. Pored svakog računara nalazila se sveska za upisivanje. Programer koji bi želeo da izvršava program rezervisao bi u svesci određeno vreme (npr. u sredi između 3 i 5 sati ujutru jer su mnogi programeri najviše voleli da rade u tišini računskog centra). Kada bi njegovo vreme došlo, programer bi krenuo u računski centar sa „špilom“ od 80-stubačnih bušenih kartica (medijum za unošenje podataka s početka računarskog doba) u jednoj ruci i zašiljenom olovkom u drugoj. Kada bi stigao u računski centar, ljubazno bi izgurao napolje svog prethodnika i seo za računar.

Ako bi programer želeo da isproba program napisan na FORTRAN-u, morao bi da uradi sledeće:

1. Otišao bi do ormana gde se čuva biblioteka programa, izvadio veliki zeleni registrator sa oznakom „FORTRAN compiler“, stavio bi kartice iz njega u čitač i pritisnuo dugme START.
2. Stavio bi svoj program napisan na FORTRAN-u u čitač kartica i pritisnuo dugme CONTINUE. Program bi tada bio učitao.
3. Kada se računar zaustavi, ponovo bi učitao svoj program. Iako je za neke prevodioce (kompajlere) dovoljno da se program učitava samo jednom, za mnoge su potrebna dva i više „prolaza“. Pri svakom prolazu treba učitati veliku grupu kartica.
4. Najzad se prevođenje programa primiće kraju. Programera obično hvata nervoza – ako prevodilac tada pronađe grešku, programer mora da je ispravi i da ponovo započne čitav postupak prevođenja. Ako ne pronađe greške, prevodilac buši nove kartice, upisujući program preveden na mašinski jezik računara.
5. Programer tada program preveden na mašinski jezik stavlja u čitač kartica zajedno s grupom kartica koje sadrže potprograme iz biblioteke i ponovo ih učitava.
6. Program počinje da se izvršava. U velikom broju slučajeva on ne radi kako treba i neočekivano se zaustavlja usred rada. Programer bi se tada malo poigrao s prekidačima na konzoli i posmatrao signalna svetla. Ukoliko bi imao sreće, otkrio bi u čemu je problem, ispravio grešku i okrenuo se ormanu s velikim zelenim registratorom da bi postupak započeo iznova. Ako ne bi imao sreće, bio bi prinuđen da štampa **status radne memorije računara** (engl. *core dump*) koji bi odneo kući da ga prostudira na miru.

Opisani scenario, uz manje izmene, bio je godinama uobičajen u mnogim računskim centrima. Uz njega su programeri morali da uče rad s računarom i da znaju šta da preduzmu kada on prestane da radi, što se često događalo. Računar je često radio u praznom kodu dok su programeri nosali kartice po prostoriji ili se zbunjeno češkali iza uveta pokušavajući da pronađu zašto njihovi programi ne rade kako treba.

Negde oko 1960. godine došlo se na ideju da se prazan hod računara smanji tako što će se automatizovati posao operatera. Poseban program, nazvan **operativni sistem** (engl. *operating system*), sve vreme se nalazio u računar. Programer je zajedno s programom unosio i kontrolne kartice koje je čitao i izvršavao operativni sistem. Slika 1-3 prikazuje primer posla koji je obavljao jedan od prvih širokoprihvaćenih operativnih sistema, FMS (FORTRAN Monitor System), na računaru IBM 709.



Slika 1-3. Primer jednog posla koji obavlja operativni sistem FMS.

Operativni sistem je učitavao karticu `*JOB` i informacije s nje koristio za brojanje. (Kontrolne kartice su označavane zvezdicom da ne bi bile protumačene kao kartice s programom ili s podacima.) Sistem je kasnije učitavao karticu `*FORTRAN` – instrukciju da s magnetske trake učita prevodilac FORTRAN-a. Učitani prevodilac bi tada preveo program pisan na FORTRAN-u. Kada bi prevodilac završio svoj zadatak, vratio bi kontrolu operativnom sistemu, koji bi tada učitao karticu `*DATA`. To je bila instrukcija za izvršavanje prevedenog programa koji potrebne podatke učitava s kartica koje se nalaze iza kartice `*DATA`.

Iako je operativni sistem automatizovao posao operatera (odadle mu i ime), on je istovremeno bio i prvi korak ka ostvarenju nove virtuelne mašine. Karticu `*FORTRAN` možete smatrati virtuelnom instrukcijom tipa „prevedi program“.

Slično tome, karticu *DATA možete smatrati virtuelnom instrukcijom tipa „izvrši program“. Nivo sa samo dve instrukcije i nije baš neki nivo, ali je važno početi.

Narednih godina, operativni sistemi su sve više usavršavani. Nivou ISA dodavane su nove instrukcije, pogodnosti i mogućnosti sve dok nije počeo da liči na nov nivo. Neke od instrukcija novog nivoa bile su iste kao i instrukcije na nivou ISA, ali druge (naročito ulazno-izlazne) bile su sasvim različite. Nove instrukcije su se često zvale **makroi operativnog sistema** (engl. *operating system macros*) ili **programerski pozivi** (engl. *supervisor calls*). Danas se uobičajeno zovu **sistemske pozive** (engl. *system calls*).

Operativni sistemi su se razvijali i u drugim pravcima. Prvi su iščitavali grupe kartica i rezultat štampali na linijskom štampaču. Takva organizacija rada poznata je kao **sistem pakete obrade podataka** (engl. *batch system*). Obično bi proteklo više sati od trenutka kada je program stavljen na učitavanje do dobijanja rezultata. Pod takvim okolnostima bilo je teško razvijati softver.

Početkom šezdesetih godina, istraživači na koledžu Dartmouth, na Tehničkom institutu iz Masačusetsa i na drugim mestima, razvili su operativni sistem koji je omogućavao da više programera istovremeno komunicira s računarom. U tim sistemima udaljeni terminali su s centralnim računarom povezivani pomoću telefonskih linija. Računaru su istovremeno pristupali mnogi korisnici. Programer je mogao da unese program i da skoro trenutno dobije odštampane rezultate u kancelariji, u svojoj garaži ili na bilo kom mestu gde je instalirao terminal. Takvi sistemi su se zvali **sistemi s podeljenim vremenom** (engl. *timesharing systems*).

Naše zanimanje za operativne sisteme ograničićemo na one njegove delove koji interpretiraju instrukcije i mogućnosti nivoa 3 koji ne postoje na nivou ISA – ne zanimaju nas aspekti deljenja vremena. Iako to ne naglašavamo, imajte na umu da operativni sistem, osim interpretiranja instrukcija i mogućnosti dodatnih nivou ISA, radi još štošta drugo.

Prenošenje funkcionalnosti u mikrokod

Kada je mikroprogramiranje već postalo obična stvar (sedamdesetih godina), projektanti su shvatili da nove instrukcije mogu dodavati jednostavnim proširivanjem mikroprograma. Drugim rečima, mogli su da dodaju „hardver“ (nove mašinske instrukcije) prostim programiranjem. To otkriće je izazvalo pravu eksploziju skupova mašinskih instrukcija, što je bio rezultat međusobnog takmičenja programera da naprave što veći i bolji skup. Mnoge instrukcije nisu bile u pravom smislu nove – njihov efekat se mogao postići i korišćenjem postojećih instrukcija, ali su nove instrukcije po pravilu davale rezultat nešto brže od postojećih. Na primer, mnogi računari imaju instrukciju INC (uvećanje za jedan, engl. *INCrement*), koja je zadati broj uvećavala za jedan. Pošto su takvi računari imali i opštu instrukciju ADD, uvođenje nove instrukcije za dodavanje broja jedan (ili, npr. broja 720) nije bilo neophodno. Međutim, instrukcija INC je po pravilu radila nešto brže od instrukcije ADD, pa je zadržana.

Iz sličnih razloga mikroprogramu su dodavane i mnoge druge instrukcije. To su često bile:

1. Instrukcije za množenje i deljenje celih brojeva.
2. Instrukcije za računanje s brojevima u formatu pokretnog zareza.
3. Instrukcije za pozivanje procedura i vraćanje rezultata iz njih.
4. Instrukcije za ubrzavanje rada u petljama.
5. Instrukcije za rad sa znakovnim nizovima.

Štaviše, kada su konstruktori računara uvideli koliko se lako mogu dodavati nove instrukcije, počeli su da traže nove mogućnosti koje bi dodali svojim mikroprogramima. Evo nekoliko primera takvih dodataka:

1. Ubrzanje računanja s nizovima (indeksiranje i indirektno adresiranje).
2. Mogućnost premeštanja programa u memoriji nakon početka njegovog izvršavanja (mogućnost relokacije).
3. Sistemi prekida koji šalju signal računaru čim se završi neka ulazna ili izlazna operacija.
4. Mogućnost da se privremeno zaustavi jedan program i započne drugi pomoću malog broja instrukcija (zamena procesa).
5. Specijalne instrukcije za obradu zvuka, slike i multimedijских datoteka.

Tokom godina, dodavane su i mnoge druge osobine i mogućnosti, najčešće da bi se ubrzala određena aktivnost.

Napuštanje mikroprogramiranja

Mikroprogrami su se prilično uvećali tokom zlatnog doba mikroprogramiranja (šezdesetih i sedamdesetih godina). Zbog toga su se izvršavali sve sporije i sporije. Konačno su neki stručnjaci shvatili da će računari raditi brže ukoliko se eliminišu mikroprogrami, smanji skup instrukcija i direktno izvršavaju preostale instrukcije (tj. ako se hardverski kontroliše putanja podataka). Projektovanje računara je tako u izvesnom smislu zatvorilo krug, vraćajući se na mesto gde je bilo pre nego što je Wilkes smislio mikroprogramiranje.

Ipak, taj krug se i dalje okreće. Programi pisani na Javi obično se izvršavaju tako što se prevode u međujezik (Javin bajtkod) koji se zatim interpretira.

U ovom izlaganju istakli smo proizvoljnost granice između hardvera i softvera jer se ona stalno pomera. Današnji softver može postati hardver sutrašnjice i obrnuto. Štaviše, nejasne su i granice između pojedinih nivoa. S programerskog stanovišta, nevažno je kako se određena instrukcija implementira (osim, možda, ako postoje razlike u brzini izvršavanja). Onaj ko programira na nivou ISA, može da upotrebi njegovu instrukciju za množenje kao da je hardverski ugrađena i da o njoj više ne brine (čak ne mora ni znati da li je ona stvarno hardverska). Hardver jednog programera je softver drugog. Na sve ove teme vратиćemo se kasnije.

1.2 KLJUČNE TAČKE RAZVOJA ARHITEKTURE RAČUNARA

Da bi se stiglo do digitalnog računara kakav postoji danas, projektovano je na stotine različitih vrsta. Većina je davno pala u zaborav, ali je nekoliko vrsta imalo i značajan uticaj na savremene ideje. U ovom odeljku ukratko ćemo prikazati neke epizode istorijskog razvoja računara da bismo potpunije razumeli kako smo stigli do onoga što imamo danas. Ne treba posebno isticati da se ovde samo dotičemo najsvetlijih primera, zapostavljajući brojne detalje. Na slici 1-4 nabrojani su neki od ključnih računara o kojima će biti govora u ovom odeljku. Slaterov tekst (1987) dobro je štivo za upoznavanje sa osobama koje su započele eru računara. Njihove kratke biografije, uz predivne kolor-fotografije Louisa Fabiana Bachracha, potražite u Morganovoj knjizi (1997).

Godina	Ime	Autor (proizvođač)	Napomena
1834.	Analytical Engine	Babbage	Prvi pokušaj da se napravi digitalni računar
1936.	Z1	Zuse	Prva mašina za računanje s relejima koja je radila
1943.	COLOSSUS	Britanska vlada	Prvi elektronski računar
1944.	Mark I	Aiken	Prvi američki računar opšte namene
1946.	ENIAC I	Echert/Mauchley	Savremeni računar; istorija počinje ovde
1949.	EDSAC	Wilkes	Prvi računar sa uskladištenim programom
1951.	Whirlwind I	Masačusetski tehnički institut	Prvi računar koji je radio u realnom vremenu
1952.	IAS	Von Neumann	Većina današnjih računara koristi ovu konstrukciju
1960.	PDP-1	DEC	Prvi mini računar (prodato 50 komada)
1961.	1401	IBM	Izuzetno popularan mali poslovni računar
1962.	7094	IBM	Dominirao u naučnim proračunima ranih šezdesetih godina
1963.	B5000	Burroughs	Prvi računar projektovan za jezik visokog nivoa
1964.	360	IBM	Prva proizvodna linija projektovana kao porodica računara
1964.	6600	CDC	Prvi naučni superračunar
1965.	PDP-8	DEC	Prvi mini računar za široko tržište (prodato 50.000 komada)
1970.	PDP-11	DEC	Dominantan mini računar sedamdesetih godina
1974.	8080	Intel	Prvi 8-bitni računar opšte namene na čipu

Slika 1-4. Neke ključne tačke razvoja savremenih digitalnih računara.

1974.	CRAY-1	Cray	Prvi vektorski superračunar
1978.	VAX	DEC	Prvi 32-bitni supermini računar
1981.	IBM PC	IBM	Počinje era savremenih ličnih računara
1981.	Osborne-1	Osborne	Prvi prenosivi računar
1983.	Lisa	Apple	Prvi lični računar s grafičkim korisničkim okruženjem (GUI)
1985.	386	Intel	Prvi 32-bitni prethodnik serije Pentium
1985.	MIPS	MIPS	Prvi komercijalni RISC računar
1987.	SPARC	Sun	Prva RISC radna stanica zasnovana na tehnologiji SPARC
1990.	RS6000	IBM	Prvi superskalarni računar
1992.	Alpha	DEC	Prvi 64-bitni lični računar
1993.	Newton	Apple	Prvi džepni računar

Slika 1-4. Neke ključne tačke razvoja savremenih digitalnih računara. (nastavak)

1.2.1 Nulta generacija – mehanički računari (1642–1945)

Računsku mašinu koja stvarno radi prvi je napravio francuski naučnik Blaise Pascal (1623–1662), pa je u njegovu čast jedan programski jezik nazvan Pascal. Uređaj koji je Pascal napravio 1642, kada je imao samo 19 godina, trebalo je da pomogne njegovom ocu, porezniku francuske vlade. Uređaj je bio potpuno mehanički, sastavljen od zupčanika, i pokretao se pogodno postavljenom ručicom.

Pascalova računska mašina mogla je samo da sabira i oduzima, ali je trideset godina kasnije veliki nemački matematičar, baron Gottfried Wilhelm von Leibniz (1646–1716) napravio drugu mehaničku mašinu koja je mogla i da množi i da deli. Leibniz je u stvari još pre tri veka napravio mehanički ekvivalent današnjeg džepnog kalkulatora.

Tokom sledećih 150 godina nije se dogodilo ništa značajno, sve dok profesor matematike na Kembridžu, Charles Babbage (1792–1871), pronalazač brzina, nije konstruisao i izgradio svoju **diferencnu mašinu** (engl. *difference engine*). Taj mehanički uređaj, koji je kao i Pascalova mašina mogao samo da sabira i oduzima brojeve, trebalo je da preračunava tabele brojeva koje su se koristile za pomorsku navigaciju. Čitava konstrukcija mašine bila je podređena izvršavanju jednog jedinog algoritma – izračunavanju vrednosti polinoma metodom konačnih razlika. Najzanimljivija osobina diferencne mašine bio je način na koji je isporučivala rezultat: ona ga je pomoću čelične matrice utiskivala u bakarnu ploču, što je na neki način preteča svih neizbrisivih medijuma za zapisivanje podataka, kao što su bušene kartice ili kompakt diskovi.

Iako je diferencna mašina radila prilično dobro, Babbage uskoro nije više bio zadovoljan mašinom koja je izvršavala samo jedan algoritam. Počeo je da ulaže sve svoje vreme i sve veće količine porodičnog imetka (da ne pominjemo 17.000 funti koje mu je dodelila vlada) na projektovanje i konstruisanje njenog naslednika – **analitičke mašine** (engl. *analytical engine*). Analitička mašina je imala četiri komponente: skladište (memoriju), radni deo (računsku jedinicu), ulazni odeljak (čitač bušenih

kartica) i izlazni odeljak (bušač kartica i štampač). Skladište ulaznih podataka i rezultata sadržalo je 1000 reči sa po 50 decimalnih cifara. Radni deo je mogao da prihvati operande iz skladišta, da ih sabira, oduzima, množi ili deli i da rezultat na kraju vrati u skladište. Slično diferencnoj mašini, i analitička mašina je bila potpuno mehanička.

Analitička mašina je imala tu prednost što je bila opštenamenska. Ona je učitala instrukcije s bušenih kartica i izvršavala ih. Neke instrukcije su mašini naređivale da uzme dva broja iz skladišta, da ih prenese u radni deo, da s njima nešto uradi (npr. da ih sabere) i da rezultat vrati u skladište. Druge instrukcije su joj mogle narediti da ispita broj i da s njim uradi jednu ili drugu operaciju, zavisno od toga da li je broj pozitivan ili negativan. Kada bi se u mašinu stavile bušene kartice s različitim programima, analitička mašina je mogla da obavlja različita izračunavanja, što diferencna mašina nije mogla.

Pošto se analitička mašina mogla programirati pomoću jednostavnog assembler-skog jezika, za nju je trebalo napraviti softver. Za taj posao Babbage je angažovao mladu ženu, Adu Augustu Lovelace, ćerku čuvenog britanskog pesnika, lorda Byrona. Ada Lovelace je tako postala prvi svetski programer. Programski jezik Ada[®] dobio je ime u njenu čast.

Nažalost, kao i mnogi savremeni konstruktori, ni Babbage svoj hardver nikada nije potpuno očistio od grešaka. Problem je bilo to što je njegova mašina zahtevala na hiljade precizno izrađenih klinova, točkova i zupčanika kakve tehnologija devetnaestog veka nije mogla da proizvede. Bez obzira na to, on je sa svojim idejama bio daleko ispred svog vremena, pa je čak i danas konstrukcija mnogih savremenih računara ostala vrlo slična analitičkoj mašini, tako da se može reći da je Babbage (pra)otac savremenog digitalnog računara.

Sledeći veći skok u razvoju računara dogodio se krajem tridesetih godina prošlog veka, kada je nemački student tehnike Konrad Zuse napravio niz automatskih računskih mašina koristeći elektromagnetske releje. On nije mogao da dobije podršku vlade za svoj projekat jer je rat već počeo, a državne birokrate su bile uverene da on neće trajati dovoljno dugo da mašina bude dovršena pre njegovog kraja. Zuse nije znao za Babbageov rad, a njegove mašine su uništene u savezničkom bombardovanju Berlina 1944. godine, tako da njegov doprinos nije imao nikakvog uticaja na dalji razvoj računara. Ipak, treba ga pomenuti kao jednog od pionira u ovoj oblasti.

Nešto kasnije, u SAD su se pojavila dva čoveka koji su takođe konstruisali kalkulator: John Atanasoff s Državnog koledža Ajove i George Stibbitz iz Belovih laboratorija. Atanasoffljeva mašina je bila zapanjujuće moderna za svoje vreme. Koristila je binarnu aritmetiku i imala je kondenzatorsku memoriju koja je zbog električne disipacije povremeno osvežavana, što je postupak koji se zvao „čišćenje memorije“. Savremeni čipovi s dinamičkom memorijom (DRAM) rade na isti način. Mašina, nažalost, nikada nije stvarno puštena u rad. Atanasoff je na izvestan način prošao slično kao i Babbage: obojica su bili vizionari koje je porazila neodgovarajuća hardverska tehnologija tog vremena.

Stibbitzov računar, iako primitivniji od Atanasoffljevog, ipak je radio. Stibbitz je javno demonstrirao svoj rad na konferenciji u koledžu Dartmouth 1940. godine. Konferenciji je prisustvoao i John Mauchley, potpuno nepoznat profesor fizike s Pensilvanijskog univerziteta. O njemu će računarski svet tek čuti.

Dok su se Zuse, Stibbitz i Atanasoff bavili konstruisanjem automatskih kalkulatora, mladi Howard Aiken „peške“ je obavljao mučna numerička izračunavanja za svoj diplomski rad na Harvardu. Aiken je tek po diplomiranju shvatio važnost mašinskog izračunavanja. Otišao je u biblioteku, pronašao Babbageov rad i odlučio da od niza releja napravi računar opšte namene kakav Babbage nije mogao da sklopi od nazubljenih točkića.

Aikenova prva mašina, Mark I, dovršena je na Harvardu 1944. godine. Imala je 72 reči sa po 23 decimalne cifre i svaku instrukciju je izvršavala za 6 sekundi. Za učitavanje podataka i saopštavanje rezultata korišćena je bušena papirna traka. U vreme kada je Aiken dovršio njenog naslednika, mašinu Mark II, računari s relejima već su bili zastareli. Počela je era elektronike.

1.2.2 Prva generacija – elektronske cevi (1945–1955)

Drugi svetski rat bio je podsticaj za razvoj elektronskih računara. Na njegovom početku nemačke podmornice su pravile darmar u britanskoj mornarici. Nemački admirali su iz Berlina, preko radija slali komande podmornicama a Britanci su ih mogli presretati, što su i činili. Problem je bilo to što su poruke bile šifrovane uređajem **ENIGMA**, čijeg prethodnika je konstruisao jedan amater, bivši američki predsednik Thomas Jefferson.

Rat je tek bio počeo kada je Britanska obaveštajna služba dobila primerak mašine ENIGMA od Poljske obaveštajne službe koja ju je ukrala od Nemaca. Međutim, za dešifrovanje kodiranih poruka bilo je potrebno mnogo računanja, a da bi od njih bilo ikakve koristi, one su se morale razumeti vrlo brzo posle presretanja. Britanska vlada je opremila laboratoriju za dešifrovanje poruka u kojoj je u najvećoj tajnosti izgrađen elektronski računar COLOSSUS. U njegovom projektovanju aktivno je učestvovao i slavni britanski matematičar Alan Turing. COLOSSUS je bio spreman za rad 1943. godine, ali je sa aspekta razvoja računarstva to bio ćorsokak, jer je britanska vlada sve informacije o tom projektu proglasila najstrožom vojnom tajnom za period od 30 godina. COLOSSUS pominjemo samo zato što je to bio prvi elektronski digitalni računar na svetu.

Osim što je uništio Zuseov trud i podstakao izgradnju COLOSSUS-a, rat je uticao i na razvoj računarstva u SAD. Američkoj vojsci bile su potrebne tablice dometa za navođenje njene teške artiljerije. Ona ih je pravila tako što je angažovala stotine žena da ih izračunavaju pomoću ručnih kalkulatora (smatralo se da žene ovaj posao rade preciznije od muškaraca). Postupak izračunavanja je dugo trajao, a greške su se neizbežno događale.

John Mauchley, koji je znao za radove Atanasoffa i Stibbitza, shvatio je koliko su vojsci potrebni mehanički kalkulatori. Kao i mnogi naučnici na polju računarstva posle njega, i on je podneo zahtev za finansiranje projekta elektronskog računara. Njegov predlog je prihvaćen 1943. godine i Mauchley i njegov diplomac J. Presper Eckert počeli su da prave elektronski računar koji su nazvali **ENIAC (Electronic Numerical Integrator And Computer)** – elektronski numerički integrator i računar).

On se sastojao od 18.000 elektronskih cevi i 1500 releja. ENIAC je težio 30 tona i „vukao“ 140 kilovata iz električne mreže. Što se arhitekture tiče, mašina je imala 20 registara, od kojih je svaki mogao da primi desetocifreni decimalan broj. (Decimalni registar je vrlo mala memorijska komponenta koja može da čuva jedan broj sa određenim maksimalnim brojem decimalnih cifara, nešto slično „kilometraži“ u automobilu.) ENIAC se programirao podešavanjem 6000 višepoložajnih prekidača i povezivanjem mnoštva utičnica pomoću čitave šume kratkospojnih kablova.

Mašina nije dovršena do 1946. godine, kada je već bilo kasno da se na bilo koji način iskoristi za svoju osnovnu svrhu. Međutim, pošto se rat završio, Mauchley i Eckert su dobili mogućnost da organizuju letnju školu i tako prikažu svoj rad kolegama naučnicima. Letnja škola je začela eksploziju zanimanja za izgradnju velikih digitalnih računara.

Posle te istorijske letnje škole, mnogi istraživači su se opredelili za pravljenje elektronskih računara. Prvi takav računar bio je EDSAC (1949) koji je izgradio Maurice Wilkes na Kembridžu. Zatim su sledili JOHNIAC iz korporacije Rand, ILLIAC sa Univerziteta Ilinoisa, MANIAC iz laboratorija Los Alamos i WEIZAC sa Weizmanovog instituta u Izraelu.

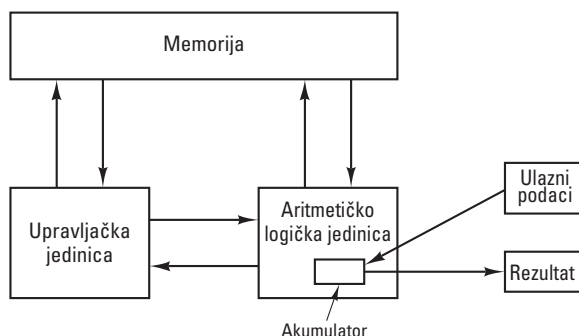
Eckert i Mauchley su uskoro počeli da rade na nasledniku, računaru **EDVAC (Electronic Discrete Variable Automatic Computer)** – elektronski automatski računar diskretne promenljive). Međutim, taj projekat je propao kada su napustili Pensilvanijski univerzitet da bi u Filadelfiji osnovali samostalnu kompaniju Eckert-Mauchley Computer Corporation (tada još niko nije mislio na Silicijumsku dolinu). Posle niza integracija, iz ove kompanije je izrasla savremena korporacija Unisys.

Što se tiče pravne strane, Eckert i Mauchley su podneli prijavu za patent tvrdeći da su oni izmislili digitalni računar. Kad čovek bolje razmisli, uopšte ne bi bilo loše imati takav patent. Posle više godina pravničkog natezanja, sud je doneo odluku da Eckert i Mauchley ne mogu dobiti patent jer je digitalni računar izmislio John Atanasoff iako ga nikad nije patentirao.

Dok su Eckert i Mauchley radili na EDVAC-u, jedan od njihovih saradnika na projektu ENIAC, John von Neumann, prešao je na Institut za napredna istraživanja u Princetonu da bi napravio sopstvenu verziju EDVAC-a, **računar IAS**. Von Neumann je bio svestrani genije poput Leonarda da Vinčija. Govorio je mnoge jezike, bio je stručnjak za fiziku i matematiku, i sećao se svega što je ikada čuo, video ili pročitao. Mogao je napamet da izgovori tekst knjige koju je pročitao godinama pre toga. U doba kada je počeo da se zanima za računare bio je već najugledniji matematičar na svetu.

Ubrzo mu je postalo jasno da je programiranje računara pomoću brojnih prekidača i kablova spor, mučan i krut postupak. Shvatio je da se program u memoriji računara može predstaviti u digitalnom obliku, gde bi se našao zajedno s podacima. Isto tako, uvideo je i da se nezgrapno serijsko decimalno računanje koje se koristilo u ENIAC-u, gde je svaku cifru predstavljalo 10 elektronskih cevi (1 uključena i 9 isključenih), moglo zameniti paralelnom binarnom aritmetikom, nečim što je Atanasoff davno iskoristio.

Osnovna zamisao koju je on prvi opisao sada je poznata kao **Von Neumannova mašina**. Ona je iskorišćena za računar EDSAC, prvi računar koji je memorisao programe i još uvek je osnova za skoro sve digitalne računare – čak i danas, posle više od pola veka. Von Neumannova mašina (i mašina IAS, napravljena u saradnji s Hermanom Goldstineom) imala je toliki uticaj na razvoj računarstva da je vredi ukratko opisati. Iako se ta mašina uvek vezuje za ime Von Neumanna, važan doprinos u njenom ostvarivanju takođe su dali Goldstine i drugi. Blok-dijagram njene arhitekture prikazan je na slici 1-5.



Slika 1-5. Originalna Von Neumannova mašina

Von Neumannova mašina ima pet osnovnih delova: memoriju, aritmetičko-logičku jedinicu, upravljačku jedinicu, ulazne i izlazne uređaje. Memorija se sastoji od 4096 reči od po 40 bitova, od kojih svaki može biti 0 ili 1. Svaka reč sadrži dve 20-bitne instrukcije ili 40-bitni označen broj. Osam bitova svake instrukcije određuju njen tip, a preostalih 12 bitova definišu jednu od 4096 reči iz memorije. Aritmetičko-logička jedinica i upravljačka jedinica zajedno čine „mozak“ računara. U savremenim računarima one se nalaze na jednom čipu zvanom **centralni procesor** (engl. *Central Processing Unit, CPU*).

U aritmetičkoj logičkoj jedinici nalazi se specijalan unutrašnji 40-bitni registar nazvan **akumulator**. Tipična instrukcija dodaje memorisanu reč u akumulator ili sadržaj akumulatora smešta u memoriju. Mašina ne podržava operacije s brojevima u formatu pokretnog zareza jer je Von Neumann smatrao da svaki pristojan matematičar može da pamti decimalni zarez (u stvari, binarni zarez).

Otrpili su u isto vreme kada je Von Neumann pravio mašinu IAS, istraživači na Masačusetskom tehničkom institutu (MIT) pravili su još jedan računar. Za razliku od IAS-a, ENIAC-a i drugih sličnih mašina koje su radile s dugačkim rečima i služile za složena numerička izračunavanja, mašina razvijena na MIT-u, Whirlwind I, radila je sa 16-bitnim rečima i bila je namenjena upravljanju u realnom vremenu. Taj projekat je podstakao pronalaženje memorije s magnetnim jezgrom (Jay Forrester), a kasnije i razvoj prvog komercijalnog mini računara.

Dok se sve ovo događalo, IBM je bio mala kompanija koja se bavila proizvodnjom bušača kartica i mašina za njihovo sortiranje. Iako je kompanija IBM delom finansirala Aikena, nije bila previše zainteresovana za računare sve dok 1953. godine nije na tržište

izbacila model 701, davno pošto je kompanija Eckerta i Mauchleyja postala broj jedan na tržištu s računarom UNIVAC. Model 701 je imao 2048 36-bitnih reči, s dve instrukcije po jednoj reči. To je bio prvi iz serije naučnih računara koji će dominirati sledećom decenijom. Tri godine kasnije pojavio se model 704 koji je u početku imao 4096 reči osnovne memorije, 36-bitne instrukcije i jednu novost – hardver za rad s brojevima u formatu pokretnog zareza. Godine 1958, IBM je počeo proizvodnju svog poslednjeg računara s elektronskim cevima, modela 709, koji je bio unapređeni model 704.

1.2.3 Druga generacija – tranzistori (1955–1965)

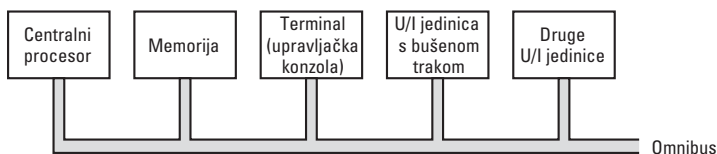
Tranzistor su 1948. godine u Belovim laboratorijama prvi napravili John Bardeen, Walter Brattain i William Shockley, za šta su 1956. godine dobili Nobelovu nagradu za fiziku. Tranzistor je za samo 10 godina revolucionarno izmenio industriju računara, tako da su krajem pedesetih godina računari sa elektronskim cevima smatrani prevaziđenim. Prvi računar s tranzistorima izrađen je u laboratoriji Linkoln Masačusetskog tehničkog instituta. To je bila 16-bitna mašina, napravljena po uzoru na Whirlwind I. Nazvana je **TX-0 (Transistorized eXperimental computer 0)** – tranzistorizovani eksperimentalni računar 0) i predstavljala je probni uređaj za razvijanje mnogo složenijeg modela TX-2.

S računarom TX-2 nije se mnogo odmaklo kada je jedan od inženjera koji su radili u laboratoriji, Kenneth Olsen, 1957. godine osnovao kompaniju Digital Equipment Corporation (DEC) za proizvodnju komercijalnog računara veoma sličnog modelu TX-0. Prošle su četiri godine dok je taj računar, PDP-1, ugledao svetlost dana, uglavnom zato što su investitori DEC-a čvrsto verovali da za računare ne postoji tržište. Uostalom, i sâm T. J. Watson, bivši predsednik IBM-a jednom je izjavio da svetsko tržište računara zauzima samo pet ili šest procenata ukupnog tržišta. DEC je zbog toga uglavnom prodavao male štampane ploče.

Kada se model PDP-1 konačno pojavio 1961. godine, imao je 4096 18-bitnih reči i mogao je da izvršava 200.000 instrukcija u sekundi. Performanse su mu bile dvostruko slabije od IBM-ovog modela 7090, tranzistorizovanog naslednika modela 709 i u to vreme najbržeg računara na svetu. PDP-1 je koštao 120.000 dolara; model 7090 je koštao milione. DEC je prodao više desetina modela PDP-1 i tako je rođena industrija mini računara.

Jedan od prvih modela računara PDP-1 dat je Masačusetskom tehničkom institutu, gde je ubrzo privukao pažnju nekoliko nadobudnih genijalaca, tako čestih na MIT-u. Jedna od brojnih novotarija modela PDP-1 bio je ekran rezolucije 512 × 512 tačaka. Nije dugo potrajalo, a studenti su programirali PDP-1 za „zvezdane ratove“ i tako je se pojavila prva video igrice.

Nekoliko godina kasnije DEC je predstavio model PDP-8, 12-bitni računar, ali mnogo jeftiniji od modela PDP-1 (16.000 dolara). PDP-8 je imao udarnu inovaciju: jedinstvenu magistralu, tzv. omnibus, prikazanu na slici 1-6. **Magistrala** (engl. *bus*) skup je paralelnih žica kojima se povezuju komponente računara. Primenjena arhitektura je jasno odstupila od arhitekture mašine IAS, usmerene na memoriju, i od tog doba je bila prihvaćena za skoro sve male računare. DEC je prodao 50.000 modela PDP-8, što ga je učvrstilo na prvom mestu među proizvođačima mini računara.



Slika 1-6. PDP-8 omnibus.

U međuvremenu, IBM je na pojavu tranzistora reagovao tako što je kao tranzistorizovanu verziju modela 709 zamislio model 7090, a kasnije i model 7094. Model 7094 imao je sistemski ciklus od 2 mikrosekunde i osnovnu memoriju od 32.536 36-bitnih reči. Modeli 7090 i 7094 označili su kraj mašina tipa ENIAC, ali su ipak tokom šezdesetih godina dominirali naučnim računarstvom.

U vreme kada je IBM postala glavna kompanija u oblasti naučnog računarstva sa svojim modelom 7094, zarađivala je i velike količine novca od malog poslovnog računara 1401. Taj model je mogao da čita magnetske trake i da upisuje podatke na njih, da čita kartice i da ih buši, i da rezultate štampa skoro isto tako brzo kao model 7094 – i to sve samo za delić njegove cene. Za naučne proračune je bio katastrofalan, ali je odlično vodio poslovne knjige.

Model 1401 bio je neobičan po tome što uopšte nije imao registre, čak ni fiksnu dužinu reči. Memorija mu se sastojala od 4000 8-bitnih bajtova, mada su kasniji modeli podržavali i tada nezamislivih 16.000 bajtova. Svaki bajt je sadržao 6-bitni znak, administrativni bit i bit kojim je označavan kraj reči. Instrukcija MOVE je, na primer, dobivši izvorišnu i odredišnu adresu, prebacivala bajtove sa izvorišta na odredište sve dok ne bi naišla na bajt u kome je bit predviđen za označavanje kraja reči bio 1.

Godine 1964, mala i nepoznata kompanija Control Data Corporation (CDC) predstavila je računar 6600 koji je bio skoro za red veličine brži od moćnog modela 7094 i svakog drugog tadašnjeg računara. On je za računardžije bio „ljubav na prvi pogled“ i kompaniji CDC uspeh je bio osiguran. Tajna njegove brzine i razlog što je bio mnogo brži od modela 7094 ležali su u njegovom centralnom procesoru koji je u velikoj meri koristio paralelan rad. On je sadržao više funkcionalnih jedinica za sabiranje, množenje i deljenje, a sve su mogle raditi paralelno (istovremeno). Iako je za izvlačenje maksimuma iz takve mašine bilo potrebno pažljivo programiranje, uz malo truda se moglo postići da ona istovremeno izvršava 10 instrukcija.

I kao da to nije bilo dovoljno, model 6600 je imao i niz malih pomoćnih računara, tako da je podsećao na Snežanu i sedam patuljaka; to je značilo da je centralni procesor nesmetano mogao da „melje“ brojeve, prepuštajući detalje upravljanja i ulazno-izlazne operacije malim računarima. Iz sadašnje perspektive moglo bi se mirne duše reći da je model 6600 bio deset godina ispred svog vremena. Mnoga ključna rešenja koja nalazimo u savremenim računarima direktno potiču od modela 6600.

Konstruktor modela 6600, Seymour Cray, bio je legendarna osoba, unekoliko sličan Von Neumannu. Čitav svoj život je posvetio pravljenju sve bržih računara, nazvanih **superračunari** (engl. *supercomputers*), uključujući modele 6600, 7600 i

Cray-1. Izumeo je i čuveni algoritam za kupovinu kola: odete do najbližeg prodavca, uđete i pokažete prvi automobil, govoreći: „Uzimam ovaj“. Taj algoritam troši najmanje vremena na nevažne stvari (kao što je kupovina kola) ostavljajući vam maksimum vremena za rešavanje važnih (kao što je projektovanje superračunara).

Doba razvoja obiluje i drugim računarima, ali se jedan od njih ističe iz sasvim drugih razloga i zato ga treba pomenuti: model Burroughs B5000. Konstruktori mašina kao što su PDP-1, 7094 i 6600 bili su preokupirani hardverom, bilo zbog njegove cene (DEC) ili zbog njegove brzine (IBM i CDC). Na softver uopšte nisu obraćali pažnju. Konstruktori modela B5000 primenili su drugačiji pristup. Oni su napravili mašinu specijalno namenjenu programiranju na Algolu 60, prethodniku jezika C i Java, i u hardver su ugradili mnoge funkcije da bi programskom prevodiocu olakšali posao. Tako se spoznalo da i softver nešto vredi. Nažalost, to je gotovo odmah i zaboravljeno.

1.2.4 Treća generacija – integrisana kola (1965–1980)

Zahvaljujući otkriću Roberta Noycea iz 1958. godine – silicijumskim integrisanim kolima – desetine tranzistora moglo se smesti na jedan čip. Takvo zgusnuto pakovanje omogućilo je gradnju računara koji su bili manji, brži i jeftiniji od njihovih prethodnika s tranzistorima. U nastavku opisujemo neke značajnije modele računara ove generacije.

IBM je 1964. godine bila vodeća računarska kompanija i imala je veliki problem sa svoja dva veoma uspešna računara: modelima 7094 i 1401. Oni su međusobno bili vrlo nekompatibilni: jedan je bio visokobrzinski „mlin za brojeve“ i koristio je paralelnu binarnu aritmetiku sa 36-bitnim registrima, dok je drugi bio proslavljeni obrađivač ulazno-izlaznih podataka koji je serijsku decimalnu aritmetiku primenjivao na memorisane reči promenljive dužine. Mnogi IBM-ovi kupci imali su oba računara i nikako im se nije dopadalo to što treba da imaju dva potpuno različita programerska odeljenja.

Kada je došlo vreme da ove dve serije proizvoda zamene novi modeli, IBM je povukao radikalan potez. Uveo je jedinstvenu proizvodnu liniju System/360, zasnovanu na integrisanim kolima, koja je trebalo da zadovolji i naučne i poslovne potrebe kupaca. System/360 sadržao je mnogo inovacija, a najvažnije je to da je porodica od nekoliko mašina različite veličine i moći koristila isti assemblerski jezik. Kompanija je mogla da zameni model 1401 novim računarom 360 Model 30, a model 7094 računarom 360 Model 75. Model 75 je bio veći i brži (a i skuplji), ali je softver pisan za bilo koji računar iz serije mogao u načelu da radi na bilo kom drugom računaru iz iste serije. U praksi je to značilo da će softver pisan za manji model bez problema raditi i na većem, ali da program pisan za veći računar možda neće moći da stane u memoriju manjeg. Ipak, i to je bilo veliko poboljšanje u odnosu na situaciju s modelima 7094 i 1401. Koncept porodice računara odmah se primio i za samo nekoliko godina većina proizvođača je nudila porodice sličnih računara koji su se razlikovali samo po ceni i performansama. Neke osobine prvih članova porodice 360 prikazane su na slici 1-7. Ostali modeli su uvedeni kasnije.

Svojstvo	Model 30	Model 40	Model 50	Model 65
Uparedne performanse	1	3,5	10	21
Trajanje sistemskog ciklusa (nanosekunde)	1000	625	500	250
Maksimalna memorija (bajtovi)	65.536	262.144	262.144	524.288
Preuzeti bajtovi po ciklusu	1	2	4	16
Maksimalan broj kanala za podatke	3	3	4	6

Slika 1-7. Početna ponuda IBM-ove proizvodne linije 360.

Druga velika inovacija u računarima serije 360 bilo je **multiprogramiranje** (engl. *multiprogramming*), tj. mogućnost da se u memoriji istovremeno drži više programa, tako da dok jedan od njih čeka da se završe ulazno-izlazne operacije, drugi za to vreme može da koristi procesor. To je rezultovalo boljim iskorišćenjem centralnog procesora.

Modeli iz serije 360 bili su i prvi računari koji su mogli da emuliraju druge računare (da simuliraju njihov rad). Manji računari su mogli da emuliraju model 1401, a veći model 7094, tako da su kupci mogli nastaviti da koriste svoje stare binarne programe pri prelasku na računar iz serije 360. Neki modeli su programe pisane za računar 1401 izvršavali toliko brže od njega da mnogi kupci nisu ni menjali stare programe.

Emuliranje je s tim modelima išlo lako jer su svi početni modeli iz serije 360 – a i većina kasnijih – bili mikroprogramirani. IBM je trebalo da napiše samo tri mikroprograma: osnovni skup instrukcija za seriju 360 i po jedan skup instrukcija za modele 1401 i 7094. Takva fleksibilnost je i bila jedan od osnovnih motiva za uvođenje mikroprogramiranja.

Modeli iz serije 360 razrešili su dilemu „binarno-paralelno ili serijski-decimalno“ jer se usvojilo kompromisno rešenje: računari su snabdeveni sa po 16 32-bitnih registara za binarnu aritmetiku, ali im je memorija organizovana po bajtovima, kao kod modela 1401. Isto tako, koristili su i serijske instrukcije u stilu modela 1401 za premeštanje zapisa promenljive dužine u memoriji.

Još jedna važna osobina računara iz serije 360 bio je (za to vreme) ogroman adresni prostor od 2^{24} (16.777.216) bajtova. Uz ondašnju cenu od nekoliko dolara po bajtu, tolika memorija je izgledala nezamislivo velika. Seriju 360 je, nažalost, zamenila serija 370, a zatim i serije 4300, 3080 i 3090 (sve sa istom arhitekturom). Sredinom osamdesetih godina ograničenje memorije je postalo ozbiljan problem, pa je IBM morao delimično da odustane od kompatibilnosti kada je prešao na 32-bitne adrese neophodne za adresiranje novih 2^{32} bajta memorije.

Gledajući unazad, moglo bi se primetiti da je verovatno već trebalo da imaju i 32-bitne adrese jer su imali 32-bitne reči i registre, ali u to vreme niko nije ozbiljno mogao zamisliti računar sa 16 miliona bajtova memorije. Kriviti IBM za nemanje vizije, bilo bi kao okrivljavati današnjeg proizvođača ličnih računara što ima samo 32-bitne adrese. Za nekoliko godina, ličnim računarima možda će trebati mnogo više od 4 milijarde bajtova memorije, pa će 32-bitne adrese postati neprihvatljivo kratke.

I svet mini računara se dobro pokazao u trećoj generaciji, kroz seriju DEC-ovih modela PDP-11, 16-bitnog naslednika modela PDP-8. Serija PDP-11 umnogome je

podsećala na mlađeg brata serije 360, slično odnosu modela PDP-1 i 7094. Modeli serija 360 i PDP-11 imaju registre koji rade s rečima i memoriju organizovanu po bajtovima, a svi pokrivaju znatan raspon parametra cena/performanse. Modeli iz serije PDP-11 pokazali su se kao veoma dobri, naročito u akademskoj sredini i održali su komaniju DEC na samom vrhu među proizvođačima mini računara.

1.2.5 Četvrta generacija – vrlo visok stepen integracije (1980-?)

Tehnologija, nezgrapno, ali tačno nazvana **vrlo visok stepen integracije** (engl. *Very Large Scale Integration*, VLSI) osamdesetih godina omogućila je da se na jedan čip smeste desetine hiljada, zatim stotine hiljada i na kraju milioni tranzistora. Ovakav napredak tehnologije uskoro je doveo do proizvodnje manjih i bržih računara. Pre modela PDP-1 računari su bili toliko veliki i skupi da su kompanije i univerziteti morali da predviđaju posebne prostorije za njih, tzv. **računske centre**. S pojavom mini računara, svako odeljenje je moglo samostalno da kupi računar za svoje potrebe. Osamdesetih godina cene su pale toliko nisko da je i pojedinačni korisnik mogao sebi da priušti računar. Tako je započela era ličnih računara.

Lični računari su korišćeni sasvim drugačije od velikih računara. Oni su uporebljavani za obradu teksta, za tabelarne proračune i za brojne visokointeraktivne aplikacije (npr. igrice) s kojima se veliki računari nisu najbolje slagali.

Prvi lični računari obično su prodavani u „kompletu“. Svaki komplet je sadržao ploču sa štampanim kolima, pregršt čipova među kojima je obično bio Intelov čip 8080, par kablova, napojnu jedinicu i možda disketnu jedinicu od 8 inča. Sastavljanje računara od delova prepuštano je kupcu. Uz računar nije isporučivan nikakav softver. Ako vam je softver bio potreban, morali ste ga sami napisati. Kasnije je Gary Kildall napisao operativni sistem CP/M koji je stekao veliku popularnost na računarima s procesorom 8080. To je bio pravi operativni sistem za podršku (disketnoj) jedinici, sa sistemom datoteka i komandama koje je korisnik s tastature zadavao komandnom procesoru (komandnom okruženju).

Pojavio se i lični računar druge vrste, prvo Apple, a kasnije i Apple II (oba su konstruisali Steve Jobs i Steve Wozniak, i to u garaži). Ti računari su postali veoma popularni među kućnim korisnicima i u školama, pa je Apple preko noći postao ozbiljan takmac.

IBM, tada vodeća snaga u industriji računara, posle mnogo oklevanja i osvrtnja konačno je prelomio i odlučio da se upusti u posao s ličnim računarima. Umesto da svoj lični računar projektuje od početka, koristeći samo sopstvene komponente, što bi predugo trajalo, IBM je uradio nešto što uopšte ne liči na ovu kompaniju. Jedan od rukovodilaca, Philip Estridge, snabdeven je džakom punim novca i oteran daleko od birokrata iz sedišta firme u Armonku (država Njujork), uz napomenu da se ne vraća dok ne donese kakav god upotrebljiv lični računar. Estridge je osnovao radionicu daleko od sedišta firme, u Boca Ratonu (Florida), izabrao Intelov čip 8088 za centralni procesor i napravio IBM-ov lični računar (IBM Personal Computer, IBM PC) od komponentata koje je našao na tržištu. On je predstavljen 1981. godine i odmah je postao najprodavaniji računar u istoriji.

IBM je povukao još jedan neobičan potez, ali je zbog toga kasnije zažalio. Umesto da projekat računara drži u tajnosti (ili barem da ga zaštiti patentima), što je inače radio, objavio je kompletne planove, zajedno sa električnim šemama u knjizi koju je prodavao za 49 dolara (!). Zamisao je bila da se tako drugim kompanijama omogući pravljenje dodatnih kartica za IBM PC kako bi se povećala njegova fleksibilnost i popularnost. Na nesreću, pošto je projekat sada bio dostupan svima i pošto su se svi delovi mogli lako naći na tržištu, brojne kompanije su počele da prave PC **klonove**, često jeftinije od originala. Tako je nastala čitava industrija.

Iako su i druge kompanije pravile lične računare koristeći druge procesore (npr. Commodore, Apple i Atari), dominacija industrije ličnih IBM računara bila je toliko velika da su sve one naprosto zbrisane. Ostala ih je samo nekolicina i one danas posluju na marginalnim tržištima.

Jedan takav računar je, iako jedva, ipak preživeo, a to je Appleov Macintosh. Macintosh je predstavljen 1984. godine kao naslednik zlosrećne Appleove Lise, koja je bila prvi računar s **grafičkim korisničkim okruženjem** (engl. *Graphical User Interface*, GUI) sličnim današnjem Windowsu. Lisa je propala jer je bila preskupa, ali je jeftiniji Macintosh koji se pojavio godinu dana kasnije postigao veliku slavu pa su mu brojni obožavaoci postali vrlo privrženi.

Rano tržište ličnih računara uskoro je iznedrilo do tada neiskazanu želju za posedovanjem prenosivog računara. U to doba koncept „prenosivog računara“ imao je isto toliko smisla kao i sintagma „prenosivi frižider“ danas. Prvi stvarno prenosivi lični računar bio je Osborne-1, koji je, budući težak 11 kg, više zasluživao atribut „prtljažni“. Ipak, on je dokazao da je koncept prenosivog računara realan. Osborne-1 je postigao skroman komercijalni uspeh, ali je godinu dana kasnije kompanija Compaq na tržište izbacila svoj prvi prenosivi IBM PC klon i ubrzo postala lider na polju prenosivih računara.

Prva verzija ličnog IBM računara prodavana je sa operativnim sistemom MS-DOS, koji je isporučivala tada mala kompanija Microsoft. Kako je Intel proizvodio sve snažnije mikroprocesore, IBM i Microsoft su unapređivali MS-DOS i napravili mu naslednika – operativni sistem OS/2, koji je radio u grafičkom okruženju sličnom okruženju Appleovog Macintosha. U međuvremenu Microsoft je razvio i sopstveni operativni sistem, Windows, koji se izvršavao preko MS-DOS-a, za slučaj da OS/2 ne bude prihvaćen. Da skratimo priču, OS/2 zaista nije zaživeo, IBM i Microsoft su se uz mnogo buke javno „razveli“ i Microsoft se okrenuo Windowsu, i preko njega postigao veliki uspeh. Kako su malečki Intel i još manji Microsoft uspeli da s prestola svrgnu IBM, jednu od najvećih, najbogatijih i najmoćnijih kompanija u istoriji, nešto je što se bez sumnje detaljno proučava u poslovnim školama širom sveta.

Držeći u ruci uspešan čip 8088, Intel je krenuo da pravi njegove naprednije bolje verzije. Posebno je pažnje vredan procesor 386, predstavljen 1985. godine, koji je u suštini bio prvi Pentium. Iako su savremeni Pentiumi mnogo brži od procesora 386, oni su u pogledu arhitekture samo unapređeni procesori 386.

Sredinom osamdesetih godina pojavila se nova jednostavnija i brža arhitektura RISC, i zamenila je staru, komplikovanu arhitekturu CISC. Devedesetih godina pojavili su se superskalarni mikroprocesori. Računari s njima mogli su istovremeno da izvršavaju više instrukcija, često redosledom drugačijim od njihovog redosleda u programu. Pojmove CISC, RISC i superskalarni definisaćemo u poglavlju 2 i objašnjavati ih kroz čitavu knjigu.

Sve do 1992. godine, lični računari su bili 8-bitni, 16-bitni ili 32-bitni. Tada je kompanija DEC predstavila revolucionarni 64-bitni računar Alpha, pravu 64-bitnu RISC mašinu koja je po performansama daleko iza sebe ostavila sve tadašnje lične računare. Alpha je postigla skroman uspeh i prošla je skoro decenija dok 64-bitni računari nisu bili na pravi način prihvaćeni, a i tada uglavnom samo kao vrhunski serveri.

1.2.6 Peta generacija – nevidljivi računari

Godine 1981. japanska vlada je saopštila da je odvojila 500 miliona dolara kao pomoć domaćim kompanijama koje razvijaju računare pete generacije – zasnovane na primeni veštačke inteligencije, što će predstavljati pravi kvantni skok u odnosu „glupave“ računare četvrte generacije. Gledajući godinama kako japanske kompanije malo po malo preuzimaju tržište u mnogim oblastima (od kamera, preko stereo-uređaja do televizije), američke i evropske proizvođače računara u trenu je zahvatila panika i svi su od svojih vlada odmah zatražili finansijsku pomoć za ostvarivanje istog cilja. Uprkos pompeznoj najavi, japanski projekat računara pete generacije u osnovi je propao i tiho je napušten. S njim se u izvesnom smislu dogodilo što i s Babbageovom analitičkom mašinom – ideja je bila vizionarska, ali toliko ispred svog vremena da tehnologija nije mogla da je realizuje.

Bilo kako bilo, ono što se može nazvati petom generacijom ipak se dogodilo i to na prilično neočekivan način: računari su se smanjili. Appleov Newton, predstavljen 1993. godine, pokazao je da se može napraviti računar velik koliko prenosivi kasetofon. Korisnik je u Newton unosio podatke pišući ih rukom, što se pokazalo priličnom preprekom, ali su kasniji računari ove klase, nazvani **lični digitalni pomoćnici** (engl. *Personal Digital Assistants*, PDAs), dobili poboljšano korisničko okruženje i postali veoma popularni. Mnogi od njih su sada moćni koliko i lični računari od pre nekoliko godina.

Međutim, čak ni lični digitalni pomoćnici nisu stvarno revolucionarni. U tom smislu veći utisak ostavljaju „nevidljivi“ računari ugrađeni u aparate, satove, kreditne kartice i u brojne druge uređaje (Bechini et al., 2004). Takvi procesori u mnogim primenama obezbeđuju povećanu funkcionalnost uz niže troškove. Diskutabilno je da li oni stvarno predstavljaju novu generaciju čipova (svuda su oko nas od sedamdesetih godina), ali ipak su potpuno izmenili način rada hiljada aparata i drugih uređaja. Njihov uticaj već postaje globalan i u narednim godinama će i dalje rasti. Jedan neobičan aspekt ovih ugrađenih računara jeste to što su njihov hardver i softver **uporedo projektovani** (engl. *codesigned*) (Henkel et al., 2003). Na njih ćemo se vratiti kasnije.

Ako kao prvu generaciju smatramo računare sa elektronskim cevima (npr. ENIAC), kao drugu tranzistorizovane računare (npr. IBM 7094), kao treću prve računare sa integrisanim kolima (npr. IBM 360), a kao četvrtu lične računare (npr. sa Intelovim procesorima), peta generacija bi više predstavljala promenu njihove namene nego neku specifičnu novu arhitekturu. Računari će u budućnosti biti svuda, ugrađeni u razne proizvode, i zaista – nevidljivi. Oni će predstavljati rutinski deo svakodnevice, počev od otvaranja vrata, paljenja (gašenja) svetla, pa do manipulacije novcem i hiljada drugih stvari. Ovaj model je predložio pokojni Mark Weiser i nazvao ga **sveprisutno računarstvo** (engl. *ubiquitous computing, pervasive computing*) (Weiser, 2002). Smatra se da će on promeniti svet baš kao što ga je izmenila industrijska revolucija. Na ovom mestu nećemo više govoriti o sveprisutnom računarstvu, a detalje o njemu potražite kod Lyytinen i Yooa, 2002; Sahae i Mukherjeeja, 2003; i kod Sakamure, 2002.

1.3 RAČUNARSKI ZVERINJAK

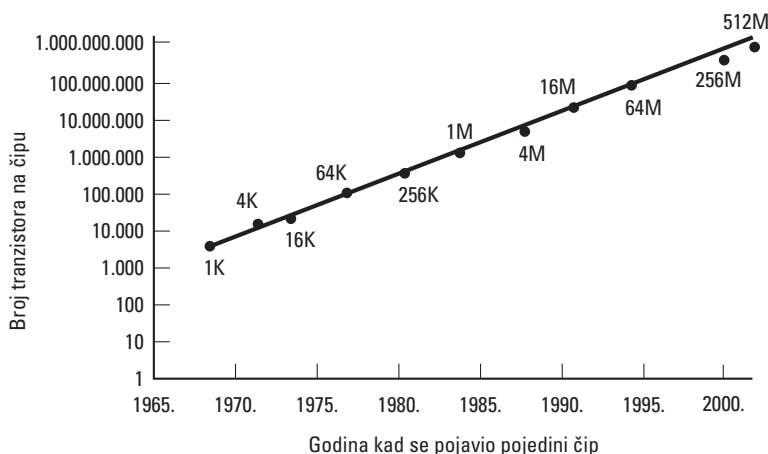
U prethodnom odeljku ukratko smo prikazali prošlost računarskih sistema. U ovom ćemo pretesti njihovu sadašnjicu i baciti pogled u budućnost. Iako su najpoznatiji lični računari, danas postoje i druge njihove vrste, pa ih treba pogledati.

1.3.1 Tehnološke i ekonomske sile

Industrija računara napreduje brže od bilo koje druge industrije. Njena pokretačka snaga su proizvođači čipova koji svake godine uspevaju da smeste na čip sve više i više tranzistora. Što više tranzistora, tih elektronskih prekidača, to je veća memorija, odnosno snažniji procesor. Gordon Moore, suosnivač i bivši predsednik Intela, jednom se ovako našalio – da se avionska industrija razvija kao industrija računara, napravili bi avion koji obleti zemlju za 20 minuta uz potrošnju od dvadesetak litara goriva i košta samo 500 dolara, a ne bi bio veći od kutije za cipele.

Dok je pripremao govor za jednu industrijsku grupaciju, Moore je primetio da se nova generacija memorijskih čipova po pravilu uvodi svake treće godine. Pošto nova generacija uvek donosi četiri puta više memorije od prethodne, on je zapazio da se broj tranzistora na čipu povećava linearno i da će se takav trend nastaviti decenijama. Ovo njegovo zapažanje postalo je poznato kao **Mooreov zakon**. Danas se on tumači tako da se broj tranzistora na čipu udvostručava svakih 18 meseci, što je ekvivalentno godišnjem povećanju broja tranzistora za oko 60%. Veličina memorijskih čipova u zavisnosti od datuma njihovog uvođenja u proizvodnju (slika 1-8) potvrđuje važenje Mooreovog zakona tokom perioda od preko tri decenije.

Mooreov zakon nije teorijski izveden – to je empirijsko zapažanje brzine kojom fizičari i procesni inženjeri unapređuju svoju specifičnu tehnološku oblast, a koje omogućava da se i u budućnosti očekuje sličan trend. Neki analitičari industrijskog razvoja smatraju da će se važenje Mooreovog zakona produžiti barem tokom još jedne decenije. Tada će tranzistori biti toliko mali da će postati nepouzdati, mada razvoj



Slika 1-8. Mooreov zakon predviđa da godišnje povećanje broja tranzistora na čipu iznosi 60%. Tačke na dijagramu predstavljaju veličinu memorije u bitovima.

kvantnog računarstva to može suštinski da izmeni (Oskin et al., 2002). Drugi analitičari, međutim, očekuju da će se mnogo ranije pojaviti problemi s disipacijom toplote i „curenjem“ struje te drugi nepoželjni efekti koje će biti potrebno rešavati (Bose, 2004; Kim et al., 2003).

Mooreov zakon ostvaruje ono što bi ekonomisti nazvali **povoljno kruženje** (engl. *virtuous circle*). Napredovanje tehnologije (tranzistora/čipova) donosi bolje proizvode i niže cene. Niže cene vode novim primenama (niko nije pravio računarske igrice onda kada je računar koštao 10 miliona dolara). Nove primene otvaraju nova tržišta i podstiču osnivanje novih kompanija da ih iskoriste. Nove kompanije znače povećanu konkurenciju, što podstiče ekonomsku potrebu za boljim tehnologijama. Time se krug zatvara.

Drugi činilac tehnološkog razvoja je Nathanov prvi zakon softvera (po Nathanu Myhrvoldu, bivšem Microsoftovom visokom rukovodiocu). On tvrdi: „Softver je kao gas koji se širi dok ne ispuni sud u kome se nalazi“. Davnih osamdesetih godina, tekst je obrađivan programima kao što je troff (korišćen je i za ovu knjigu). Troff zauzima nekoliko kilobajta memorije. Savremeni programi za obradu teksta zauzimaju megabajte memorije. Za naredne će, nema sumnje, biti potrebni gigabajti. (Prefiksi kilo, mega i giga u prvoj aproksimaciji znače hiljadu, milion i milijardu, a detalje pogledajte u odeljku 1.5.) Softver kome se stalno dodaju nove mogućnosti (slično broskom koritu na koje se neprestano lepe školjke) stalno traži sve brži procesor, više memorije i više ulazno-izlaznih kapaciteta.

Iako je brzina rasta broja tranzistora po čipu već godinama dramatična, ni razvoj drugih računarskih tehnologija nije bio sporiji. Na primer, 1982. godine predstavljen je računar IBM PC/XT sa čvrstim diskom od 10 megabajta. Dvadeset godina kasnije, naslednici ovog računara već uobičajeno imaju disk od 100 gigabajta. Priraštaj od četiri reda veličine za 20 godina znači godišnji priraštaj od 58%. Izražavanje poboljšanja u tehnologiji proizvodnje diskova samo njihovim kapacitetom, varljivo je

jer su u igri i drugi parametri, kao što su brzina prenosa podataka, vreme njihovog pronalaženja i cena. Gotovo svaka primenjena metrika pokazaće da se odnos cena/performance od 1982. godine povećavao po godišnjoj stopi od barem 50%. Ovakav rast performansi diskova, zajedno sa činjenicom da je vrednost diskova koji se isporučuju iz Silicijumske doline prevazišla vrednost isporučenih procesorskih čipova, naveo je Ala Hoaglanda da posumnja u ispravnost njenog imena: trebalo bi je nazvati Dolinom oksida gvožđa (pošto se na diskovima on koristi kao medijum za zapisivanje podataka).

Još jedna oblast koja se spektakularno razvijala jesu telekomunikacije i tehnologija umrežavanja. Za manje od dve decenije, od modema brzine 300 bitova/s, preko analognih modema brzine 56.000 bitova/s, stigli smo do mreža od optičkih vlakana brzine 10^{12} bitova/s. Transatlantski optički telefonski kablovi, TAT-12/13, koštaju oko 700 miliona dolara, traju 10 godina i mogu istovremeno da prenesu 300.000 poziva po ceni manjoj od centa za 10-minutni međukontinentalni razgovor. Dokazana je mogućnost rada optičkih komunikacionih sistema brzine prenosa 10^{12} bitova/s na daljinama od preko 100 km bez pojačavača. Eksponencijalni rast Interneta ne treba ni pominjati.

1.3.2 Spektar računara

Richard Hamming, svojevremeno istraživač u Belovim laboratorijama, jednom prilikom je primetio da kvantitativna promena od jednog reda veličine izaziva kvalitativan skok. Tako trkački automobil koji juri brzinom 1000 km/h kroz pustinju u Nevadi predstavlja sasvim drugu vrstu automobila od normalne mašine koja se autoputem kreće brzinom 100 km/h. Slično tome, oblakoder od 100 spratova nije samo uvećana stambena zgrada od 10 spratova. Kada govorimo o računarima, ne mislimo na desetostruki faktor uvećanja, već na uvećanje od milion puta tokom tri decenije.

Rast koji predviđa Mooreov zakon može se iskoristiti na više načina. Jedan je pravljenje sve jačih računara po istoj ceni. Drugi je pravljenje istog računara koji će iz godine u godinu koštati sve manje. Računarska industrija je iskoristila i jedan i drugi pristup, a i više od toga, zbog čega danas imamo veoma različite računare. Jedna sasvim gruba klasifikacija današnjih računara prikazana je na slici 1-9.

Vrsta	Cena (USD)	Primer primene
Za jednokratnu upotrebu	0,5	Čestitke
Mikrokontroler	5	Ručni satovi, automobili, kućni uređaji
Računar za igranje	50	Kućne video igrice
Lični računar	500	Stoni ili prenosivi računar
Serverski	5000	Mrežni server
Skup radnih stanica	50.000–500.000	Poslovni mini superračunar
Centralni računar	5.000.000	Paketna obrada podataka u banci

Slika 1-9. Današnji spektar raspoloživih računara. Cene treba primiti s velikom rezervom.

U narednim odeljcima opisaćemo svaku od navedenih kategorija i ukratko objasniti njene osobine.

1.3.3 Računari za jednokratnu upotrebu

Sasvim u dnu lestvice nailazimo na pojedinačne čipove nalepljene sa unutrašnje strane čestitki, koji reprodukuju melodiju „Srećan rođendan...“, „Evo neveste...“ ili neku drugu prigodnu pesmicu. Nisam još naleteo na luksuzni telegram sa izjavom saučešća koji svira posmrtni marš, ali izgleda da ni na to nećemo dugo čekati. Za svakoga ko je odrastao uz centralni računar od više miliona dolara, sama ideja o računaru za jednokratnu upotrebu ima smisla koliko i avion za jednokratnu upotrebu.

Međutim, računari za jednokratnu upotrebu već su oko nas i tu će i ostati. Najvažnije otkriće na polju ovakvih računara verovatno je čip za **radiofrekventnu identifikaciju** (engl. *Radio Frequency Identification*, RFID). Danas se za nekoliko centi može napraviti RFID čip tanji od 0,5 mm koji sadrži miniijturni primopredajnik i jedinstven 128-bitni broj, a radi bez baterije. Kada od spoljne antene primi signal, taj energetski podsticaj je dovoljan da čip anteni emituje svoj identifikacioni broj. Iako je čip tako sićušan da se nema šta videti, posledice njegove primene nisu nimalo naivne.

Počnimo od jedne svakodnevne primene: uklanjanja bar-kodova s proizvoda. Već su obavljani eksperimenti u kojima su proizvodi u samouslugama umesto bar-kodova bili snabdeveni RFID čipovima. Kupac bi izabrao proizvode, stavio ih u kolica i progurao kolica kroz prolaz pored kontrolne „kase“. Tu bi čitač sa antenom poslao signal tražeći od proizvoda da se identifikuju, što su oni i činili emitujući povratne signale. Kupac bi se takođe identifikovao preko čipa na svojoj bankovnoj ili kreditnoj kartici. Na kraju meseca, samousluga bi kupcima slala detaljne račune za proizvode kupljene u tom mesecu. Ako kupac ne bi imao važeću bankovnu ili kreditnu RFID karticu, oglasio bi se alarm. Ovaj sistem ne samo da otklanja potrebu za kasirkom i čekanjem u redu, već obezbeđuje samouslugu i štiti od krađe jer se artikli skriveni po džepovima i torbama takođe identifikuju.

Zanimljivo je da se ovim sistemom, uz 128-bitni broj, može pojedinačno identifikovati svaki artikal, dok se bar-kodom identifikuje samo tip artikla. To znači da svaka kutija, recimo, aspirina na polici u nekoj superapoteci ima drugačiji RFID kôd. Ukoliko bi proizvođač aspirina naknadno otkrio neku grešku u proizvodnji, morao bi da svim superapotekama širom sveta naloži da sistem podese tako da se uključi alarm kad neko kupi pakovanje aspirina čiji RFID broj pripada određenom opsegu, čak i ako se to dogodi mesecima kasnije u nekoj udaljenoj zemlji. Aspirini izvan defektne grupe ne bi pozivali na uzbunu.

Obeležavanje pakovanja aspirina, keksa i hrane za pse, međutim, tek je početak. Zašto se zaustaviti na hrani za pse, kada možemo da obeležimo psa? Vlasnici kućnih ljubimaca već zahtevaju od veterinara da njihovim životinjama ugrade RFID čip kako bi ih lakše pronašli ako se izgube ili ako ih neko ukrade. I stočari žele da obeleže svoja grla. Očigledan sledeći korak su uvek brižni roditelji koji od pedijatar traže da njihovoj deci ugrade RFID čip za slučaj da ih neko kidnapuje ili se sama izgube. Kada

smo već kod toga, zašto se novorođenoj deci ne bi odmah ugrađivali čipovi i tako izbegla zbrka sa zamenom beba u porodilištima? Vlada i policija bi, nema sumnje, našli stotine razloga za neprestano i istovremeno praćenje kretanja svih građana. Nadamo se da su posledice mogućih „primena“ RFID čipa sada malo jasnije.

Jedna (nešto manje kontroverzna) primena RFID čipova mogla bi biti praćenje kretanja vozila. Kada kompozicija železničkih vagona sa ugrađenim RFID čipovima prođe pored čitača, računar vezan za njega može da sastavi listu vlasnika tih vagona. Takav sistem omogućava saznavanje lokacije svakog železničkog vagona, što pomaže snabdevačima, njihovim mušterijama i železnici. Sličan sistem se može primeniti i na kamionski transport. Kod putničkih automobila sistem bi mogao naći primenu za elektronsko naplaćivanje putarine.

Transport prtljaga avionom i drugi sistemi transporta prtljaga takođe mogu imati koristi od RFID čipova. Eksperimentalni sistem koji je testiran na londonskom aerodromu Hitrou omogućio je putnicima da se oslobode nošenja teškog prtljaga. Koferi putnika koji su se opredelili za ovu uslugu obeleženi su RFID čipovima, upućeni posebnom rutom kroz aerodromsku zgradu i direktno isporučeni u njihove hotele. U druge primene RFID čipova spadaju: obeležavanje automobila na proizvodnoj traci prema boji kojom treba da budu ofarbani, proučavanje migracija životinja, odeća koja sama saopštava veš-mašini temperaturu pranja i štošta drugo. Neki čipovi se mogu integrisati sa senzorima tako da se u njihove manje značajne bitove beleže temperatura, pritisak, vlažnost i drugi parametri okoline.

Napredniji RFID čipovi imaju i stalnu memoriju. Ta mogućnost je navela Evropsku centralnu banku da RFID čipove umeće i u novčanice evra. Čipovi treba da beleže mesta gde su bili. To će falsifikovanje evra učiniti gotovo nemogućim, ali će omogućiti i praćenje opranog novca i novca od ucena i pljački, i možda omogućiti njegovo daljinsko obezvređivanje. Kada novac više ne bude anoniman, standardna policijska procedura u budućnosti biće utvrđivanje gde je neki sumnjivi novac prethodno boravio. Zašto treba ugrađivati čipove u ljude, kada su njihovi džepovi puni čipova? Naglašavamo, kad javnost na pravi način sazna šta se sve može postići pomoću RFID čipova, verovatno će to pobuditi izvesnu raspravu.

Tehnologija proizvodnje RFID čipova se brzo razvija. Najmanji čipovi su pasivni (ne koriste baterije) i svoj jedinstveni broj emituju samo kad postoji antenski signal. Veći čipovi su aktivni, imaju bateriju i rudimentaran računar, i mogu da obave određena izračunavanja. Inteligentne kartice koje se koriste u finansijskim transakcijama spadaju u ovu kategoriju.

RFID čipovi se razlikuju po tome da li su pasivni ili aktivni, ali i po opsegu radio-frekvencija na koje se odazivaju. Oni koji rade na nižim frekvencijama odlikuju se malom brzinom prenosa podataka, ali su osetljiviji na antenski signal. Kod onih koji rade na višim frekvencijama situacija je obrnuta: brzina prenosa podataka je veća, ali je domet otkrivanja kraći. Čipovi se razlikuju još po mnogo čemu i sve vreme se izmišlja nešto novo. Internet je prepun informacija o RFID čipovima, a www.rfid.org je dobra polazna adresa za istraživanje.

1.3.4 Mikrokontroleri

Sledeći na lestvici su računari koji se ugrađuju u uređaje i koji se inače ne prodaju kao računari. Ugrađeni računari, ponekad zvani **mikrokontroleri**, upravljaju uređajem i komuniciraju s korisnikom. Mikrokontroleri se mogu naći u različitim uređajima, uključujući i uređaje s narednog spiska. U zagradi je navedeno nekoliko konkretnih primera za svaku kategoriju.

1. Kućni aparati (radio-sat, veš-mašina, mašina za sušenje veša, mikrotalasna pećnica, alarm).
2. Komunikacioni uređaji (bežični telefon, mobilni telefon, faks, pejdžer).
3. Računarski periferni uređaji (štampač, skener, modem, CD uređaj).
4. Uređaji za razonodu (video-rikorder, DVD, muzički stub, MP3 plejer, lokalni TV konvertor).
5. Uređaji za rad sa slikama (televizor, digitalni fotoaparat, kamkorder, aparat za fotokopiranje).
6. Medicinski uređaji (rentgen, magnetna rezonanca, kardio-holter, digitalni termometar).
7. Oružje (krstareće rakete, interkontinentalni balistički projektili, torpeda).
8. Uređaji za prodavnice (automati koji rade na novčiće, kase).
9. Igračke (lutka koja govori, igračka konzola, teledirigovani automobilčić ili brodić).

Neki sasvim čudan automobil mogao bi sadržati i do 50 mikrokontrolera koji upravljaju njegovim podsistemima, kao što su sistem za deblokiranje kočnica, ubrizgavanje goriva, radio i GPS uređaj. Mlaznjak bi ih mogao imati i više od 200. U jednoj porodičnoj kući, takvih mikrokontrolera moglo bi biti i više stotina a da ih ukućani i ne budu svesni. Za nekoliko godina praktično će sve što radi na struju ili baterije imati mikrokontroler. Broj prodatih mikrokontrolera iz godine u godinu premašuje broj svih drugih prodatih računara (osim onih za jednokratnu upotrebu) za više redova veličine.

Dok su RFID čipovi zaista minimalni sistemi, mikrokontroleri su mali, ali potpuni računari. Svaki mikrokontroler ima procesor, memoriju i ulazno-izlazne mogućnosti. Ulazno-izlazne mogućnosti obično obuhvataju registrovanje uključenosti dugmadi i prekidača uređaja i upravljanje njegovim svetlosnim i zvučnim signalima, ekranom i motorom. Softver, u većini slučajeva u obliku memorije samo za čitanje, ugrađen je u čip još tokom proizvodnje. Mikrokontroleri spadaju u jednu od dve opšte klase: za opštu upotrebu i za specijalne namene. U prvu klasu spadaju mali, ali uobičajeni računari; mikrokontroleri druge klase imaju arhitekturu i skup instrukcija koji su specijalno prilagođeni određenoj aplikaciji (na primer, multimediji). Postoje 4-bitne, 8-bitne, 16-bitne i 32-bitne verzije mikrokontrolera.

Ipak, čak se i mikrokontroleri opšte namene znatno razlikuju od standardnih PC računara. Njihova cena je veoma važna. Kompanija koja nabavlja milione takvih proizvoda može da promeni dobavljača samo zbog razlike od jednog centa po komadu. To proizvođače mikrokontrolera tera da arhitekturu biraju rukovodeći se mnogo

više proizvodnim troškovima nego cenom čipa koji košta na stotine dolara, premda cena mikrokontrolera uveliko zavisi i od toga s koliko bitova istovremeno radi, koliko memorije ima, i od drugih činilaca. Da biste stekli bar neku predstavu, reći ću da 8-bitni mikrokontroler (ako ih odjednom kupite dovoljno) verovatno možete dobiti za oko 10 centi po komadu. Zbog takve cene možete da ugradite računar u radio-sat koji košta samo 10 dolara.

Naizgled svi mikrokontroleri rade u realnom vremenu. Kada dobiju podsticaj, očekuje se da odmah reaguju. Na primer, kada korisnik pritisne neko dugme, često se upali signalna lampica i ne sme postojati nikakva neobjašnjiva vremenska zadržka između pritiskanja dugmeta i paljenja lampice. Potreba da se radi u realnom vremenu često utiče na izbor arhitekture.

Za sisteme u koje se ugrađuju mikrokontroleri često se postavljaju fizička ograničenja u pogledu veličine, težine, potrošnje energije i sl. Zbog toga se pri projektovanju njihovih mikrokontrolera to mora imati u vidu.

1.3.5 Računari za igranje

Na sledećoj stepenici nalaze se računari za igranje. To su obični računari, s posebnim grafičkim i zvučnim mogućnostima, ali sa ograničenim softverom i malim mogućnostima za proširenje. Oni su nastali od jeftinih procesora namenjenih igranju akcionih igara, kao što je ping-pong, na TV ekranu. Tokom godina su se razvili u mnogo moćnije sisteme koji su stali u isti red s ličnim računarima ili ih čak nadmašili u nekim aspektima.

Da biste stekli predstavu o tome šta se nalazi unutar računara namenjenih za igranje, razmotrite specifikacije tri popularna proizvoda. Sonyjev PlayStation 2 sadrži 128-bitni Sonyjev procesor (nazvan Emotion Engine) na 295 MHz, koji se zasniva na RISC mikroprocesoru MIPS IV. PlayStation 2 ima i 32 MB memorije, specijalan grafički čip brzine 160 MHz, specijalan 48-kanalni audio čip i DVD plejer. Microsoftov XBOX sadrži Intelov procesor Pentium III na 733 MHz sa 64 MB memorije, specijalan grafički čip brzine 300 MHz, specijalan 256-kanalni audio čip, DVD plejer i čvrsti disk od 8 GB. Nintendov GameCube sadrži specijalan 32-bitni mikroprocesor (nazvan Gekko) na 485 MHz, izveden iz IBM-ovog RISC procesora PowerPC, 24 MB memorije, specijalni grafički čip brzine 200 MHz, 64-kanalni audio čip i Microsoftov optički disk od 1,5 gigabajta.

Iako ove mašine nisu tako moćne kao lični računari proizvedeni u isto vreme, one ne zaostaju mnogo za njima, a u izvesnim aspektima su i bolje (npr. 128-bitni mikroprocesor u konzoli PlayStation 2 ima veće registre od mikroprocesora bilo kog PC računara, iako mu je radni takt mnogo niži). Glavna razlika između igračkih mašina i PC računara i nije mikroprocesor, nego činjenica da su igračke mašine zatvoreni sistemi. Korisnici ih ne mogu proširivati dodatnim karticama, premda neki imaju USB priključak ili FireWire vezu. Što je još važnije, mašine za igru su pažljivo optimizovane za jednu jedinu namenu: visokointeraktivne 3D igrice uz visokokvalitetan stereo zvuk. Sve ostalo je sporedno. Zbog navedenih ograničenja u pogledu hardvera i

softvera, malog radnog takta, male količine memorije, nepostojanja monitora visoke rezolucije i (po pravilu) nepostojanja čvrstog diska, ove mašine su mnogo jeftinije od ličnih računara. Uprkos navedenim ograničenjima, do danas je prodato na milione ovakvih mašina.

Kompanije koje proizvode velike igračke mašine, prave i prenosive verzije koje rade na baterije. One su ipak bliže ugrađenim sistemima o kojima smo ranije govorili nego ličnim računarima.

1.3.6 Lični računari

Stigli smo do ličnih računara na koje većina ljudi pomisli kad čuje reč „računar“. U njih spadaju stoni i prenosivi modeli. Obično se isporučuju sa stotinama megabajta memorije, čvrstim diskom na koji može da se smesti oko 100 gigabajta podataka, kombinovanim CD/DVD-uređajem, modemom, zvučnom i mrežnom karticom, monitorom visoke rezolucije i drugim perifernim komponentama. Oni imaju složen operativni sistem, brojne mogućnosti za proširivanje i gomilu softvera na raspolaganju. Neki pod izrazom „PC računar“ podrazumevaju mašine sa Intelovim mikroprocesorom, a „radnim stanicama“ nazivaju računare opremljene skupim RISC čipovima, kao što je Sun-ov UltraSPARC. Konceptijski, razlika između njih je minimalna.

Srce svakog ličnog računara je (matična) ploča sa štampanim kolima u dnu kućišta. Ona obično sadrži mikroprocesor, memoriju, razne ulazno-izlazne uređaje (kao što su zvučna kartica i modem), kao i priključke za tastaturu, miša, disk, mrežu i nekoliko podnožja za proširivanje. Na slici 1-10 prikazana je jedna ploča.

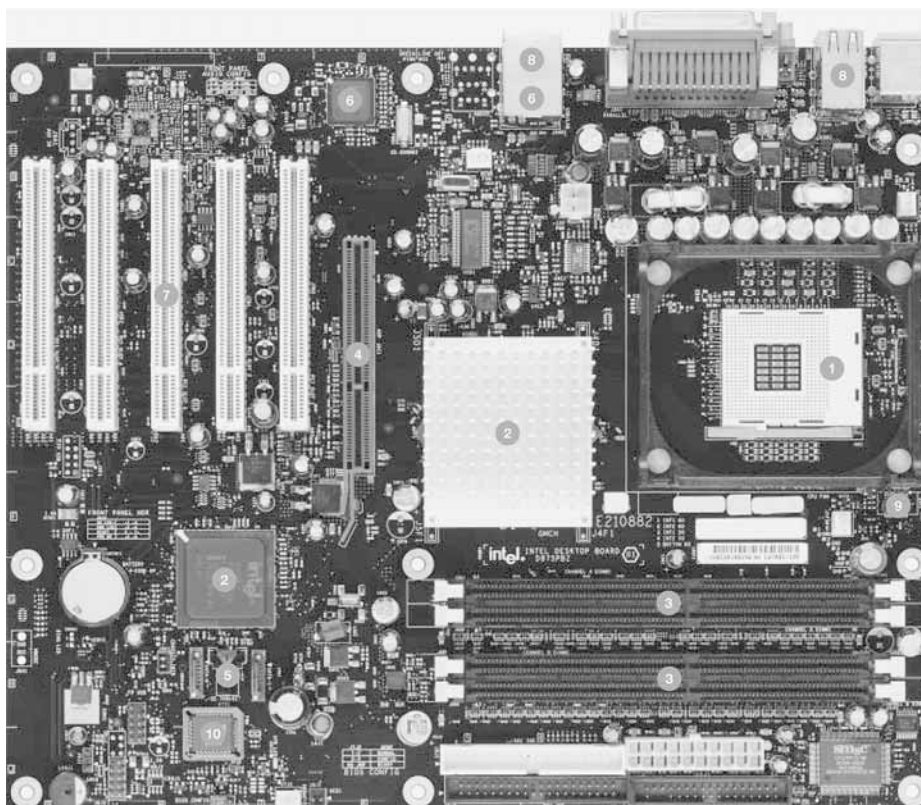
Prenosivi računar (laptop) u osnovi je PC računar u manjem pakovanju. U njemu su iste hardverske komponente, samo manje. Na prenosivim računarima radi isti softver kao i na stonim računarima.

Još jedan srodan uređaj je i lični digitalni pomoćnik (PDA). Iako je on manji čak i od prenosivog računara, ima mikroprocesor, memoriju, tastaturu, ekran i većinu drugih elemenata ličnog računara. Pošto je većina čitalaca dobro upoznata s ličnim računarima, u uvodu nećemo o njima više ništa reći.

1.3.7 Serveri

„Nabudženi“ lični računar ili radna stanica često služi kao mrežni server u lokalnoj mreži (po pravilu, unutar male kompanije) ili na Internetu. Serveri mogu imati jedan ili više procesora, gigabajte memorije, stotine gigabajta na čvrstom disku i mogućnost brzog rada u mreži. Neki od njih mogu da obavljaju na hiljade transakcija u sekundi.

Ipak, iz perspektive arhitekture, server s jednim procesorom ne razlikuje se previše od ličnog računara s jednim procesorom. On je samo brži, veći, ima više prostora na disku i možda brži pristup mreži. Serveri rade pod istim operativnim sistemima kao i lični računari, najčešće pod nekom varijantom UNIX-a ili Windowsa.



Slika 1-10. Štampana matična ploča nalazi se u središtu svakog ličnog računara. Na slici je fotografija Intelove matične ploče D875PBZ. Sva prava na fotografiju zadržava korporacija Intel, 2003, a ovde je upotrebljena uz njenu dozvolu.

- | | | |
|-----------------------------------|-----------------------|-----------------------|
| 1. Podnožje za procesor Pentium 4 | 5. Priključak za disk | 8. USB 2.0 priključci |
| 2. Čip 875P za podršku | 6. Gigabitni Ethernet | 9. Hlađenje |
| 3. Podnožja za memorijske čipove | 7. Pet PCI utičnica | 10. BIOS |
| 4. AGP utičnica | | |

1.3.8 Skup radnih stanica

Zbog sve povoljnijeg odnosa cena/performance kod radnih stanica i ličnih računara, konstruktori sistema su od nedavno počeli da ih uveliko povezuju u **skupove radnih stanica** (engl. *Clusters of Workstations*, COWs), ponekad zvane **klasteri** (engl. *clusters*). Oni su sastavljeni od standardnih ličnih računara ili radnih stanica povezanih gigabitnim mrežama i izvršavaju specijalan softver koji svim računarima u skupu omogućava da rade na zajedničkom problemu, često iz oblasti nauke ili inženjerstva. Ti računari su obično ono što bismo nazvali **potrošna roba** (engl. *Commodity Off The Shelf*, COTS), tj. mogu se nabaviti od svakog proizvođača računara kao dnevne novine.

Glavna razlika je u mrežnom priključku velike brzine, ali je ponekad i to standardna komercijalna mrežna kartica. Klasteri se lako mogu proširivati, od samo nekoliko, pa do hiljadu računara. Obično je samo novac ograničavajući činilac. Zahvaljujući niskoj ceni komponenata, sada i pojedinačna odeljenja u organizacijama mogu da imaju takve računare.

Konfiguracija COW se koristi i za Web servere na Internetu. Kada na Web lokaciju stiže hiljade zahteva u sekundi, često je najekonomičnije rešenje džinovski klaster od stotinu, pa čak hiljadu servera. Dolazni zahtevi se tada raspodeljuju na više servera kako bi se istovremeno mogli obraditi. Kada se koristi na ovaj način, COW se često zove i **farma servera** (engl. *server farm*).

1.3.9 Centralni računari

Evo nas kod centralnih računara (engl. *mainframes*) – računara veličine sobe koji postoje još od šezdesetih godina. U mnogim slučajevima oni su direktni potomci IBM-ovih centralnih računara iz serije 360 od pre nekoliko decenija. Uglavnom nisu mnogo brži od moćnih servera, ali uvek imaju veći ulazno-izlazni kapacitet i često su opremljeni prostranim farmama diskova na koje se mogu smestiti hiljade gigabajta podataka. Iako su skupi, često se održavaju u radnom stanju da bi se opravdala ogromna ulaganja u softver, podatke, radne procedure i osoblje. Mnogim kompanijama je jeftinije da povremeno plaćaju par miliona dolara za nov centralni računar, nego da reprogramiraju sve svoje aplikacije za rad na manjim računarima.

Ovo je klasa računara koja je prouzrokovala čuveni problem „2000. godina“ jer su programeri na COBOL-u iz šezdesetih i sedamdesetih godina godinu u datumu predstavljali pomoću dve decimalne cifre (u nameri da uštede memoriju). Nisu ni pomišljali da će njihov softver živeti tri ili četiri decenije. Iako se predviđana katastrofa nije dogodila zahvaljujući ogromnom trudu uloženom u rešavanje problema, mnoge kompanije su ponovile istu grešku tako što su dodale godini još dve cifre. Zato sam slobodan da predvidim kako će se 31. decembra 9999. godine okončati civilizovan svet kakav poznajemo. Toga dana će istovremeno otkazati svi, 8000 godina stari programi napisani na COBOL-u.

Uz to što se na njima izvršava softver star preko 30 godina, recimo i to da od nedavno, zahvaljujući Internetu, centralnim računarima nadene su nove namene. Sada rade kao moćni Internet serveri, na primer, za obradu ogromnog broja transakcija elektronske trgovine u sekundi, naročito u poslovnim oblastima gde se barata sa ogromnim bazama podataka. Iako su glavne teme ove knjige lični računari, serveri i mikrokontroleri, u poglavlju 5 još malo ćemo govoriti i o centralnim računarima.

Do nedavno je postojala još jedna kategorija računara, čak mnogo moćnija od centralnih računara, a to su **superračunari** (engl. *supercomputers*). Oni su imali izuzetno brze mikroprocesore, mnogo gigabajta radne memorije i vrlo brze diskove i mreže. Korišćeni su za obimna naučna i tehnička izračunavanja, kao što su simuliranje sudara galaksija, pravljenje novih lekova ili modelovanje toka vazduha oko avionskih krila. Međutim, poslednjih godina klasteri računara obezbeđuju istu računarsku snagu po mnogo nižoj ceni, pa su pravi superračunari na pragu izumiranja.

1.4 ODABRANI PRIMERI PORODICA RAČUNARA

U ovoj knjizi ćemo obraditi tri vrste računara: lične računare, servere i ugrađene računare. Lični računari nas zanimaju jer svaki čitalac nesumnjivo ima jedan takav. Serveri nas zanimaju jer izvršavaju sve usluge na Internetu. Najzad, ugrađeni računari, iako skriveni od pogleda korisnika, upravljaju automobilima, televizorima, mikro-talasnim pećnicama, veš-mašinama i skoro svim drugim električnim uređajima koji koštaju više od 50 dolara.

U ovom odeljku, ukratko ćemo predstaviti tri računara koje ćemo koristiti kao primere kroz čitavu knjigu, po jedan iz svake od tri kategorije. To su računari s procesorima Pentium 4, UltraSPARC III i 8051.

1.4.1 Uvod u Pentium 4

Robert Noyce, pronalazač silicijumskog integrisanog kola, Gordon Moore (čuven po svom zakonu) i Arthur Rock, investitor iz San Franciska, 1968. godine osnovali su kompaniju Intel da bi proizvodili memorijske čipove. U prvoj godini poslovanja Intel je prodao čipove u vrednosti od samo 3000 dolara, ali je posle toga posao krenuo.

Krajem šezdesetih godina kalkulatori su bili nezgrapne elektromehaničke mašine veličine sadašnjeg laserskog štampača i teške dvadesetak kilograma. U septembru 1969. godine, japanska kompanija Busicom uputila je zahtev Intelu da za nju proizvede 12 specijalnih čipova za ugradnju u budući elektronski kalkulator. Intelov inženjer Ted Hoff koji je bio zadužen za ovaj projekat zaključio je da će 4-bitni procesor opšte namene smešten na jedan jedini čip obaviti isti posao uz manje komplikacija i troška. Tako je 1970. godine rođen prvi mikroprocesor na jednom čipu – procesor 4004 sa 2300 tranzistora (Faggin et al., 1996).

Treba primetiti da ni Intel ni Busicom nisu bili svesni šta su postigli. Kada je Intel odlučio da ipak pokuša da upotrebi čip 4004 i u drugim projektima, on je ponudio da Busicomu vrati sve što je ova firma uložila u Intel za razvijanje tog čipa (60.000 dolara) kao naknadu za sva prava na njega. Intelova ponuda je prihvaćena i on je odmah počeo da razvija 8-bitnu verziju čipa 8008, koja je predstavljena 1972. godine. Na slici 1-11 prikazana je porodica Intelovih čipova, počevši od prvih: 4004 i 8008.

Intel nije očekivao veliku potražnju čipa 8008, pa je izgradio proizvodnu liniju malog kapaciteta. Potražnja je, na opšte zaprepašćenje, bila iznad svih očekivanja, pa je Intel započeo projektovanje novog mikroprocesora koji bi prevazišao ograničenja čipa 8008 (16 KB memorije) nastala zbog malog broja nožica. Tako je rođen 8080, mali mikroprocesor opšte namene, koji je predstavljen 1974. godine. Slično računaru PDP-8, i ovaj proizvod je munjevito osvojio tržište i odmah postao roba široke potrošnje. Samo, umesto da kao DEC prodaje hiljade komada, Intel je prodavao na milione.

Godine 1978. pojavio se 8086, pravi 16-bitni mikroprocesor na jednom čipu. 8086 je projektovan po uzoru na 8080, ali nije bio potpuno kompatibilan s njim.

Čip	Datum	MHz	Tranzistori	Memorija	Napomene
4004	4/1971	0,108	2300	640	Prvi mikroprocesor na čipu
8008	4/1972	0,108	3500	16 KB	Prvi 8-bitni mikroprocesor
8080	4/1974	2	6000	64 KB	Prvi mikroprocesor opšte namene na čipu
8086	6/1978	5-10	29.000	1 MB	Prvi 16-bitni mikroprocesor na čipu
8088	6/1979	5-8	29.000	1 MB	Korišćen u IBM-ovom PC računaru
80286	2/1982	8-12	134.000	16 MB	Postoji zaštita memorije
80386	10/1985	16-33	275.000	4 GB	Prvi 32-bitni mikroprocesor
80486	4/1989	25-100	1,2M	4 GB	Ugrađenih 8 KB keša
Pentium	3/1993	60-233	3,1M	4 GB	Dva procesna toka; kasniji modeli imaju skup MMX instrukcija
Pentium Pro	3/1995	150-200	5,5M	4 GB	Ugrađena dva nivoa keša
Pentium II	5/1997	233-450	7,5M	4 GB	Pentium Pro plus skup MMX instrukcija
Pentium III	2/1999	650-1400	9,5M	4 GB	SSE instrukcije za 3D grafiku
Pentium 4	11/2000	1300-3800	42M	4 GB	Hipernitni rad; dodatne SSE instrukcije

Slika 1-11. Porodica Intelovih mikroprocesora. Radni takt je izražen u MHz (megahercima), pri čemu 1 MHz označava milion ciklusa u sekundi.

Posle čipa 8086 pojavio se čip 8088 iste arhitekture koji je izvršavao iste programe, ali je umesto 16-bitne imao 8-bitnu magistralu, zbog čega je bio sporiji i jeftiniji od svog prethodnika. Kada je IBM izabrao čip 8088 kao mikroprocesor za svoj prvobitni IBM PC, taj čip je ubrzo postao standard u industriji ličnih računara.

Procesori 8088 i 8086 nisu mogli adresirati više od jednog megabajta memorije. Početkom osamdesetih godina to je postajalo sve ozbiljniji problem, pa je Intel projektovao procesor 80286, kompatibilan s procesorom 8086. Osnovni skup instrukcija u osnovi je bio isti kao kod procesora 8086 i 8088, ali je organizacija memorije bila sasvim drugačija – prilično nezgrapna, zbog zahteva za kompatibilnošću sa starijim čipovima. Procesor 80286 iskorišćen je za računare IBM PC/AT i za osrednje modele PS/2. Slično procesoru 8088, i model 80286 je postigao veliki uspeh, uglavnom zato što su korisnici smatrali da je to brži procesor 8088.

Sledeći logičan korak bio je pravi 32-bitni procesor, model 80386, predstavljen 1985. godine. Slično procesoru 80286, i on je manje-više bio kompatibilan sa starijim verzijama sve do modela 8080. Ta kompatibilnost bila je blagoslov za korisnike koji su još uvek želeli da na njemu izvršavaju stariji softver, ali i velika smetnja onima koji su težili za jednostavnom, čistom i savremenom arhitekturom, neopterećenom greškama i tehnologijom prošlih vremena.

Nakon četiri godine, pojavio se procesor 80486. To je bila znatno brža verzija procesora 80386, s podjedinicom za računanje u formatu pokretnog zareza i keš memorijom od 8 kilobajta na samom čipu. **Keš memorija** ili jednostavno **keš** (engl. *cache memory*) korišćena je za čuvanje najčešće upotrebljivanih memorijskih reči u samom mikroprocesoru ili u njegovoj neposrednoj blizini kako bi se izbeglo (sporo) pristupanje glavnoj memoriji. Čip 80486 imao je i ugrađenu podršku za višeprocorski rad kako bi se proizvođačima omogućilo da sklapaju sisteme u kojima više procesora deli zajedničku memoriju.

U to doba Intel je na neprijatan način saznao da se brojevi (npr. 80486) ne mogu pravno zaštititi (izgubivši sudski proces u vezi sa zaštitom imena proizvoda), pa je sledeća generacija procesora dobila ime **Pentium** (od grčke reči za pet). Za razliku od procesora 80486, koji je imao jedan unutrašnji kanal za podatke (engl. *pipeline*), Pentium je dobio dva takva kanala pa je bio dvostruko brži (o kanalima za podatke govorićemo u poglavlju 2).

Kasnije, tokom proizvodnje ovog procesora, Intel je dodao i specijalan **skup instrukcija za podršku multimediji** (engl. *MultiMedia eXtension*, MMX). One su bile namenjene ubrzanju obrade zvuka i slike da bi se izbegla ugradnja specijalnog koprocesora za obradu multimedijiskog sadržaja.

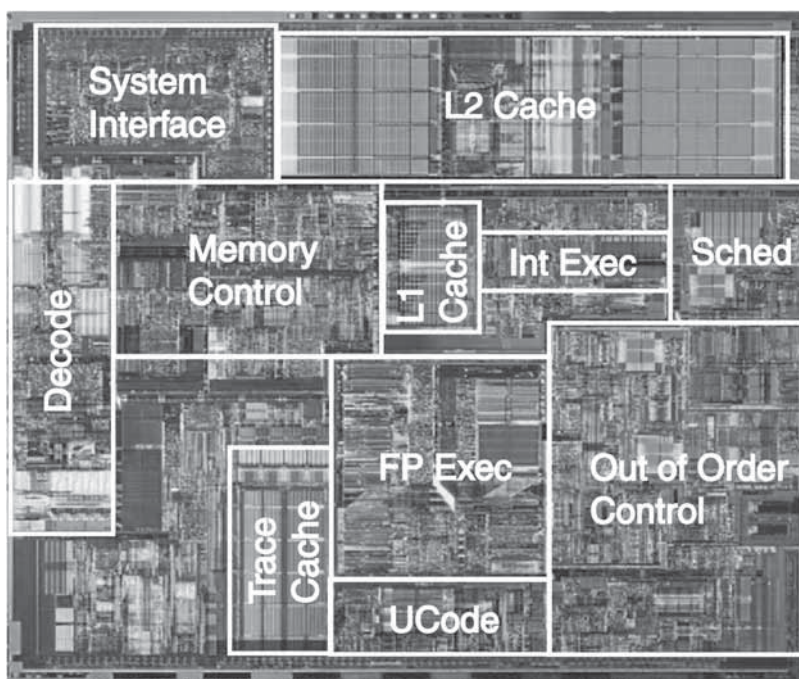
Kada se pojavila sledeća generacija procesora, razočarali su se korisnici koji su očekivali da će to biti „Sexium“ (*sex* je latinska reč za šest). Ime Pentium je već bilo toliko poznato da su marketinški stručnjaci želeli da ga zadrže, pa je novi čip nazvan Pentium Pro. Bez obzira na neznatnu promenu imena, konstrukcija je bila radikalno drugačija. Umesto dva i više kanala za podatke, Pentium Pro je dobio sasvim drugačiju unutrašnju organizaciju i mogao je istovremeno da izvršava do pet instrukcija.

Druga novost u procesoru Pentium Pro bila je dvostepena keš memorija. Procesorski čip je imao 8 kilobajta brze memorije za čuvanje najčešće korišćenih instrukcija i drugih 8 kilobajta za čuvanje najčešće korišćenih podataka. U pakovanju s Pentiumom Pro (ali ne na samom čipu) nalazila se i druga keš-memorija od 256 kilobajta.

Iako je Pentium Pro imao veliki keš, nije dobio MMX instrukcije (pošto Intel nije mogao da proizvede tako veliki čip po prihvatljivoj ceni). Kada je tehnologija napredovala dovoljno da su se na istom čipu mogle smestiti i MMX instrukcije i keš, kombinovani proizvod je predstavljen svetu kao Pentium II. Da bi se ubrzao rad sa 3D grafikom, kasnije je dodato još instrukcija za podršku multimediji. One su nazvane **SSE instrukcije** (engl. *Streaming SIMD Extensions*) (Raman et al., 2000). Novi čip je postao poznat kao Pentium III, ali je u suštini to i dalje bio Pentium II.

Sledeći procesor Pentium zasnovan je na potpuno drugačijoj unutrašnjoj arhitekturi. Da bi obeležio taj događaj, Intel je s rimskih brojeva prešao na arapske i nazvao ga Pentium 4. Kao što se i očekivalo, Pentium 4 je bio brži od svih svojih prethodnika.

U verziju koja je imala radni takt od 3,06 GHz uvedena je i zanimljiva nova mogućnost – istovremeni rad s više programskih niti. Ona je omogućavala programima da svoj rad podele u dva programska toka kojima je Pentium 4 mogao paralelno da upravlja, što je ubrzavalo izvršavanje. Osim toga, dodat je još jedan podskup SSE instrukcija za dopunsko ubrzanje obrade zvuka i slike. Na slici 1-12 prikazana je fotografija čipa Pentium 4. On je dimenzija 16 mm × 13,5 mm – jedan izuzetno veliki čip.



Slika 1-12. Čip Pentium 4. Sva prava na fotografiju zadržava korporacija Intel, 2003. ovde je upotrebljena uz njihovu dozvolu.

Osim glavne serije svojih procesora za lične računare koju smo opisali, Intel je proizvodio i verzije Pentium čipova namenjene specijalnim tržištima. Početkom 1998. godine Intel je predstavio novu seriju procesora nazvanu **Celeron**, jeftiniju i po performansama slabiju verziju Pentiuma II, namenjenu jeftinim PC računarima.

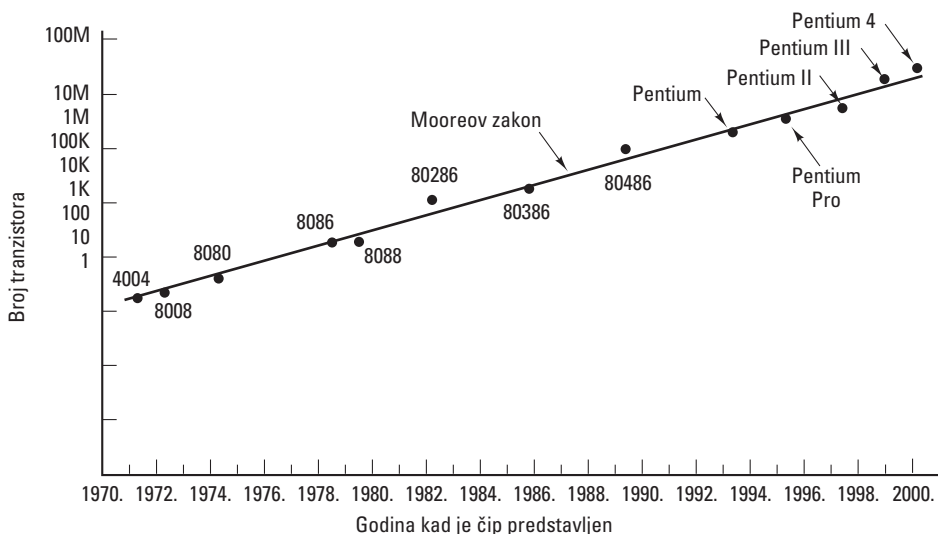
Pošto Celeron ima istu arhitekturu kao i Pentium II, nećemo ga u ovoj knjizi više razmatrati. Juna 1998. godine Intel je proizveo specijalnu verziju Pentiuma II za zahtevnije tržište. Ovaj procesor, nazvan **Xeon**, imao je veću keš memoriju, bržu magistralu i bolju podršku za višeprocessorski rad, ali je u suštini ostao Pentium II, tako da ni njega nećemo posebno izdvajati. Pentium III je takođe imao svoju verziju Xeon.

Novembra 2000. godine Intel je predstavio Pentium 4 koji je izvršavao iste programe kao Pentium III i Xeon, ali je imao sasvim novu konstrukciju. U verziji koja je imala radni takt od 3,06 GHz uveden je istovremeni rad s više programskih niti (o tome ćemo govoriti u poglavlju 8).

Godine 2003. Intel je proizveo procesor Pentium M (M označava Mobile), čip namenjen prenosivim računarima. On je bio deo arhitekture Centrino, a ciljevi su bili manja potrošnja radi dužeg trajanja baterija, manji i lakši računari i ugrađena podrška za bežično umrežavanje prema (WiFi) standardu IEEE 802.11. Intel je nameravao da ponudi skupove čipova i za druge buduće specifične primene – na primer, za uređaje za kućnu zabavu i prenosive računare po standardu IEEE 802.16 (WiMax).

Svi Intelovi čipovi su kompatibilni sa starijim verzijama sve do procesora 8086. Drugim rečima, Pentium 4 može bez izmene da izvršava programe pisane za procesor 8086. Takva kompatibilnost je za Intel uvek bila projektni imperativ kako bi se korisnicima omogućilo da i na novim procesorima izvršavaju softver u koji su uložili novac. Razume se da je Pentium 4 za tri reda veličine složeniji od procesora 8086, pa može da uradi mnogo toga što procesor 8086 ne može. Da bi se postigla kompatibilnost, dobijena je arhitektura koja nije tako elegantna kao što bi mogla biti da je neko konstruktorima Pentiuma 4 dao 42 miliona tranzistora i naredio im da sve urade iz početka.

Zanimljivo je primetiti da Mooreov zakon ne važi samo za broj bitova memorije, već i za procesorske čipove. Kada na polulogaritamskoj skali prikazemo broj ugrađenih tranzistora (slika 1-12) u zavisnosti od datuma proizvodnje čipova, uverićemo se da Mooreov zakon važi i tu. Ta zavisnost je prikazana na slici 1-13.



Slika 1-13. Mooreov zakon za (Intelove) procesorske čipove.

Iako će Mooreov zakon verovatno važiti još neko vreme, ubrzano se pojavljuje sve veći problem s disipacijom toplote. Manji tranzistori omogućavaju rad pri višim učestalostima, za šta su neophodni veći naponi. Snaga i proizvedena toplota rastu s kvadratom napona, tako da brži rad izaziva i veći problem sa odvođenjem toplote. Na radnom taktu od 3,6 GHz, Pentium 4 ima snagu od 115 vati. To znači da je zagrejan približno kao sijalica od 100 vati. Povećavanje radnog takta samo dodatno pogoršava stvari.

Novembra 2004. godine Intel je prekinuo razvoj Pentiuma na 4 GHz zbog problema sa odvođenjem toplote. Veliki ventilatori mogu da pomognu, ali šum koji prave nije baš omiljen među korisnicima, a vođeno hlađenje koje se koristi u centralnim računarima nije baš neko rešenje za stone računare (a gotovo nikakvo za prenosive).

Zaključak je da se nezadrživi rast radnog takta mora privremeno obuzdati, bar dok Intelovi inženjeri ne otkriju kako da na efikasan način odvedu oslobođenu toplotu. Umesto povećaja radnog takta, Intel bi u budućnosti mogao da stavi dva procesora na isti čip, zajedno s velikom deljenom keš memorijom. Zbog zavisnosti snage od napona i radnog takta, dva procesora na jednom čipu manje su snage od jednog dvostruko bržeg procesora. Prema tome, razvoj koji predviđa Mooreov zakon u budućnosti se može bolje iskoristiti ukoliko se bude na čip ugrađivala sve veća i veća keš memorija, nego ako se povećava radni takt (pošto memorija ne troši toliko snage).

1.4.2 Predstavljajanje čipa UltraSPARC III

Sedamdesetih godina prošlog veka UNIX je bio popularan na univerzitetima, ali ga nikako nije bilo na ličnim računarima, pa su poklonici UNIX-a morali da koriste (često preopterećene) vremenski deljene mini računare kao što su PDP-11 i VAX. Godine 1981, Nemač Andy Bechtolsheim, diplomac sa Stanforda koga je nerviralo što mora da ode u računski centar kad god hoće da koristi UNIX, odlučio je da reši taj problem tako što će sâm napraviti radnu stanicu pod UNIX-om od komponenata koje se mogu naći na tržištu. Nazvao ju je SUN-1 (Stanford University Network).

Bechtolsheim je ubrzo privukao pažnju 27-godišnjeg Indusa Vinoda Khoslae, koji je goreo od želje da se kao milioner penzionise u tridesetoj godini. Khosla je ubedio Bechtolsheima da osnuju kompaniju za proizvodnju i prodaju radnih stanica Sun. Khosla je unajmio Scotta McNealyja, drugog diplomca sa Stanforda, da vodi proizvodnju. Za pisanje softvera angažovali su Billa Joyja, glavnog projektanta Berkeley UNIX-a. Njih četvorica su 1982. godine osnovali kompaniju Sun Microsystems.

Prvi proizvod firme Sun, radna stanica Sun-1 s procesorom Motorola 68020, postigao je munjevit uspeh, baš kao i naredni modeli Sun-2 i Sun-3, takođe s Motorolinim procesorima. Za razliku od ličnih računara iz tog vremena, ove mašine su bile mnogo moćnije (otuda i naziv „radna stanica“) i od samog početka su bile predviđene za rad u mreži. Svaka radna stanica Sun bila je opremljena mrežnim Ethernet priključkom i imala je TCP/IP softver za povezivanje na mrežu ARPANET, preteču Interneta.

Godine 1987 firma Sun, koja je tada prodavala sisteme u vrednosti od pola milijarde dolara godišnje, odlučila je da napravi sopstveni mikroprocesor koji bi se zasnivao na revolucionarnom novom projektu izrađenom na Univerzitetu Kalifornije u Berkliju (RISC II). Taj mikroprocesor, nazvan **SPARC** (Scalable Processor ARChitecture, proširiva arhitektura procesora), bio je jezgro radne stanice Sun-4. Nije mnogo prošlo, a sve radne stanice SUN koristile su procesor SPARC.

Za razliku od mnogih drugih računarskih kompanija, firma Sun je odlučila da ne proizvodi sama procesor SPARC. Ona je za njegovu proizvodnju ovlastila nekoliko proizvođača poluprovodnika, nadajući se da će konkurencija učiniti svoje i da će čip postajati sve bolji i jeftiniji. Proizvođači su napravili niz različitih čipova koji su se zasnivali na različitim tehnologijama, radili na različitim radnim taktovima, i naravno – različito koštali. Među njima su bili čipovi MicroSPARC, HyperSPARC, SuperSPARC i TurboSPARC. Iako su se malo razlikovali, svi su na binarnom nivou međusobno bili kompatibilni i mogli su izvršavati iste korisničke programe bez ikakvih izmena.

Sun je uvek težio da arhitekturu procesora SPARC održi otvorenu, s mnogo dobavljača delova i sistema, kako bi podstakao razvoj industrije i stvorio konkurenciju u svetu ličnih računara u kome su već dominirali Intelovi mikroprocesori. Da bi stekao poverenje kompanija koje su se zanimale za procesor SPARC, ali nisu želele da ulažu novac u konkurentski proizvod, Sun je stvorio industrijski konzorcijum SPARC International koji će upravljati razvojem budućih verzija arhitekture procesora SPARC. Shodno tome, važno je uočiti razliku između SPARC arhitekture koja predstavlja specifikaciju skupa instrukcija i drugih mogućnosti vidljivih programerima, i konkretne realizacije te arhitekture. U ovoj knjizi bavićemo se i opštom SPARC arhitekturom, i (u okviru teme o procesorskim čipovima u poglavljima 3 i 4) konkretnim SPARC čipovima ugrađenim u radne stanice Sun.

Prvi SPARC je bio 32-bitna mašina koja je radila na 36 MHz. Procesor nazvan **celobrojna jedinica** (engl. *Integer Unit*, IU) bio je mali, sa samo tri osnovna formata za instrukcije i ukupno 55 instrukcija. Pored toga, jedinica za računanje u formatu pokretnog zareza imala je i svojih 14 instrukcija. Ovo se može uporediti sa razvojem Intelovih procesora, koji je počeo 8-bitnim i 16-bitnim čipovima (8088, 8086, 80286), da bi 32 bita dostigao tek sa procesorom 80386.

Prvi SPARC-ov raskid s prošlošću desio se 1995. godine, kada je stvorena deveta verzija SPARC arhitekture, prava 64-bitna arhitektura sa 64-bitnim adresama i registrima. Prva Sunova radna stanica u koju je implementirana arhitektura V9 (verzija 9) bila je stanica **UltraSPARC I**, predstavljena 1995. godine. (Tremblay i O'Connor, 1996). Iako je bila 64-bitna, ona je na binarnom nivou bila potpuno kompatibilna s postojećim 32-bitnim SPARC radnim stanicama.

UltraSPARC je bio namenjen novim tržištima. Dok su dotadašnji računari radili sa alfanumeričkim podacima i izvršavali program za obradu teksta i program za unakrsna izračunavanja, UltraSPARC je obrađivao slike, zvuk, video zapis i multimediju u celini. Osim 64-bitne arhitekture, bilo je tu još novina kao što su 23 nove instrukcije, među kojima i neke za pakovanje i raspakivanje piksela iz 64-bitnih reči, za menjanje veličine i rotiranje slika, premeštanje blokova te za komprimovanje i dekomprimovanje video zapisa u realnom vremenu. Ove tzv. **VIS** instrukcije (Visual Instruction Set, skup „vizuelnih“ instrukcija) bile su namenjene za opšti rad s multimedijom, analogno Intelovom skupu MMX instrukcija.

UltraSPARC je bio namenjen najzahtevnijim aplikacijama, (npr. velikim Web serverima s više desetina procesora i fizičkom memorijom do 8 TB (1 TB terabajt, jednak je 10^{12} bajtova). Međutim, njegove slabije verzije mogle su se koristiti i u prenosivim računarima.

UltraSPARC I su (prirodno) nasledili UltraSPARC II, UltraSPARC III i UltraSPARC IV. Ovi modeli su se prvenstveno razlikovali po radnom taktu, ali je u svaki dodato još ponešto. Kada budemo govorili o SPARC arhitekturi u ovoj knjizi, uglavnom ćemo kao primer koristiti 64-bitni V9 model UltraSPARC III Cu. UltraSPARC IV je u osnovi procesor u kome dva UltraSPARC-a III na zajedničkom čipu dele istu memoriju. O njemu ćemo govoriti u okviru teme o multiprocesorima u poglavlju 8.

1.4.3 Čip 8051

Naš treći primer se veoma razlikuje od prvog (Pentium 4, namenjen ličnim računarima) i drugog (UltraSPARC III, namenjen serverima). To je čip 8051, koji se koristi u ugrađenim sistemima. Priča o čipu 8051 počinje 1976. godine, u vreme kada je 8-bitni čip 8080 već oko dve godine bio na tržištu. Proizvođači kućnih aparata počeli su da ugrađuju čip 8080 u svoje uređaje, ali da bi sistem bio potpun, nije bio dovoljan samo čip 8080, već i jedan ili više memorijskih, kao i jedan ili više ulazno-izlaznih čipova. Troškovi za ugradnju barem tri čipa i njihovo međusobno povezivanje nisu bili mali, pa su računari ugrađivani u veće i skuplje modele aparata. Mnogi proizvođači su predlagali Intelu da ceo računar (procesor, memoriju i ulazno-izlaznu jedinicu) stavi na isti čip kako bi smanjio troškove.

Intel je odgovorio tako što je napravio čip 8748 – mikrokontroler sa 17.000 tranzistora, procesorom sličnim procesoru 8080, jednim kilobajtom programske memorije samo za čitanje, 64 bajta memorije za čitanje i upisivanje namenjene promenljivama, 8-bitnim meračem vremena (tajmerom) i 27 ulazno-izlaznih vodova za kontrolne prekidače, dugmad i signalne lampice. Iako primitivan, ovaj čip je postigao komercijalan uspeh, što je Intel podstaklo da 1980. godine na tržište izbací čip 8051. Nov čip je imao 60.000 tranzistora, mnogo brži procesor, 4 KB memorije samo za čitanje, 128 bajtova memorije za čitanje i upisivanje, 32 ulazno-izlazna voda, serijski priključak i dva 16-bitna merača vremena. Uskoro su usledili i drugi članovi **porodice čipova MCS-51**, kako ju je Intel nazvao (slika 1-14).

Čip	Programska memorija	Tip memorije	RAM	Merači vremena (tajmeri)	Programski prekidi
8031	0 KB		128	2	5
8051	4 KB	ROM	128	2	5
8751	8 KB	EPROM	128	2	5
8032	0 KB		256	3	6
8052	8 KB	ROM	256	3	6
8752	8 KB	EPROM	256	3	6

Slika 1-14. Članovi porodice čipova MCS-51.

Svi ovi čipovi imali su memoriju samo za čitanje (namenjenu programu) i malu količinu **RAM** memorije (**Random Access Memory**, memorija s direktnim pristupom) za čitanje i upisivanje (namenjene podacima). Kod čipova 8031 i 8032 programska memorija je bila izvedena kao spoljna, s mogućnošću uključivanja i dodatnih 8 KB ako je neophodno. Memorije **ROM** (**Read Only Memory**, memorija samo za čitanje) i **EPROM** (**Erasable Programmable ROM**, izbrisiva programabilna memorija samo za čitanje) obradićemo u poglavlju 3. Zasad je dovoljno reći da su mikrokontroleri 8051 i 8052 izvedeni na jednom čipu i korišćeni su u kućnim aparatima. Svaka grupa čipova namenski je pravljená za poznatog kupca (npr. za određenog proizvođača kućnih aparata) i svaki čip je sadržao program koji je isporučivao kupac.

Da bi napravio softver, kupac je morao imati i razvojni sistem. Tu su na scenu stupali čipovi 8751 i 8752. Oni su bili mnogo skuplji od čipova 8051 i 8052, ali ih je kupac mogao programirati za potrebe ispitivanja softvera. Kada bi se našla greška u kodu, program se iz čipova 8751 i 8752 mogao izbrisati tako što su se čipovi izlagali ultraljubičastoj svetlosti, posle čega bi se u njih mogao upisati nov program. Kada bi ispitivanje softvera bilo završeno, softver bi bio prosleđen proizvođaču čipova, koji bi tada napravio namenske čipove 8051 ili 8052 s datim kodom.

U pogledu arhitekture, priključivanja i programiranja, svi pripadnici porodice MCS-51 bili su vrlo slični. Zbog jednostavnosti najviše ćemo pominjati čip 8051, ističući po čemu se razlikuje od drugih čipove gde je to potrebno.

Nekima će uzimanje za primer 8-bitnog čipa starog barem 20 godina zvučati pomalo čudno, ali za to postoje dobri razlozi. Godišnje se proda mikrokontrolera oko 8 milijardi, a taj broj i dalje raste. Ovaj broj za red veličine prevazilazi broj godišnje prodatih Pentiuma. Tek je 2001. godine broj 8-bitnih mikrokontrolera prevazišao broj prodatih 4-bitnih mikrokontrolera. Trenutno prodaja 8-bitnih mikrokontrolera prevazilazi sve druge mikrokontrolere zajedno, a porodica čipova MCS-51 najpopularnija je porodica 8-bitnih čipova. Kada uzmemo u obzir važnost ugrađenih sistema, svako ko proučava arhitekturu računara mora da poznaje čipove koji se u njima koriste, a čip 8051 jedan je od najpopularnijih za tu svrhu.

Mnogo šta je doprinelo uspešnosti čipa 8051. Prva je i najvažnija njegova cena. U zavisnosti od broja naručenih komada, čip 8051 može se dobiti za oko 10 do 15 centi, pa čak i jeftinije ako je narudžbina velika. Nasuprot tome, 32-bitni mikrokontroleri po pravilu koštaju oko 30 puta više, dok su 16-bitni negde između. Za proizvode čija je cena na slobodnom tržištu manja od 50 dolara, smanjenje proizvodnih troškova i za par dolara znači mnogo u pogledu cene i prodaje. To je glavni razlog popularnosti čipa 8051 – on je izuzetno jeftin.

Drugo, više od polovine kompanija koje prave čip 8051 proizvode ga po Intelovoj licenci. Njihovi proizvodi pokrivaju širok raspon brzine, od prvobitnih 12 MHz do 100 MHz, a prave se različitim tehnologijama i drugačije se pakuju. Cena je niska upravo zbog konkurencije, a i veliki kupci više vole da ne zavise isključivo od jednog snabdevača.

Treće, pošto se čip 8051 već dugo primenjuje, za njega postoji mnogo softvera, uključujući asemblere, programske prevodioce za programski jezik C i druge jezike, biblioteke svih vrsta, programe za otkrivanje grešaka, simulatore, softver za testiranje i štošta drugo. Na tržištu je i više razvojnih sistema koji ubrzavaju projektovanje hardvera i softvera koji se ugrađuje. Na kraju, mnogi programeri i inženjeri dobro poznaju čip 8051, što olakšava pronalaženje kvalifikovane radne snage.

Ovakva popularnost raste sama od sebe. Razvojni inženjeri koji rade na ugrađenim sistemima često biraju čip 8051 upravo zbog njegove rasprostranjenosti, da bi, na primer, isprobali novu tehnologiju kojom se štedi energija (Martin et al., 2003) ili otpornost na greške (Lima et al., 2002).

Na Internetu se može pronaći mnogo informacija o čipu 8051. Dobra polazna adresa je www.8051.com. Pored toga, o njemu se još uvek pišu knjige (Ayala, 2004; Calcutt et al., 2004; MacKenzie et al., 2005 i Mazidi et al., 2005).

1.5 METRIČKE JEDINICE

Da bismo izbegli zabunu, treba jasno i glasno reći da se u ovoj knjizi, kao i inače u računarstvu, koriste metričke jedinice (umesto tradicionalnih anglosaksonskih jedinica u stilu aršin-livra-pinta-dan hoda). Osnovni prefiksi metričkih jedinica prikazani su na slici 1-15. Oni se obično skraćuju na početno slovo, iza koga sledi veliko slovo za jedinicu ako je ona veća od 1 (KB, MB itd.). Od ovoga se (iz istorijskih razloga) izuzima kb/s (kilobiti u sekundi). Tako komunikacioni kanal brzine 1 Mb/s propušta 10^6 bitova/s, a radni takt periode ciklusa od 100 ps otkucava svakih 10^{-10} sekundi. Pošto oba prefiksa mili i mikro počinju slovom „m“, dogovoreno je da skraćenica za mili bude „m“, a za mikro – „μ“ (grčko slovo mi).

Eksponencijalni množitelj	Decimalni format	Prefiks	Eksponencijalni množitelj	Decimalni format	Prefiks
10^{-3}	0,001	mili	10^3	1.000	Kilo
10^{-6}	0,000001	mikro	10^6	1.000.000	Mega
10^{-9}	0,000000001	nano	10^9	1.000.000.000	Giga
10^{-12}	0,000000000001	piko	10^{12}	1.000.000.000.000	Tera
10^{-15}	0,000000000000001	femto	10^{15}	1.000.000.000.000.000	Peta
10^{-18}	0,000000000000000001	ato	10^{18}	1.000.000.000.000.000.000	Eksa
10^{-21}	0,0000000000000000000001	cepto	10^{21}	1.000.000.000.000.000.000.000	Ceta
10^{-24}	0,000000000000000000000001	jokto	10^{24}	1.000.000.000.000.000.000.000.000	Jota

Slika 1-15. Osnovni prefiksi metričkih jedinica.

Treba naglasiti i da pri izražavanju veličine memorije, diskova, datoteka i baza podataka, u uobičajenoj praksi jedinice imaju nešto drugačije značenje. Tako kilo označava 2^{10} (1024) umesto 10^3 (1000) jer je kapacitet memorije uvek stepen broja dva. Memorija od 1 KB sadrži 1024 bajta, a ne 1000 bajtova. Slično tome, memorija od 1 MB sadrži 2^{20} (1.048.576) bajtova, memorija od 1 GB sadrži 2^{30} (1.073.741.824) bajtova, a baza podataka od 1 TB sadrži 10^{40} (1.099.511.627.776) bajtova. Međutim, komunikacioni kanal brzine 1kb/s može da propusti 1000 bitova u sekundi, a lokalna mreža brzine 10 Mb/s propušta 10.000.000 bitova/s jer ove dve brzine nisu stepeni broja dva. Mnogi, nažalost, lako mešaju ove sisteme, naročito kada se radi o veličini diskova. Da bismo izbegli dvosmislenost, u ovoj knjizi ćemo koristiti simbole KB, MB, GB i TB za 2^{10} , 2^{20} , 2^{30} i 2^{40} bajtova, a simbole kb/s, Mb/s, Gb/s i Tb/s za 10^3 , 10^6 , 10^9 i 10^{12} bitova/s.

1.6 PREGLED SADRŽAJA KNJIGE

Ovo je knjiga o računarima s više nivoa (što obuhvata skoro sve savremene računare) i o njihovoj organizaciji. Analiziraćemo prilično detaljno svaki od četiri nivoa računara – nivo digitalne logike, nivo mikroarhitekture, nivo ISA i nivo operativnog sistema. Neke od osnovnih tema koje ćemo razmotriti obuhvataju opštu organizaciju nivoa (kao i motive za baš takvu organizaciju), raspoložive vrste instrukcija i podataka, organizaciju memorije i adresiranje i način na koji se nivo implementira u praksi. Proučavanje ovih i sličnih tema zaokružuje oblast koja se zove organizacija računara ili arhitektura računara.

Prvenstveno će nas zanimati koncepti, a ne detalji ili formalna matematička izvođenja. Zbog toga će neki primeri biti pomalo uprošćeni kako vas detalji ne bi sprečili da jasnije uočite osnovnu zamisao.

Da biste stekli izvesnu predstavu o tome kako se principi izneti u ovoj knjizi ne samo mogu primeniti, već se i primenjuju u praksi, kroz čitavu knjigu ćemo kao praktične primere navoditi procesore Pentium 4, UltraSPARC III i 8051. Ova tri čipa su izabrana iz više razloga. Prvo, svi se široko koriste i čitalac je verovatno došao u dodir s barem jednim od njih. Drugo, svaki od njih ima svoju jedinstvenu arhitekturu koja predstavlja osnov za poređenje. Knjige koje obrađuju samo jednu vrstu računara često na čitaoca ostavljaju lažan utisak da su naišli na pravi računar, što je apsurdno ako imamo u vidu mnoge kompromise i proizvoljne odluke koje su konstruktori prinuđeni da donose. U ovoj knjizi čitaoca podstičemo da kritički prouči i sve druge računare i pokuša da razume zašto sve izgleda onako kako izgleda, kao i to da je u datim okolnostima sve moglo izgledati i drugačije.

Od samog početka treba da bude jasno da ovo nije knjiga o programiranju Pentiuma 4, UltraSPARC-a III ili čipa 8051. Njih ćemo koristiti kao pogodne primere tamo gde nađemo za shodno, ali ih nećemo podrobno opisivati. Čitalac koji želi da se detaljno upozna s jednim od tih čipova treba da potraži informacije koje je o njima objavio proizvođač.

U poglavlju 2 predstavljene su osnovne komponente računara – procesori, memorije i ulazno-izlazni uređaji. Ono treba da ponudi pregled arhitekture sistema i da bude uvod u naredna poglavlja.

Svako od poglavlja 3, 4, 5 i 6 bavi se jednim od nivoa prikazanih na slici 1-2. Naš pristup je odozdo naviše jer se i računari po tradiciji tako projektuju. Projekat nivoa k umnogome je određen svojstvima nivoa $k-1$, pa ćete teško razumeti bilo koji nivo ukoliko već prilično dobro ne poznajete prethodni nivo koji i predstavlja osnovni motiv za sledeći. Isto tako, i didaktički je ispravno najpre objasniti jednostavnije niže nivoe, pa postepeno prelaziti na sve složenije nivoe.

Poglavlje 3 govori o nivou digitalne logike – pravom hardveru računara. U njemu se objašnjava šta su logička kola i kako se ona mogu kombinovati da rade nešto korisno. U tom poglavlju predstavljamo i Bulovu (logičku) algebru, alatku za analiziranje digitalnih kola. Objašnjavamo i računarske magistrale (sabirnice), naročito popularnu PCI magistralu, a navodimo i brojne primere iz industrijske prakse, uključujući i tri pominjana primera računara.

U poglavlju 4 govorimo o arhitekturi mikroarhitekturnog nivoa i načinima upravljanja njime. Pošto je funkcija ovog nivoa da interpretira instrukcije nivoa 2 u višem sloju, usredsređićemo se na tu temu i ilustrovaćemo je primerima. Tu je i objašnjenje nivoa mikroarhitekture nekih realnih računara.

Poglavljje 5 obrađuje nivo ISA koji većina proizvođača predstavlja kao mašinski jezik. Ovde ćemo detaljno ispitati računare koje smo uzeli za primere.

Poglavljje 6 pokriva neke od instrukcija, organizaciju memorije i mehanizme upravljanja koji postoje na nivou operativnog sistema računara. Za primere smo izabrali Windows XP (popularan na zahtevnijim serverima s Pentiumom 4) i UNIX, koji se koristi na UltraSPARC-u III.

Poglavljje 7 opisuje nivo asemblera. U njemu se objašnjavaju i assembler i proces prevođenja. Tu je svoje mesto našao i postupak povezivanja programa s programskim bibliotekama.

Poglavljje 8 objašnjava paralelne računare, jednu od sve važnijih današnjih tema. Neki od paralelnih računara imaju više procesora koji dele zajedničku memoriju. Drugi imaju više procesora koji nemaju zajedničku memoriju. Neki od njih su super-računari, neki su potpuni sistemi na čipu, a treći su skupovi radnih stanica.

U poglavlju 9 dat je spisak referenci za dalje čitanje, kritički sređen po temama i abecedno uređen spisak citirane literature. To je najvažnije poglavljje knjige.

VEŽBANJA

1. Objasnite sopstvenim rečima svaki od sledećih izraza:
 - a. Prevodilac.
 - b. Interpreter.
 - c. Virtuelna mašina.
2. Kakva je razlika između interpretiranja i prevođenja?
3. Može li se zamisliti da programski prevodilac (kompajler) generiše rezultat za nivo mikroarhitekture umesto za nivo ISA? Objasnite prednosti i nedostatke ovog predloga.
4. Možete li da zamislite računar s više nivoa u kome nivo uređaja i nivo digitalne logike nisu najniži nivoi? Obrazložite.
5. Zamislite računar s više nivoa u kome su svi nivoi različiti. Svaki nivo ima instrukcije koje su m puta moćnije od onih s nivoa ispod njega; to znači da jedna instrukcija nivoa r može da uradi koliko i m instrukcija nivoa $r - 1$. Ako se program nivoa 1 izvršava za k sekundi, za koliko će se vremena izvršiti ekvivalentni programi na nivou 2, 3 i 4, ukoliko se pretpostavi da je potrebno n instrukcija nivoa r za interpretiranje jedne instrukcije nivoa $r + 1$?
6. Neke instrukcije na nivou operativnog sistema računara identične su instrukcijama ISA jezika. Te instrukcije ne izvršava operativni sistem, već direktno mikroprogram. Kada uzmete u obzir svoj odgovor na prethodno pitanje, recite zašto mislite da je tako?

7. Zamislite računar sa identičnim interpreterima na nivoima 1, 2 i 3. Interpreter „potroši“ n instrukcija da bi preuzeo, ispitao i izvršio jednu instrukciju. Instrukcija nivoa 1 se izvršava za k nanosekundi. Koliko će izvršenje jedne instrukcije trajati na nivoima 2, 3 i 4?
8. Na koji način hardver i softver jesu ili nisu ekvivalentni?
9. Babbageova diferencna mašina imala je fiksni program koji se nije mogao menjati. Je li to suštinski isto kao i savremeni kompakt disk čiji se sadržaj ne može menjati? Obrazložite odgovor.
10. Jedna od posledica Von Neumannove ideje da program smesti u memoriju jeste i to što se takav program može menjati, baš kao i podaci. Možete li da smislite primer situacije u kojoj bi to bilo korisno? (Pomoć: razmišljajte o aritmetici nizova.)
11. Odnos performansi modela 75 i modela 30 iz IBM-ove serije računara 360 iznosi 50, iako je radni takt prvog modela samo pet puta veći. Kako objašnjavate ovo neslaganje?
12. Dva osnovna projekta sistema prikazani su na slikama 1-5 i 1-6. Opišite kako bi se na svakom od dva sistema mogle odvijati ulazno-izlazne operacije. Koji od dva sistema ima potencijalno bolje ukupne performanse?
13. Pretpostavite da svaki od 300 miliona građana Amerike dnevno potroši dva pakovanja robe s RFID oznakama. Koliko se RFID oznaka mora proizvesti godišnje da bi se zadovoljila ova potražnja? Ako je cena jedne oznake 1 cent, koliki su ukupni troškovi za proizvodnju oznaka? Uzimajući u obzir bruto nacionalni proizvod (GDP), da li će ova suma novca postati prepreka za njihovo korišćenje na svakom pakovanju trgovačkog proizvoda?
14. Navedite tri kućna uređaja kao kandidate za ugrađivanje mikroprocesora.
15. U jednom trenutku tranzistor na mikroprocesoru imao je prečnik od 0,1 mikrometara. Prema Mooreovom zakonu, koliki će biti tranzistor na modelu iz sledeće godine?
16. Pravni spor o tome ko je izmislio računar počeo je aprila 1973. godine, a sudio je Earl Larson. On je pred sobom imao prijavu o kršenju patentnih prava koju je podnela kompanija Sperry Rand, naslednik patenata za ENIAC. Njeno stanovište je bilo da svako ko pravi računar treba da joj isplati nadoknadu jer ona poseduje ključne patente. Tokom suđenja je izneto preko 30.000 dokaza. Dokumentacija procesa imala je više od 20.000 stranica. Proučite pažljivo ovaj slučaj služeći se obimnim informacijama o njemu koje postoje na Internetu i napišite izveštaj o njegovim tehničkim aspektima. Šta su tačno Eckert i Mauchley patentirali i zašto je sudija stekao utisak da se njihov sistem zasniva na Atanasoffljevom radu?
17. Navedite tri osobe za koje mislite da su najviše uticale na stvaranje savremenog računarskog hardvera, ukratko opišite njihov doprinos i obrazložite zašto ste ih odabrali.
18. Ponovite vežbanje 17, ali za računarski softver.