

# R

**RAČUNALA**, općenit naziv pomagala za računanje. S pojavom elektroničkih računala *računalo* (ili *kompjutor*, prema lat. *computator*, engl. *computer*) jest poseban naziv za elektroničke uređaje koji primaju, obrađuju i prikazuju podatke, dakle obavljaju znatno složenije funkcije od samog računanja. Iz tog su naziva potekle i mnoge izvedenice: računarstvo, računalstvo, računarska, računalna ili računačka tehnika, upravljanje računalom i mnoge druge. Za jednostavnija pomagala za računanje ustaljen je naziv *kalkulator*.

Suvremena elektronička računala osnova su za primjenu računarstva i informatike u gotovo svim područjima ljudskog djelovanja, a posebno u provedbi tzv. treće industrijske revolucije (v. *Analogno računalo*, TE 1, str. 296; v. *Digitalna računala*, TE 3, str. 313; v. *Digitalna računala, programiranje*, TE 3, str. 329).

Razvoj različitih pomagala za izvođenje računskih operacija ima dugu povijest. Najznačajnija su od njih kineska i rimska varijanta abaka. Kineski je abak okvir s napetim žicama na kojima su nанизane kuglice, a rimski se abak sastojao od ploče podijeljene u pruge po kojima se micalo pločice ili kameničići (otuda lat. *calculare* računati, od *calculus* kameničić). U XVII. stoljeću B. Pascal i G. W. Leibniz razvili su mehaničke naprave za računanje na osnovi kojih su nastala današnja mehanička i elektromehanička računala (kalkulatori). Engleski matematičar Ch. Babbage počeo je 1834. raditi na uređaju što ga je nazvao *analitičkim strojem*. On je upotrijebio bušene kartice što ih je 1801. izumio Francuz J. M. Jacquard za upravljanja tkalačkim stanom za proizvodnju složenih uzoraka. Babbage je zamislio da stroj čita dva niza bušenih kartica: jedan s naredbama i drugi s podacima te da se rezultati računanja buše na treći niz kartica. Babbage, međutim, nije ostvario svoje zamisli, koje su bile daleko ispred mogućnosti tadašnje tehnologije. H. Aiken je predložio (1937) uredaj nazvan *Automatic Sequence Controlled Calculator* (nazvan kasnije Mark 1), zasnovan na idejama Babbagea i ostvaren pomoću elektromehaničkih računala tvrtke IBM. Taj se stroj počeo graditi 1939., a završen je 1944. godine.

Istdobno s razvojem računala Mark 1, započet je u kolovozu 1942. godine projekt za gradnju prvoga elektroničkog digitalnog računala, nazvana ENIAC (Electronic Numerical Integrator and Computer) na Sveučilištu Pennsylvania (SAD). Postojanje tog računala objavljeno je 15. veljače 1946., i taj se dan smatra početkom razdoblja elektroničkih digitalnih računala. Na čelu razvojne grupe bili su J. P. Eckert i J. W. Mauchly, a matematički savjetnik grupe bio je John von Neumann. ENIAC je u biti elektronička verzija stroja Mark 1 s više paralelnih jedinica za računanje. Za vrijeme rada na projektu ENIAC pojatile su se dvije nove concepcije. Prva je potekla iz konstatacije da su elektronički sklopovi dovoljno brzi za obavljanje operacija, pa se ne mora graditi mnogo paralelnih jedinica za računanje, nego se operacije mogu provoditi s jednom tzv. *aritmetičkom jedinicom*. To je jako pojednostavnilo gradnju računala, iako je kasnije načelo paralelnih jedinica ponovo oživjelo. Druga je concepcija, smještaj programa u memoriju, što omogućuje izvođenje niza naredaba u proizvoljnom redoslijedu, mnogo značajnija i bitno je djelovala na dalji razvoj računala. Kao rezultat tih dva razmatranja predložen je novi stroj EDVAC (Electronic Discrete Variable Arithmetic Computer). Prijedlog za EDVAC bio je grupni rad i teško je jednoj osobi pripisati specifične ideje. U skici prijedloga za EDVAC (lipanj 1945.) John von Neumann je ujedinio sve ideje i postavio osnovnu logičku strukturu računala s memoriranim programom.

Osnovni su dijelovi takva računala: *ulazni dio*, pomoću kojeg se može unositi proizvoljan broj operanara naredaba; *memorija*, u koju se mogu unositi i iz koje se mogu dobiti operandi i naredbe, i to u proizvolnjem poretku; *aritmetičko-logička jedinica*, koja može obavljati aritmetičke i logičke operacije s operandima koji se uzimaju iz memorije, s tim da se rezultati također pohranjuju u memoriju; *izlazni dio*, preko kojeg se rezultati, tzv. izlazni podaci

na nekom izlaznom sredstvu, predaju korisniku; *upravljačka jedinica*, koja može interpretirati naredbe dobivene iz memorije i upravljati ostalim dijelovima računala, s tim da može odabirati alternative dalje akcije već prema dobivenom rezultatu.

Osnovna struktura koja proizlazi iz tih uvjeta poznata je kao *von Neumannova struktura* i sva se računala proizvedena od tog vremena zasnivaju na toj strukturi. Od tada do danas uvedena su samo neka poboljšanja von Neumannove osnovne concepcije, a impresivna poboljšanja svojstava računala treba pripisati uglavnom razvoju tehnologije elektroničkih komponenata.

Projekt EDVAC bio je doista usporen zbog raspada grupe na Sveučilištu Pennsylvania, pa se računalo EDVAC pojavilo tek 1952. i bilo u upotrebi do 1962. Unutarnja je memorija tog računala izvedena s akustičkim linijama za kašnjenje. Drugi je pokušaj gradnje računala s elektroničkom memorijom mali sustav izrađen 1948. na Manchesterskom sveučilištu. U njemu je za pamćenje upotrijebljena katodna cijev. U svibnju 1949. prvi svoj program izvelo je računalo EDSAC (Electronic Delay Storage Automatic Calculator) na Sveučilišnom matematičkom laboratoriju u Cambridgeu. To je računalo imalo 32 akustičke memorije od kojih je svaka mogla pamtit 32 broja. Posebna je memorija postojala za naredbe. Dio osnovnih programa bio je zično ugrađen u računalo.

J. P. Eckert i J. W. Mauchly osnovali su vlastito poduzeće za izradu računala. Prvi im je rad bio sustav BINAC za tvrtku Northrop Aircraft, koji je bio isporučen 1948., a stvarno je postao operativan 1950. U ožujku 1951. grupa Eckert-Mauchly dala je na tržište prvo komercijalno računalo UNIVAC I (Universal Automatic Computer). Prvi je primjerak nabavio Bureau of Census USA. Taj je stroj imao 5000 elektronskih cijevi i mnoštvo poluvodičkih dioda. Memorija je stroja bila akustička sa stotinu živinih linija za kašnjenje. Kapacitet je bio tisuću riječi sa po 12 dekadnih mjesti.

Ubrzo nakon pojave stroja UNIVAC I, na tržištu su se pojavile memorije s magnetskim jezgrama i one su omogućile izgradnju računala u današnjem smislu. Godine 1954. pojavio se na tržištu stroj IBM650 male memorije s magnetskim jezgrami i vanjskom memorijom na magnetskom bubnju kapaciteta tisuću riječi. Takvih je strojeva izrađeno oko tisuću, pa su oni bili vrlo važni u razvoju programske tehnike. Prvo tranzistorizirano računalo instalirano je u krajem 1959. godine. Razvila ga je tvrtka NCR (National Cash Register), a izgradila tvrtka General Electric. Centralni procesor imao je oko 4000 tranzistora i 8000 dioda. Memorija s magnetskim jezgrama imala je maksimalni kapacitet od 48000 znakova.

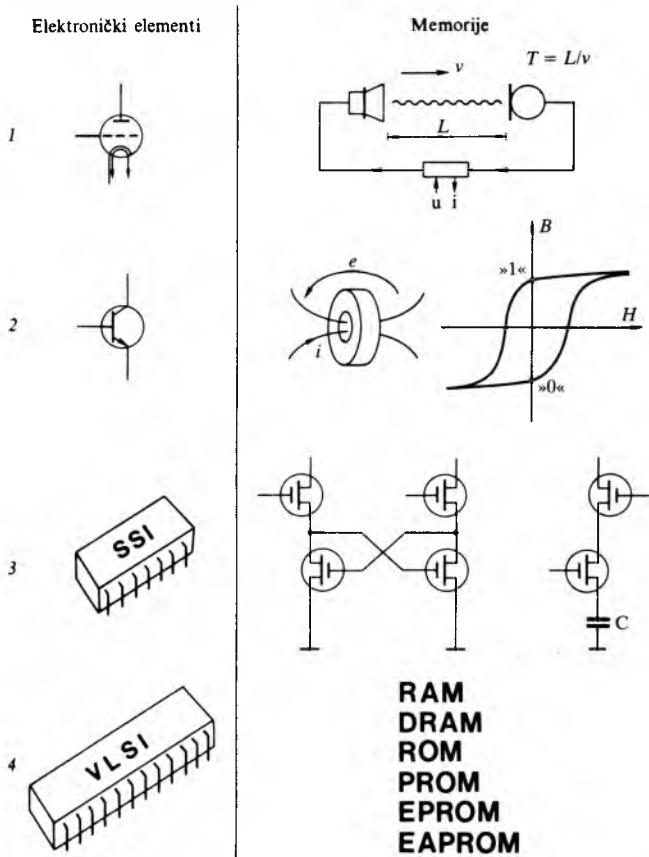
Početkom 1960. na tržištu su se pojavila tzv. *miniračunala*. Nositac je tog pristupa tvrtka Digital Equipment. To je bio početak masovne proizvodnje standardiziranih jedinica. Dalji razvoj išao je tako naglo da je već početkom 1970. postojao kalkulator na silicijskoj pločici, a odmah nakon toga je tvrtka INTEL dala na tržište mikroprocesore. Tehnikom visokog stupnja integracije uspijelo se tijekom 30-godišnjeg razvoja smanjiti računalo od obujma  $90\text{ m}^3$  i priključnu snagu 200 kW na uredaj od nekoliko kubnih decimetara i snage od stotinjak vata.

Elektronička računala često se razvrstavaju u pet tehnoloških generacija (sl. 1). Računala *prve tehnološke generacije* su ona s elektronskim cijevima. Memorije su bile akustičke. Podaci su se pamtili u obliku ultrazvučnog vala u stupcu živinih para, na magnetostrikcijski pobudenjem niklenoj žici, u piezoelektrički pobudenjem posebno izbrušenom kristalu kremena. Takve su memorije imale maleni kapacitet od nekoliko tisuća bitova, a bile su dosta skupe i nepouzdane. Računala s tranzistorima tvore *drugu tehnološku generaciju*. Opremljena su memorijama s feritnim magnetskim jezricama. Te su memorije imale velik kapacitet (do nekoliko desetaka milijuna bitova u većim računalima), a cijena i pouzdanost bile su za ono dobro razumno. U toj su generaciji u velikoj upotrebi masovne memorije na magnetskim vrpčama, diskovima i bubnjevima. Integrirani sklopovi niskog stupnja integracije sastavni su dijelovi računala *treće tehnološke generacije*. Pojavljuju se integrirane poluvodičke memorije koje istiskuju feritne memorije. Sklopovi visokog stupnja integracije, osobito mikroprocesori i memorije, osobitosti su računala *četvrte tehnološke generacije*. Memorije kapaciteta od nekoliko desetaka milijuna bitova stanu na pločicu ploštine  $\sim 200\text{--}400\text{ cm}^2$ . Programirljive, ispisne memorije (ROM, EPROM, EAROM) i vrlo sposobni mikroprocesori promijenili su način izrade računala i omogućili izgradnju vrlo moćnih stolnih računala (engl. *personal computer*).

## RAČUNALA

Tome su pridonijele i vrlo moćne masovne memorije u obliku audiokaseta, savitljivih diskova, tzv. disketa, winchester-diskova, koji imaju velik kapacitet uz male dimenzije te videodiskova kapaciteta  $\sim 10^9$  znakova na ploči veličine gramofonske ploče. Sada se razmatra što će biti peta tehnološka generacija. Smatra se da će računala u petoj generaciji biti izgrađena pomoću organskih sastavnih dijelova, iako je još moguća međugeneracija od submikrometarskih poluvodičkih elemenata.

Vrlo je važna i druga klasifikacija računala. Ona se temelji na arhitekturi i organizaciji računala. Arhitektura računala umijeće je određivanja potreba korisnika za nekom strukturu i zatim oblikovanje i razrada te strukture što je više moguće u skladu s iskazanim potrebama, te tehničkim i ekonomskim mogućnostima. Izraz arhitektura također znači opis svojstava sustava kako ih vidi stvaratelj programa za upotrebu stroja. Očito je da će se tijekom razmatranja ostvarenih računala te dvije klasifikacije isprepletati.



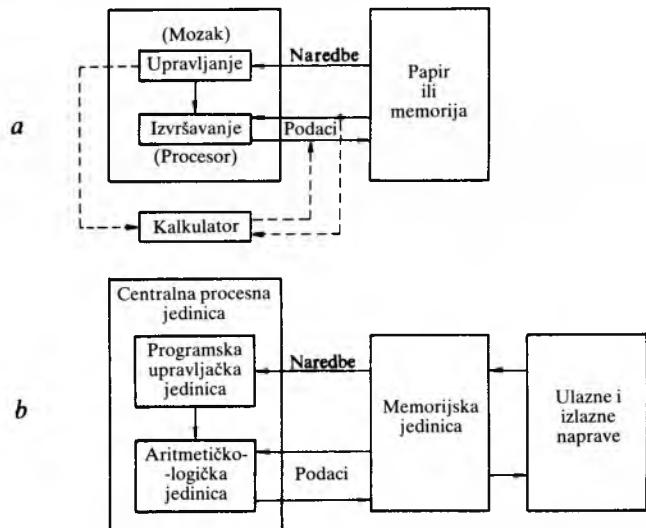
## PROTEINI ?? ITD

Sl. 1. Prikaz osnovnih svojstava tehnoloških generacija računala. U prvom stupcu su elektronički elementi na kojima se osnivalo računalo, a u drugom memorije

**Glavne komponente pomagala za računanje.** Pomagala za računanje stvorena su veoma rano. Prva su pomagala pojednostavnila, ali nisu zamijenila računanje. Pomagala su služila kao sredstvo za pamćenje znamenaka ili su se u daljem razvoju, pomakom mehaničkih dijelova, izvodili zahvati analogni računskim operacijama. Razvoj znanosti i njezine primjene tražio je sve više računanja, te je ručno računanje došlo do svoje granice. Osnovna je granica ručnog računanja brzina kojom ljudi mogu računati. Zbrajanje ili oduzimanje izvodi se u desetak sekundi, a množenje i dijeljenje može potrajeti i dulje. Danas se potrebna računanja ne mogu ručno izvršiti u vremenu u kojem su potrebna. Povećanje broja ljudi koji računaju ne vodi uvijek prema cilju, jer mnoga izračunavanja treba obaviti prema redoslijedu. Poseban je problem što je čovjek pri računanju sklon pogreškama. Rezultati su uvijek nepouzdani, pa su potrebne usporedbe i razrađen sustav provjere. Ako rezultate računanja treba prepisati (npr. izrada računskih tablica), tada je to nov izvor pogrešaka koje se vrlo teško pronađe, a jedna ili dvije pogreške u tablicama mogu imati čak i teške posljedice.

G. P. Prony vodio je 1794. projekt francuske vlade za ručno izračunavanje niza matematičkih tablica. Tu su bili

uključeni logaritmi prirodnih brojeva od 1...200000 na 19 decimalnih mesta, sinus, tangensi, njihovi logaritmi itd. Projekt je trajao dvije godine i radilo ga je 70...100 ljudi. Pristup računanju bio je algoritamski, tj. suradnici su znali samo zbrajati i oduzimati. Upute za rad davala je grupa matematičara. Svaki su podatak računala dva čovjeka. Ako se njihovi rezultati nisu slagalii, račun se ponavlja. Procjenjuje se da su tablice (nikad nisu bile objavljene) sadržavale oko 88 milijuna znamenaka. Ako se procijeni da je taj rad trajao ukupno 600 dana po deset sati dnevno i da je u prosjeku radilo 80 ljudi, izlazi da bi za taj posao jedan čovjek trebao teorijski 480000 sati, odnosno približno 54 godine. Ako se zamisliti da bi taj posao obavilo današnje malo računalo, koje zbrajanje ili oduzimanje na 19 decimala može izvršiti za  $\sim 60 \mu\text{s}$ , ukupno bi računanje trajalo  $\sim 1000$  sekundi, a ispis rezultata  $\sim 30$  sati. Papir je prvi osnovni element za ručno računanje, i on je spremnik podataka. Na papiru su pohranjeni početni podaci, međurezultati i konačni rezultati. Upute za rad, a to je program, također mogu biti zapisane na papiru. Osnovne postavke računanja zna onaj koji računa. On je procesor. Čovjek, koji je razumno biće, obavlja niz radnji: čita, piše, računa, odlučuje o redoslijedu. U tom poslu čovjek može upotrijebiti pomagala, džepni kalkulator. To će ubrzati posao. Sve opisane radnje pokazane su na sl. 2a. Usporedba s računskim strojem (sl. 2b) pokazuje analogiju. Zaključak je: svaki računski sustav mora imati procesor za analizu i izvođenje programa, memoriju za pohranu programa i podataka, te ulazne i izlazne naprave. To je srž von Neumannovih zaključaka o strukturi računala.



Sl. 2. Osnovni sustavi za računanje. a) čovjek, b) računski stroj

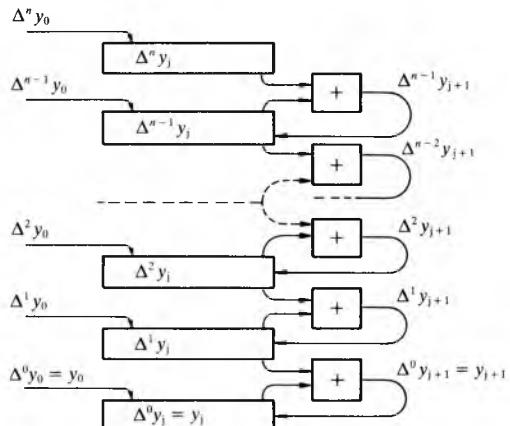
**Babbageova računala.** Ch. Babbage zanimalo se za računska pomagala. Prva njegova ideja bila je izgradnja tzv. diferencijskog stroja (*Difference engine*) pa je 1822. izradio mali model koji se mogao upotrebljavati. Podloga je diferencijskog stroja račun konačnih diferencija.

Za izračunavanje je potrebno imati zbrajala i registre za pohranu diferencija (sl. 3). Početne vrijednosti unose se u registre, a zatim se obavljaju pribrajanja. Ukupno je potreban  $n+1$  registar i  $n$  zbrajala ako se računanje želi provesti paralelno. Takvi su sustavi izumljivi za upotrebu u brzim interpolatorima. Potrebno je naglasiti da je Babbage nastojao načiniti mehaničke računske strojeve i da je glavni razlog neuspjeha nedovoljno razvijena tehnika izrade dijelova.

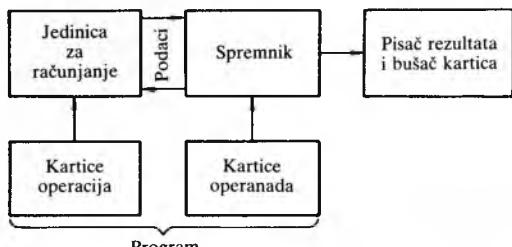
Već je 1834. Babbage počeo rad na *analytičkom stroju* (engl. *Analytical engine*), da bi izbjegao ograničenja koja ima diferencijski stroj. Osnovna zamisao analitičkog stroja, koji on nije uspio dovršiti, pokazana je na sl. 4.

Jedinica za računanje (*The mill*) bila je zamišljena za sve četiri aritmetičke operacije i računanje drugog korijena. Redoslijed operacija čita se iz kartica operacija. Kartice

operanada unose podatke u spremnik (*The store*) i tamo su na raspolaganju jedinici za računanje. Rezultati se ispisuju ili buše na kartice. Mogućnost izrade grafova bila je također razmatrana. Analitički stroj nije bio dovršen zbog već spomenute nedovoljno razvijene tehnike za izradu mehaničkih dijelova.



Sl. 3. Struktura Babbageova diferencijskog stroja



Sl. 4. Struktura Babbageova analitičkog stroja

Bušene kartice koje je Babbage predviđao razvio je Jacquard za upravljanje radom tkalačkih strojeva. Babbageove su kartice kartonske ili metalne, izbušene na potrebnim mestima i međusobno povezane žicom ili vrpcem u niz. Kartice se prebacuju preko šesteropravne prizme, koja okretanjem dovodi karticu po karticu pod mehanizam za čitanje. Čini se da je bila predviđena mogućnost gibanja kartica u oba smjera.

Osnovni je Babbageov doprinos zamisao računskog stroja za sve četiri računske operacije i računanje drugog korijena, upravljanje radom stroja bušenim karticama i stvaranje programa koji bi se mogao upotrijebiti za više skupova brojaka.

**Računalo ENIAC.** Prvi pokušaj izgradnje elektroničkog digitalnog računala načinio je J. V. Atanasoff na tadašnjem Iowa State College (sada: Iowa State University) već sredinom tridesetih godina. U toj se ustanovi već ranih dvadesetih godina proučavala mogućnost upotrebe kartičnih naprava za statističke primjene. Atanasoff se najprije bavio problemima rješavanja linearnih jednadžbi u operacijskim istraživanjima primjenom analognih računala. Uočivši velika ograničenja koja imaju analogna računala, Atanasoff se već 1935. godine okrenuo digitalnoj tehnici.

Svijest o nepogodnosti mehaničkih strojeva i poznavanje Eccles-Jordanova bistabilnog multivibratora, konstruiranog još 1919. god. (v. *Elektronika, Sklopovi*, TE 4, str. 552) uputili su ga na proučavanje mogućnosti primjene elektronike. Zaključio je da je binarni brojni sustav najpovoljniji za primjenu. U razdoblju od 1936. do 1937. godine Atanasoff je zamislio sustav s memorijom i logičkim sklopovima i razradio detalje tijekom 1938. Od 1939. radio je na ostvarenju sustava zajedno sa C. E. Berryjem. Načinili su uzorak računskog sklopa. Nakon toga su počeli razradu računala koje bi moglo rješavati sustav od 30 linearnih jednadžbi. Elektro-

nički dio sustava bio je dovršen 1942, ali je tada bio projekt napušten.

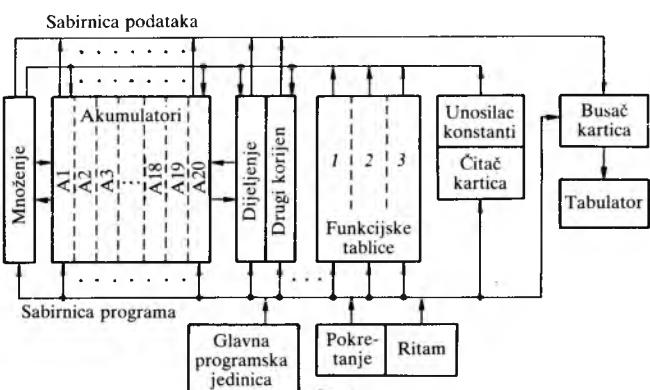
U Njemačkoj su mogućnost zamjene mehanike u računala posebno razmatrali negdje od 1934. H. Schreyer i K. Zuse. Schreyer je u svojoj disertaciji 1941. obradio elektromehaničke sklopove za tu namjenu. Iako je Zuse izgradio nekoliko računala (Z1-Z4), sva su ta računala bila elektromehanička i razorenja su u ratu. Njegov plan da 1942. počne rad oko elektroničkog računala s 1500 elektronskih cijevi bio je odbačen.

Za vrijeme rata u Engleskoj, a i u drugim zemljama, radilo se na kriptografskim napravama. Prve su bile elektromehaničke. Zatim je razrađen sustav s elektronskim cijevima, nazvan *Colossus*, koji je bio stavljen u upotrebu u prosincu 1943. Colossus je imao oko 2000 elektronskih cijevi. Podaci su se unosili na bušenoj papirnatoj vrpci. Uspjeh je toga stroja bio tolik da su odmah naručena sljedeća tri u poboljšanoj verziji i pod nazivom *Mark 2*. Prvi je od tih dovršen u lipnju 1944. Do kraja rata bilo je instalirano oko deset sustava Colossus, a nekoliko je bilo naručeno.

Pod pritiskom ratnih potreba počelo se 1942. na Moore School of Electrical Engineering, University of Pennsylvania, zajedno s Ballistic Research Laboratory, U. S. Army Ordnance Department, baviti oko izrade strojeva za izračunavanje balističkih tablica. Nosioci su razvoja bili J. W. Mauchly i J. P. Eckert. Veza s naručiocem bio je H. H. Goldstein. Stvarni je rad počeo Mauchlyjevim memorandumom 1942. i detaljnijim planovima iznesenim u izvještaju 1943. Zamisljeno računalo (sl. 5) nazvano je ENIAC. To je računalo, stavljeno u pogon 15. veljače 1946., uz razne nadopune radio do listopada 1953. Računalo je sadržavalo više od 18000 elektronskih cijevi, 70000 otpornika, 10000 kondenzatora, 6000 sklopaka, 1500 releja. Stroj je bio sastavljen od 30 jedinica smještenih u obliku slova U (ukupan obujam oko 90 m<sup>3</sup>). Za žarenje katodâ elektronskih cijevi bila je potrebna snaga od 80 kW, za anodne izvore 40 kW, a za ventilatore 20 kW. Jedno pribrajanje trajalo je ~3 ms. Registri su bili izrađeni kao prstenasta brojila (to je niz od deset bistabila povezanih u prsten i samo se jedan od njih nalazi u stanju jedan). Prikaz je brojeva bio *unarani*.

Unarni kod je položajan. Položaj jedinica u nizu nula ili položaj prijelaza od niza nula na niz jedinica određuje vrijednost. Na primjer, nula: 100000, jedan: 010000, dva: 001000 itd. Moguće su i druge kombinacije.

Program se stvarao, osim postavljanjem sklopaka, prospajanjem pojedinih jedinica. To je žično ostvaren program. Glavna programska jedinica sastojala se od deset jedinica koje broje programske korake i uklapaju programske veze. Programiranje na stroju ENIAC bilo je mukotrpno. S današnjeg gledišta to je bio programirljiv kalkulator.



Sl. 5. Struktura računala ENIAC

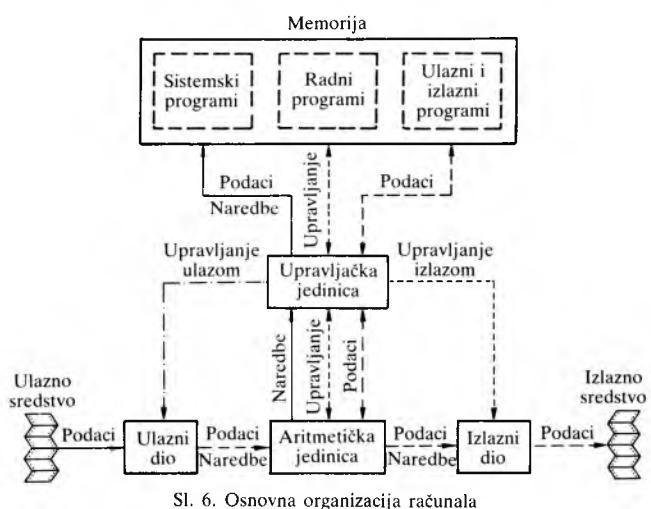
**Struktura digitalnih računala.** Osnovni su dijelovi elektro-ničkog računala: ulazne jedinice, memorije, upravljačke jedinice, aritmetička jedinica i izlazne jedinice. Podaci se uvode u računalo pomoću ulaznih sredstava: tipaka i sklopaka (ručno unošenje), ili bušenih vrpca i kartica, ili magnetskih

površina (vrpca, pločica, bubanj, disk). Rezultati se prikazuju pomoću izlaznih sredstava: bušenih vrpca i kartica, otisnutih ispisu, magnetskih površina (vrpca, pločica, bubanj, disk), prikaza na raznim zaslonima (grafičke stanice), crtačih naprava, numerički upravljanjih strojeva, procesnih upravljačkih naprava. Cjelokupnim radom upravlja program. To je logički sročen niz naredaba potrebnih za ispunjenje postavljene zadaće. Suvremena računala rade s memoriranim programom. Program je izrađen u jednom od četiri jezika. To su strojni, mnemonički, viši simbolički (FORTRAN, ALGOL, COBOL, ADA, BASIC, PASCAL, APL i dr.) i problemski orientiran jezik (ECAP, OSMP, APT i dr.).

Računalo razumije samo strojni jezik. Viši jezici olakšavaju programiranje. Prijevod iz mnemoničkog jezika (engl. *assembly language*) u strojni obavlja *mnenomici procesor* (engl. *assembler*). Prijevod iz višega u strojni jezik obavljaju posebni programi, *kompilatori* (engl. *compiler*). Kompilatori kao međukorak mogu sadržavati mnemonički procesor. Za programe napisane u simboličkom jeziku kaže se da su u simboličkom kodu. Nakon prijevoda program je u *premjestivom* (engl. *relocatable*) kodu. Program *punilac* (engl. *loader, collector, mapper*) pretvara premjestivi kod u *apsolutni strojni kod*. U apsolutnom kodu ustanovljene su stvarne adrese izvođenja programa, vezne adrese za skokove u potprograme i sl. Postupak pretvaranja simboličkog u premjestivi kod izvodi se bez sudjelovanja korisnika.

U memoriji računala nalaze se *sistemski* (ili *sustavski*) programi (upravljaju radom cjelokupnog računala) i *radni* (aplikacijski) program (jedan ili više njih) s prostorom za ulazne podatke i prostorom za izlazne podatke.

Osnovna organizacija računala naziva se *von Neumannovom organizacijom* (sl. 6).



Sl. 6. Osnovna organizacija računala

Ulez i izlez podataka spore su radnje. Ulezne podatke koji su na bušenoj kartici ili vrpci pogodno je prenijeti na magnetsku vrpcu, disk ili bubanj i zatim ih brzo prenijeti u memoriju na znak upravljačkog programa. Isto se tako mogu podaci pripremljeni za ispis najprije prenijeti na masovne magnetske memorije, a zatim ispisati na retkovnom pisaču. Za izvršavanje opisanih ulaznih i izlaznih operacija nije potrebna aritmetička jedinica, pa ulazne i izlazne jedinice mogu samostalno prenositi podatke. Sklopovi za analizu naredaba i za aritmetičke operacije dotele su nezaposleni. Ulezne i izlazne jedinice u takvim sustavima sadrže vlastitu memoriju i potreban upravljački sklop.

Rad je računala potpuno određen programom. To je logički slijed naredaba potrebnih za izvršenje zadatka. Sve se naredbe mogu svrstati u tri glavne skupine: operacijske, prijenosne i uvjetne naredbe.

*Operacijske naredbe* nareduju izvođenje aritmetičkih i logičkih operacija: PRIBROJI, ODUZMI, MNOŽI, DIJELI, I, ILI, NE, NILI, NI, EXILI itd. *Prijenosne naredbe* nareduju

prijenos podataka iz jednog dijela u drugi dio sustava (memorije – memorija, memorija – registri itd.). *Uvjetne naredbe* nareduju ispitivanje pojedinih uvjeta: paran, neparan, jednak nuli, pozitivan, negativan itd. i određuju promjenu adrese za sljedeću naredbu.

Naredbe i podaci jesu binarni brojevi. Pojedine naredbe nareduju da se određeni sadržaj smatra podatkom. U osnovi se svaki sadržaj smatra naredbom.

**Osnove izvođenja programa.** Osnova je izvođenja programa *pojedinačna obrada*. Računalo može izvoditi samo jedan program. Za svaki program koji se želi izvesti mora korisnik ili operator obaviti sve potrebne početne poslove. Pojedinačna je obrada karakteristična za sustave koji svojom arhitekturom pripadaju tzv. *prvoj arhitektonskoj generaciji*, bez obzira pripada li sklopovlje prvoj tehnološkoj generaciji (elektronske cijevi) ili čak četvrtvoj (mikroračunala).

Dalji razvoj arhitekture računala omogućio je automatizaciju usluga na računalu. Jedinice računala rade prema uputama *monitora*. To je *sistemski program* koji je stalno u glavnoj memoriji i obavlja niz poslova koje bi inače morao obaviti korisnik ili operator. Tada je moguće skupiti niz programa u grupu i čitavu grupu postaviti u ulaznu jedinicu. Stroj samostalno očitava i izvodi program po program. Takav se rad naziva *grupna obrada* (engl. *batch processing*) i dostignuće je sustav koji se po arhitekturi svrstava u *drugu arhitektonsku generaciju*. Vrijeme koje protekne od trenutka očitavanja programa pa do dobivanja izlaznog dokumenta ovisi o količini programa koji u ulaznom repu čekaju na izvođenje, njihovoj prednosti i njihovoj duljini. Korisnik ne može komunicirati s programom za vrijeme izvođenja. Tehničar sustava može na temelju podataka na upravljačkom stolu intervenirati, mijenjati prednost i sl. Sve se to događa bez znanja korisnika, koji bi obavljanje svog programa mogao narediti preko vrlo udaljenog, tzv. *daljinskoga grupnog terminala* (engl. *remote batch terminal*).

Daljinski terminal može biti izведен slično pisačem stroju (ispis se dobiva na papir ili se pojavljuje na zaslonu katodne cijevi ili na kakvom drugom zaslonu). Tada korisnik izravno unosi svoje naredbe u stroj. Osim toga, korisnik može uvesti program u izvođenje i komunicirati s njime u toku rada. Takav način rada zove se *konverzacijska obrada* (engl. *conversational, demand, interactive processing*). Pored spremnosti računala, osnovni je uvjet za konverzacijsku obradu odgovarajuća konstrukcija programa, koji mora imati predviđene odlomke za analizu upita i upute za donošenje odluka. Za vrijeme izvođenja i sam korisnik sudjeluje kao periferna jedinica. Vrijeme reakcije korisnika utječe na brzinu izvršenja programa. Sustav kojim se izvodi konverzacijski program treba osigurati dovoljno kratko vrijeme odziva. Inače rad postaje mučan i psihološki nepogodan. Vrijeme odziva je vremenski razmak koji protekne od zadnjeg upisanog znaka do prvog isписанog znaka. U to se vrijeme ne uračunava vrijeme potrebno za izračunavanje prema programu. Vrijeme odziva u granicama od 3–5 s smatra se prihvatljivim. Konverzacijski rad ima primjene u raznim područjima: nastava pomoću računala, inženjerske i medicinske primjene, informacijski sustavi. Kao i za grupnu obradu, konverzacijski terminali mogu biti vrlo udaljeni od računskog centra. Takav udaljeni rad najčešći je oblik rada u konverzacijskom postupku. Sustavi računala *treće arhitektonske generacije*, uz još mnoga druga svojstva, omogućili su istodobno konverzacijski rad s više terminala. Osim toga oni su omogućili najprije stvaranje *terminalnih mreža*, a kasnije i *računalnih mreža*. Za konverzacijski rad postoje konverzacijske verzije uobičajenih viših programskih jezika, npr. FORTRAN, ali za takav rad postoje i posebni jezici, kao npr. BASIC, APL i drugi. Višeprogramski i višeprocesorski rad, posebno na velikim i brzim računalima, povećali su mogućnost priključka više terminala.

Mnogi programi i korisnički zahtjevi ne mogu se izvesti grupnom obradom. Sve terminale treba poslužiti jednakopravno, bez uočljivog čekanja na dodjelu procesora. Rad s *vremenском подјелом* (engl. *time sharing*) daje nizu korisnika

takvu mogućnost istodobne upotrebe sustava da se svakome od njih čini da je jedini korisnik. S obzirom na mogućnost upotrebe, tri su glavne kategorije sustava s vremenskom podjelom: a) *Pasivni sustav*, u kojem se može provoditi pažljivo specificirana i planirana akcija. U takvu sustavu postoji mogućnost izvođenja nekoliko zadatača, pa je to *višezadačni sustav* (engl. *multitasking*). Opseg je aktivnosti ograničen. Korisnik komunicira s programom koji je stalno u sustavu i unosi predviđenu kategoriju podataka, te postiže predviđene odgovore. Takvi su sustavi npr. bankovni sustav, rezervacijski sustav avionskih, hotelskih i sl. ustanova, medicinski informacijski sustav i sl. b) *Jednojezični sustav*, u kojem se mogu istodobno izvoditi razni programi, ali svi u istom jeziku. c) *Višejezični sustav*, u kojem se istodobno mogu izvoditi razni programi u raznim jezicima.

Za uspješan rad sustav s vremenskom podjelom treba biti opremljen brzim masovnim memorijama za *prebacivanje programa* (engl. *swapping*) iz glavne memorije u pomoćnu i obratno. Rad s vremenskom podjelom moguć je na računalima opremljenim *operacijskim sustavom*. To je još jedna od karakteristika sustava treće arhitektonske generacije. Budući da su veliki sustavi asinkroni, mnogo se akcija odvija istodobno uz upotrebu raznih dijelova računala. Istodobnost u radu i umnogostručavanje pojedinih važnih dijelova u računalu, tzv. *parallelizam*, još je jedna značajka računala treće arhitektonske generacije.

Rad u *realnom vremenu* (engl. *real time*) zahtijeva da računalo bude dio zatvorene petlje koja nadzire neki proces. Program za rad u realnom vremenu stalno je aktivan i neprekidno se nalazi u glavnoj memoriji. Nijedan drugi program (osim drugog programa za rad s realnim vremenom s većom prednostju) ne može prekinuti akciju koju je počeo program za rad u realnom vremenu. Primjeri su upotrebe: vođenje proizvodnih procesa, traganje i vođenje raketa, te drugi procesi koji zahtijevaju neposrednu akciju. Već spomenuti rezervacijski i informacijski sustavi obično rade onako kako je svojstveno radu u realnom vremenu.

U sustavu za rad u realnom vremenu mogu se istodobno nalaziti dvije vrste programa. Prvi radi u realnom vremenu i to je *glavni program* (engl. *foreground*). Ako glavni program miruje i čeka da se ponovno aktivira, tada računalo izvodi *sporedni program* (engl. *background*). Taj se program može u svakom trenutku prekinuti.

Sustav je za rad u realnom vremenu interaktivan, ali konverzira češće s posebnim napravama nego s čovjekom. Treba napomenuti da jednostavniji sustavi s pojedinačnom obradom izravno priopćavaju korisniku rezultat čim on nastane, pa su to zapravo računala koja rade u realnom vremenu.

**Neke osobitosti u organizaciji računala.** Sva su računala zasnovana na istim načelima, ali različito ostvaruju zadatke, dakle postoji razlika u arhitekturi i organizaciji.

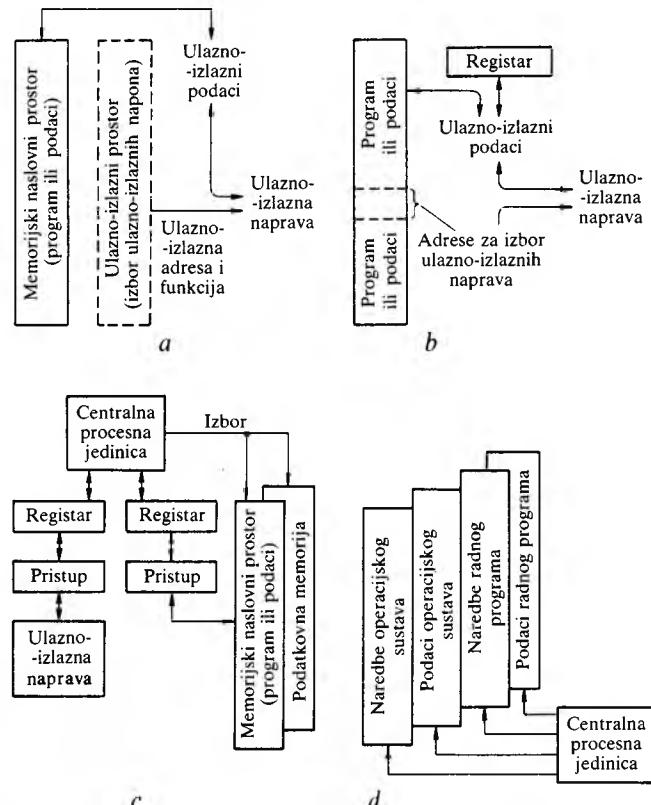
Računalo mora komunicirati s okolinom. U računalo se unose naredbe i podaci, a okolini se predaju rezultati u traženom obliku. Taj se promet između memorije i okoline ostvaruje na nekoliko načina.

Prvi se način ostvaruje upotrebom posebnih ulazno-izlaznih naredaba. Tada se stvara poseban signal koji naznačuje da je to ulazno-izlazni prijenos. Podatak na adresnoj sabirnici naznačuje koji je ulazno-izlazni kanal izabran i u koju svrhu, a na *podatkovnoj* se sabirnici nalazi podatak (sl. 7a). Kaže se da se prijenos odvija u *ulazno-izlaznom naslovnom prostoru* (a ne u memorijskom naslovnom prostoru).

Dругi se način osniva na upotrebi naredaba za pohranu ili dohvata iz memorije u okviru jedinstvenog *memorijskog naslovnog prostora*. Dio se tog prostora sklopovski odvoji za ulazno-izlazni prijenos (sl. 7b). Odabrana adresa kaže koja je naprava prizvana i što treba raditi na podatkovnoj sabirnici. Tada se ne može sav memorijski prostor upotrijebiti za potrebe memorije.

Treći način (sl. 7c) upotrebljava posebne *ulazno-izlazne pristupe* (engl. *port*). Svaki je pristup povezan s pripadnim registrom u centralnom procesoru, a naredba u programu

određuje da li se taj podatak prenosi u memoriju ili obrnuto. Može postojati poseban signal (jedan ili više bitova) koji kaže s kojom se memorijom, sistemskom, korisničkom ili podatkovnom odvija promet (sl. 7d). Signalni za odabiranje memorije mogu postojati i u računalima koja rade na prva dva načina.



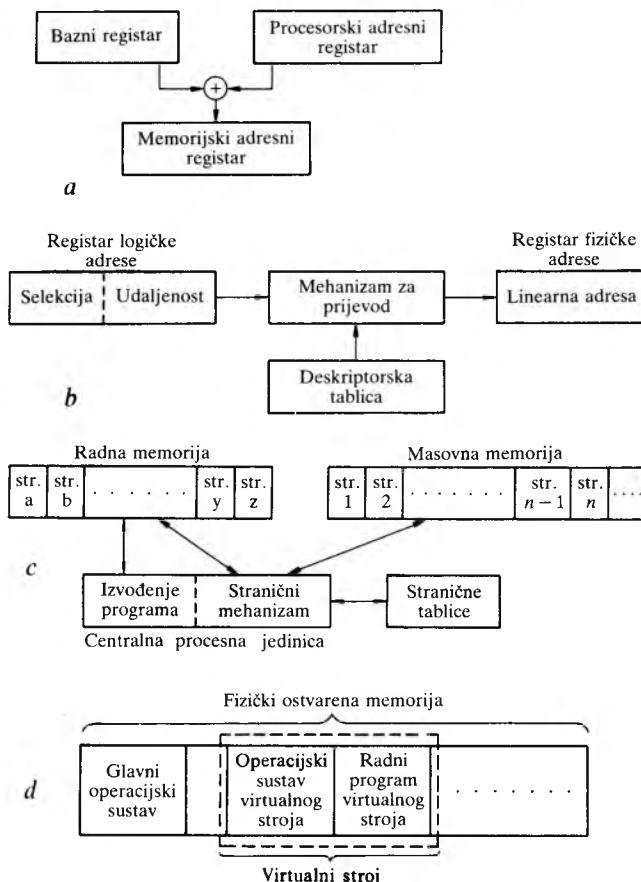
Sl. 7. Organizacija memorije i povezanost s okolinom. a) sustav s posebnim ulazno-izlaznim naredbama, b) izdvajanje dijela memorijskog naslovnog prostora za adresiranje ulazno-izlaznih naprava, c) upotreba ulazno-izlaznih registara, d) namjenska organizacija memorija

Centralna procesna jedinica može *izravno* dohvati svaki položaj u glavnoj memoriji koji je unutar naslovnog prostora određen brojem adresnih bitova. Memorijski se naslovni prostor može razdijeliti na *odsječke* (engl. *segment*) i poseban je *bazni registar* uperen na početak potrebnog odsječka (sl. 8a). Tada se položaj u memoriji određuje zbrojem sadržaja baznog registra i sadržaja adresnog registra. Sadržaj je adresnog registra udaljenost od položaja što ga određuje bazni registar. U tom je primjeru *stvarni memorijski naslovni prostor* jednak (ako bazni registar ne postoji) ili veći od onoga koji procesorski adresni registar može izravno naslovit. U nekim računalima, a posebno u najnovijim mikroprocesorima, unutarnji adresni registar može izravno naslovit mnogo veći *unutarnji naslovni prostor* (npr.  $2^{46}$  okteta = 64 teraokteta) nego što je predviđeno za *stvarni fizički ili linearni memorijski naslovni prostor* (npr.  $2^{32}$  okteta = 4 gigaokteta).

U računarstvu se obično upotrebljavaju tzv. binarni višekratni brojčanih jedinica, slični onima za decimalne jedinice. To su: *kilo* (znak K) u značenju  $2^{10} = 1,024 \cdot 10^3$ , *mega* (znak M) u značenju  $2^{20} = 1,048576 \cdot 10^6$ , *giga* (znak G) u značenju  $2^{30} = 1,073741824 \cdot 10^9$  i *tera* (znak T) u značenju  $2^{40} = 1,09951162776 \cdot 10^{12}$  imenovanih jedinica. Osim brojčane jedinice *bit* (v. *Digitalna računala*, TE 3, str. 315; v. *Impulsna i digitalna tehnika*, TE 6, str. 445) često se upotrebljava i brojčana jedinica *byte* (izg. bajt), što ju je u literaturu uvela tvrtka IBM. Byte je broj bitova koji se odjednom dohvata. U strojevima IBM to je 8 bitova i jednako je kodu za jedan znak. Ako byte ima 8 bitova, naziva se *oktet*, i to je standardni izraz Međunarodnog savjetodavnog odbora za telefoniju i telegrafiju (CCITT, Comité Consultatif International Téléphonique et Télégraphique).

Unutarnji se memorijski naslovni prostor naziva *virtualnim* ili *logičkim memorijskim naslovnim prostorom*. Mehanizam za prijevod adresa iz jednog prostora u drugi ugrađen je u

procesorsku jedinicu (sl. 8b). Podaci o rasporedu u virtualnom naslovnom prostoru i njegovoj vezi s fizičkim prostorom nalaze se u *deskriptorskoj tablici*, koja može biti *globalna*, za sve zadatke, ili *lokalna*, samo za pojedine zadatke. U oba se slučaja cijelokupni program nalazi u fizički ostvarenoj radnoj memoriji (tj. u onoj koju centralni procesor može izravno nasloviti).



Sl. 8. Pristup memoriji. a) iznadno naslovljavanje, b) prijevod iz virtualnog naslovnog prostora u fizički, c) virtualna memorija, d) virtualni stroj

Ponekad je potrebno omogućiti rad programima koji mogu za jedan ili više redova veličine premašiti fizički ostvarenu glavnu memoriju. Tada se stvara *virtualni memoriski prostor* (sl. 8c). Cijelokupni je radni program tada smješten na vanjskoj masovnoj memoriji (npr. diskovna memorija), a pojedini se *odsječi* ili *stranice* (engl. *page*) prenose u glavnu memoriju prema potrebama i zahtjevima programa. Stranice koje trenutno nisu potrebne vraćaju se u masovnu memoriju. O cijelokupnom prometu brine se posebni *stranični mehanizam* unutar centralne procesne jedinice.

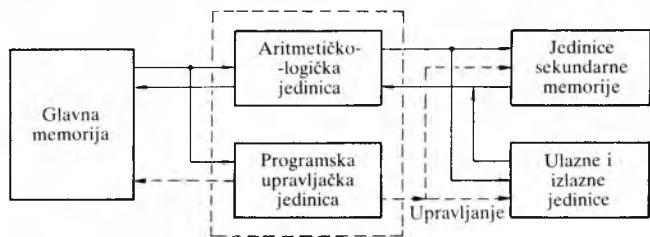
U naprednim složenim računalima sav se rad odvija pod nadzorom i uz pomoć *operacijskog sustava*. Vrlo je često potrebno pod okriljem jednog operacijskog sustava omogućiti u nekom određenom dijelu radne memorije rad i izvršavanje radnih programa pod nadzorom i uz pomoć nekoga drugog operacijskog sustava, tj. unutar jednog računala ostvariti rad nekoga drugog, zamišljenog računala. Takav se rad naziva *virtualnim strojem* (engl. *virtual machine*) (sl. 8d). Potrebno je istaknuti da se programska mora omogućiti da *povlaštene povrhate* operacijskog sustava virtualnog stroja mora izvršiti glavni operacijski sustav.

#### ARHITEKTURA I ORGANIZACIJA DIGITALNIH RAČUNALA

**Računala prve arhitektonске generacije.** Za vrijeme izgradbe računala ENIAC spoznalo se da su potrebna računala s pohranjenim programom, odnosno računala von Neumanove strukture. To su računala EDVAC, EDSAC, BINAC

itd. Sva su ta računala prve arhitektonске generacije. To su bila računala izvedena u tehnici elektronskih cijevi, pa ujedno pripadaju prvoj tehnološkoj generaciji. Računala koja imaju strukturu sličnu onoj prve arhitektonске generacije izvode se i u novijim tehnološkim generacijama. Većina su računala prve generacije unikati, razvijeni u sveučilišnim laboratorijima.

Osnovna je značajka računala prve arhitektonске generacije središnji položaj aritmetičko-logičke jedinice. Sav promet između memorije i ulazno-izlaznih jedinica odvija se preko akumulatorskog registra (sl. 9). Znak ili podatak koji se dohvata iz izlazne ili ulazne jedinice, odnosno iz jedinice sekundarne memorije (npr. jedinica s magnetskom žicom) najprije se pohranjuje u akumulator, a zatim prenosi iz akumulatora u glavnu memoriju. Isto se tako postupa u obrnutom smjeru. Za vrijeme ulazno-izlaznog prijenosa aritmetičko-logička jedinica nije zaposlena, a za vrijeme računanja nema ulazno-izlaznog prijenosa. Sklopovi takvih računala nisu dobro iskorišteni, a dio ih je i nezaposlen.



Sl. 9. Arhitektura računala prve generacije

Računala prve generacije otvorila su veliko područje elektroničkih digitalnih računala. Teškoće je u prvom redu bila u napravama koje su se mogle upotrijebiti za memoriju. Tadašnje su naprave bile zamršene, skupe i nepouzdane. ENIAC je bio, u neku ruku, programirljiv kalkulator bez prave memorije (kasnije mu je dodana mala feritna memorija). Ostala su računala bila s programom pohranjenim u memoriji. Sadržaji u pojedinim memorijskim adresama morali su se modificirati da bi se izveli programski skokovi. Zbog toga se isti program nije mogao ponavljati. Potrebno je bilo ponovno unošenje izvornog koda, jer indeksni registri još nisu bili poznati.

S gledišta arhitekture ta su računala orijentirana na aritmetički sklop. UNIVAC je razvio nešto *asinkrona* rada na ulaznim i izlaznim operacijama. Programi su se izvodili pojedinačno. Rad je s računalom bio lokalан. Sve su te činjenice bile uočene pa se razvila druga tehnološka i druga arhitektonска generacija računala.

**Računala druge arhitektonске generacije.** Računala prve generacije pokazala su i dokazala da je moguće sagraditi matematičke strojeve upotreboom elektroničkih sklopova. U to su doba elektronički elementi bile elektronske cijevi, koje nisu mogle potaknuti stvaraocu strojeva da krenu novim putovima. Za nove arhitekture bila je potrebna veća kompleksnost računala koja se nije mogla postići sklopovima s elektronskim cijevima. Pojava poluvodičkih elektroničkih elemenata, u prvom redu tranzistora, otvorila je nove mogućnosti. U jednak obujam i uz istu disipiranu snagu mogle su se smjestiti naprave koje su imale mnogo puta veću procesnu moć. Istodobno s razvojem tranzistora razvile su se i nove memorije temeljene na feritnim prstenčićima. Tada su se već mogle izgraditi memorije velika kapaciteta i dovoljno velike pouzdanosti. Zbog toga je bilo moguće ostvariti skoro svaku zamisao u arhitekturi računala. Pojava ulazno-izlaznih procesora koja je omogućila neovisan pristup centralnog procesora i ostalih jedinica do memorije dala je novu kvalitetu: započeo je daljinski rad s računalom.

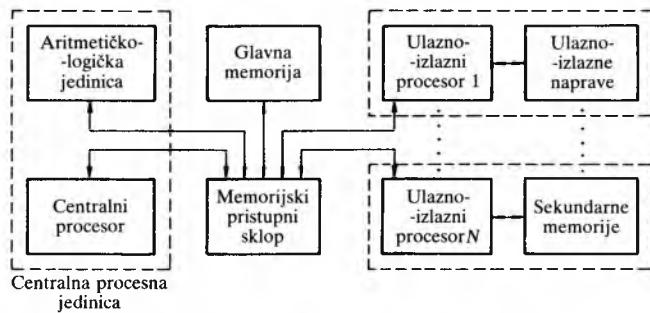
Gotovo je deset godina trebalo proteći od izuma tranzistora dok se izrada toliko usavršila da su oni postali jeftin i pouzdan elektronički element koji je mogao početi istiskivati elektronske cijevi iz upotrebe. Oko 1953. počinje izrada

računala s tranzistorima. Čini se da je prvo tranzistorizirano računalo bilo TX-0, izgrađeno u Lincoln Laboratory na Massachusetts Institute of Technology (MIT), a 1959. pojavljuju se takva računala na tržištu. To je početak računala druge tehnološke generacije. Tranzistori su omogućili izradu kompaktnijih sustava. Smanjenje dimenzija sastavnih dijelova i utroška električne energije (a to znači smanjenje razvijene topline koju iz stroja treba odvesti) omogućilo je gradnju moćnijih i kompleksnijih računala. Paralelno s razvojem tranzistora usavršene su feritne memorije. Stvaratelji računala konačno su dobili relativno brzu i pouzdanu a dosta jeftinu memoriju, tako da su se mogla početi graditi računala s velikom memorijom. Usporedno s tehničkim napretkom u izgradnji sklopova razvijala se arhitektura digitalnih računala. Uočeni su glavni nedostaci tadašnjih računala, a to je nemogućnost autonomnih asinkronih ulaznih i izlaznih operacija. Uveden je poseban *ulazno-izlazni proces* koji je omogućio ulaznim i izlaznim napravama pristup do memorije neovisno o aritmetičko-logičkoj jedinici. Aritmetičko-logička jedinica izvršava svoje zadatke, a u isto vrijeme teče ulazno-izlazni prijenos podataka. Budući da je rad pojedinih jedinica postao *asinkron*, razradena je *tehnika prekidâ* (engl. *interrupt*), uvedena na računalu UNIVAC 1103, pomoću koje jedinice javljaju centralnom procesoru da traže posluživanje. Razradeni su posebni *sistematski nadzorni programi*, tzv. *monitori*, koji upravljaju radom računala. Osim toga u arhitekturi stroja načinjen je još dalji napredak. Uveden je *indeksni registar* koji služi za modifikaciju memorijске adrese, pa se izbjegla potreba promjene sadržaja memorije. Razraden je *potprogramski skok* i povratak u glavni program, pa se isti program može automatski uzastopce ponavljati, a više se raznih programa sastavljuju kao nezavisne jedinice u jednu grupu i izvode automatski jedan za drugim. To je *grupno* (engl. *batch*) izvođenje programa. Broj korisnika računala postao je velik i počela je intenzivna izmjena iskustava. Rezultat je velik napredak u izradi *sistemske programske opreme* (engl. *system software*). Osim spomenutih monitora, to su mnemonički i jezični procesori, programi puniovi, uslužni programi za rad s periferijom i masovnim memorijama. Svi ti programi, zajedno s programima za primjenu kao što su matematički i statistički postupci, programi za razvrstavanje i spajanje (engl. *sort, merge*) te razni drugi programi za poslovnu, tehničku i informacijsku namjenu, slagali su se u *programske knjižice*. Kupci su uz stroj kupovali i takve knjižice programa. Sva opisana dostignuća atributi su druge arhitektonske generacije računala. Treba istaknuti da strojeva druge arhitektonske generacije ima i u izvedbi prve tehnološke generacije i u izvedbi kasnijih tehnoloških generacija. Pripadnost jednoj arhitektonskoj generaciji strojeva ne znači pripadnost istoimenoj tehnološkoj generaciji. Razvoj izrade sastavnih dijelova omogućio je brži napredak i ostvarenja koja možda u starijoj tehnici ne bi bila moguća.

Druga se arhitektonska generacija računala počela proizvoditi kao tzv. *porodica računala*. Korisnik prema svojim potrebama i mogućnostima odabire jedinice računala i slaze sustav. Unutar jedne porodice mogu se odabrat i različite centralne jedinice. Programi pripremljeni za manji sustav izvode se bez promjene na većemu.

Zbog masovne upotrebe računala, koja se upravo pojavila tijekom druge arhitektonske generacije, razvili su se programski jezici koji su, za razliku od mnemoničkih i simboličkih kodova, bili strojno neovisni. To su npr.: FORTRAN (1957), ALGOL (1958), COBOL (1960), BASIC i APL (1962), IPL (1957), LISP 1.0 (1960), SNOBOL (1964), GPSS (1961), SIMSCRIPT (1962) i SIMULA (1967). Strojno neovisni jezici omogućili su izmјenu iskustava i pridonijeli naglom širenju upotrebe računala.

Središnja točka u računalima prve arhitektonske generacije bio je registar za prijenos podataka u aritmetičkoj jedinici. Sustavi drugih generacija takav centralizam više nemaju (u računalu UNIVAC I već je bilo nekoliko registara koji su sudjelovali u prijenosu podataka). Osnovni oblik arhitekture računala druge generacije prikazuje sl. 10.



Sl. 10. Osnovna arhitektura računala druge generacije

Glavnoj se memoriji pristupa preko posebnog pristupnog sklopa. Radom pristupnog sklopa upravlja centralni procesor. Preko pristupnog sklopa s memorijom se povezuju centralni procesor, aritmetičko-logička jedinica i ulazno-izlazni procesor. Dopuštenje za pristup glavnoj memoriji daje centralni procesor. On ujedno izdaje naloge za početak rada ulazno-izlaznih procesora, koji sada rade neovisno o aritmetičko-logičkoj jedinici.

**Računala treće arhitektonske generacije.** Mnoga su se svojstva računala druge arhitektonske generacije mogla ostvariti upotrebom tranzistora. Oni su omogućili povećanje sposobnosti računala i ugradnju novih svojstava. Tranzistori su zajedno s feritnim memorijama omogućili izgradnju računala koja su bila osnova za stvaranje računskih centara s jedne strane, a za vodenje znanstvenih i tehničkih procesa s druge. Sistemska programska oprema snažno se razvila i pokazala dalje mogućnosti upotrebe i razvoja računala. Napredak poluvodičke elektronike omogućio je izgradnju *integriranih sklopova* koji na istoj silicijskoj pločici imaju ostvaren jedan ili više logičkih sklopova (v. *Poluvodiči*, TE 10, str. 658). To je potaklo izgradnju još sposobnijih računala. Ujedno je povećan broj jedinica unutar jednog računala i ostvarena paralelnost sklopova. Pojavila su se višeprocesorska računala i višeprogramska rad. Računala poslužuju sve veći teritorij i postavljena je osnova za razvoj terminalnih i računalnih mreža. Integrirana je tehnologija počela ostvarivati i poluvodičke memorije, koje zbog svoje cijene i kapaciteta bitno mijenjaju tehniku izgradnje računala. Treća arhitektonска generacija otvara put prema novoj, četvrtoj generaciji računala koja će svojstva treće generacije uspjeti ostvariti kao računala malih dimenzija i niske cijene.

Nije jednostavno razlučiti računala na ona iz druge i ona iz treće arhitektonske generacije. Ne postoji čvrsta granica pa i prijelaz iz druge u treću generaciju nije oštar. Neka računala druge već imaju neka svojstva treće generacije. Obično se smatraju računalima treće arhitektonske generacije ona koja su izgrađena sredinom šezdesetih godina i kasnije. Kako se u isto doba pojavljuju i sklopovi treće tehnološke generacije (integrirani sklopovi), često se neopravданo smatra da je svako računalo izgrađeno s integriranim sklopovima računalo treće arhitektonske generacije.

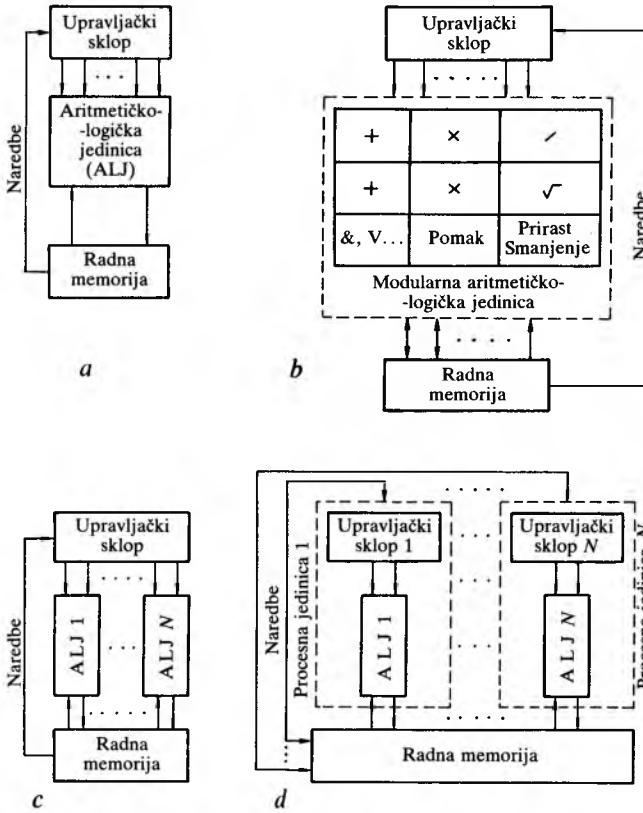
Svojstva su treće arhitektonske generacije: *smanjenje dimenzija i cijene* zbog upotrebe integriranih sklopova; *potiskivanje feritnih memorija* ili memorija s tankim magnet-skim filmom upotrebom novih poluvodičkih memorija (niža cijena poluvodičkih memorija omogućuje izgradnju sustava s vrlo velikim naslovnim prostorom); *upotreba mikroprogramiranja*, što pojednostavljuje konstrukciju i povećava prilagodljivost računala raznim zahtjevima; *izgradnja sustava s naglašenom paralelnosti* za istodobnu obradu više programa ili više skupova podataka (upotrebljava se višeprogramska rad, višeprocesni rad, aritmetičke i naredbene cjevovodne jedinice); *izrada operacijskih sustava* (poseban skup tzv. sistemskih programa) koji imaju sveopći nadzor nad dodjelom sredstava računala (engl. *computer, system resources*), rasporeduju izradu pojedinih zadataka (engl. *task scheduling*), rješavaju sukobe u zahtjevima za dodjelu pojedinih sredstava, upravljaju dodjelom memorije u logičkom ili fizičkom naslovnom prostoru.

## RAČUNALA

Paralelni sklopovi, istodobni rad i operacijski sustav najvažnija su obilježja treće arhitektonске generacije računala. Računala prve generacije (bez obzira na tehnologiju izrade) izvode programe pojedinačno. Računala druge generacije imaju sistemski program monitor, koji omogućuje grupno izvođenje programa. Računala treće arhitektonске generacije imaju *operacijski sustav*, a on omogućuje, uz sve ostalo, i rad u *vremenskoj podjeli*, te se brine za paralelni rad sustava u obliku višeprogramskog i više procesnog rada. U doba treće generacije stvaraju se *mreže računala*.

Osnovna su obilježja paralelnosti i istodobnosti u radu: *preklapanje dohvata*, analize i izvođenja pojedine naredbe unutar pojedinog programa izradom naredbenih i aritmetičkih cjevovoda; *postojanje višestrukih aritmetičko-logičkih jedinica* za sve ili za pojedine računske operacije.

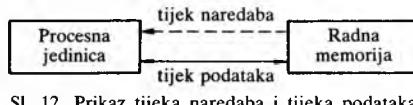
Osnovni oblici računala bez ugrađene paralelnosti i s ugrađenom paralelnosti u aritmetičko-logičkoj jedinici prikazani su na sl. 11. Na sl. 11a vidi se prospojni put koji povezuje jedinstvenu aritmetičko-logičku jedinicu s radnom memorijom. Na sl. 11b aritmetičko-logička je jedinica rastavljena na niz neovisnih jedinica koje samostalno izvršavaju svoje zadatke i samostalno prenose podatke u memoriju ili iz nje. Memorija se također izvodi kao niz jedinica s neovisnim pristupom. Primjer na sl. 11c ima nekoliko potpunih i neovisnih aritmetičko-logičkih jedinica upravljenih iz jednog upravljačkog sklopa. To je primjer *vektorskih ili grupnih računala*. Sustav na sl. 11d sastoji se od više samostalnih procesora od kojih svaki može izvesti po dio nekog programa ili cijelokupni program.



Sl. 11. Osnovni oblik računala. a) bez ugradene paralelnosti, b) s ugradenom paralelnosti u aritmetičko-logičkoj jedinici, c) s više aritmetičko-logičkih jedinica, d) s više centralnih jedinica

Primjena svih dostignuća treće generacije omogućuje ostvarenje vrlo kompleksnih i vrlo brzih računala, tzv. *superračunala*. Noviji razvoj zamjenjuje superračunala *raspodijeljenim računalima*, odnosno *raspodijeljenom obradom* kad god je to moguće. Sklopovi četvrte tehnološke generacije dosta su pridonijeli mogućnosti stvaranja raspodijeljenih sustava.

Osnovni su pothvati neke procesne jedinice: dohvata naredaba i operanada iz radne memorije, izvođenje naredaba i pohranu rezultata u memoriju. Koraci povezani s obradom naredbe stvaraju *naredbeni ciklus*. Naredbe se dohvataju uzastopice prema redoslijedu određenom programom i stvaraju *tijek naredaba* koji je usmjeren iz radne memorije u procesnu jedinicu. Operandi stvaraju *tijek podataka* koji je dvosmjeran. Prikaz tih dvaju tijekova za neko jednostavno računalo vidi se na sl. 12. Pojmovi tijeka naredaba i tijeka podataka potrebni su u razmatranju klasifikacije procesorskog paralelizma. Takvu je često upotrebljavaju klasifikaciju načinio M. J. Flynn i ona je zasnovana na broju istodobnih tijekova naredaba i podataka u procesoru za vrijeme izvođenja programa. U postupku ocjene paralelizma u sustavu potrebno je pretpostaviti da procesor radi punim kapacitetom, tako da se iskorištava njegov puni stupanj paralelizma. Tom zahtjevu treba biti prilagođen i radni program. Sa  $m_i$  i  $m_d$  označiti će se minimalni brojevi tijekova naredaba, odnosno podataka koji se aktivno obrađuju u bilo kojem od glavnih koraka u obradi naredbe. To su: izrada adrese sljedeće naredbe, dohvata naredbe, prepoznavanje naredbe, izrada adrese operanda, dohvata operanda, izvođenje naredbe, pohranu rezultata. Veličine  $m_i$  i  $m_d$  označavaju *mnogostruktost* procesora i mjera su njegova paralelizma. U postupku određivanja  $m_i$  i  $m_d$  treba naći minimalan broj tijekova u bilo kojoj točki, jer se ukupna sposobnost paralelne obrade određuje prema dijelu koji ima najniži stupanj paralelnosti.



Sl. 12. Prikaz tijeka naredaba i tijeka podataka za jednostavno računalo

Računala se mogu s obzirom na stupanj paralelnosti razvrstati u četiri osnovne grupe (neko računalo ne mora idealno odgovarati toj klasifikaciji): 1) *Jednostruki tijek naredaba, jednostruki tijek podataka* SISD (engl. *Single Instruction stream, Single Data stream*):  $m_i = m_d = 1$ . Računala vrste SISD (sl. 11a) imaju centralnu procesnu jedinicu opremljenu jednom aritmetičko-logičkom jedinicom koja je namijenjena izvođenju skalarnih operacija. 2) *Jednostruki tijek naredaba, višestruki tijek podataka*, SIMD (engl. *Single Instruction stream, Multiple Data stream*):  $m_i = 1, m_d > 1$ . U tu se grupu svrstavaju računala s jednom programskom upravljačkom jedinicom, a s umnogostručenim izvršnim (računskim) jedinicama. Računalo ILLIAC IV posebno je pogodan primjer računala vrste SIMD jer ima 64 računske jedinice. U tu se grupu mogu svrstati i razni asocijativni procesori, te procesori s memorijom koja se naslovljuje sadržajem (asocijativne memorije). Takvi sustavi istodobno obrađuju, na temelju jedne naredbe, više skupina podataka. Primjer takva računala je STARAN, koje je izradila tvrtka *Goodyear Aerospace Corporation*. Sustavi na sl. 11b i 11c pripadaju skupini SIMD. Računalima vrste SIMD mogu se smatrati i neki cjevovodni sustavi, jer se u jednom cjevovodu izvodi jedna naredba na nizu podataka. 3) *Višestruki tijek naredaba, jednostruki tijek podataka*, MISD (engl. *Multiple Instruction stream, Single Data stream*):  $m_i > 1, m_d = 1$ . Prilično je teško naći sustav koji odgovara uvjetima MISD. U tu se grupu često svrstavaju cjevovodni sustavi ako se promatraju s gledišta da se u svakom odsječku cjevovoda jedan te isti tijek podataka podvrgava drugoj mikronaredbi. 4) *Višestruki tijek naredaba, višestruki tijek podataka*, MIMD (engl. *Multiple Instruction stream, Multiple Data stream*):  $m_i > 1, m_d > 1$ . To su računala koja mogu istodobno izvoditi nekoliko programa, pa su to više procesorska računala (sl. 11d).

Paralelizam se u računalima uvodi da bi se povećala propusnost. Kao mjera propusnosti može se upotrijebiti *širina pojasa podataka*, tj. neka veličina  $b_D$  koja iskazuje najveći broj rezultata koje promatrani sustav može izraditi u jedinici vremena. Takva često upotrebljavana mjera jest broj operacija u pomicnom brojnom zarezu u sekundi, tzv. *flops* (engl.

*floating point operations per second*). Superračunalima se mogu smatrati ona kojima ta mjera nadmašuje iznos od 20 megaflopsa. Vrhunski modeli imaju propusnost od  $\sim 10$  gigaflopsa. Često nudena mjera broj naredaba u sekundi, tzv. *ips* (engl. *instructions per second*) ne daje uvid ni u paralelnost ni u kompleksnost sustava. Suvremeni mikroprocesorski sustavi mogu postići propusnost veću od milijun naredaba u sekundi, ali ako se promatraju operacije u pomicnom brojnom zarezu, tada propusnost može iznositi desetak tisuća flopsa.

### Mikroprogramirani sustavi

Mikroprogramirano je upravljanje rasporedivanje upotrebe osnovnih sredstava računala: multivibratorskih sklopova, logičkih i aritmetičkih sklopova, sklopaka i propusta, registara i spremnika, sabirnice, memorija itd. na temelju posebnog programa, tzv. *mikroprograma*. Mikroprogram je većinom pohranjen u posebnoj upravljačkoj memoriji. Danas su mnoga računala, posebno mikroprocesorska, izvedena mikroprogramska. Samo manji dio proizvođača daje podatke jesu li njihova računala mikroprogramirana. Za ona koja su predstavljena takvima i za koja postoji mogućnost izradbe vlastitog programa vrlo su rijetko na raspolaganju tvornički izradena programska pomagala. Podaci o strukturi i načinu izrade mikroprograma često se smatraju tvorničkom tajnom.

**Osnovno o mikroprogramiranju.** Svako digitalno računalo zahtijeva detaljnije upute kada i koji pothvat treba poduzeti. Te se upute nazivaju programom i to je razumno sročen slijed naredaba. Rezultat izvršavanja toga slijeda, programa za rad, mora biti jednak onom učinku koji je stvaralač programa zamislio. Naredbe se računalu predaju u onom obliku koji ono razumije. To su *strojne naredbe* sastavljene od niza binarnih znamenaka. Svaka dopuštena kombinacija ima određeno značenje koje računalo mora otkriti i prema njoj postupiti. Značenje može biti npr.:

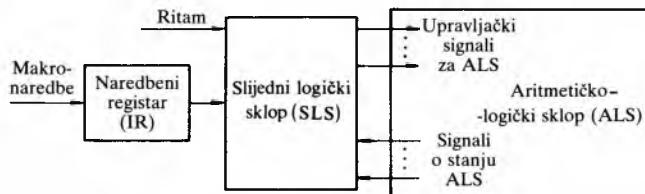
00010  $\Rightarrow$  očisti i napuni akumulator,

10000  $\Rightarrow$  skoči na zadanu adresu u memoriji,

01000  $\Rightarrow$  pomakni sadržaj akumulatora udesno, itd.

Svaka strojna naredba nalaže upravljačkoj i aritmetičko-logičkoj jedinici izvršavanje novoga, unaprijed predviđenog slijeda pothvata, kojega korisnik ne mora biti svjestan i ne može na njega utjecati. Tako, npr., kad je zadana prva naredba, novi program mora prvo postaviti sve bistabile u akumulatoru u logičko stanje nula, zatim mora odrediti adresu operanda i narediti memorijskoj upravljačkoj jedinici dohvata iz memorije. Nakon dovršena dohvata upravljačka jedinica treba narediti da se taj podatak pribroji u akumulator i obavijestiti da je tražena naredba dovršena. U većine je računala takav program ostvaren pogodnim povezivanjem potrebnih sklopova. Bilo kakva promjena programa vrlo je teška, ona je praktički gotovo neizvediva. Analizom pojedinih strojnih naredaba i pregledom upotrebe pojedine vrste sklopova može se zaključiti da bi se sklopovi računala mogli ekonomičnije izvesti ako bi se ostvarilo da upute za upotrebu pojedinog sklopa budu pohranjene u posebnoj memoriji. Upute u toj posebnoj memoriji nazivaju se *mikronaredbe*, a memorija u kojoj su mikronaredbe pohranjene naziva se *upravljačka memorija* (engl. *control store*). Budući da su naredbe za upotrebu sklopova za izvršenje neke strojne naredbe nazvane mikronaredbama, u ovom odlomku naziv *mikronaredba* označava strojnu naredbu. Općenito naziv mikronaredba označava nakupinu od nekoliko strojnih naredaba. Mikronaredbe opisuju radove s registrima i drugim sklopovima u aritmetičko-logičkoj i ostalim jedinicama računala, pa se jezik za opis mikronaredbe naziva *registarskim jezikom*.

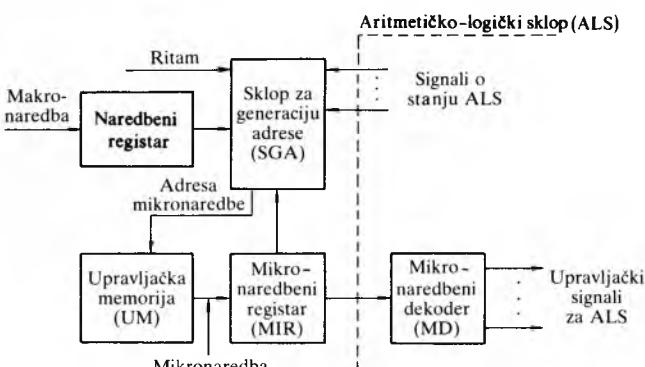
Sklopovi koji izvode makronaredbe mogu se prikazati kao na sl. 13. Makronaredba se pohranjuje u *naredbeni register* (engl. *instruction register*). Na temelju njegova sadržaja i na temelju signala o stanju aritmetičko-logičke jedinice slijedni logički sklop daje slijed upravljačkih signala za procesor, koji upravljaju prijenosom podataka između pojedinih registara,



Sl. 13. Sklopoško izvođenje makronaredbe

odnosno upravljaju i aktiviraju aritmetičke i logičke sklopove. U mikroprogramiranom je sustavu slijedni logički sklop zamijenjen upravljačkom memorijom i sklopom za generaciju adresa, kao što je pokazano na sl. 14.

Makronaredba pohranjena u naredbenom registru dovodi se do sklopa za generaciju adresa i zajedno sa signalima o stanju aritmetičko-logičkog sklopa i sa signalima što ih daju u tu svrhu odabrani bitovi mikronaredbe pohranjeni u mikronaredbenom registru određuje adresu mikronaredbe u upravljačkoj memoriji. Novo odabrana mikronaredba prenosi se u mikronaredbeni registar. Mikronaredbeni dekoder dekodira dobivenu mikronaredbu i daje na svom izlazu potrebne upravljačke signale za aritmetičko-logički sklop. Istodobno se na temelju signala o stanju aritmetičko-logičkog sklopa i sadržaja mikronaredbenog registra određuje adresu sljedeće mikronaredbe. Taj se postupak ponavlja dok se ne izvedu sve mikronaredbe koje pripadaju makronaredbi pohranjenoj u naredbenom registru. Nakon toga započinje postupak dohvata nove makronaredbe itd.



Sl. 14. Mikroprogramsko izvođenje makronaredbe

Promjenom sadržaja upravljačke memorije može se za promjenjenu kombinaciju bitova u naredbenom registru, dakle za makronaredbu nekog drugog računala, postići potreban slijed mikronaredaba za tu novu makronaredbu, a da se nisu promjenili sklopovi u aritmetičko-logičkom sklopu i njegovoj upravljačkoj jedinici. Na mikroprogramiranom se računalu može promjenom sadržaja upravljačke memorije omogućiti izvođenje makronaredaba predviđenih za neko drugo računalo. Taj se postupak naziva *emulacija* ili *oponašanje*.

Zahtijevane pothvate izvode u računalu *sklopovi* (engl. *hardware*). *Program* je redoslijed izvođenja pothvata. *Programska oprema* (engl. *software*) skup je svih potrebnih programa. U mikroprogramiranim računalnim sustavima posebni upravljački *mikroprogram* sjedinjuje pojam i funkcije sklopova i programske opreme te se pohranjuje u *ispisnu memoriju* (ROM, tabl. 2). Takav proizvod, pohranjeni mikroprogram zajedno s memorijskim sklopom, zove se *programski sklop* (engl. *firmware*).

Zamisao mikroprogramskih sustava iznio je 1951. M. V. Wilkes. Iako je ta konceptacija odmah bila zapažena, nije se mogla u to doba pogodno ostvariti. Visoka cijena i sporost postojećih memorija bila je glavna zapreka. Ipak je tehniku mikroprogramiranja bila tada djelomično upotrijebljena (diodne matrice za 4800 bitova) u računalu *Whirlwind I*. Upotreba se mikroprogramiranja proširila pošto je ta tehniku sredinom šezdesetih godina bila upotrijebljena u računalima

serije IBM S/360. Ta su računala mogla, između ostalog, emulirati naredbe sustava IBM 7090. Posebno se tehnika mikroprogramiranja razvila pomoću sklopova treće i četvrte tehnološke generacije. Osobito su važne bile poluvodičke memorije. Mikroprocesori su mikroprogramirani sustavi, iako to korisnik nije u stanju uočiti.

Makronaredbe (strojne naredbe) i mikronaredbe (registarske naredbe) dva su različita postupka, ali postoji mogućnost da se izrada mikroprograma zabunom izjednači s programiranjem u strojnem jeziku. Mikroprogram je razina programa najbliža sklopovima. Ne samo da se mikronaredbama mogu interpretirati vrlo složeni strojni jezici (tj. vrlo složene makronaredbe) već se mogu interpretirati i naredbe programskih jezika vrlo visoke razine. Isto vrijedi i za interpretaciju problemski orijentiranih jezika. Upotreboom mikronaredaba mogu se izravno ostvariti zamršeni algoritmi. Sve to traži veliku upravljačku memoriju.

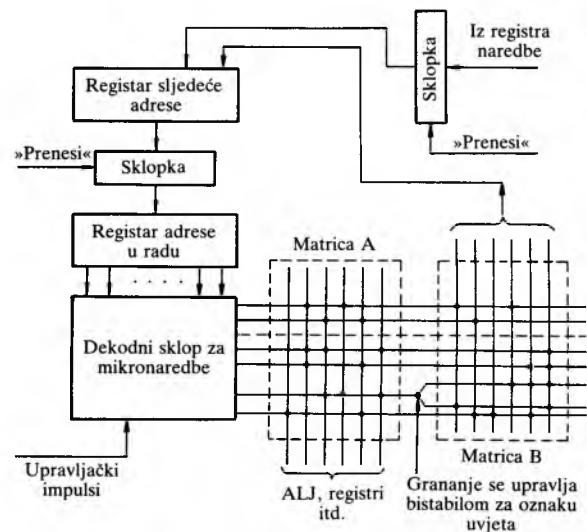
Osobitosti su mikroprogramiranja prema strojnem programiranju: *Razina upravljanja*; mikronaredbe upravljaju primitivnim sklopovskim sredstvima, dok makroprogram radi sa strukturama podataka što ih je izabrao i oblikovao korisnik. *Izbor naredaba*; skup je mikronaredaba općenito malen, dok je skup makronaredaba velik. *Paralelnost*; ovisno o izvedbi moguće je istodobno izvesti nekoliko mikronaredaba. *Vrste memorija*; upravljačke su memorije brze i skupe memorije, pa su zato ograničena kapaciteta.

Budući da za izvođenje makronaredbe treba izvesti nekoliko mikronaredaba, posebna se pažnja mora posvetiti vremenu potrebnom za dohvat mikronaredbe. Na temelju toga može se zaključiti da upravljačka memorija mora imati vrijeme čitanja oko deset puta kraće od odgovarajućeg vremena radne memorije.

Mikroprogramiranje ima sljedeća svojstva: a) *Niska cijena*; mala i srednja mikroprogramirana računala s velikim skupom makronaredaba mogu se jeftinije ostvariti nego ona u kojima se makronaredba ostvaruje pomoću sklopova; potreban veliki broj slijednih sklopova u aritmetičko-logičkim jedinicama isplati se graditi samo za vrlo brza velika računala. b) *Prilagodljivost*; mikroprogram se može relativno lako mijenjati, što omogućuje uvođenje novog skupa makronaredaba. Promjena se može izvršiti i na mjestu gdje računalo radi. Tako se može računalo prilagoditi novim potrebama i proizvesti mu uporabni vijek. c) *Jednostavan razvoj i održavanje*; mikroprogramirana računala imaju manje sklopova, pa su jednostavnija i lakše ih je razviti. Zbog manje sklopova mikroprogramirana su računala pouzdanija i jednostavnija za održavanje. d) *Brzina*; iako brzina izvođenja pojedinačne makronaredbe ne mora biti jednaka onoj u sklopovskim izvedbama, izvođenje nekog mikroprogramiranog algoritma može često biti brže i uspješnije.

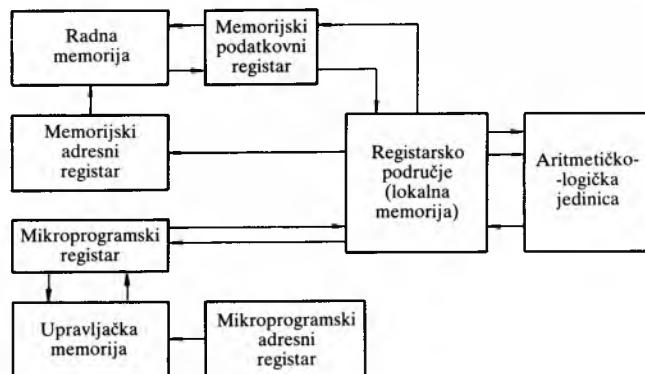
**Mikroprogramirana računala i organizacija mikroprograma.** Osnovna ideja mikroprogramiranog računala prema M. V. Wilkesu pokazana je na sl. 15. Osnovni dio, upravljačka memorija, sastoji se od dviju ispisnih memorija, matrice A i matrice B. Žicu rijeći tih matrica odabire dekodni sklop na temelju adrese za mikronaredbu u radu. Bitovi matrice A upravljaju radom aritmetičko-logičke jedinice, registara itd. Bitovi matrice B nadopunjaju sadržaj registra adrese sljedeće mikronaredbe tako da se zajedno s operacijskim dijelom makronaredbe izrađuje potrebna adresa. Svi ti sklopovi zajedno mogu se nazvati mikroupravljačkom jedinicom, dok se sklopovi kojima upravlja matrica A nazivaju registarskom upravljačkom jedinicom. Upravljački signali vode mikroupravljačku jedinicu kroz potrebna stanja i određuju trenutak prijenosa podataka u registarsku upravljačku jedinicu i u vlastite registre. Sljedeća adresa ovisi o postavljenom operacijskom kodu i o stanju stroja u koraku što ga upravo izvodi. Grananje se upravlja bistabilom uz označku mjesta.

Na temelju tih načela izrađuje se pojednostavnjena opća arhitektura mikroprogramiranog računala (sl. 16). U nekom stvarnom računalu dodat će se još neki potrebni sklopovi i možda će se modificirati pojedini spojni putovi, ali osnovna slika ostat će ista.

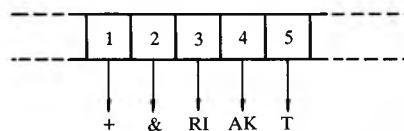


Sl. 15. Wilkesovo načelo mikroprogramiranog računala

Mikronaredbe se najčešće razvrstavaju u dva razreda: 1) *Vertikalne mikronaredbe* upravljaju osnovnim operacijama aritmetičko-logičkog sklopa. Njima se naređuju osnovne operacije nad operandima u registrima, pa donekle sliče strojnim (makro)naredbama. Duljina vertikalne mikronaredbe obično je između 10 i 30 bitova. 2) *Horizontalne mikronaredbe* istodobno upravljaju nizom pothvata i sredstava sustava. One omogućuju paralelnost u računalu. Horizontalna mikronaredba može istodobno upravljati radom više aritmetičko-logičkih sklopova, prijenosom kroz nekoliko registara itd. Zbog obilja podataka što ih treba sadržavati horizontalna je mikronaredba duga; često ima ~90 bitova.



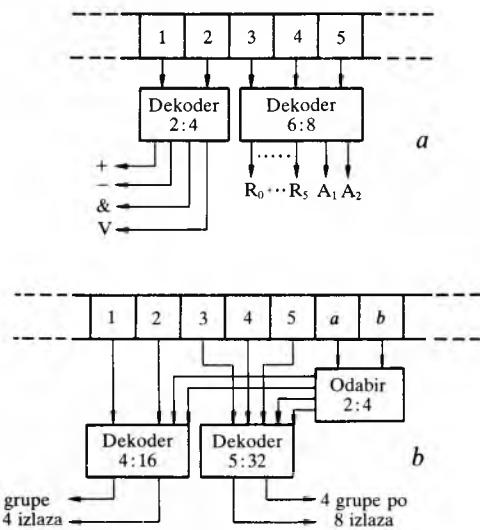
Sl. 16. Opća arhitektura mikroprogramiranog sustava



Sl. 17. Horizontalni ili razvijeni oblik mikronaredbe; svaki bit upravlja jednim pothvatom ili jednim sredstvom

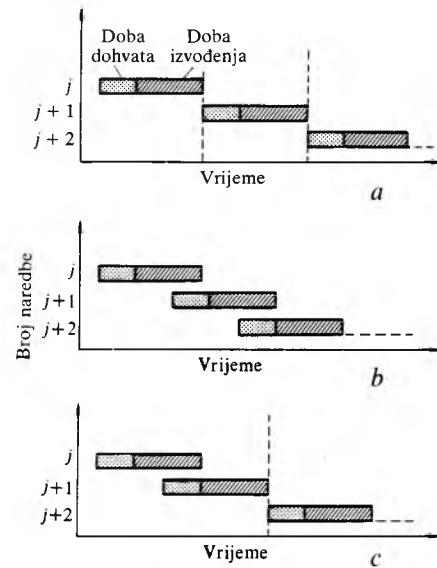
Mikronaredbe su, dakle, vertikalne ili horizontalne. Od luka kamo pojedina mikronaredba pripada donosi se na temelju broja sredstava kojima ona upravlja. Za vertikalne mikronaredbe kaže se da su *zbijena oblika*, a horizontalne da su *razvijena oblika*. Stupanj zbijenosti ovisi o načinu dekodiranja informacije sadržane u mikronaredbi. Jednostavno razvijena mikronaredba (sl. 17) pokazuje da svaki bit upravlja jednim izvorom ili jednom operacijom. Ako se predviđi jedna razina dekodiranja (sl. 18a), onda isti broj bitova upravlja s više sredstava ili pothvata. U tom primjeru jednorazinsko dekodiranje omogućuje, s istim brojem bitova, odabir četiriju

operacija, šest registara i dvaju akumulatora. Dodavanjem dvobitovnog dekodera na višoj razini (sl. 18b) postiže se mogućnost generiranja četiriju grupa sa po četiri upravljačka signala na izlazu jednog dekodera i četiriju grupa sa po osam signala na izlazu drugog dekodera. Postignut je zbijen oblik mikronaredbe. Vremenska analiza pokazuje da će najmanje vremensko kašnjenje biti postignuto s razvijenom horizontalnom mikronaredbom, a najveće s višerazinskom zgusnutom vertikalnom mikronaredbom.



Sl. 18. Shema dekodiranja. a jednorazinsko, b dvorazinsko dekodiranje

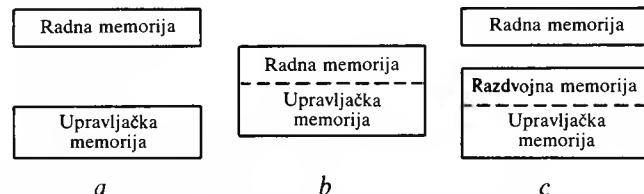
Adresa sljedeće mikronaredbe ovisi, između ostalog, o prethodnom stanju. Kao i za svaku drugu vrstu naredaba, razlikuju se dva glavna vremenska perioda u toku rada: period dohvata i period izvođenja. S obzirom na ta dva perioda, rad je serijski, paralelan i miješan (serijsko-paralelan). Serijski je rad pokazan na sl. 19a. Dohvat sljedeće mikronaredbe počinje pošto je prethodna potpuno izvršena. Pri paralelnom radu (sl. 19b) dohvati se preklapa s izvođenjem. Pri miješanom radu (sl. 19c) neke mikronaredbe imaju preklapljen dohvat, a druge ga nemaju. Odluka ovisi o funkciji mikronaredbe.



Sl. 19. Dohvat i izvođenje mikronaredaba, a serijsko, b paralelno, c miješano serijsko-paralelno izvođenje

Proučavanje organizacije prijenosa i dekodiranja mikronaredbe pokazuje razlike u izvedbama strojeva. U jednofaznom sustavu cijela se mikronaredba dekodira odjednom. U polifaznom sustavu pojedini se dijelovi mikronaredbe dekodiraju u pojedinoj fazi.

Mikronaredbe koje čine mikroprogram smještene su u upravljačkoj memoriji. Budući da postoje različiti razredi mikronaredaba, varirat će i organizacija upravljačke memorije. Ako se umjesto jednostavnoga višerazinskog dekodiranja razdvoje mikronaredbe na više razina ili na više stranica, postat će organizacija upravljačke memorije također višerazinska. S obzirom na razmjještaj, radna i upravljačka memorija mogu biti razdvojene, u istom naslovnom prostoru ili *hierarhijski organizirane*. Te su tri mogućnosti pokazane na sl. 20. U organizaciji prvog tipa upravljačka je memorija obično mnogo brža nego radna. Toj grupi pripadaju računala IBM S/360, modeli 30 do 65 i mnogi drugi. U drugoj vrsti organizacije obje memorije imaju ista svojstva i dijele zajednički naslovni prostor. Raspodjela naslovnog prostora može se postići dodatnim sklopovima ili pomoću mikroprograma. To je moguće izvesti ako se raspolaže velikom, brzom i jeftinom memorijom. Takva je organizacija npr. u računalima IBM S/370, model 145. Treća je vrsta organizacije samo modifikacija druge vrste. Upravljačka je memorija toliko brza da se ne isplati vezati je s radnom memorijom u isti naslovni prostor, nego se ona povezuje u naslovni prostor *razdvojne memorije* (engl. *cache memory, buffer storage*). Prijenos mikroprograma iz glavne u razdvojnu memoriju obavlja posebna sklopovski izvedena jedinica. Ako su tražene mikronaredbe već pohranjene u razdvojnoj memoriji, tada je to, s gledišta mikronaredbe, sustav druge vrste. Takva je organizacija primjenjena npr. u računalima IBM S/360, model 85, i S/370, model 165.



Sl. 20. Jedna od klasifikacija organizacije upravljačke memorije. a razdvojene memorije, b druga vrsta: zajednički naslovni prostor, c treća vrsta: hierarhijska organizacija

Iako su mnoga računala mikroprogramirana, korisnik obično to ne zna i ne dopušta mu se vlastito mikroprogramiranje. Ondje gdje je i moguće vlastito mikroprogramiranje, rijetko postoje potrebna dobra programska pomagala. Mikroprogramirana računala kojima je mikroprogram smješten u radnoj memoriji, organizacija druge ili treće vrste, pa imaju mogućnost promjene mikroprograma tijekom izvođenja ili pod nadzorom posebnog programa, nazivaju se *dinamički mikroprogramirajuća računala*. Ako proizvođač opskrbљuje korisnika programskim pomagalima za izradbu mikroprograma, računalo je *korisnički mikroprogramirajuće*. Vrlo je korisno ako je računalo *korisničko-dinamički mikroprogramirajuće*.

Jezici za izradu mikroprograma različitog su stupnja zamršenosti. Svrstavaju se u nekoliko razina: a) visoka razina: procedurno orijentirani strojno neovisni; procedurno orijentirani strojno ovisni; makroregistarски; registarски; b) niska razina: mnemonički mikrojezik; mikrojezik.

Mnemonički mikrojezik sliči općem mnemoničkom jeziku. Opći je format za vertikalnu mikronaredbu:

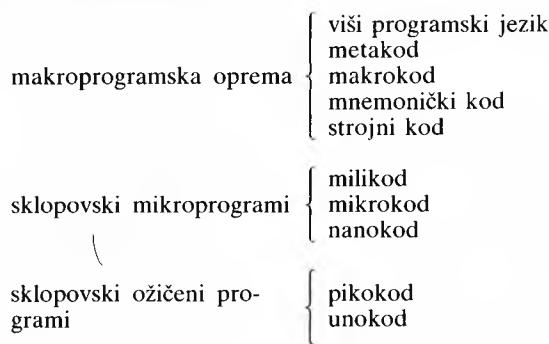
oznaka	operacija	operandi	napomena
--------	-----------	----------	----------

Za horizontalnu je mikronaredbu to obrazac s određenim funkcijanskim poljima. Registarски jezik omogućuje pisanje mikronaredbe u obliku koji sliči jeziku visoke razine. Zapravo, to je jezik vrlo sličan mnemoničkom. Korisnik mora biti dobro upoznat sa sklopovima računala i njihovom povezanošću. Registarски je jezik povoljan za horizontalno mikroprogramiranje. Makroregistarски jezik ima dodatne mnemonike koji olakšavaju programiranje. U primjeni viših procedurno orijentiranih jezika (FORTRAN, PL/1, PASCAL

itd.) potrebni su vrlo zamršeni kompilatori i općenito se ne postiže suviše uspješan kod.

Izradi programskih pomagala za mikroprogramiranje povećuje se velika pažnja zbog komplikiranosti programiranja. Za izradu mikroprograma treba više vremena nego za izradu makroprograma.

Program napisan u jednom od jezika zahtijeva jezičnu obradu kojom se postiže odgovarajući kod. Dobiveni se kodovi mogu svrstati u razine:

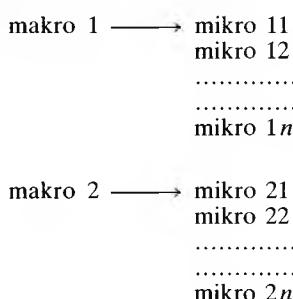


upravljački signali.

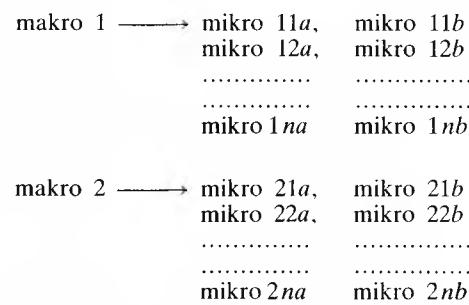
U prvoj su grupi kodovi u kojima se izrađuju razni programi za pojedini zadatak. Metakod je produkt metajezika (jezik kojim se opisuje neki drugi viši jezik) i obično je to unutarnji kod kompjulatora višeg programskog jezika. Makrokod uvodi mnemonike više razine i strukturno nije različit od mnemoničkog koda. Strojni kod je stvarna naredba. Bez obzira na koji je način pojedini kod pripremljen, oni se upotrebljavaju za izradu programske opreme. Sljedeća razina su sklopovski mikroprogrami. Osnovni oblik je *mikrokod*. Niža razina u dvorazinskom sustavu je *nanokod*. Međutim, postoji težnja da se pomoću naredaba na razini sklopovskih programa emulira neki kompleksniji sklop i da upravljanje tim sklopom bude u jednom problemski orientiranom kodu. Potrebno je interpretirati neke posebne simboličke naredbe, npr. one koje su prilagođene upravljanju nekim motorom. Takav će se kod ugraditi u višu razinu upravljačke memorije i on će izravno nasloviti mikroprogramsku memoriju. Takav se kod naziva *milikod*. Sklopovi u upravljačkoj i aritmetičko-logičkoj jedinici ostvaruju neki ožičeni program. Izlaz iz sklopovskih programa naziva se *pikokod*. Ako se zamisli da signali pikokoda upravljaju nekim izbornim sklopkama ili multipleksorima, dolazi se do još niže razine signala nazvane *unokod*. Unokod upravlja pojedinim osnovnim sklopovima, npr. bistabilima, logičkim sklopkama itd.

Mikroprogrami se smještaju u pripadnu memoriju na nekoliko načina:

1) Najjednostavnija je i najčešća struktura upravljačke memorije uobičajeno memorjsko polje u kojemu je jedna mikronaredba u svakoj riječi upravljačke memorije:



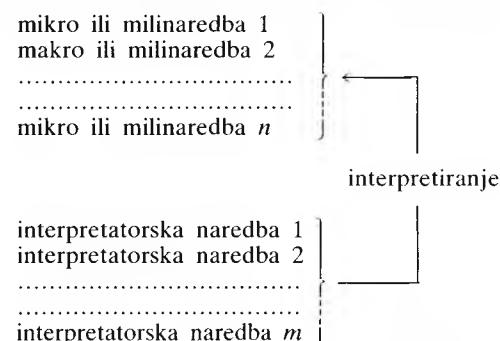
2) Kad se mikronaredbe, npr. dvije, slažu u jednu riječ. Time se smanjuje vrijeme dohvata uzastopnih mikronaredaba, što je posebno dobro ako je upravljačka memorija spora. Posebni sklopovi razvrstavaju naredbe *a* i *b* u mikronaredbeni registar:



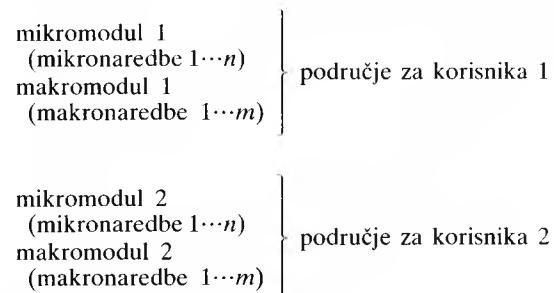
3) Upravljačka se memorija može podijeliti na dva odvojena dijela: upravljački dio, koji sadrži adrese, i izvršni dio, u kojemu su izvršne naredbe. Mikronaredbe su u upravljačkom dijelu vertikalne i mogu, osim adrese, sadržavati još poneku uputu, npr. skok u potprogram. Mikronaredbe su u izvršnom dijelu horizontalne. Već prema sadržaju upravljačkog dijela, to mogu biti nanonaredbe:



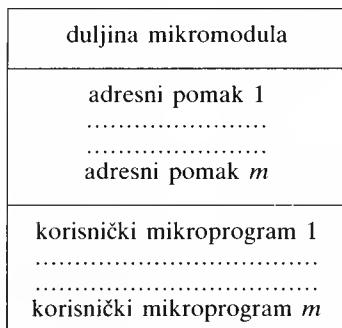
4) Dvorazinska organizacija slična je prethodnoj, samo se operacije drukčije odvijaju. U vertikalno organiziranoj memoriji više razine nalazi se mikroprogram ili miliprogram. Horizontalno organizirana memorija niže razine jest interpretator koji izvodi mikronaredbe ili milinaredbe. Dakle, memorija više razine ne sadrži adrese, nego operacije, a izvršna memorija pretražuje i interpretira njezin sadržaj:



*Dinamički mikroprogramirani sustavi* trebaju posebnu organizaciju memorije, jer mikroprogram mora biti povezan s makroprogramom u zajedničkoj radnoj memoriji. Organizacija mora biti takva da ne uzrokuje povećan prazni hod operacijskog sustava za korisnike koji, npr., ne žele iskoristiti mogućnost dopunskega mikroprogramiranja. Posebno se složeni zadaci pojavljuju ako se upotrebljava sustav koji dopušta višeprogramske radove. Osnovni je raspored u memoriji:

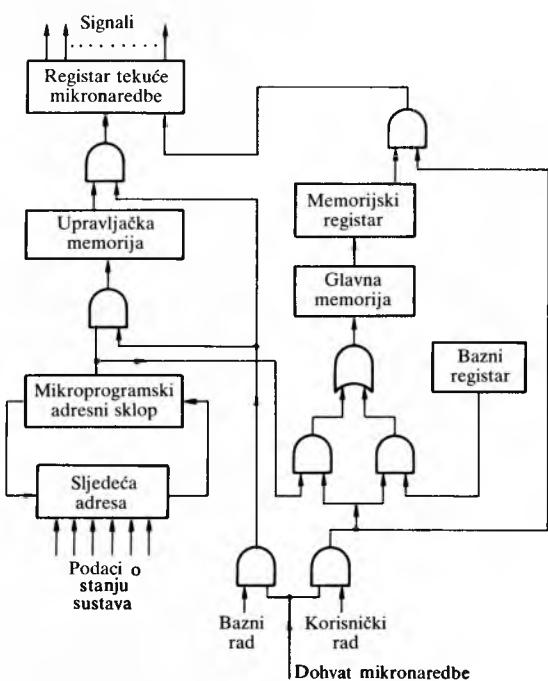


Svakom korisniku dodjeljuje se jedno od područja memorije za smještaj *programskog modula*. Mikronaredbe su grupirane u dijelu koji se naziva *mikromodulom*, a strojne naredbe u *makromodulu*. Svaki korisnik ne mora zahtijevati potpuni mikromodul, pa se mora izmisliti način koji će biti prilagodljiv, a neće nepotrebno trošiti memorijski prostor. Takva je organizacija mikromodula npr.:



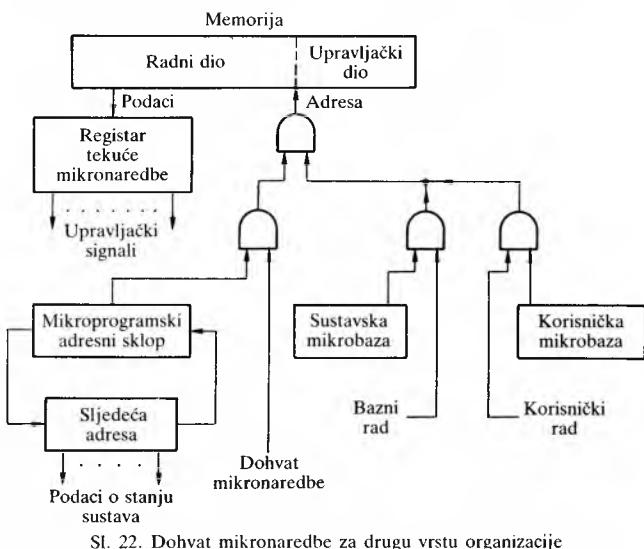
Čelna riječ modula sadrži podatak o ukupnom broju riječi što ih u memoriji zauzima mikromodul. Operacijski sustav provjerava tu prvu riječ i na temelju njena sadržaja zaključuje da li je dotični korisnik uključio mikromodul u svoj program ili nije. Prema tome, korisnik koji ne želi poseban mikrokod ne trpi zbog toga nikakve posljedice. Nakon toga slijedi prostor za  $m$ -adresni vektor. Kad se u mikroprogramu nađe na nestandardni operacijski kod, onda se taj kod upotrijebi kao kazaljka za adresni vektor. Ako je sadržaj na toj adresi jednak nuli, to je za tog korisnika ilegalni makrokod. Operacijski sustav neće dopustiti dalji rad. Vidi se da korisnik može upotrebljavati *bazni stroj* (bez dopunskog mikroprogramskog modula) ili *modificirani stroj* (s uključenim dopunskim mikroprogramskim modulom).

Dinamičko mikroprogramiranje može se provesti i u slučaju organizacije memorije prve vrste prema sl. 20a. Povezivanje mikroprogramskog i makroprogramskog modula bit će onda drugačije. Međutim, koncepcija baznog i modificiranog stroja ostaje. Iz tog razloga mora se modificirati sklop za dohvata mikronaredbe. Na sl. 21 pokazan je dio za dohvata mikronaredbe za organizaciju prve vrste. Postoje dva načina dohvata mikronaredbe: bazni i korisnički. Bazni dohvat osposobljava spojne putove do upravljačke memorije i



Sl. 21. Dohvat mikronaredbe za prvu vrstu organizacije

adresiranje je relativno, tj. s obzirom na početnu adresu u toj memoriji. Za vrijeme korisničkog dohvata (tj. dohvata korisničkih mikronaredaba) mikronaredba se dohvaca iz glavne memorije i pohranjuje u registar tekuće mikronaredbe mimo upravljačke memorije. Adresa mikronaredbe u glavnoj memoriji stvara se tako da se sadržaj mikroprogramskog adresnog sklopa dodaje sadržaju baznog registra. Bazni registar sadrži početnu adresu korisničkog programskega modula. Dohvat mikronaredaba za organizaciju druge vrste obavlja se prema sl. 22. Glavna memorija razdijeljena je na radni i upravljački dio. Registr je tekuće naredbe, na pogodan način, priključen na memorijsku sabirnicu. Adresni sklop izraduje relativnu adresu mikronaredbe. Ta se adresa dodaje adresi pohranjenoj u registru sustavske mikrobaze za bazni rad, odnosno u registru korisničke mikrobaze za korisnički rad (upotreba korisničkog mikroprograma). Prijelaz iz baznog u korisnički rad i obratno zbiva se pod nadzorom posebnih mikronaredaba.



Sl. 22. Dohvat mikronaredbe za drugu vrstu organizacije

Registri za različite namjene tvore registarsko polje ili *lokalnu memoriju*. Dio registra služi za određivanje granica odsječaka memorije i korisnik ne smije moći mijenjati njihov sadržaj. Zato je povoljno razdijeliti lokalnu memoriju u područja. Aktiviranje pojedinog područja obavlja poseban *registar područja*. Područni se registar može adresirati samo u baznom radu. Isto tako sadržaj registara bitnih za upotrebu sustava može se mijenjati samo u baznom radu. Bazni je rad većinom dopušten samo nekim ovlaštenim korisnicima.

## MIKROPROCESORSKA RAČUNALA

**Računala četvrte arhitektonске generacije.** Skloovi niskog stupnja integracije preobrazili su računala. Oni su omogućili stvaranje moćnih računala treće arhitektonске generacije. Sama tehnika integriranja brzo je napredovala i već je 1971. najavljen *mikroprocesor*, najveće dostignuće silicijске integrirane tehnike. U mikroprocesoru središnji je sklop računala izrađen na jednoj silicijskoj pločici. Iako s gledišta arhitekture mikroprocesor nije unio ništa nova, on je potpuno promjenio način upotrebe računala zbog svojih malih dimenzija, niske toplinske disipacije i niske cijene. Računala izgrađena s mikroprocesorom i drugim pripadnim elementima, posebno treba istaknuti ROM i EPROM (tabl. 2), postala su mala, jeftina i prilagodljiva posebnim zahtjevima. Pojedini se uređaji i strojevi opremanju *ugradenim* (engl. *embedded*) računalima potpuno prilagođenim potrebama stroja. Ta računala obavljaju svoje poslove na temelju unaprijed ugrađenog programa, pa korisnik ne treba biti upućen u programiranje. Pokretanje je, većinom, automatsko i nevidljivo za onog koji radi sa strojem. Mikroprocesori su danas tako usavršeni

da se njihova svojstva uspoređuju sa svojstvima velikih računala. Ugrađena računala, kao i sva druga, sastoje se od svih jedinica koje je von Neumann opisao. Dopunsko je dostignuće silicijске proizvodne tehnike stvaranje cijelokupnog računala s mikroprocesorom, radnom i programskom memorijom, ulaznim i izlaznim pristupima, analogno-digitalnim pretvaračem, brojilima, komunikacijskim sklopom itd. na jednoj pločici. To su prava računala četvrte arhitektonске generacije, tzv. *mikroraćunala*.

**Razvoj mikroprocesorskih računala i mikroraćunala.** Napredak elektroničke tehnologije očituje se u povećanju broja osnovnih elektroničkih elemenata, odnosno sklopova koji se mogu smjestiti na određenu površinu silicijске pločice (engl. *chip*; v. *Poluvodiči*, TE 10, str. 643). Pri tome se ne smije povećati toplina koju ta pločica predaje okolini. To znači da se površina pojedinog elementa smanjuje, a proporcionalno njoj i snaga. Kako je napon napajanja uglavnom jednak za mnogo vrsta integriranih sklopova, smanjuje se i struja koja protječe kroz pojedini tranzistor. Većina mikroprocesorskih pločica ima stranice od ~8 mm i na njima se nalazi čak 50000 do 100000 tranzistora, odnosno površina je jednog tranzistora u prosjeku  $1000 \times 2000 \mu\text{m}^2$ . Struja koja u prosjeku teče kroz pojedini tranzistor (površine  $\sim 30 \mu\text{m} \times 60 \mu\text{m}$ ) u prosjeku je  $\sim 5 \mu\text{A}$ . Veći mikroprocesori ili njihovi dijelovi sastavljaju se od više pločica, pa se na skupini od tri pločice nalazi oko 225000 tranzistora tipa MOSFET (v. *Impulsna i digitalna tehnika*, TE 6, str. 435). Najnovije izvedbe imaju povećanu pločicu na kojoj smještaju oko 275000 tranzistora izrađenih postupkom CHMOS III (engl. *Complementary High-speed Metal Oxide Semiconductor*) uz upotrebu 1,5-mikrometarske geometrije i dvostrukje metalizacije. Mikroprocesor je smješten u povećanu kućištu sa 100 nožica.

Razvoj je mikroprocesora veoma skup. Budući da se mikroprocesori proizvode i prodaju u velikim serijama, postiže se niska cijena. Procjenjuje se da je svjetska proizvodnja veća od 100 milijuna mikroprocesora.

U pogledu arhitekture centralne procesne jedinice većina mikroprocesora ne predstavlja neki napredak. Mnogi se od njih mogu svrstati u jednostavne sustave druge generacije (ili čak djelomično prve), iako se uz njih grade sustavi koji su napredni sustavi treće generacije. Prilikom razmatranja mikroprocesora treba razdvojiti i ocjenu njihove arhitekture i organizacije od tehnoloških dostignuća. Mikroprocesori su zbog svoje niske cijene i jednostavnosti znatno utjecali na primjenu računala u raznim granama ljudske djelatnosti. Mogućnosti su primjene računala zbog toga jako povećane.

Doba mikroprocesora, može se kazati, počinje s oglasom koji je tvrtka INTEL objavila u časopisu Electronic News (15. studenog 1971). Tim je oglasom najavljen četverobitovni mikroprocesorski sklop i prateći sklopovi: sklop ispisne memorije, sklop radne memorije i sklop registara. Ti su sklopovi činili tzv. sustav 4000, poznat pod nazivom MCS-4.

Razvoj mikroprocesora (taj je naziv INTEL uveo 1972) bio je potaknut zahtjevom japanskog proizvođača kalkulatora BUSICOM, koji je 1969. zatražio da INTEL razvije dvanaest sklopova za izradbu kalkulatora koji se može programirati. Svaki je sklop trebao sadržavati 3000 do 5000 tranzistora tipa MOSFET. Suradnik tvrtke INTEL M. E. Hoff analizirao je zahtjev i uočio da je traženi sustav suviše zamršen, posebno u usporedbi s jednostavnim ustrojstvom sustava DEC PDP-8. Tada je, odvojeno od izvornog poticaja, u tvrtki INTEL nastavljen rad pa je razvijen MCS-4.

Nakon uspjeha s četverobitovnim sustavom INTEL je nastavio razvoj i u travnju 1972. pojavio se na tržištu osmobiljni sustav s mikroprocesorom 8008. Ta je godina promijenila smjer razvoja digitalne elektronike. Gradnja se kombinacijskih sklopova i zamršenih digitalnih sustava više nije isplatila. Mnoge probleme rješava primjena mikroprocesorske tehnike. To je zahtijevalo da dosadašnji specijalisti na području sklopova nauče izradivati programe. Razvoj nekog digitalnog sustava uključuje projektiranje sklopovlja i programa. Način rada u elektroničkom laboratoriju potpuno je izmijenjen.

Ubrzo su i drugi proizvođači integriranih sklopova uočili perspektivnost mikroprocesorske proizvodnje, pa već 1974. postoji ~19 raznih mikroprocesora, 1975. ~40, a krajem 1976. ~54 razna mikroprocesora.

INTEL 8008 (kao i 4004) imao je 45 raznih naredaba, a prosječno vrijeme izvođenja naredbe bilo je  $\sim 30 \mu\text{s}$ . Osim toga, bilo je potrebno mnogo dopunskih sklopova da bi sustav bio cijelovit. U travnju 1974. INTEL je napravio osmobiljni mikroprocesor 8080 koji je imao 30 naredaba više od svog prethodnika, a vrijeme izvođenja naredbe  $\sim 2 \mu\text{s}$ . Osim toga, mikroprocesor 8080 može nasloviti memorijski prostor od 64 kilookteta, dok je mikroprocesor 8008 mogao samo 16 kilookteta. Procesor 8080 sastavljen je od 5000 tranzistora. U to doba pojavljuju se glavni konkurenti. Za mikroprocesor MOTOROLA 6800 treba samo jedan izvor za napajanje. Novi proizvođač ZILOG stavlja 1976. godine na tržište novi sustav Z80. Taj mikroprocesor ima 158 naredaba, a među njima su i naredbe procesora 8080. Dakle, Z80 emulira 8080, tj. izvodi njegove naredbe.

U daljem razvoju slijede sustavi Texas Instruments 9980, INTEL 8088 i MOTOROLA 9980. To su mikroprocesori s osmobilnjim podatkovnim sabirnicama (kao što postoje u mikroprocesorima 8080, Z80, 6800 itd.), ali interno se podaci obraduju kao šesnaestobitovni.

Do sada opisani mikroprocesori radili su s memorijom koja nije bila u okviru njihove silicijске pločice. Da bi se stvorilo računalo, treba uzeti sklop mikroprocesora i sklop memorije (uz ostale potrebne sklopove). INTEL je 1976. godine proizveo sklop mikroraćunala 8048. Taj sklop sadržava na jednoj pločici osmobiljni mikroprocesor, ulazne i izlazne pristupe, radnu memoriju i ispisnu programsku memoriju. Nakon toga izrađeno je mikroraćunalo 8748, koje, za razliku od 8048, ima ugrađen EPROM umjesto ROM. Slična mikroraćunala izradili su i drugi proizvođači, npr. ZILOG ima mikroraćunalo Z8671.

Mikroprocesor, radne memorije, ispisne memorije, ulazni i izlazni sklopovi itd. proizvodi su četvrte tehnološke generacije integriranih sklopova (sklopovi velikog i veoma velikog stupnja integracije, LSI i VLSI; v. *Impulsna i digitalna tehnika*, TE 6, str. 451; v. *Poluvodiči*, TE 10, str. 663). Takvi sklopovi služe za izradu cijelokupnih računala i njihovih dijelova. Njihova arhitektura i organizacija pripada drugoj i trećoj generaciji (ili čak prvoj). Izrazit je primjer mikroprocesor IM6100, proizvod tvrtke INTERSIL. Taj mikroprocesor emulira naredbe računala DEC PDP-8/E, a pomoću dijelova iz sustava 6000 ono se može izgraditi jednostavno.

Mikroraćunala koja na jednoj silicijskoj pločici imaju sve bitne dijelova računala mogu se zaista nazvati računalima četvrte generacije, bez obzira na to kojoj arhitektonskoj generaciji stvarno pripadaju.

Razvoj mikroprocesorskih sklopova nastavlja se i dalje. Pojavljuju se šesnaestobitovni mikroprocesori. INTEL je 1978. dao na tržište mikroprocesor 8086 (odnosno sustav iAPX 86), koji može nasloviti memorijski prostor od milion okteta. Pločica mikroprocesora sadrži oko 29000 tranzistora. Tome mikroprocesoru konkuriraju ZILOG 8000, MOTOROLA 68000 i drugi. MOTOROLA 68000 je već interno tridesetdvobitovni mikroprocesor. INTEL je 1981. godine proizveo potpuni tridesetdvobitovni mikroprocesorski sustav iAPX 432. Naslovni memorijski prostor ima 16 milijuna znakova, brzina je rada oko dva milijuna naredaba u sekundi. Sama centralna procesna jedinica izrađena je na dvije pločice i ima poseban ulazno-izlazni procesor. Glavni procesor sadržava oko 200000 tranzistora. Sasvim novi tridesetdvobitovni mikroprocesori naslovaju memorijski prostor od nekoliko gigaokteta. Izvođenje traje  $\sim 100$  nanosekunda, pa se postiže propusnost između 5 i 33 milijuna naredaba u sekundi (engl. *Million Instructions Per Second*, MIPS). Razvoj je mikroprocesora prikazan u tabl. 1.

Mikroprocesorske se jedinice danas razvrstavaju u tri mikroprocesorske generacije. INTEL 4004, 4040, 8008 pripadaju prvoj generaciji. Ostali osmobiljni, dvanaestobitovni i početni šesnaestobitovni mikroprocesori svrstavaju se u drugu

generaciju. Novi šesnaestobitovni i tridesetdvobitovni mikroprocesori pripadaju trećoj generaciji. Arhitekture prve i druge mikroprocesorske generacije te mikroračunala pripadaju drugoj arhitektonskoj generaciji računala. Arhitektura treće generacije mikroprocesora uključuje niz svojstava karakterističnih za računala treće arhitektonske generacije, pa su to prava računala treće generacije. Neka od njih imaju posebno naprednu arhitekturu.

Tablica 1  
PRIKAZ RAZVOJA MIKROPROCESORA

Proizvođač	Tip	Godina	Opis
INTEL	4004	1971.	Prvi mikroprocesor
INTEL	8008	1972.	Prvi osmobiljni mikroprocesor
INTEL	8080	1974.	Prvi N-kanalni mikroprocesor
MOTOROLA	6800	1974.	Prvi jednonaponski mikroprocesor
Texas Instruments	TMS 1000	1974.	Najprodavaniji četverobiljni mikroupravljački sklop
National Semiconductor	PACE	1974.	Prvi šesnaestobitovni mikroprocesor na jednoj pločici
RCA	1802	1974.	Prvi mikroprocesor CMOS
INTEL	8048	1976.	Prvo osmobiljni mikroračunalo
INTEL	8088	1979.	Prvi osmobiljni mikroprocesor s unutrašnjom šesnaestobitovnom organizacijom
INTEL	2920	1979.	Prvi mikroprocesor za obradu analognih signala
INTEL	iAPX 432	1981.	Prvi tridesetdvobitovni mikroprocesor

**Mikroračunala.** Mikroračunalo je integrirani sklop koji na silicijskoj pločici uobičajenih dimenzija sadrži mikroprocesor, programsku i podatkovnu memoriju, ulazno-izlazne sklopove, vremenske i upravljačke sklopove. Neka su mikroračunala opremljena analogno-digitalnim pretvaračima i analognim multiplexorima. Mikroračunala u pogledu unutrašnje arhitekture nisu, barem za sada, neko novo dostignuće, ali su tehnološki gledano veliko dostignuće. Njihova primjena dolazi osobito do izražaja u upravljanju raznim procesima.

U većini je mikroračunala programska memorija stalna ispisna memorija ROM (tabl. 2). Naslovni je prostor te memorije 2 do 4 kilookteta. Odmah se može zaključiti da će biti potrebno naručiti mnogo mikroračunala ako se želi postići niska cijena. Budući da tijekom razvoja neke primjene treba ispravljati i nadopunjavati programe, moguće je dobiti posebne izvedbe mikroračunala s izvedenim memorijskim vodovima ili izvedbe s programirljivim ispisnim memorijama EPROM (tabl. 2). Mikroračunala s takvom memorijom vrlo su pogodna za nove i manje serije.

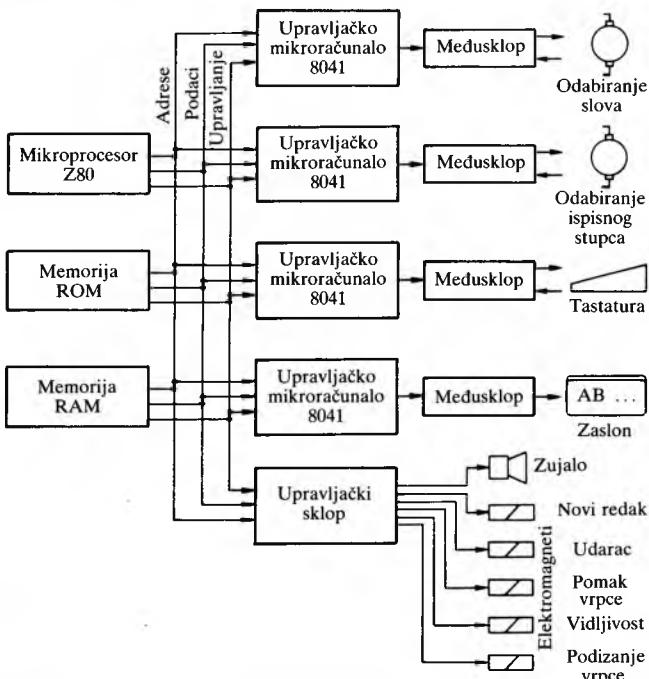
Tablica 2  
KRATICE MEMORIJA U RAČUNALIMA

Kratika	Značenje
RAM	Random Access Memory, memorija s izravnim pristupom
DRAM	Dynamic RAM, dinamički RAM
SRAM	Static RAM, staticki RAM
ROM	Read Only Memory, ispisna (permanentna) memorija (memorija samo za čitanje)
EAROM	Electrically Alterable ROM, električki promjenljiv ROM, riječ po riječ ili bit po bit
PROM	Programmable ROM, programirljivi ROM
EPROM	Erasable PROM, izbrisivi PROM
EEPROM	Electrically EPROM, električki izbrisiv PROM: a) riječ po riječ ili bit po bit, tada je isto što i EAROM b) čitav sadržaj odjednom

Mikroračunala su posebno pogodna za primjenu u napravama koje moraju uvijek obavljati iste poslove, npr. mjerne ili upravljačke naprave. Radni je program u takvom *ugradenom računalu* (engl. *embedded computer*) stalan i rukovalac naprave ne treba biti upućen u njegovu upotrebu.

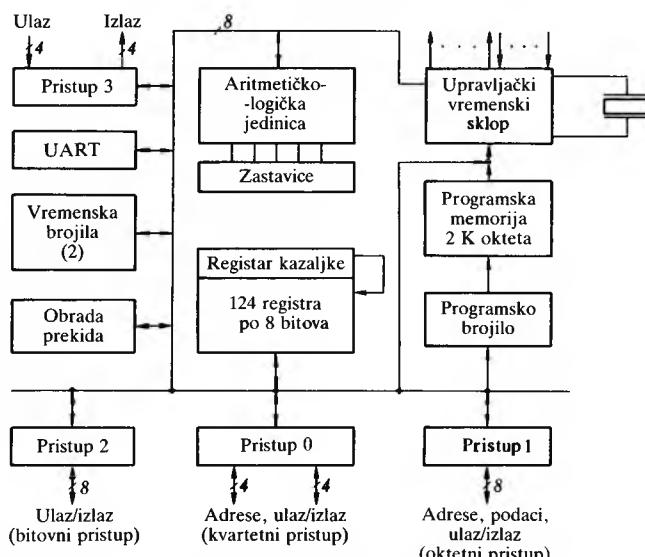
Poseban su oblik *upravljačka* ili *prozivna mikroračunala* (engl. *microcontrollers*). Ta računala miruju sve dok ih glavno računalo ne prozove. Tada ona obave predviđeni posao i vrate se u stanje čekanja novog naloga. Njihov rad sliči radu računala koje je u stanju čekanja (to je većinom naredba WAIT). Primjer zajedničkog rada mikroprocesora i mikrou-

pravljača prikazuje sl. 23. To je pisaci stroj u kojemu glavni mikroprocesor (Z80) upravlja cijelokupnim radom. Mikroupravljači (8041) nadziru rad otisne ploče, mehanizma za udarac, pomake, rad tastature i zaslona za prikaz unijetih znakova. Radni program broji znakove u retku, brine se o desnom i lijevom rubu, pamti broj redaka i naređuje izmjenu papira itd.



Sl. 23. Blok-sHEMA elektroničkog pisaćeg stroja

**Sustav ZILOG Z8.** U toj skupini mikroračunala postoji nekoliko izvedaba. Sve imaju istu osnovnu arhitekturu, ali imaju različite memoriske izvedbe. Osnovna je arhitektura pokazana na sl. 24. Na pločici je ugrađena ispisna memorija kapaciteta 2 kilookteta (jedna izvedba ima ispisnu memoriju kapaciteta 4 kilookteta). Memorije su smještene u kućište s nožištem s više izvoda, a na stražnjoj strani kućišta može biti i podnožje za izbrisivu programirljivu ispisnu memoriju (tzv. *piggy-back* izvedba). Osim ispisne programske memorije, mikroračunalo ima ugrađen osmobiljni mikroprocesor s pripadnim vremensko-upravljačkim sklopovima, regalarsko polje od 124 registra koje može služiti i kao radna memorija, te četiri ulazno-izlazna pristupa. Svakom pristupu pripadaju



Sl. 24. Arhitektura mikroračunala ZILOG Z8

registri u registarskom polju. Pomoću tih registara programira se funkcija i način rada pojedinog pristupa. Ako se želi upotrijebiti vanjska radna i *podatkovna memorija*, adrese se stvaraju na pristupima 0 i 1. Pristup 1 je oktetni i služi kao donji oktet adrese ili za prijenos oktetno oblikovanih podataka. Pristup 0 može se programirati da služi kao osam gornjih bitova adrese ili kao četiri gornja bita adrese i četiri bita za prijenos nekih podataka. Pristup 2 je oktetno oblikovan i služi za prijenos podataka. Pristup 3 je posebno opremljen i razrađen. Može se programirati za unos i za izlaz četiriju bitova. Funkcije njegovih izvoda programski se određuju pomoću sadržaja pripadnih programskih registara. Jedna je od funkcija rad kao serijski ulazno-izlazni sklop za ugrađeni univerzalni asinkroni primopredajnik (*Universal Asynchronous Receive Transmit, UART*) i rad s vremenskim brojilima za generiranje potrebnog ritma. Nadalje, jedan se od izvoda dade programirati za davanje upravljačkog signala za uključivanje dodatne *podatkovne memorije*. Izvodi tog pristupa dadu se programirati za primanje i analizu prednosti zahtjeva za prekid. Preko izvoda pristupa 3 ulazi se u vremenska brojila, pa se može dovesti vanjski ritam i podijeliti ga predviđenim brojem i dovesti na izlaz, ili se unutarnji ritam podijeljen traženim brojem pojavljuje na tom izlazu. Osim toga, svim se pristupima može programski odrediti hoće li im ili neće biti priključen unutarnji povlačni otpor (engl. *pull-up resistor*).

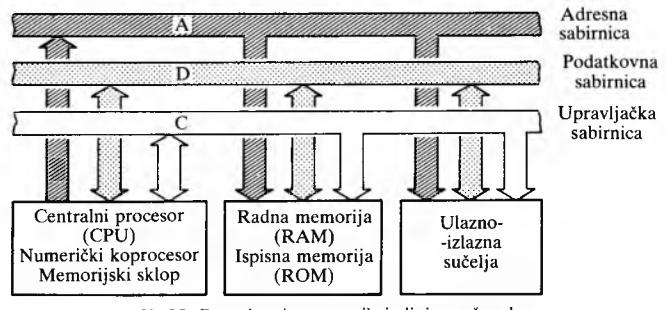
Jedna od izvedaba mikroračunala Z8 ima u ugrađenoj ispisnoj memoriji vrlo sposoban mali monitorski sustav i interpretator za vrlo pojednostavljen dijalekt jezika BASIC. Taj oblik jezika BASIC omogućuje vrlo sažeto pisanje programa, manipulacije s bitovima i registrima, te jednostavno miješanje s odsjećima programa pisanim u strojnem jeziku. Ugrađeni univerzalni asinkroni primopredajnik (UART) daje jednostavnu mogućnost priključka vanjskog terminala, jer su potrebni programi za rad s terminalom već uključeni u monitor.

**Stolna i osobna računala.** Računala druge arhitekturne generacije bila su većinom oprema računskih centara. Mogućnosti koje su pružala računala za razne upravljačke i mjerne primjene bile su velike, ali se glomazna oprema računskih centara teško mogla prilagoditi tim potrebama. Potreba da računalo bude u neposrednom kontaktu s korisnikom bila je rano uočena i oko 1963. na Sveučilištu MIT razvijeno je malo prilagodljivo računalo LINC (*Laboratory Instrument Computer*). Taj je rad utjecao na istodoban razvoj tzv. miniračunala PDP-5 (*Programmed Data Processor*), što ga je na tržište dala tvrtka Digital Equipment Corporation (DEC). Već 1965. ta tvrtka daje na tržište računalo PDP-8 koje je prošlo niz preobražaja i spojilo tehnike druge i treće generacije.

Mikroračunala, posebno PDP-8, pokazala su prednosti za neke posebne primjene. Često su se u industrijsku, znanstvenu, zdravstvenu i sličnu opremu ugrađivala takva tzv. namjenska (decidirana) računala. Mikroprocesori, a posebno razne poluvodičke memorije velika kapaciteta, omogućili su stvaranje vrlo sposobnih i produktivnih računala malih dimenzija i niske cijene. Takva su računala nazvana *stolna računala* (engl. *desk computer, top computer*). Kako su takva računala vezana za pojedinu osobu, a njihova je niska cijena omogućila da se nabave i upotrebljavaju u kućnim uvjetima, za njih se uvriježio i naziv *osobna računala* (engl. *personal computer*). Treba posebno naglasiti da današnja mala osobna računala mogu imati veću proizvodnu moć nego neka velika računala druge generacije. Posebno su tome pridonijeli novi modeli mikroprocesora, tzv. mikroprocesori treće generacije.

Na sl. 25 pokazano je povezivanje osnovnih jedinica računala pomoću *sabirnice* (engl. *bus*, izvedeno iz lat. *omnibus*, svima). Načelno u računalu postoje tri osnovne sabirnice. *Adresna sabirница* raspoređuje podatke prema adresi položaja u memoriji ili napravi koja treba nešto obaviti. *Podatkovna sabirница* služi za prijenos podataka. Ona je dvosmjerna i služi za promet podataka između jedinica računala, odnosno za promet između računala i okoline. *Upravljačka sabirница* raspoređuje upravljačke signale za sve

sklopove računala. Na sabirnice su priključene tri glavne grupe sklopova, većinom izrađenih u tehnici VLSI. Prvu grupu čine centralni procesor, numerički koprocesor (služi za ubrzavanje izvođenja složenih računskih postupaka) i sklopovi za upravljanje pristupom do memorija. U drugoj su grupi memorije. Radna memorija RAM služi, u skladu s von Neumannovim načelima, za pohranu korisničkih programa i dijela sustavnih programa, ulaznih podataka i rezultata rada. Ispisna memorija ROM služi za pohranu osnovnih programa i pokretačkih sustavnih programa koji omogućuju započinjanje radā.



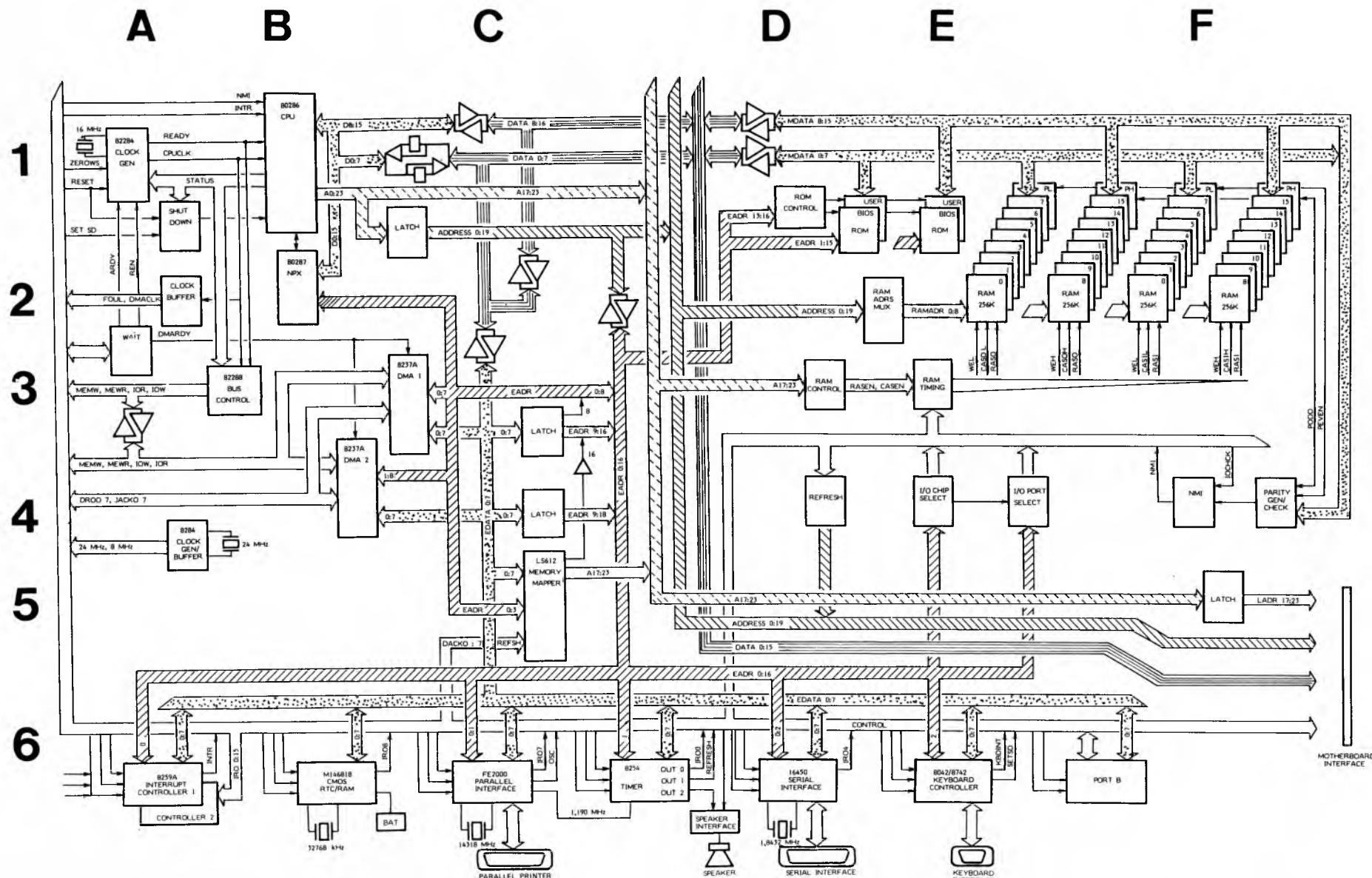
Sl. 25. Povezivanje osnovnih jedinica računala

U trećoj su grupi razni ulazno-izlazni sklopovi, odnosno njihova sučelja. To su sučelja između centralne procesne jedinice i vanjskih naprava računala, kao što su: disketna memorija, winchester-diskovna memorija, ulazne naprave (tastatura i sl.), pisač, zaslon s katodnom cijevi, komunikacijski podsustav itd.

Na sl. 26 pokazana je detaljnija, ali još uvjek pojednostavljena shema stolnog računala RIZ-Olivetti M28 (radi lakšeg tumačenja cijela je shema prividno podijeljena u polja označena kombinacijom slova i brojaka. U polju A1 nalazi se generator procesorskog ritma 82284, te sklop za upravljanje kada se prekida rad (sklopovi s naznakom kodnog broja proizvod u tvrtke INTEL). U polju B1 nalazi se centralni procesor, moći mikroprocesor 80286, koji omogućuje više-programski i višezadatačni rad. Radi ubrzanja aritmetičkih operacija centralnom je procesoru dodan numerički koprocesor 80287 (u polju B2). Numerički koprocesor prati rad centralnog procesora i kad uoči da se pojavila koprocesorska naredba, on počeka da centralni procesor dohvati adresu podatkovnog polja u memoriji. Nakon toga numerički koprocesor javi centralnom procesoru da je razumio zadatak i samostalno ga izvodi. U poljima C1 i D1 nalaze se sabirnički propusti i pojačala. Ispisne memorije ROM nalaze se u polju E1, a dinamičke radne memorije RAM u polju F1. Unutar računala ukupno postoji prostor za 512 tisuća riječi po 16 bitova (= 1 megaoktet). Sklopovi koji uređuju promet između raznih jedinica i memorije nalaze se u poljima C3, D3 i E3. Sklop za obnavljanje sadržaja raznih memorija (one su dinamičke) nalazi se u polju D4. Pregledom polja s memorijama uočava se da je na svakih osam memorijskih sklopova (kapaciteta po 256 kilobitova) dodan još jedan, sklop za paritetnu zaštitu sadržaja memorije. Paritetna zaštita povećava sigurnost rada memorije.

Mikroprocesor 80286 omogućuje računanje adresa do 1 gigaoktet (=  $2^{30}$  oktet). Adrese iz toga virtualnoga memorijskog naslovnog prostora pretvaraju se u adrese fizički ostvarena memorijskog prostora. Taj prostor može iznositi do 16 megaokteta (=  $2^{24}$  oktet). U tom je stroju ostvareno mnogo manje. Pretvorba adresa iz virtualnoga naslovnog prostora u fizički ostvaruje se u procesoru, a sklop u polju C5 pomaže u pronalaženju pojedinih odsječaka po 64 kilookteta.

U šestom retku na sl. 26 nalaze se razna ulazno-izlazna sučelja i drugi sklopovi. U polju A6 nalazi se sklop za primanje zahtjeva za prekid. U polju B6 sklop za vrijeme i datum ima bateriju, da se stanje ne bi izgubilo prilikom isključenja stroja. U polju C6 nalazi se paralelno sučelje za priključak pisača i sklop za davanje ritmičkih vremenskih

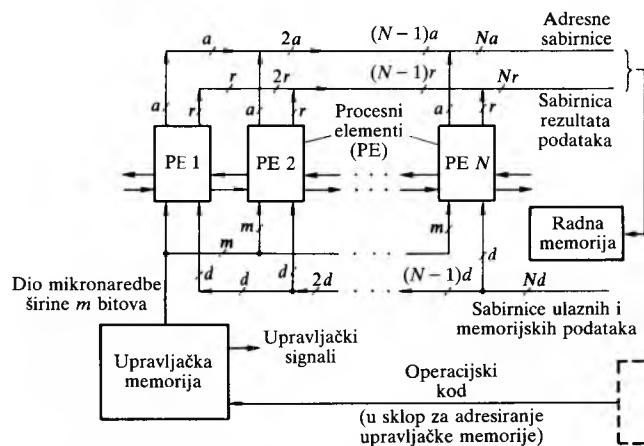


Sl. 26. Pojednostavljena shema stolnog računala RIZ-Olivetti M38. Stupac A: CLOCK GEN, generator ritma; SHUT DOWN, iskapanje; CLOCK BUFFER, pojačalo ritma; WAIT, čekanje; BUS CONTROL, sabirničko upravljanje; INTERRUPT CONTROLLER, upravljanje prekida; stupac B: CPU, centralna procesna jedinica; NPX, numerički procesor; CMOS RTC/RAM, CMOS brojilo realnog vremena/memorija; BAT, baterija; stupac C: LATCH, zadržni spremnik; DMA, izravni memorijski pristup; MEMORY MAPPER, memorijsko preslikavanje; PARALLEL INTERFACE, paralelno sučelje; PARALLEL PRINTER INTERFACE, paralelno sučelje pisača; TIMER, vremensko brojilo; SPEAKER INTERFACE, medusklop za zvučnik; SPEAKER, zvučnik; stupac D: ROM/RAM CONROL, upravljanje za ROM/RAM; REFRESH, obnavljanje; SERIAL INTERFACE, serijsko sučelje; RAM ADRS MUX, adresni RAM; stupac E: BIOS/USER, osnovni ulazno-izlazni sustav; RAM TIMING, ritam za RAM; I/O CHIP (PORT) SELECT, izbor ulazno-izlaznog sklopa (pristupa); KEYBOARD CONTROLLER, upravljanje radom tastature; KEYBOARD INTERFACE, medusklop tastature; stupac F: NMI, neprekidni zahvat za prekid, PARITY GEN/CHECK, generator i provjera pariteta; PORT, pristup; MOTHERBOARD INTERFACE, sabirnički priključci

impulsa. U polju D6 nalazi se zvučnički sklop i serijsko sučelje prema preporuci CCITT V. 24. U polju E6 nalazi se sučelje za priključak tastature, a u polju F6 je dodatni pristup koji sudjeluje u generiranju upravljačkih impulsa za razne sklopove sustava. Na desnom rubu polja F6 naznačen je izravni priključak na sabirnice računala. To omogućuje priključak raznih sklopova i sučelja, npr. za diskovnu memoriju itd.

**Mikroprocesorski režnjevi.** Razvoj integriranih procesora započeo je razvojem četverobitovnog mikroprocesora INTEL 4004. Iako je to bilo ogromno dostignuće, tražio se mikroprocesor s dužom riječi. Ubrzo se pojavljuje osmobitovni mikroprocesor INTEL 8008. Nakon niza poboljšanja danas ima nekoliko tipova takvih mikroprocesora od istog i drugih proizvođača. TEXAS, INTEL, MOTOROLA stavlaju na tržiste mikroprocesore sa šesnaestobitovnom aritmetikom i osmobitovnom podatkovnom sabirnicom za promet s memorijom (TEXAS 9980, INTEL 8088, MOTOROLA 6809). Dalji napredak omogućio je izradbu pravih šesnaestobitovnih mikroprocesora, kao što su National Semiconductor PACE (1974) i SUPER PACE, zatim TEXAS 9900, INTEL 8086, ZILOG 8000, MOTOROLA 68000 itd. Posljednji je iznutra tridesetdvobitovni, a memorijska sabirница mu je šesnaestobitovna. Najnoviji mikroprocesori INTEL 432, INTEL 386, MOTOROLA 68020 itd. potpuno su tridesetdvobitovni procesori. Sav se prikazani razvoj dogodio u jednom desetljeću. Mikroprogrammska tehnika odigrala je važnu ulogu u razvoju spomenutih sustava.

Razvijeni mikroprocesorski sustavi imali su i imaju veliku važnost u razvoju digitalnih računala. Ali, stvaraoci računalnih sustava bili su skučeni u izboru makronaredaba, odnosno duljine riječi u aritmetičko-logičkom sklopu i u veličini memorijskog naslovnog prostora. Zato se već rano pojavila ideja o razrezivanju aritmetičko-logičkog sklopa na *režnjeve* (engl. *bit slice*) po dva, četiri ili osam bitova.



Sl. 27. Povezivanje mikroprocesorskih režnjeva.  $N$  broj režnjeva,  $a$  adresni bitovi,  $r$  bitovi rezultata,  $d$  bitovi podataka,  $m$  bitovi mikrokoda

Režjan je *procesni element* za određeni broj bitova. Sadrži pripadni dio aritmetičko-logičkog sklopa uključujući dijelove stanovitog broja registara. Put podataka određen je mikronaredbom, kojoj su upravljački bitovi paralelno povezani do svih upotrijebljenih procesnih elemenata. Procesni elementi imaju poprečne veze za povezivanje signala prijenosa, odnosno bitova desnog ili lijevog pomaka. Načelo je prospajanja pokazano na sl. 27. Mikroprogram se nalazi u upravljačkoj memoriji, a radni je program pohranjen u radnoj memoriji.

### IZRAZITO PARALELNA RAČUNALA

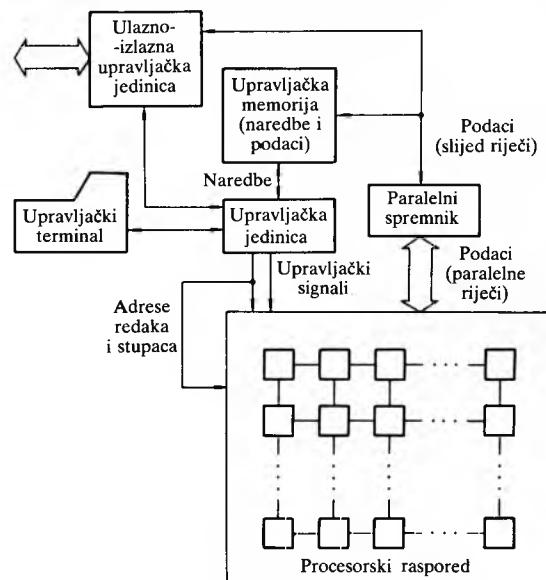
Računala su prolazila i još prolaze kroz razne faze razvoja. Jedan od poticaja razvoja, a možda i glavni, bila je želja da se stvore sposobnija računala. Osim želje za smanjenjem dimenzija i utroška energije, postojala je osnovna želja za povećanjem njihove propusnosti i brzine računanja. Ta se želja ostvarivala na razne načine. Jedan je od njih izgradnja

sve bržih sklopova. Drugi je povećanje propusnosti istodobnom upotrebotom više jednakih sklopova, dakle, izgradnjom paralelnih sustava. Memorijske se jedinice ubrzavaju dohvatom riječi koja je višekratnik osnovne riječi. Međutim, i u paralelnim sustavima i u onima s centralnim procesorom uočena je nezaposlenost nekih dijelova računala dok drugi rade. Odatle i ideja da se posao mogne porazdijeliti i slijedno organizirati. Budući da svaki stupanj obavlja posao u nekom dijelu ukupnog vremena potrebnog za ukupni rad, pojavit će se zamisao o *cjevovodnom sustavu*. Ti sustavi imaju propusnost približno jednaku onoj paralelnih sustava, ako postoji dovoljno gust niz ulaznih podataka. Rezultati se tada postižu u vremenu koje je jednako prolazu kroz najsporiji stupanj. Rezultat je svih tih napora izgradnja brzih i vrlo sposobnih računala, koja se nazivaju *superračunala*. Posebna su vrsta cjevovodnih računala *sistolika računala*, gdje su u pojedinom stupnju upotrijebljeni elementi sposobni provesti potpuno izračunavanje.

Paralelna računala uglavnom pripadaju razredu SIMD. ILLIAC i slični sustavi su *matrična računala* (engl. *array processors*). Oni se sastoje od više procesnih elemenata razvrstanih u stanovit poredak. PEPE i drugi takvi sustavi pripadaju razredu *grupnih računala* (engl. *ensemble processors*) jer je to nakupina procesora okupljena oko jedne upravljačke jedinice. *Asocijativna računala* (engl. *associative computers*) imaju procesore koji pretražuju memoriju prema nekom uzorku. Ona mogu biti matrična i grupna. Neka računala mogu programski mijenjati način unutarnjeg povezivanja. To su *rekonfigurabilna računala* (engl. *reconfigurable computers*).

### Mnogoprocesorski sustavi

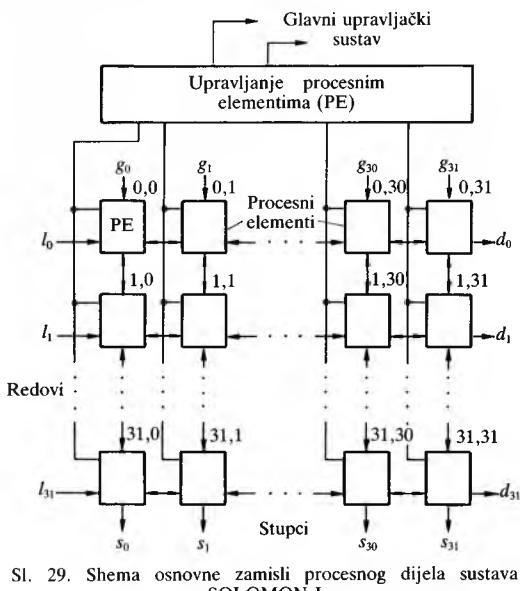
Krajem pedesetih ili početkom šezdesetih godina započeo je intenzivan rad na istraživanju i razvoju paralelnih sustava, a sredinom šezdesetih godina oni su se počeli upotrebljavati. Pristup rješavanju osnovnog zadatka, povećanju propusnosti računala, bio je različit i ovisio je o namjeni računala. Osnovni oblik organizacije računala s grupom paralelno vezanih procesora pokazan je na sl. 28. Procesorska grupa sadrži potreban (ili mogući) broj procesnih elemenata raspoređenih prema zahtjevima koje takvo računalo treba ispuniti. *Upravljačka jedinica* (engl. *control unit*) prima naredbe iz *upravljačke memorije* (engl. *control memory*). Naredbe se dekodiraju i za one koje treba izvesti u *procesorskem polju* (engl. *processor array*) stvara se adresa retka i stupca. Odabrani procesorski elementi izvest će zadanu naredbu ako su njihovi *zastavni registri* (engl. *flag register, mode register*) u stanju koje odgovara onome postavljenom u naredbi.



Sl. 28. Osnovna organizacija izrazito paralelnih računala

Naredbe se šalju i u ulazno-izlaznu upravljačku jedinicu, na upravljački stol. Naredbe koje se izravno izvode u upravljačkoj jedinici spremaju rezultate u upravljačku memoriju. Upravljačka memorija sadrži sustavski dio programa i zajedničke podatke. Rad je toga sustava serijski i on, osim računanja nekih zajedničkih vrijednosti, određuje tijek programa. *Paralelni spremnik* (engl. *parallel buffer*) ostvaruje paralelnu komunikaciju s poljem procesora, a serijsku s ostalim dijelom sustava. Ulazno-izlazna upravljačka jedinica obavlja poslove ulaza i izlaza podataka u sustav. Budući da je to uobičajena serijska jedinica, često će se ostvariti poseban izravni paralelni spojni put s procesorskim poljem.

**Sustav SOLOMON.** Jedno od prvih izrazito paralelnih računala bio je sustav SOLOMON I. (Simultaneous Operation Linked Ordinal Modular Network). Osnovna je zamisao pokazana na sl. 29. Procesni se dio sastoji od mreže *procesnih elemenata*. Ukupno su trebala biti 1024 takva elementa povezana u polje sa  $32 \times 32$  elementa. Svaki je procesni element bio zamišljen kao bitovno-serijski sklop za izvođenje aritmetičkih i logičkih naredaba s parom binarnih znamenaka. Svaki procesni element mogao je sadržavati podatkovnu memoriju od 16384 bita u jedinicama od po 4096 bitova. Duljina se riječi mogla programski odabrati u rasponu od 1...128 bitova. Svaki se procesni element povezuje s četiri susjedna. Rubni se procesni elementi mogu povezati na više načina, već prema naredbi u programu. Ako se povežu stupci  $s_i$  sa  $g_i$  (sl. 29), dobivaju se vertikalni cilindri procesnih elemenata. Isto tako, povezivanje  $d_i$  sa  $l_i$  daje horizontalne cilindre. Takva se povezivanja mogu upotrijebiti istodobno. Nadalje, procesni elementi mogu stvoriti linearni niz odabranih procesnih elemenata ako se  $d_i$  poveže sa  $l_{i+1}$ . Ako se osim toga još poveže  $d_{31}$  sa  $l_0$ , tada se dobiva pun kružni oblik procesnih elemenata.

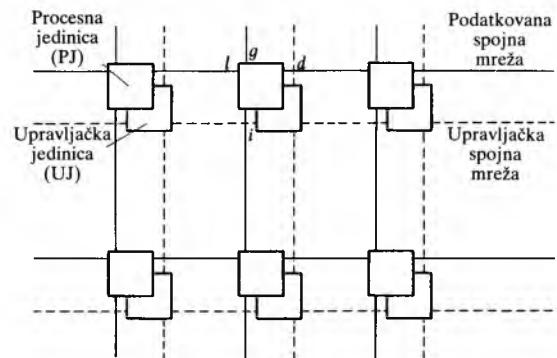


Sl. 29. Shema osnovne zamisli procesnog dijela sustava SOLOMON I

Svaki procesni element sadrži dvobitovni registar koji naznačuje jedno od četiri moguća stanja. Ta se stanja programski unose u registar. Programska naredba naznačuje koje je stanje potrebno, te pripadni procesni elementi izvode zadanu operaciju.

**Hollandovi strojevi.** Oko 1958. predložio je J. H. Holland stroj u kojem je funkcija upravljanja raspoređena u mnoštvu procesnih modula procesnih elemenata. Rani radovi na tom području analizirali su broj potrebnih jediničnih sklopova po modulu. Zaključilo se da je za izradbu takva stroja potrebno oko 1000 sklopova po modulu. Smatralo se da bi bilo moguće izraditi stroj koji bi imao 10000 modula. Pa ipak, praktički prihvatljiva ostvarenja još nema. Novi LSI sklopovi (v. *Impulsna i digitalna tehnika*, TE 6, str. 451) daju velike mogućnosti za ostvarivanje Hollandove zamisli. Osnovna je

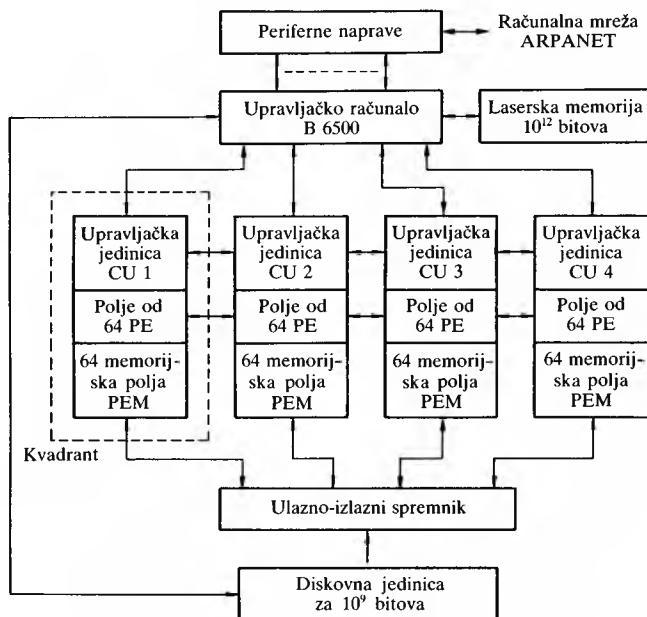
njegova zamisao pokazana na sl. 30. Posve teorijski, to je zamišljeno kao nakupina procesnih elemenata koji se sastoje od dviju funkcionalnih jedinica, a povezani su s dvjema spojnim mrežama. Dvije funkcionalne jedinice u procesnom elementu mogu se nazvati *podatkovnom jedinicom* i *upravljačkom jedinicom*. Svaki procesni element ima i vlastiti naredbeni sklop, pa može izvoditi naredbu koja pripada samo njemu bez obzira na susjede. Isto tako, svaki procesni element ima i svoj aritmetičko-logički sklop, te svoje sklopove za usmjeravanje i povezivanje. Pomoći registri su jednobitovni i dvobitovni, te služe za davanja obavijesti o stanju jedinice. Iz tog opisa proizlazi da Hollandov stroj pripada vrsti MIMD. Dapače, u Hollandovu se stroju u isto vrijeme može naći više međusobno neovisnih aktivnih programskih cjelina (glavnih programa ili potprograma).



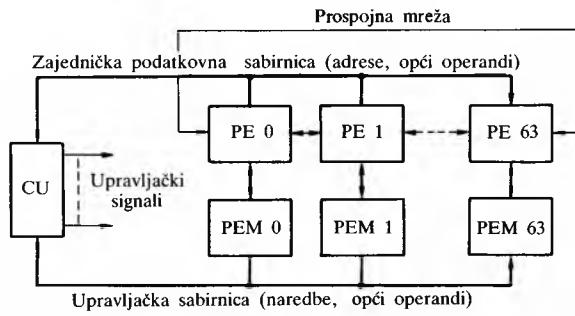
Sl. 30. Shema osnovne zamisli Hollandova stroja

**Sustav ILLIAC.** Računala koja nose ime ILLIAC (engl. *Illinois Automatic Computer*) bila su razrađena na Sveučilištu u Illinoisu (SAD). Sva su računala iz te grupe izrazito paralelna, ali se međusobno razlikuju u tehnici izvedbe i namjeni, pa prema tome i u svojstvima. Sustav ILLIAC I raden je oko 1952., i to je proizvod prve tehnološke generacije. Izrađen je s elektronskim cijevima, a mogao je izvesti oko 11000 aritmetičkih operacija u sekundi. Sustav ILLIAC II razvijen je 1963. i tu su primjenjeni elementi druge tehnološke generacije. To je računalo već moglo obaviti do pola milijuna operacija u sekundi. ILLIAC III je raden oko 1966., i to je, za razliku od prethodnih, specijalizirano računalo. Kao namjena isticala se nenumerička obrada, npr. obrada slike. Sustav ILLIAC IV pravi je izrazito paralelni sustav, temeljen na zamislima razvijenim u okviru projekta SOLOMON. Rad je započeo 1967. Oko 1972. godine izrađen je pokusni sustav s jednim kvadrantom i stavljen u rad u Ames Research Center, koji pripada organizaciji NASA. ILLIAC IV bio je spojen na računalnu mrežu ARPANET i tako bio dostupan širokom krugu korisnika. U rujnu 1981. ILLIAC IV je isključen iz rada i zamijenjen drugim računalom. Osnovna je zamisao sustava ILLIAC IV pokazana na sl. 31. Središnji je dio računala kvadrant koji sadrži upravljačku jedinicu, polje od 64 procesna elementa i polje od 64 procesorske memorije. Upravljačke jedinice i polja procesnih elemenata u kvadrantima mogu se međusobno povezati i načiniti (ako je potrebno) sustav od 128 procesnih elemenata. Središnji je dio računala povezan s periferijskim napravama pomoći upravljačkog računala B6500. Osim nadzornih poslova te prijenosa podataka i naredaba u upravljačke jedinice, u računalu B6500 kompiliraju se programi koji će se izvoditi u sustavu. Osim toga to računalo upravlja radom diskovne jedinice s nepomičnim glavama kapaciteta  $10^9$  bitova i laserskom memorijom kapaciteta  $10^{12}$  bitova.

U laserskoj se memoriji upotrebljavaju snop iz argonskog lasera kojim su se u tankom metalnom sloju bušile rupice mikroskopske veličine. Sloj je bio nanesen na poliesterni nosač i pričvršćen na rotirajući bubanj. Svaki je nosač mogao pohraniti  $2,9 \cdot 10^9$  bitova. Nosači su bili pohranjeni u spremištu s prostorom za 400 nosača. Traženi se nosač mogao dohvatiti za 5 s, a podatak na nosaču za 200 ms. Učestalost je pisanja ili čitanja iznosila 4 megabit-a u sekundi, u dva kanala. Ta se memorija može smatrati prototipom tzv. videobubnja, a time i prototipom videodisksa.



Sl. 31. Shema osnovne zamisli računala ILLIAC IV

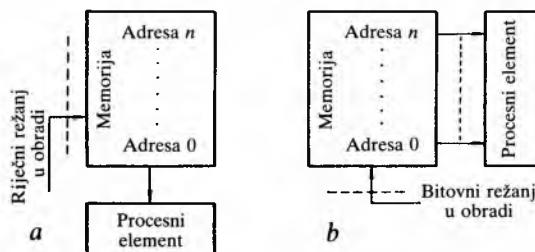


Sl. 32. Shema sastava jednog kvadranta

Sastavni su dijelovi jednog kvadranta priključeni na unutarnji sabirnički sustav (sl. 32). Upravljačka jedinica preko zajedničke podatkovne sabirnice dostavlja u procesne elemente adrese podataka u procesorskoj memoriji i neke opće operande. S druge strane, upravljačka jedinica može preko upravljačke sabirnice razaslati podatke i naredbe u procesorske memorije, koje se međusobno povezuju prospojnom mrežom onako kao u sustavu SOLOMON (sl. 29). Takvo spajanje omogućuje prijenos podataka između upravljačkih elemenata razmaka od  $\pm 1$  ili  $\pm 8$  koraka u 100 ns. Ostali se razmaci sastavljaju od tih  $\pm 1$  i  $\pm 8$  koraka. Jedan od poslova u kojima je bio upotrijebljen ILLIAC IV bila je simulacija ulaska u atmosferu letjelice SHUTTLE.

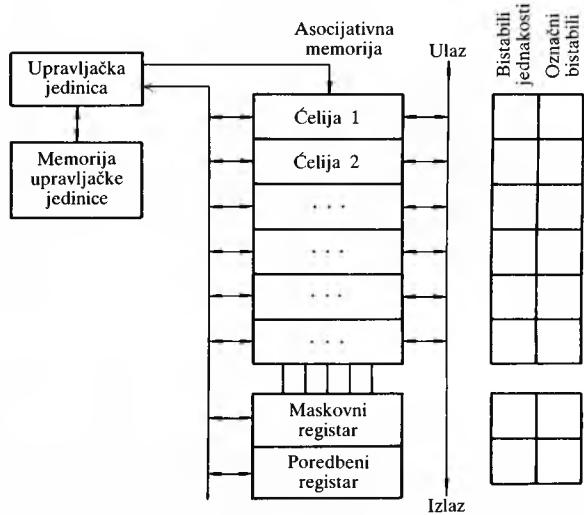
#### Asocijativni procesori

Uobičajeni je način rada računala dohvati riječ po riječ iz memorije i izvođenje naredbe s tom riječu. Kad je organizirana paralelna obrada, više se riječi obraduje istodobno. Obrada riječ po riječ simbolički je pokazana na sl. 33a.



Sl. 33. Primjena paralelnosti, a paralelnost u riječnom režimu, b paralelnost u bitovnom režimu

Međutim, upotrebojem asocijativno organizirane memorije odabire se u memoriji jedan bitovni režanj (sl. 33b), i tome se bitovnom položaju u cijeloj memoriji istodobno ispituje podudarnost s uzorkom u poredbenom registru. Procesor koji radi s asocijativnom memorijom naziva se asocijativnim procesorom. Pojam asocijativnog procesora općenito uključuje asocijativnu memoriju kojoj je dodana mogućnost izvođenja transformacije (aritmetički i logički) više skupina podataka na temelju samo jedne naredbe. Asocijativni su procesori takve vrste strojevi razreda SIMD. Osnovna je organizacija asocijativnog procesora pokazana na sl. 34. Upravljačka jedinica nadzire rad asocijativne memorije. To je obični slijedni procesor s posebnim pristupom do asocijativne memorije i s nekoliko posebnih naredaba za upravljanje.



Sl. 34. Načelna shema primjera organizacije asocijativnog procesora

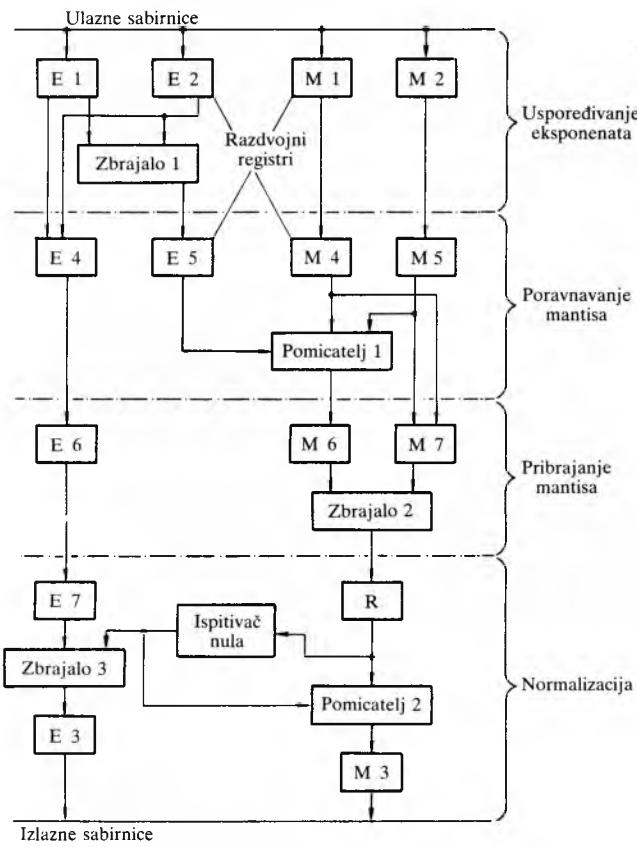
Radni se program upravljačke jedinice nalazi u posebnoj memoriji. Upravljačka jedinica može samostalno izvesti neke ulazno-izlazne operacije. Ako je sadržaj ćelije u bitovima za koje maskovni register dopušta ispitivanje jednak sadržaju poredbenog registra, postavlja se pripadni bistabil jednakosti. Označni bistabili označuju i upravljaju redoslijedom ispitivanja ćelija. U potpuno paralelnoj organizaciji, međutim, sve bi se ćelije ispitale odjednom.

#### Cjevovodni sustavi

Računalo se može shvatiti kao nakupina jedinica od kojih svaka obavlja neki potpuni zadatak. Taj zadatak ima svoje dijelove i u sljedeći se dio ulazi tek kad je prethodni završio posao. Zaposleni dio radi, a nezaposleni dijelovi miruju. Ako se načini sustav u kojem neće biti nezaposlenih dijelova, onda bi takav sustav morao imati propusnost povećanu za onolikouputa od koliko je dijelova sastavljen zadatak. Takav se sustav naziva *cjevovod* (engl. *pipeline*). Prema zadatku za koji je cjevovodni sustav ostvaren, on može biti aritmetički, naredbeni ili s nekom posebnom namjenom.

To se može ilustrirati zbrajalom za brojeve s pomicnim brojnim zarezom. Ono se sastoji od četiri osnovna stupnja. U prvom se uspoređuju eksponenti. U drugom se mantise pomeravaju. U trećem se normalizira rezultat. Odmah je očito da je samo jedan od stupnjeva zaposlen. Neka se to zbrajalo modificira dodavanjem razdvojnih registara kao na sl. 35. Razdvojni registri  $E_1$  do  $E_7$ ,  $M_1$  do  $M_7$  i  $R$  služe da preuzmu podatke od prethodnog stupnja i čuvaju ih sve dok su potrebni. Podaci se dovode do ulaznih registara  $E_1$ ,  $E_2$ ,  $M_1$ ,  $M_2$ . Čim su podaci smješteni u ulazne registre, sabirnica je slobodna za druge poslove. Nakon što se eksponenti usporede, međurezultati se pohranjuju u registre  $E_4$ ,  $E_5$ ,  $M_4$ ,  $M_5$ , koji razdvajaju prvi stupanj od drugoga. U tom trenutku ulazni registri i cijeli prvi stupanj postaju slobodni i mogu preuzeti

novi par podataka. Drugi stupanj poravnava mantise i pohranjuje međurezultate u registre  $E_6$ ,  $M_6$  i  $M_7$  na granici drugog i trećeg stupnja. Tada treći stupanj pribraja mantise i stavlja rezultate u registre  $E_7$  i  $R$ , drugi stupanj preuzima posao od prvoga, a prvi preuzima nove podatke. Nakon što se rezultati pribrajanja normaliziraju i preuzmu od sabirničkih registara  $E_3$  i  $M_3$ , podaci i međurezultati pomiču se prema izlazu za jedan stupanj. U tom se cjevovodnom zbrajalu nalaze u obradi četiri skupine podataka, ali svaka u drugoj fazi. Ukupno je vrijeme prolaza jednak vremenu prolaza kroz jedan stupanj, što je u tom primjeru četvrtina ukupnog vremena prolaza kroz cjevovod. Naravno, takvo se vrijeme prolaza kroz zbrajalo postiže samo ako na sabirnici uvijek postoje spremni podaci za rad.

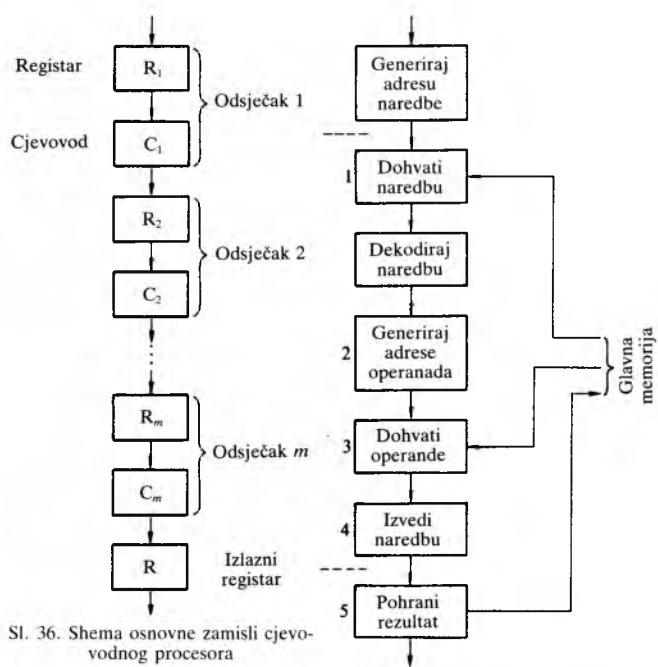


Sl. 35. Shema cjevovodnog zbrajala

Na temelju toga primjera može se nacrtati osnovna struktura nekog cjevovodnog sustava (sl. 36). Cjevovod se sastoji od  $m$  odsječaka. Svaki se odsječak sastoji od procesne jedinice  $C$  i ulaznog registra  $R$ . Cjelokupno je cjevovod zaključen izlaznim registrom. Cjevovod sastavljen od  $m$  odsječaka imat će ukupno  $m + 1$  registar. Budući da propusnost nekog cjevovodnog sustava određuje njegov najsporiji odsječak, to se prilikom razrade moraju odsječci tako odabrati da svi imaju približno jednako vrijeme zadržavanja.

Postoje dvije glavne vrste cjevovodnih sustava: *naredbeni cjevovodi* i *aritmetički cjevovodi*. Primjer je aritmetičkog cjevovoda već razmotreno zbrajalo. Slično tome, u naredbenim cjevovodima naredba takođe prolazi kroz pojedine odsječke cjevovoda, ali u sklopu za dekodiranje naredbe. Čim jedno polje u naredbi bude obradeno, preostali se dio prenosi u sljedeći odsječak, a u oslobođenim se donosi nova naredba. Obično se ne stavljuju više od četiri naredbe u cjevovod zbog vjerojatnosti pojave naredbe skoka. Kad naide naredba skoka, cjevovod se mora isprazniti, te započinje dohvati i rad s novim nizom naredaba. Primjer naredbenog cjevovoda pokazan je na sl. 37.

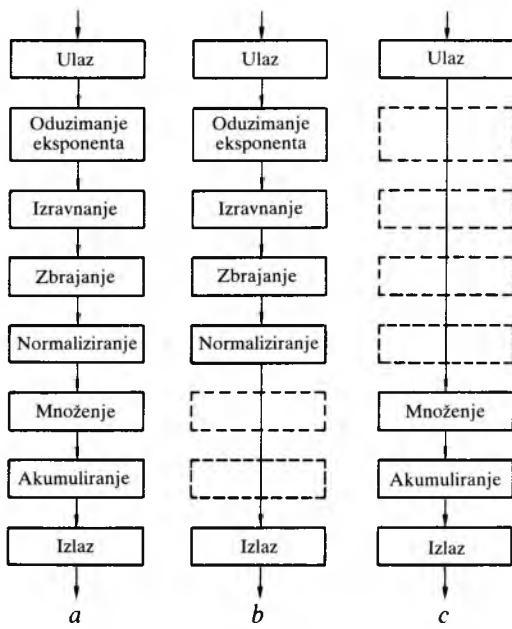
Prilikom izvođenja aritmetičkih operacija može se isti cjevovod upotrijebiti za razne operacije. To znači da će



Sl. 36. Shema osnovne zamisli cjevovodnog procesora

Sl. 37. Sedmoodsječni naredbeni cjevovod (brojke označuju korake izvođenja u naredbenom cjevovodu računala UNIVAC 1100/xx)

cjevovod biti sastavljen od odsječaka za razne poslove, a upotrijebit će se samo oni koji su zaista potrebni. Na sl. 38 načelno je pokazan osmoodsječni aritmetički cjevovod u računalu TI ASC (Texas Instruments Advanced Scientific Computer). Cjevovod je projektiran da daje šezdesetčetverobitni rezultat svakih 60 ns. Prvi je ulazni odsječak primač operanada. U drugom se odsječku uspoređuju eksponenti. Treći odsječak poravnava mantise prema rezultatima iz drugog odsječka. U četvrtom se odsječku pribrajavaju šezdesetčetverobitne mantise. Peti odsječak služi za normalizaciju rezultata. Šesti je odsječak za množenje. Množe se trideset-dvobitovni brojevi u pomicnom ili nepomicnom brojnom zarezu i rezultat se dobiva svakih 60 ns. Upotrijebljen je postupak s pohranom prijenosa. Izlaz su tog odsječka dvije



Sl. 38. Shema osmoodsječnog aritmetičkog cjevovoda računala TI ASC. a - osnova zamisla, b - upotreba odsječaka za pribrajanje u nepomicnom brojnom zarezu, c - upotreba odsječaka za množenje u nepomicnom brojnom zarezu

tridesetdvobitovne riječi: riječ zbroja i riječ prijenosa. Sljedeći ih sedmi odsječak za akumuliranje zbraja i stvara konačni umnožak. Odsječak za akumuliranje sličan je odsječku za zbrajanje. Već je bilo spomenuto da se u njemu stvara rezultat množenja. Tu se nalazi i rezultat skalarnog umnoška dvaju vektora. Osmi je odsječak izlazni. U njemu se obavljaju logičke operacije, ali kroz njega prolaze i svi rezultati. Primjeri na sl. 38b i 38c pokazuju kako se upotrebljavaju samo oni odsječci koji su potrebni za postavljeni zadatak. Na slici nisu naznačene postojeće ili moguće povratne veze. Ako se, npr., žele množiti ili dijeliti šezdesetčetverobitovni brojevi, tada se mora nekoliko puta proći kroz pojedine odsječke. Vodovi povratne veze (zajedno s pripadnim nadzornim sklopom) osiguravaju pravilnu upotrebu odsječaka.

Primjer cjevovoda u računalu TI ASC pokazuje da protok podataka kroz cjevovod može ovisiti o vrsti zadatka. U pojedinim odsjećcima mogu se sukobiti dvije razne skupine podataka. To se može izbjegći ako se predviđi potrebno kašnjenje između započinjanja uzastopnih procesa kojima treba cjevovod.

Minimalno vrijeme koje mora proteći između započinjanja dvaju procesa u cjevovodu naziva se *latencija* (engl. *latency*). Latencija je funkcija položaja procesa koji se nalaze u cjevovodu u bilo kojem trenutku. *Minimalna konstantna latencija* (engl. *minimum constant latency*) najmanji je broj  $L$  koji kaže da se bilo koji broj procesa može započinjati u vremenskim razmacima  $L$  a da se ne pojavi sukob. Takav rad ne mora biti optimalan. Dopoštanjem promjenljivog razmaka započinjanja procesa u cjevovodu može se postići niža latencija, tj. veća propusnost.

#### Vektorski aritmetički mnogoprocesorski sustavi

Računala vrste SIMD imaju mnogo procesnih jedinica. Svaka jedinica ima svoju memoriju i svoj aritmetički sklop. To su prava paralelna računala i procesi se odvijaju zaista istodobno. Visok trošak izgradnje mnogih aritmetičkih jedinica potrebnih za *vektorsko-matrična računala* može se smanjiti upotrebom manjeg broja snažnih cjevovodnih aritmetičkih jedinica. U sustavima VAMP (*Vector Arithmetic Multi-Processor*) posebne se memorije uz procesne elemente skupljaju u jednu veliku, zajedničku, vrlo isprepletenu memoriju. Svaki se od procesora nadomješta s nekoliko registara i tako se stvaraju *virtualni procesori*. Ti procesori imaju dovoljno pridruženih sredstava za specifikaciju posla, ali stvarno ne mogu izvoditi posao. U jednom maskovnom  $N$ -bitovnom registru pamte se aktivni virtualni procesori, a u drugom su uvjetnom  $N$ -bitovnom registru uvjeti postignuti u prethodnom radu. Podaci iz uvjetnog registra služe za određivanje aktivnih virtualnih registara u sljedećoj naredbi.

Zadaci se izvode u cjevovodnoj aritmetičkoj jedinici koja poslužuje jedan po jedan virtualni procesor, pa izvođenje naredaba i zadataka nije zapravo istodobno. Organizacija zajedničke memorije za sve procesore olakšava pristup do podataka koje treba dijeliti. Nepotreban virtualni procesor povlači se iz popisa i upotrebe sredstava računala, pa je stvarna brzina rada ovisna o broju aktivnih procesora. Arhitektura prema ideji VAMP snažno je utjecala na razvoj današnjih superračunala.

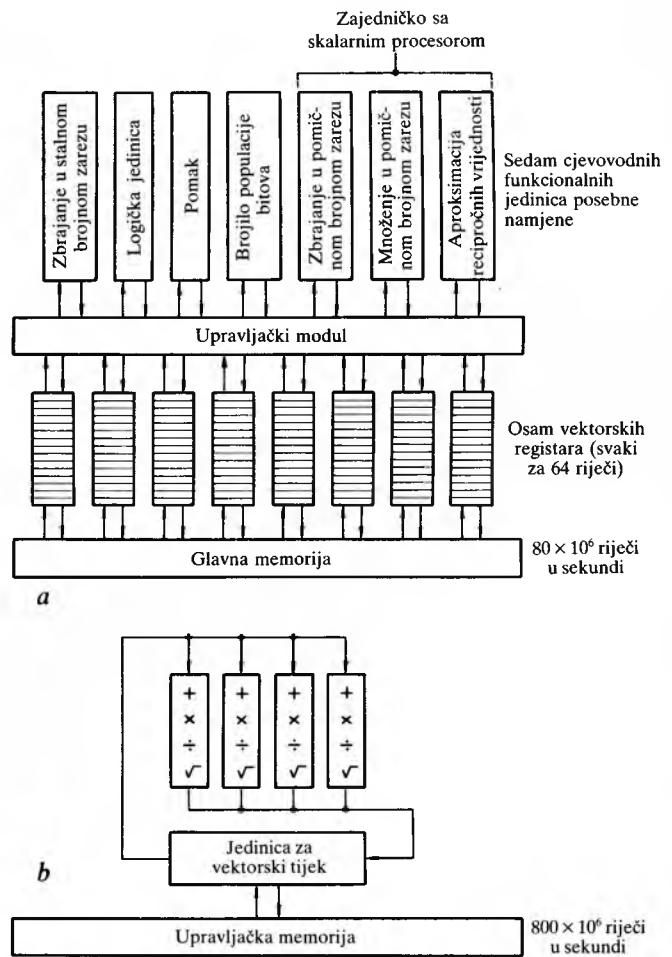
#### Superračunala

Široko rasprostranjena stolna računala izvode od nekoliko stotina do nekoliko tisuća algebarskih operacija u sekundi s brojevima prikazanim s pomicnim brojnim zarezom. Nova velika računala koja se danas nazivaju *superračunalima* izvode 100 ili više milijuna algebarskih operacija u sekundi, dakle više od 100 megaflopsa. Na tržištu se danas nalaze dva glavna predstavnika te vrste računala: CRAY-1 (tvrtka Cray Research Inc.) i CYBER 205 (tvrtka Control Data Corporation). To su vektorski cjevovodni sustavi. Njihov visokoparalelni prethodnik ILLIAC IV radio je od 1972. do 1981. godine. Propusnost računala ILLIAC IV bila je oko 20 megaflopsa. Treba napomenuti da su podaci o propusnosti superračunala

podaci proizvođača. Takva se propusnost postiže za kratke nizove podataka i pod posebnim uvjetima. Stvarna propusnost ovisi o programu koji se izvodi, a on nije nikad idealno sastavljen s obzirom na moguću paralelnost. Osim spomenutih superračunala postoji čitav niz drugih koja samo donekle zaslizuju naziv superračunalo.

Današnja superračunala upotrebljavaju integrirane logičke sklopove s vremenom porasta reda veličine 1 ns. Brzina je širenja unutar računala  $\sim 30$  cm/ns. Iz toga slijedi da se sklopovi moraju smjestiti na što manjem prostoru. To znači da raste količina razvijene topoline i hlađenje postaje vrlo ozbiljan problem. Sustav ILLIAC IV imao je hlađenje ohladenim zrakom, a sustavi CRAY-1 i CYBER 205 upotrebljavaju ohlađeni freon, koji protječe kroz cijevi za hlađenje.

Osnovna je zamisao sastava superračunala CRAY-1 i CYBER 205, koji se inače dosta razlikuju, pokazana na sl. 39. Memorije su hijerarhijski organizirane: dvije registarske memorije, glavna memorija, dopunska poluvodička memorija i diskovna memorija. Sve su funkcionske jedinice cjevovodne i mogu se istodobno upotrijebiti, ali ne mogu raditi s maksimalnom brzinom, jer imaju zajedničke dijelove, kao *podatakovne spojne putove*, memoriju itd.



Sl. 39. Shema osnovne zamisli strukture računala CRAY-1 i CYBER 205. a - vektorski procesor računala CRAY-1. b - vektorski procesor računala CYBER 205

Računalo CRAY-1 ima sedam posebnih cjevovodnih jedinica za izvođenje pojedinih operacija (tzv. populacijska jedinica broji binarne jedinice), koje su povezane s poljima vektorskih registara. Tri jedinice služe i za skalarne operacije. Polja vektorskih registara povezana su s glavnom memorijom i omogućuju prijenos 80 milijuna riječi u sekundi. Sustav CYBER 205 ima do četiri jednakih cjevovodne aritmetičke jedinice. Svaka od njih može pribrajati, množiti, dijeliti i vaditi drugi korijen. Veza je s memorijom takva da se prenosi

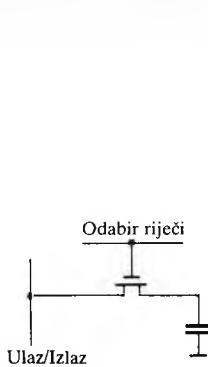
do 800 milijuna riječi u sekundi. Iako CRAY-1 ima kraći procesorski ciklus, CYBER 205 postaje brži kada duljina vektora prijeđe neku granicu.

S. Turk

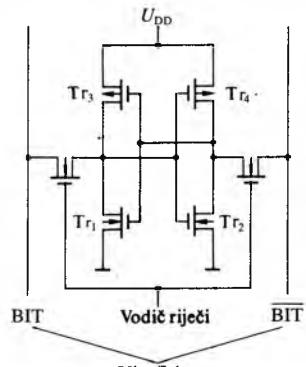
### MEMORIJE

Memorije digitalnih računarskih sustava razvijaju se vrlo brzo. Razvijaju se u dva smjera: usavršavanje i sazrijevanje postojećih sklopovskih izvedaba i proizvodnih postupaka radi poboljšanja tehničkih parametara i sniženja cijene (v. *Impulsna i digitalna tehnika*, TE 6, str. 463), te razvoj novih ili bitnije promjene postojećih rješenja.

**Dinamičke memorije** (DRAM) s jednim tranzistorom usavršene su tako da danas predstavljaju dominantno tehničko rješenje za glavne operativne memorije. Osnovna ćelija takve memorije (sl. 40) sastoji se od kondenzatora i jednog tranzistora. Podatak se pamti nabojem na kondenzatoru, kao i u drugim memorijama istog tipa. Informacija da je kondenzator nabijen (logička 1) može se pročitati tako da se pogodnim signalom na vodiču za odabir riječi uključi tranzistor MOSFET, pri čemu će se na izlazu pojaviti napon s kondenzatora. Ako kondenzator nije bio nabijen (logička 0), napon će na izlazu biti jednak nuli. Pri čitanju će se kondenzator izbiti, tj. informacija će biti izbrisana, pa se takvo čitanje zove *destruktivno čitanje*. Stoga treba informaciju nakon čitanja ponovno upisati. Jednotranzistorska ćelija zauzima manje prostora, pa može više ćelija stati na jednu pločicu poluvodiča. U redovnoj su proizvodnji čipovi od 1 megabit, a eksperimentalno su razvijeni od 4 i 16 megabita.



Sl. 40. Ćelija dinamičke memorije s jednim tranzistorm



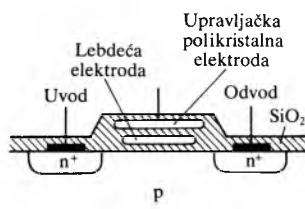
Sl. 41. Ćelija CMOS statičke memorije

**Statičke memorije** (SRAM) također su razvijene s povećanim kapacitetom i brzinom rada osnovnih jedinica. Usavršavanje proizvodne tehnologije omogućilo je da se upotrijebi i tranzistori CMOS. Osnovna memoriska ćelija statičke memorije s tranzistorima CMOS (sl. 41) ima u osnovi istu strukturu kao i osnovna ćelija s tranzistorima MOSFET (v. *Impulsna i digitalna tehnika*, TE 6, str. 463, sl. 89), samo što su u bistabilu upotrijebljeni komplementarni tranzistori, tako da je dobivena struktura CMOŠ. Time je znatno smanjen potrošak, jer, osim u trenutku promjene, kroz tranzistore ne teče struja zato što je u paru komplementarnih tranzistora ( $Tr_1/Tr_3$ ;  $Tr_2/Tr_4$ ) uvihek jedan isključen, a drugi je u zasićenju.

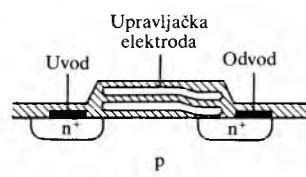
### Izbrišive programabilne permanentne memorije (EPROM)

Osnovni je element takve memorije MOSFET s lebdećom elektrodom (sl. 42). Podaci se upisuju tako da se na upravljačku elektrodu dovodi impuls relativno visoka napona (tipično 20 V) i relativno duga trajanja (tipično više milisekunda), koji uzrokuje nabijanje lebdeće elektrode elektronima. Naboј na lebdećoj elektrodi ostaje i nakon prestanka upisnog impulsa. Lebdeća je elektroda potpuno okružena izolatorom (silicij-dioksid), a izboj je zanemariv, pa često traje i više godina. Naboј na lebdećoj elektrodi mijenja, tj. povisuje prag vođenja tranzistora. Impuls za čitanje normalne amplitude od 5 V koji se dovodi na upravljačku elektrodu neće uključiti

tranzistor ako je lebdeća elektroda nabijena. Naboј se izbija, tj. informacija zapisana u memoriji briše se pomoću ultraljubičastog zračenja kojim se obasjava područje oko lebdeće elektrode. Ultraljubičasto zračenje ionizira silicij, tj. stvara parove elektron-šupljina te neutralizira uskladišteni naboј. Upravljačka i lebdeća elektroda načinjene su od polikristalnog silicija koji je dobar vodič, a istodobno propustan za ultraljubičasto zračenje. Stoga kucište u koje je takva memorija montirana mora imati kremen i prozoričić za prolaz ultraljubičastog zračenja. Takvim se postupkom odjednom briše čitav sadržaj memorije koju prije toga treba izvaditi iz uređaja i staviti u poseban uredaj za brisanje i upisivanje podataka.



Sl. 42. MOSFET s lebdecem elektrodom



Sl. 43. Osnovna ćelija električki izbrisive permanentne memorije

**Električki izbrisive permanentne memorije** (EEPROM) također se osnivaju na upotrebi lebdeće elektrode, kao i izbrisive memorije EPROM. Razlika je u tome što je širina izolatora između lebdeće elektrode i odvoda znatno smanjena (sl. 43) (tipično na manje od 0,02 μm). Upisivanje i brisanje postiže se tuneliranjem elektrona kroz tako suženi tanki sloj, tzv. Fowler-Nordheimovim mehanizmom tuneliranja. Kad se primjeni oko 20 V na upravljačku elektrodu, a odvod se drži na potencijalu mase, lebdeća elektroda se nabija elektronima. Izbijanje se dobiva upravo obrnutim postupkom: upravljačka se elektroda spoji na masu, a odvod se digne na napon od 20 V, pri čemu je uvod odspojen, tj. na lebdećem je potencijalu. Danas ima memoriskih čipova i do 1 megabit, a najavljuju se i od 4 megabita. Vrijeme je čitanja jedne riječi ~150 ns, a upisivanje ~1 ms.

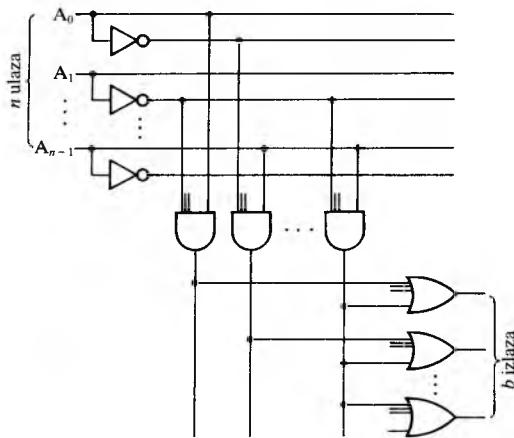
**Programabilna logička polja** (PLA). Permanentne memorije, osim za pohranjivanje podataka, mogu se upotrijebiti i za realizaciju logičkih funkcija. Svaka se logička funkcija jednoznačno definira tablicom logičkih kombinacija (ili kraće kombinacijskom tablicom) koja prikazuje stanje izlazne varijable za svaku moguću kombinaciju ulaznih varijabli. Ako se ulazne varijable dovedu kao adresu na ulaz permanentne memorije, onda stanje izlazne varijable treba zapisati u permanentnu memoriju. Ako adresa ima  $n$  bitova, može se realizirati toliko različitih logičkih funkcija od  $n$  varijabli koliko bitova ima izlazna riječ memorije.

Permanentna se memorija logički sastoji od dva dijela (v. *Impulsna i digitalna tehnika*, TE 6, str. 466): dekodera i kodera. Koder se sastoji od  $2^n$  logičkih I-sklopova, a dekoder od onolikog broja logičkih ILI-sklopova koliko bitova ima izlazna riječ memorije.

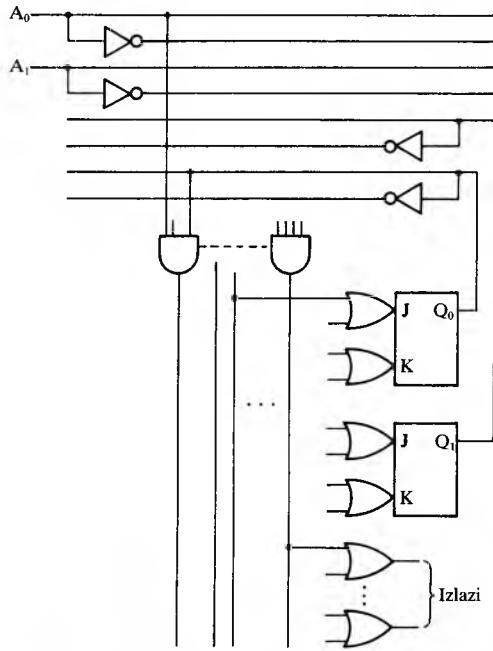
Veliki broj logičkih funkcija može se u praksi realizirati s mnogo manjim brojem I-sklopova nego što ih ima u dekoderu permanentne memorije. Stoga su na osnovi iste logičke strukture kao što je struktura permanentne memorije razvijena *programabilna logička polja* (PLA, Programmable Logic Array). Na sl. 44 prikazana je osnovna struktura takvih sklopova. Broj ulaza i izlaza kod komercijalnih sklopova najčešće je 10 do 30, a broj I-sklopova 50 do 150. Cio se integrirani sklop proizvodi u dva osnovna koraka. Najprije se u velikim serijama proizvedu I- i ILI-sklopovi, a u drugome se, prema posebnoj narudžbi korisnika, samo spoje ulazi I-sklopova s odgovarajućim ulaznim varijablama i ulazi ILI-sklopova s odgovarajućim izlazima iz I-sklopova. Sve se to radi pomoću jedne maske, kao što se radi i kod standardnih permanentnih memorija.

Da bi se izbjeglo završno prespajanje u tvornici, razvijena su i takva programabilna logička polja da korisnik može sam

programirati na licu mjesta. Za tu se svrhu upotrebljavaju rastalni osigurači ugrađeni u integrirani sklop, tj. primijenjena je ista koncepcija kao i kod memorije PROM (v. *Impulsna i digitalna tehnika*, TE 6, str. 467).



Sl. 44. Struktura programabilnog logičkog polja



Sl. 45. Programabilno sekvenčno logičko polje

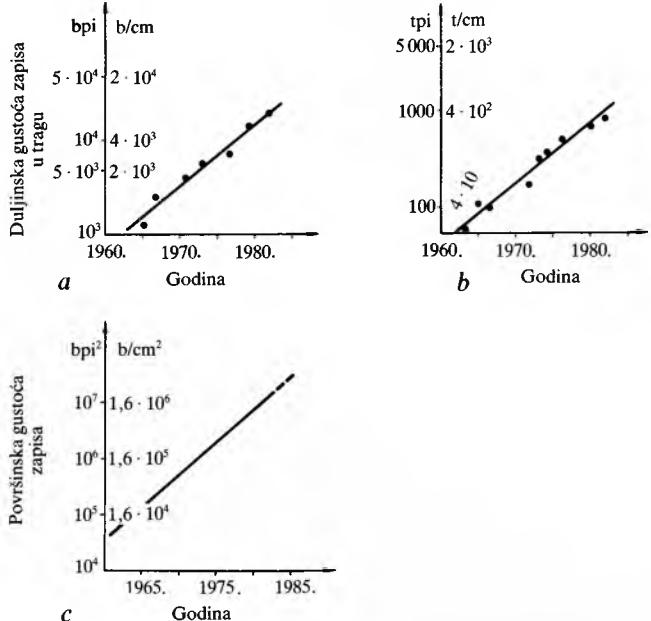
Proizvode se i takva programabilna logička polja koja imaju osim I- i ILI-sklopova također bistabile kojima su ulazi spojeni na neke izlaze ILI-sklopova, a izlazi na ulaze nekih I-sklopova (sl. 45.). Pomoću takvih sklopova mogu se ostvariti složeni kombinacijski, ali i sekvenčni sklopovi, pa se nazivaju *sekvenčna programabilna polja*.

Programabilna logička polja organizirana su u dvije razine. Za realizaciju nekih funkcija trebalo bi mnogo sklopova, što je neekonomično i sporo. Neke logičke funkcije mogu se jednostavnije realizirati logičkim sklopovima u više razina. Razvijene su stoga *višerazinske logičke matrice* (*Multilevel Gate Array* ili, kraće, samo *Gate Array*) koje se sastoje od prefabriciranih logičkih sklopova s tranzistorima smještenim kao u matrici između vertikalnih i horizontalnih vodiča. Na kraju proizvodnje spajaju se na potrebnim mjestima, prema narudžbi korisnika, na isti način kao što se to radi i za memorije ROM ili PLA.

#### Magnetske elektromehaničke memorije

Magnetske elektromehaničke memorije bile su i ostale glavnim predstavnikom memorija za praćenje vrlo velikog

broja podataka. Mogu se razvrstati u dvije osnovne grupe: diskove i vrpce ili trake, a svaka se od njih intenzivno razvija. Pokazatelj je tog razvoja i porast površinske gustoće zapisa na disku (sl. 46c). Posljednjih dvadeset godina gustoća se približno svake dvije i pol godine udvostručava, odnosno porasla je dosad  $\sim 1000$  puta, što je brže nego kod poluvodičkih paralelnih memorija. Površinska je gustoća porasla zbog povećanja gustoće bitova u tragu, odnosno gustoće tragova. Na jednu i drugu gustoću utječe niz tehnoloških parametara, u prvom redu magnetska svojstva i debljina magnetskog materijala, način (kód) magnetskog zapisa, udaljenost glave za čitanje i pisanje od površine diska te zračni raspor u glavi.



Sl. 46. Grafički prikaz porasta gustoće zapisa na magnetskim diskovima u toku razvoja. a duljinska gustoća zapisa u tragu, b radikalna gustoća tragova, c površinska gustoća zapisa ( $bpi$  bitova po inču,  $tpi$  tragova po inču,  $bpi^2$  bitova po četvornom inču)

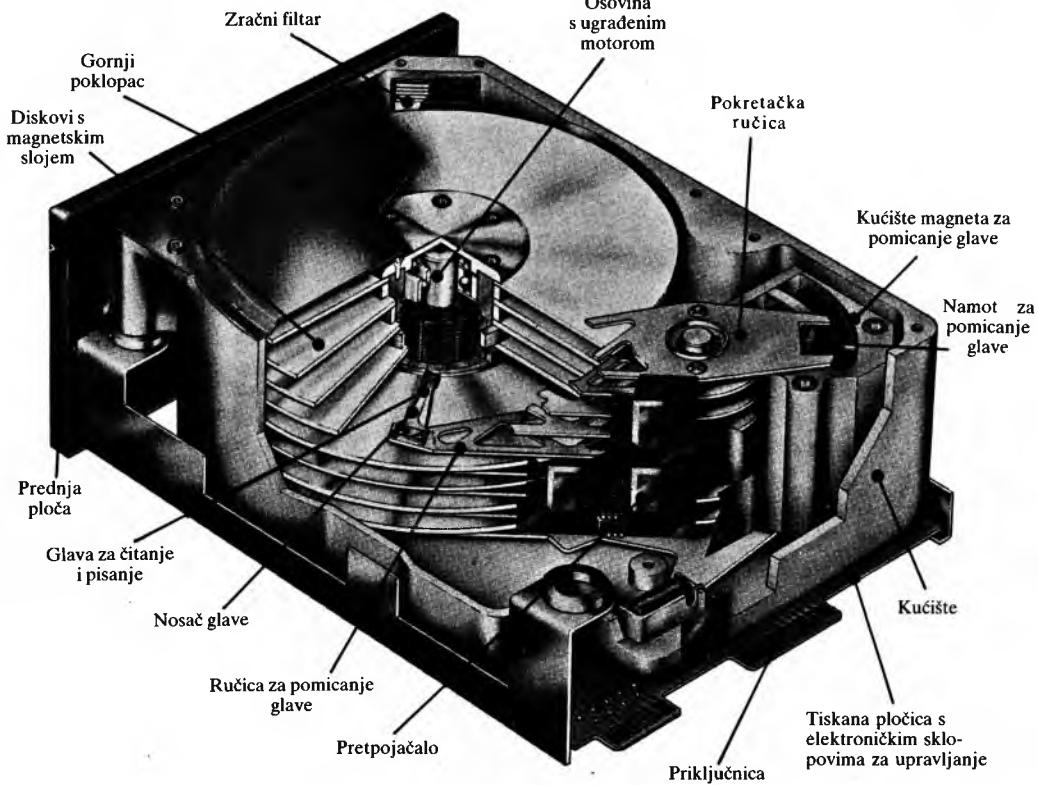
Sve donedavno osnovni je magnetski materijal za te svrhe bio željezni oksid u obliku praška, pomiješan s nekim vezivom, da bi se mogao kao sloj nanijeti na nemagnetsku površinu (aluminij za disk, a polimerni materijal za diskete i trake). Najčešće se upotrebljava  $\gamma\text{-Fe}_2\text{O}_3$  ( $\gamma$  karakterizira kristalnu strukturu) u obliku igličastih čestica, čime se postiže jaka anizotropija u ravni diska i time pravokutna petlja histereze potrebna za pouzdano pamćenje. Da bi se povećala magnetska gustoća materijala, može se umjesto magnetskih čestica povezivati vezivom, nanijeti na podlogu kontinuirani sloj oksida. Time se povećava amplituda izlaznog signala i smanjuje šum, što povećava i sigurnost čitanja. Kontinuirani magnetski sloj može se nanijeti prskanjem u vakuumu. Dodaje se nešto malo i kobalta radi povećanja koercitivne sile. Obično se traži koercitivnost  $20\cdots60$  kA/m. Za postizanje veće gustoće zapisa potreban je i tanji magnetski sloj. To, međutim, smanjuje ukupni magnetski tok, a time i amplitudu signala. Da bi se kompenzirao taj efekt, može se oksidni sloj zamijeniti feromagnetskim metalom, odnosno slitinom koja ima mnogo veći remanentni magnetizam. Do sada su razvijeni diskovi sa slojem kobalta, jer željezo i nikal ne daju dovoljno veliku koercitivnu silu. Tako postignuta gustoća zapisa veća je oko pet puta nego s feritnim česticama, i iznosi  $\sim 10$  milijuna bitova po kvadratnom centimetru. Postiže se duljinska gustoća veća od 12000 bitova po centimetru (30000 bpi).

Razvija se i vertikalni zapis, kojemu su magnetske domene za pamćenje podataka orijentirane okomito na magnetsku površinu. Očekuje se da takav način omogući dalje povećanje gustoće.

**Disk-jedinice** razvijale su se u smjeru čvrstih diskova (*Hard Disk*) s nepomičnim paketom diskova, što se pokazalo

povoljnije s gledišta postizanja velikih kapaciteta i sigurnog rada. Posebno je uspješan tip nazvan *winchester-diskom* (termin disk služi i kao kratica za disk-jedinicu), te je postao gotovo industrijskim standardom. Od prve konstrukcije te vrste (1973) na disk-jedinicama se do danas dogodio niz poboljšanja, ali je osnovna koncepcija ostala ista, pa je taj naziv danas sinonim za čvrsti disk. Prvi projekt takva diska predviđao je dvostruku jedinicu od po 30 megabytea, pa je prema pušci Winchester 30-30 i on tako nazvan.

Osnovne su karakteristike winchester-diskova sljedeće: 1) Magnetska glava za čitanje i pisanje posebno je konstruirana sa tri izbočene šine, čime je lebdenje glave na zračnom jastuku reducirano na  $\sim 0,5 \mu\text{m}$ . 2) Pritisak kojim se glava drži na razmaku od rotirajuće ploče sveden je od 3,5 N na svega 0,1 N. 3) Magnetske se glave ne izvlače iz ploča prije zaustavljanja, već se pomaknu na poseban trag na rubu ploče i tu mekano spuste. Da bi se olakšala ta operacija, površina ploča je podmazana. Na trag na koji se glave spuštaju i s kojega se dižu ne upisuju se podaci. Takvim radom, što ga u osnovi omogućuje vrlo mali pritisak na glave, znatno se smanjuje opasnost od sudara glava i ploča, koji se može dogoditi ako se zaustavi rotacija ploča prije nego što se glave maknu. Na winchester-disku se zapis u takvu slučaju obično neće izgubiti. 4) Cijela je jedinica smještena u hermetički zatvorenu kutiju s posebnim filterima za zrak. To omogućuje da se disk-jedinice upotrebljavaju s radnim stanicama i osobnim računalima koja obično rade izvan specijalno klimatiziranih prostorija kakve su potrebne za velike sustave. 5) Disk-jedinice načinjene su uglavnom sa standardnim promjerom ploča od 14; 8;  $5\frac{1}{4}$  i manje od 4 (3,5 ili 3,9) inča, odnosno približno 36, 20, 13 i manje od 10 cm. Na sl. 47 prikazan je izgled tipične winchester-disk-jedinice od  $5\frac{1}{4}$  inča. Jedinice od  $5\frac{1}{4}$  inča (13 cm) obično su smještene u standardne kutije dimenzija 15 cm  $\times$  20 cm  $\times$  8 cm.

Sl. 47. Izgled Winchester-diska od  $5\frac{1}{4}$  in (13,4 cm)

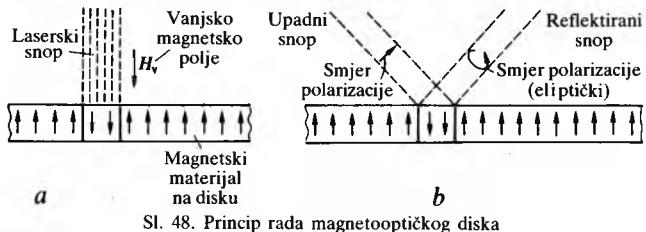
Danas se (1987) i s jedinicama od  $5\frac{1}{4}$  inča postižu i vrlo veliki kapaciteti, gotovo 800 megabytea. Osnovni su parametri takvih jedinica sljedeći: neformatizirani kapacitet 765 megabytea, broj ploča 8, srednje vrijeme pristupa 18 ms, duljinska gustoća 12200 bita po centimetru (31000 bpi), radikalna

gustoća tragova 540 bita po centimetru (1380 tpi), frekvencija čitanja 15 megabytea u sekundi.

### Optički diskovi

Optički diskovi također osiguravaju vrlo veliku gustoću zapisa. Ima ih dvije vrste: izbrisivi i neizbrisivi. Diskovi bez mogućnosti brisanja i ponovnog upisivanja informacije zapravo predstavljaju ispisnu ili permanentnu memoriju. Glavna je prednost optičkih diskova s obzirom na magnetske što im tragovi mogu biti isto tako blizu kao i bitovi u tragu, a to omogućuje postizanje većeg kapaciteta za jedan red veličine. Postoje tri tipa izbrisivih diskova: magnetooptički, promjenljive faze i promjenljive boje.

**Magnetooptički diskovi**, koji su još u intenzivnom razvoju, omogućuju znatno povećanje gustoće zapisa, tipično oko 10 puta, pa prema tome i ukupnog kapaciteta diska. Informacija se upisuje termomagnetski (sl. 48), tj. pomoću laserskog snopa, koji podiže temperaturu magnetskog medija i time reducira koercitivnu silu, tako da se može izvršiti upisivanje pomoću vanjskog magnetskog polja. Domene su orijentirane vertikalno na ravninu ploče. Cita se pomoću laserskog snopa slabijeg intenziteta, koji, kad se odbije od magnetizirane domene, promjeni polarizaciju već prema polaritetu magnetskog polja domene. Zbog ograničenja brzine premagnetiziranja, treba prije upisivanja, magnetizaciju materijala postaviti

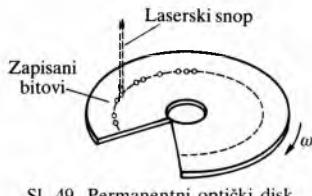


Sl. 48. Princip rada magnetooptičkog diska

u multi položaj, a tek zatim upisivati jedinice. Tada se jedan okretaj diska upotrebljava za brisanje, a drugi za upisivanje. To je velik nedostatak, pa su zato razvijene i druge metode upisivanja gdje se obje operacije mogu obaviti u jednom prolazu.

**Diskovi promjenljive faze** zapisuju informaciju tako da se laserskim snopom promijeni kristalna struktura, tj. faza, površine ploče, pa ona postane amorfna, odnosno druge kristalne strukture. U jednom i u drugom slučaju takva površina ne reflektira, odnosno manje reflektira laserski snop pri čitanju. Promjer je upisanog kružića  $\sim 1 \mu\text{m}$ . Snaga je lasera za upisivanje i brisanje različita, za upisivanje je oko 5 mW, a za brisanje 10–20 mW. Valna je duljina oko 0,8  $\mu\text{m}$ . Optički osjetljivi materijali koji se upotrebljavaju za tu svrhu nisu uvek dovoljno stabilni. Među relativno stabilnimi materijalima ubraja se slitina selena, indija i antimona, koja izdrži oko  $10^7$  upisivanja i brisanja, a očekuje se vrijeme stabilnog trajanja  $\sim 10$  godina.

**Diskovi promjenljive boje** upotrebljavaju posebnu organsku boju ili koji drugi materijal (npr. slitinu cinka i srebra) koji pod utjecajem laserskog snopa mijenja boju. Ponovno obasjavanje laserom (obično manje snage) vraća strukturu, a time i boju opet u prvobitno stanje.



Sl. 49. Permanentni optički disk

**Permanentni (neizbrisivi) optički diskovi** obrađuju se toplinom laserskog snopa kojim se formiraju nereflektirajuće udubine (rupe) u sjajno reflektirajućoj površini ploče (sl. 49). Laserski će snop npr. na površini kružića promjera 1  $\mu\text{m}$  povisiti temperaturu za 200 °C za samo 20 ns. Površina se diska nakon upisivanja zaštićuje prozirnim slojem. Proizvode se disk-jedinice za čitanje optičkih diskova u koje se umeću diskovi. Npr. jedan optički disk promjera 12 cm ima kapacitet od 600 megabytea. Nedostatak je te vrste permanentne memorije dugo vrijeme pristupa,  $\sim 0,5$  do 1,2 s.

U. Peruško

LIT.: J. C. Mirtha, Highly Parallel Information Processing Systems. U djelu: F. L. Alt and M. Rubinoff, Advances in Computers, Vol. 7. Academic Press, New York 1966. – Burroughs Corp., Digital Computer Principles. McGraw-Hill, New York 1969. – C. G. Bell, A. Newell, Computer Structures: Readings and Examples. McGraw-Hill, New York 1971. – B. Randell (ed.), The Origins of Digital Computers. Springer-Verlag, Berlin 1975. – H. S. Stone (ed.), Introduction to Computer Architecture. Science Research Associates Inc., Chicago 1975. – J. P. Hayes, Computer Architecture and Organization. McGraw-Hill, New York 1978. – C. A. Mead, L. A. Conway, Introduction to VLSI Systems. Addison-Wesley, Reading 1980. – H. T. Kung, B. Sproull, G. Stelle (ed.), VLSI Systems and Computations. Springer-Verlag, Berlin 1981. – S. P. Kartashev, S. L. Kartashev (ed.), Designing and Programming Modern Computers and Systems, Vol. I: LSI Modular Computer Systems. Prentice-Hall, Inc., Englewood Cliffs 1982. – R. D. Levine, Supercomputers. Scientific American, 246, 115–125, 1982 (No. 1 Jan.). – S. Muroga, VLSI System Design. J. Wiley, New York 1982. – D. P. Siewiorek, C. G. Bell, A. Newell, Computer Structures: Principles and Examples. McGraw-Hill, Inc., New York 1982. – J. S. Kowalik (ed.), High-Speed Computation. NATO ASI Series, Vol. 7. Springer-Verlag, Berlin 1984. – S. Hoagland, Information Storage Technology. A. Look at the Future. Computer, July 1985, 60–87. – J. M. Leod, Optical Storage. It may be close to challenging magnetics. Electronics, May 1986, 30–33. – J. Voelcher, Winchester Disks Reach for the Gigabyte. IEEE Spectrum, February 1987, 64–67. – S. Turk, Arhitektura i organizacija digitalnih računala. Školska knjiga, Zagreb 1988.

U. Peruško S. Turk

**RAČUNANJE POVRŠINA U GEODEZIJI**, računski postupci za određivanje površina zemljišta na temelju podataka mjerena na terenu ili na planu. Upotrebljavaju se sljedeće metode računanja površina: numerička, grafička, mehanička i kombinirana metoda.

Izbor metode računanja ovisi o potrebnoj točnosti, što zahtjeva i primjenu pogodne metode mjerena na terenu.

Površina se pojedinog lika (parcbla, grupa parcbla ili veća površina zemljišta) računa, u pravilu, dva puta. Konačna površina lika aritmetička je sredina rezultata obaju računanja ako je njihova razlika u granicama dopuštenog odstupanja (prema *Pravilniku o tehničkim propisima za izradu originala planova i određivanje površina parcela pri premjeru zemljišta*).

Kad se računa površina s plana, treba uzeti u obzir i promjenu površine lika zbog promjene dimenzija podloge na kojoj je izrađen plan. Da bi se utvrdile te promjene, izmjere se dimenzije okvira plana po osima  $y$  i  $x$ , tj.  $y'_{sjever}, y'_{jug}, x'_{zapad}, x'_{istok}$  (sl. 1). Za računanje se uzima obična aritmetička sredina izmjerjenih dimenzija po pojedinim osima, pa je

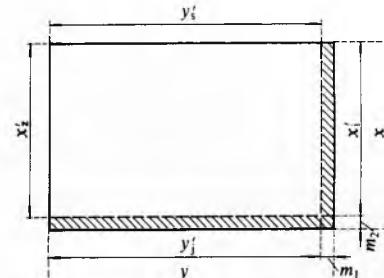
$$y' = \frac{1}{2}(y'_{sjever} + y'_{jug}), \quad (1a)$$

$$x' = \frac{1}{2}(x'_{zapad} + x'_{istok}). \quad (1b)$$

Ukupna je promjena površine plana

$$\Delta P = ym_2 + xm_1 - m_1m_2, \quad (2)$$

gdje su  $y$  i  $x$  originalne dimenzije okvira plana,  $m_1 = y' - y$  promjena dimenzije plana po osi  $y$ , a  $m_2 = x' - x$  promjena dimenzije plana po osi  $x$ . Kako je originalna površina plana poznata, može se izračunati promjena po nekoj jedinici površine, a time i promjena površine za lik izračunat na planu.

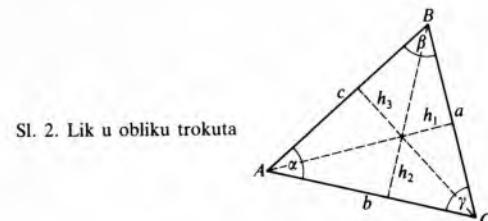


Sl. 1. Uz određivanje promjene površine plana

Upotrebljavaju se i starije jedinice za površinu: četvorni hvat ( $3,596652 \text{ m}^2$ ) i jutro (1600 četvornih hvati,  $5754,643 \text{ m}^2$ ).

**Numerička metoda** upotrebljava se za računanje površine likova pomoću originalnih mjera na terenu i pomoću koordinata prijelomnih točaka lika.

Veličine izmjerene na terenu reduciraju se na horizont. Na terenu se izmjere one veličine koje omogućavaju računanje površine lika pomoću pravilnih geometrijskih likova (trokut, pravokutnik, trapez i sl.), na dva neovisna načina. Geometrijsku pravilnost likova treba provjeriti na terenu. Zbog točnosti i ekonomičnosti preporučuje se podjeliti lik kojemu treba odrediti površinu na što manji broj geometrijskih likova.



Sl. 2. Lik u obliku trokuta

Površina lika u obliku trokuta može se izračunati na temelju izmjerene jedne stranice i visine na tu stranicu (sl. 2)

$$P = \frac{1}{2}ah_1 = \frac{1}{2}bh_2 = \frac{1}{2}ch_3. \quad (3)$$

Ako su poznate dvije stranice i kut između njih, površina je trokuta

$$P = \frac{1}{2}ab \sin \gamma = \frac{1}{2}ac \sin \beta = \frac{1}{2}bc \sin \alpha. \quad (4)$$