

MEHURČNI POMNILNIKI III

B. MIHOVILOVIĆ,
J. ŠILC,
P. KOLBEZEN

UDK: 681.327.6

INSTITUT „JOŽEF STEFAN“

V članku so opisane nekatere organizacije magnetnih mehurčnih pomnilnikov kot so: dolgi premikalni register, major/minor zanke, koincidenčna izbira podatkovnega bloka, dekoderji in dinamično urejeni podatkovni bloki; njihov razvoj in primerjave med njimi. Čeprav je danes najčesteje uporabljena major/minor zankna organizacija, niso ostale organizacije nič manj pomembne, saj vsebujejo mnoge dobre lastnosti. Tako recimo organizacija s koincidenčno izbiro podatkovnega bloka vnaša modularnost, skrajša čase dostopa, kakor tudi zmanjšuje dovzetnost do napak. Organizacija z dekoderji ima nekatere prednosti v primerjavi z major/minor zankno organizacijo, saj nudi možnost zbiranja pomnilniških lokacij z adresiranjem znotraj samega pomnilniškega elementa. Najkrajše čase dostopa in cikla pa doseže organizacija z dinamično urejenimi podatkovnimi bloki.

MAGNETIC BUBBLE MEMORIES - PART 3. In this paper the concepts of several bubble memory organizations such as: long shift register, major/minor loops, coincidence selection, decoders and dynamically - ordered shift registers have been reviewed; and their efficacies in achieving economy and improving performance have been compared. Both the conceptual and hardware developments for these memory organizations will be updated. While most of the hardware development is concentrated on the major/minor loops, it should be noted that the other three schemes are by no means less important. The coincident selection scheme provides array modularity, limits vulnerability to defects, and reduces access time for large - capacity chips. Decoders are more versatile than the major/minor loops since they offer address selection capability on the bubble chip. The dynamic ordering greatly reduces the access and cycle times.

1. UVOD

Današnji razvoj mehurčnih pomnilnikov poteka v več smereh:

- fizikalno-tehnološki razvoj stremi k izpopolnitvi sedanjih in uvajanju novih tehnologij, v želji za izboljšanjem osnovnih lastnosti magnetnih mehurčnih pomnilnikov, kot so večanje gostote, višanje propagacijske frekvence, poenostavljanje tehnologije izdelave čipa, uvajanju novih materialov,...
- razvoj v smeri načrtovanja in optimizacija organizacijskih struktur magnetnih mehurčnih pomnilnikov, ki poizkuša minimizirati čas dostopa, čas branja, čas bralno-pisalnega cikla,...
- sočasno poteka tudi razvoj v načrtovanju univerzalnejše in bolj integrirane podporne opreme (LSI krmilniki), kar bi omogočilo večjo kompatibilnost mehurčnih pomnilniških sistemov.

Preden spregovorimo o organizaciji magnetnega mehurčnega

pomnilnika preglejmo kako se odražajo osnovne funkcije pomnilnika v magnetnem mehurčnem pomnilniku [9, 10]:

- pomnjenje: logični vrednosti 1 in 0 na določeni bitni-lokaciji, sta predstavljeni s prisotnostjo oz. odsotnostjo magnetnega mehurčka na dani lokaciji,
- propagacija; dostop in organizacija: magnetni mehurček je mogoča širiti (propagirati) s pomočjo "gibljivih magnetkov", ki jih ustvarjajo z rotirajočim magnetnim poljem magneteni permalojni vzorci, katerih razporeditev določa način dostopa in organizacijo magnetnega mehurčnega pomnilnika,
- branje: za branje se izkorišča magnetorezistenčni efekt, to je spreminjanje upornosti permalojnega vzorca, ko je le-ta pod vplivom magnetnega polja, ki je posledica prehoda magnetnega mehurčka pod njim,
- brisanje: informacijo, ki je predstavljena z magnetnim mehurčkom, je mogoče zbrisati tako, da mehurček uničimo ali odstranimo z dane bitne lokacije,
- vpis: informacijo, ki je predstavljena z magnetnim mehurčkom, je mogoče vpisati tako, da generiramo nov ma-

gnetni mehurček.

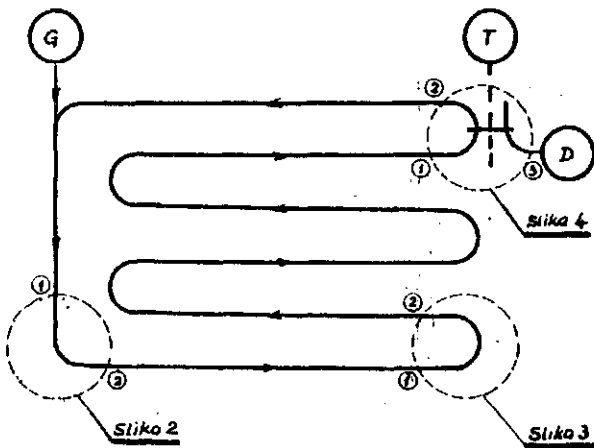
2. ORGANIZACIJA MAGNETNEGA MEHURČNEGA POMNILNIKA

Pri vseh pomnilnikih želimo organizirati shranjevanje podatkov tako, da optimiziramo število prenosnih linij, število povezav med pomnilniškim elementom in potrebno podporno opremo, kakor tudi čas dostopa in čas cikla.

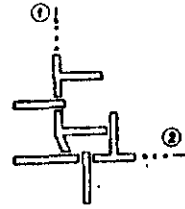
V nasprotju s koincidenčnimi pomnilniki (feritna jedra, polprevodniki, supra-prevodniki,...), kjer je osnovni mehanizem shranjevanje tokovna ali napetostna preklopna pragovna lastnost, je osnovni mehanizem uporabljen pri večini mehurčnih pomnilniških organizacij v sposobnosti voditi magnetene mehurčke od ene propagacijske sledi do druge s pomočjo ustrezno oblikovanih in magnetenih permalojnih vzorcev (prenosna vrata - transfer gate).

Vsekakor je razvoj organizacijskih struktur mehurčnih pomnilnikov rodil različne načine organizacij. Prve generacije komercialnih mehurčnih pomnilnikov (npr. 1 Kbitni Hitachijev čip, 100 Kbitni Rockwellov čip,...) so bile organizirane v obliki dolgega premikalnega registra (long shift register). Zelo kmalu pa so se pojavile zahteve po večji kapaciteti pomnilnikov, ki pa bi bili v tej organizaciji sila počasni (veliki časi dostopa in cikla). Tudi dekompozicija dolge zanke v množico med seboj ločenih krajših zank (separate short shift registers) ni dala zaželenih rezultatov, dokler niso bile razvite učinkovitejše organizacijske strukture kot so: major/minor zanke (major/minor loops), dekodirji (decoders), koincidenčna izbira podatkovnega bloka (coincident selection of data block) in dinamično urejeni podatkovni bloki (dynamically ordered shift registers)¹⁾

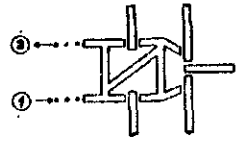
Oglejmo si nekoliko podrobneje posamezne organizacijske strukture.



slika 1.

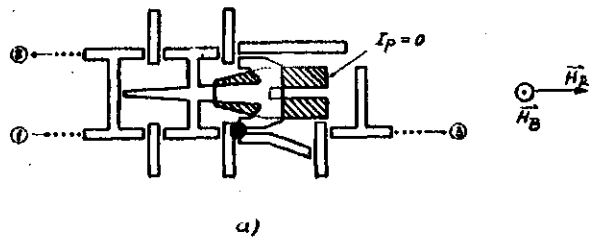


slika 2.

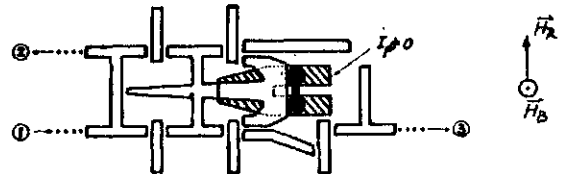


slika 3.

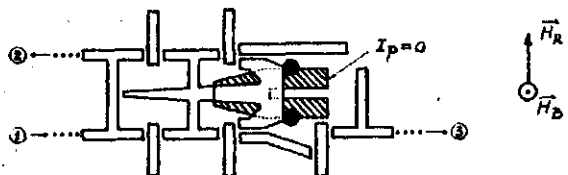
Dolgi premikalni register[1]. Takšna oblika organizacije mehurčnega pomnilnika se danes ne uporablja več, vendar jo omenjamo zato, ker vsebuje elemente, ki se pojavljajo v vseh danes uporabljenih organizacijah. Osnovna shema te organizacije je prikazana na sliki 1. Dolgo zanko oblikujejo na garnetno osnovo (nosilec mehurčkov) namešeni permalojni propagacijski vzorci, ki v interakciji z rotirajočim magnetnim poljem povzročijo, da mehurčki neprestano krožijo v tej zanki. Spreminjanje smeri širjenja magnetnih mehurčkov je izvedeno s posebno oblikovani-



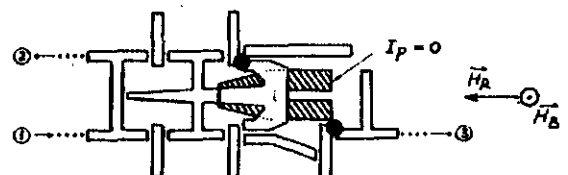
a)



b)



c)



d)

slika 4.

¹⁾ Naštete organizacijske strukture se uporabljajo pri tehnologijah s permalojnimi vzorci (permalloy bar technology), dočim se pri tehnologiji urejene mreže mehurčkov (bubble lattice file) uporablja drugačna organizacija.

mi in nameščenimi propagacijskimi vzorci. 90° -ska sprememba smeri je prikazana na sliki 2, 180° -ska pa na sliki 3. Nedestruktivno branje informacij, ki se nahajajo v tej zanki, je mogoče s pomočjo enosmernih prenosnih podvojitvenih vrat (slika 4) in detektorja.

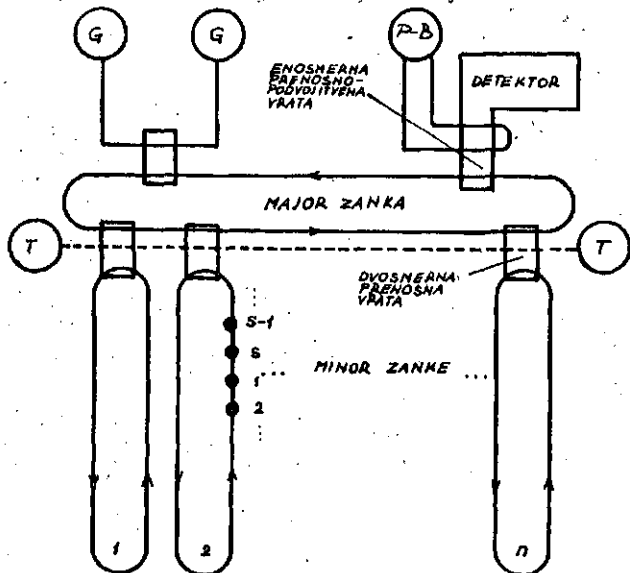
Prenos mehurčka iz zanke k detektorju omogočajo permalojni vzorci specifične oblike in pod njimi nameščena tokovna zanka. Zahteva za branje informacije pomeni prisotnost tokovnega impulza I_p (slika 4 b) v tokovni zanki, ki magnetni mehurček podvoji, "original" se vrne v zanko, "kopijo" pa sprejme detektor. Vpis nove informacije pa je mogoče preko generatorja.

Kot rečeno, ta način organizacije pomnilnika ni mogel slediti zahtevam po vse večjih kapacitetah zaradi sorazmerno dolgih časov dostopa t_a , ki so proporcionalni kapaciteti pomnilnika C in sicer:

$$t_a = \frac{C}{2} T_R \quad (1)$$

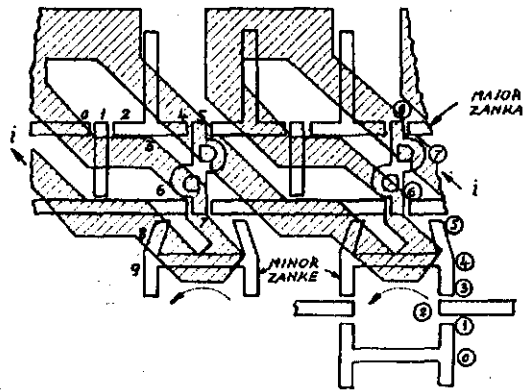
kjer je T_R perioda rotirajočega magnetnega polja H_R . Kot primer vzemimo kapaciteto mehurčnega pomnilnika 92 Kbitov in propagacijsko frekvenco 100 KHz. V tem primeru bi bil čas dostopa kar $\approx 0,5$ sek. Seveda so takšni časi mnogo predolgi zato se takšna organizacija danes ne uporablja več.

Major/minor zanke. Pri današnjih magnetnih mehurčnih pomnilnikih najčesče zasledimo major/minor zanko organizacijo (slika 5). Informacija je shranjena v številnih paralelnih zaprtozančnih premikalnih registrih, ki jih imenujemo minor zanke (minor loop). Pod vplivom rotirajočega magnetnega polja magnetni mehurčki (nosilci informacije) sinhrono krožijo v teh zankah. Njihov položaj je v danem trenutku vselej točno določen, saj se magnetni mehurčki ob vsakem zasuku polja H_R za poln kot, premakne-



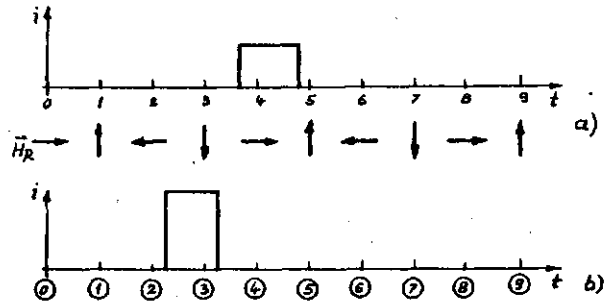
slika 5.

jo za periodo propagacijskih vzorcev p . Ta osnovni premik imenujemo korak. Operaciji vpis in branje je mogoče izvesti preko skupnega premikalnega registra, imenovanega major zanka (major loop), na katero niso vezane le vse minor zanke, temveč tudi generator (vpis informacije) in detektor (branje informacije). Povezavi detektor - major zanka in generator - major zanka sta povsem enako izvedeni kot pri organizaciji v obliki dolgega premikalnega registra (slika 4), dočim so povezave minor - major zanka izvedene s pomočjo dvosmernih prenosnih vrat brez podvajanja (slika 6). [10]



slika 6.

Posebnost takšnih vrat je specifičen δ -sko oblikovan permalojni vzorec, dodatno pa vodimo pod major zanko spirala-sto oblikovano prevodno plast, preko katere je mogoče s pomočjo tokovnega impulza izvršiti hkraten prenos iz ali v minor zanke. Opišimo najprej prenos iz major v minor zanko. Časovni potek toka i , ki je potreben za ta prenos



slika 7.

je prikazan na sliki 7 a. V času, ko je $i = 0$, se magnetni mehurček pod vplivom rotirajočega polja premika v major zanko. Predpostavimo, da se magnetni mehurček nahaja na poziciji 0 (slika 6) in se zaradi zasuka H_R za poln kot, premakne na pozicijo 4, torej naredi en korak. Ob zasuku H_R za nadaljnih 180° bi mehurček prešel pod naslednji T vzorec, kar pa mu z ustreznim dolgim ($\approx T_R/3$) in velikim (25 mA) tokovnim impulzom, ki ga pošljemo preko spiralnega vodnika, preprečimo in tako preide mehurček v pozicijo 6. Pri nadaljni rotaciji polja ($i = 0$) prevzame mehurček minor zanko. Prenos iz ene od minor zank v major zanko poteka povsem analogno kot v prejšnjem primeru, le da

je tu časovni potek toka i drugačen. Tokovni impulz katerega amplituda je nekoliko večja (50 mA), dolžina pa enaka, omogoči, da mehurček ob zasuku H_R za 180° preide iz pozicije ③ v ⑤ in ne na nasprotno stran minor zanke, kot bi bilo v primeru, če tokovnega impulza nebi bilo.

Opišimo potek osnovnih operacij in sicer: branja, brisanje in vpis informacije. Mesta magnetnih mehurčkov, ki se nahajajo na istih horizontalnih pozicijah v minorskih zankah tvorijo blok podatkov ali drugače povedano, vsak bit bloka se nahaja v drugi minor zanki. Pri branju izbranega bloka se ustrezni biti s pomočjo rotirajočega magnetnega polja sinhrono premikajo tako, da pridejo tik ob major zanko (pozicija ③ na sliki 6). Paralelni prenos vseh bitov v major zanko se izvede s pomočjo dvosmernih prenosnih vrat, tako, da v minor zankah ostanejo dotična mesta nezasedena. Bite, ki sestavljajo blok sedaj serijsko premikamo toliko časa tako, da po določenem številu korakov prvi bit bloka doseže enosmerna prenosno-podvojitvena vrata. Blok je sedaj mogoče destruktivno ali nedestruktivno brati ali pa brisati. Če želimo destruktivno branje ali brisanje, vrata magnetnih mehurčkov ne podvajajo temveč jih preko detektorja (pri brisanju detektor ni aktiviran) pošljemo v "prazno". Pri nedestruktivnem branju pa s pomočjo aktiviranih enosmernih prenosno-podvojitvenih vrat magnetne mehurčke podvojimo tako, da dobimo natančno kopijo bloka. Original, tako kot v prejšnjem primeru vodimo preko aktivnega detektorja, kopijo pa preko major zanke in dvosmernih prenosnih vrat vrnemo na prej izpraznjena mesta v minor zankah.

Če želimo vpisati novo informacijo v nezasedene lokacije v minorskih zankah, aktiviramo generator s pomočjo katerega pošljemo magnetne mehurčke v prazno major zanko. Nov blok podatkov se nato preko prenosnih vrat paralelno prenese v ustrezne lokacije minorskih zank.

Tudi pri major/minor zankni organizaciji je tako kot pri vseh ostalih organizacijah zelo pomembno, da so časi dostopa in cikla čim krajši. Za organizacijo pomnilnika, ki je shematično prikazana na sliki 5 analizirajmo čase in pogljemo v kolikšni meri nam jih je uspelo skrajšati [6]. Pomnilnik kapacitete C sestavlja n minornih zank, katerih vsaka lahko hrani s bitov informacije in pri tem ve-
lja zveza:

$$\hat{C} = 2n \times \frac{s}{2} = ns \quad (2)$$

V najneugodnejšem primeru, torej tedaj, ko je bit izbranega bloka v trenutku ko smo želeli ta blok prebrati, prešel ravno mimo dvosmernih prenosnih vrat in mora zato obkrožiti celotno minor zanko (s korakov), je čas dostopa, ki ga sestavljajo zakasnitev v minor zanki, zakasnitev med prvim prenosom in podvojitvijo (t_c) in zakasnitev detektiranja (t_d), enak:

$$t_a = s T_R + t_c + t_d \quad (3)$$

oziroma:

$$t_a = \frac{C}{n} T_R + t_o \quad (4)$$

kjer je T_R perioda propagacijske frekvence, to je čas v katerem mehurček naredi en korak. Kot je razvidno iz enačbe (4) večje število krajših minor zank zmanjšuje čas dostopa. Npr. 92 Kbitni (TI 0103) pomnilnik, ki ga sestavlja 157 minor zank, ki hranijo vsaka po 641 bitov, ki deluje s 100 KHz propagacijsko frekvenco in vsebuje 68 korakov od dvosmernih prenosnih vrat do podvojitvenih vrat in 86 korakov v detektorju, ima čas dostopa v najneugodnejšem primeru $t_a = (641 + 68 + 86) 10^{-5} = 8$ ms. To pa je kar 60-kratno skrajšanje časa dostopa glede na organizacijo v obliki dolgega premikalnega registra.

Čas potreben za branje bloka t_R , vsebuje poleg časa dostopa t_a še čas, ki je potreben za to, da se major zanka izprazni (kopija bloka se mora vrniti v minor zanke) in se glasi:

$$t_R = \left(\frac{C}{n} + 2n\right) T_R + t_o \quad (5)$$

Če t_R minimiziramo glede na n, dobimo konfiguracijo pomnilnika, ki jo sestavlja $n = \sqrt{C/2}$ minor zank v katerih je po $s = \sqrt{2C}$ bitov, tako, da dobimo minimalni čas potreben za branje bloka:

$$t_{Rmin} = 2\sqrt{2C} T_R + t_o \quad (6)$$

Sam pomnilnik pa ima dimenzijo 2×1 ($2n \times \frac{s}{2} = 2\sqrt{\frac{C}{2}} \times \sqrt{\frac{C}{2}}$).

Čas bralno-vpisovalnega cikla t_{RW} je enak zakasnitvi v minor zanki in zakasnitvama, ki ju vnašata branje in pisanje, ter je enak:

$$t_{RW} = \frac{C}{n} T_R + (2n + k_1) T_R + (2n + k_2) T_R \quad (7)$$

ali

$$t_{RW} = \left(\frac{C}{n} + 4n\right) T_R + t_o' \quad (8)$$

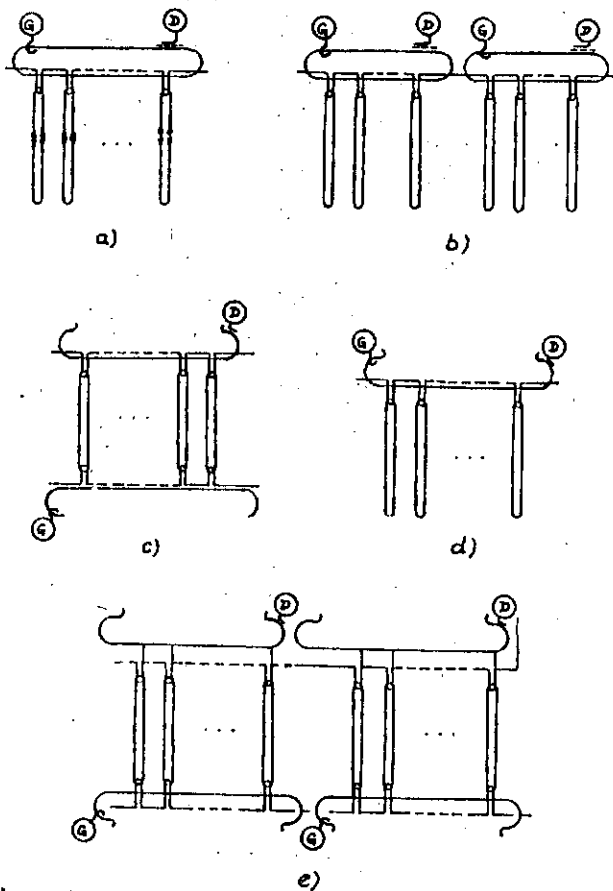
S produktom $4n T_R$ aproksimiramo čas potreben, da magnetni mehurček obide major zanko. Če t_{RW} minimiziramo glede na n, dobimo konfiguracijo pomnilnika, ki jo sestavlja $n = \sqrt{C/2}$ minor zank v katerih je po $s = 2\sqrt{C}$ bitov, tako, da dobimo minimalni čas bralno-vpisovalnega cikla:

$$t_{RWmin} = 4\sqrt{C} T_R + t_o' \quad (9)$$

Sam pomnilnik pa ima dimenzijo 1×1 ($2n \times \frac{1}{2} = \sqrt{C} \times \sqrt{C}$).

Vidimo, da v tej organizaciji (ena major zanka) naletimo na konflikt, saj sočasno ni možno minimizirati t_a in t_{RW} ,

zato so proizvajalci pristopili k različnim inačicam osnovne major/minor organizacije [1]. Nekatere od njih so prikazane na sliki 8. Slika 8 a prikazuje izvedbo kjer so uporabljene dvosmerne minor zanke, kar omogoča, da se

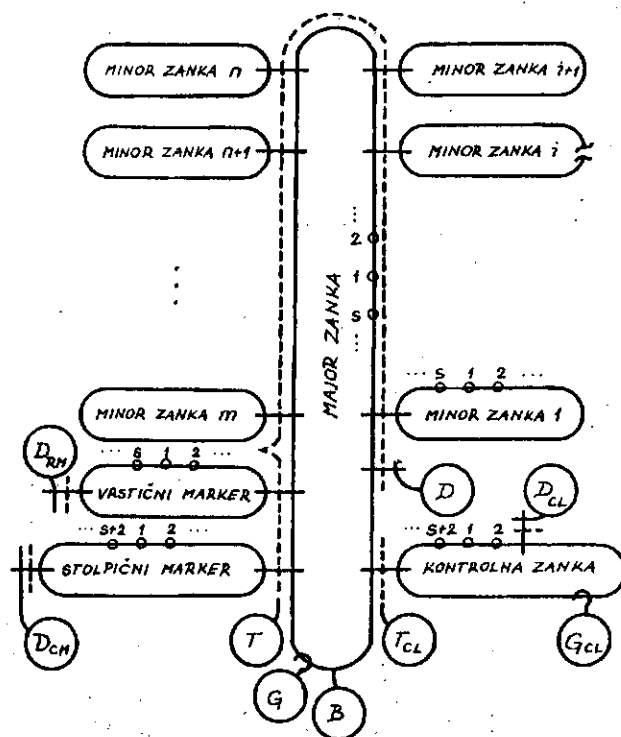


slika 8.

bit izbranega bloka približa prenosnim vratom (major zanki) po krajši poti, tako, da preide enačba (4) v obliko: $t_a = \frac{C}{2n} T_R + t_o$. Podoben efekt lahko dobimo tudi z organizacijo prikazano na sliki 8 b, saj smo z ločenim shranjevanjem lihih in sodnih bitov bloka ustvarili dve podatkovni strukturi (katerih kapaciteta je $C/2$), ki sta vsaka zase organizirani tako kot prikazuje slika 5, le da sta povezani preko skupnega prenosa. Organizacijam na slikah 8 c, d, e je skupno to, da namesto zaključeni major zank uporabljamo odprte zanke - imenovane major trakovi in namesto dvosmernih prenosnih vrat (brez podvajanja), enosmerna oziroma dvosmerna prenosno podvojitvena vrata in jih imenujemo organizacija s podvajanjem bloka.

Ob izdelavi pomnilniškega čipa so zaradi samega tehnološkega postopka vnešene določene napake (tipično 10 napak na cm^2). Pomnilniški element je kljub temu mogoče uporabljati, če so napake takega tipa, da so poškodovane le nekatere minor zanke, saj so že v naprej vnešene dodatne redundančne (minor) zanke, ki nadomestijo poškodovane.

Odkrivanje in nadomeščanje slabih minor zank z redundantnimi, ki je avtomatsko, je izvedeno s pomočjo specifičnih organizacij; to so v bistvu konvencionalne major/minor organizacije katerim so dodane dodatne zanke. Ena od takšnih izvedb prikazuje slika 9 in vsebujejo: s-bitno major zanko, n s-bitnih minor zank, $(m - n)$ s-bitnih redundantnih (minor) zank, $(s + 2)$ -bitno kontrolno zanko (control loop), s-bitni vrstični marke (row marker loop) in $(s + 2)$ -bitni stolpični marker (column marker loop). Dodatne zanke (kontrolna, vrstični in stolpični marker) vršijo naslednji funkciji: odkrivanje defektnih minor zank in nadomeščanje le-teh z dobrimi redundantnimi ter hranjenje informacije o defektnosti minor zank, s tem pa krmiljeno branje in pisanje (kontrolna zanka); ter časovni nadzor nad branjem in pisanjem (vrstični in stolpični marker).



- D_{RM} - detektor vrstičnega markerja
- D_{CM} - detektor stolpičnega markerja
- T - prenos major/minor in obratno
- G - spontani generator
- B - brisanje
- T_{CL} - prenos major/kontrolna zanka
- G_{CL} - generator kontrolne zanke
- D_{CL} - detektor kontrolne zanke
- D - detektor

Slika 9

Pri začetnem testiranju pomnilniškega elementa vpišemo v vsako minor zanko na k-to mesto magnetni mehurček. Da bi si zagotovili zanesljiv vpis, po s korakih vpis ponovimo, tako, da ponovno vpišemo magnetni mehurček na k-to mesto

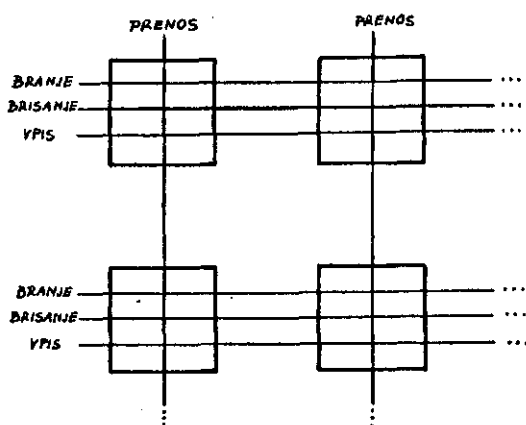
v vsako minor zanko. Po ponovni "zavrtitvi" minor zank prenesemo testni blok v major zanko. Vsaka morebitna napaka v minor zanki zadrži magnetni mehurček, tako, da na ustreznem mestu v testnem bloku ni prisoten magnetni mehurček. Testni blok, ki vsebuje informacijo o dobrih (mehurček prisoten) in slabih (mehurček odsoten) minor zankah, prenesemo v kontrolno zanko, kjer ga shranimo. (Ker je major zanka dolga $s + 2$ bitov, sta dve poziciji nezasedeni).

Pri vpisovanju informacije v minor zanke se vsak blok preko major zanke sukcesivno prenaša v minor zanke in sicer v časovnih intervalih $t_1 = s T_R$, $t_2 = (2s + 2) T_R$, $t_3 = (3s + 4) T_R, \dots$, katerih dolžina je $(s + 2) T_R$. Podrobneje si oglejmo dogajanje v časovnem intervalu $(0, t_1)$. V tem času generiramo n -bitni blok, $n \leq s$. j -ti bit tega bloka naj bi se vpisal v i -to minor zanko, za katero je, tako kot za vse ostale, v kontrolni zanki informacija o tem ali je zanka dobra ali ne. Predpostavimo, da i -ta minor zanka ni dobra, torej na i -tem mestu v kontrolni zanki ni mehurčka. Ker istočasno z generacijo j -tega bita bloka detektiramo i -ti bit kontrolne zanke, nam odsotnost i -tega mehurčka v kontrolni zanki povzroči, da se vsi predhodno generirani biti $1, 2, \dots, j-1$ in tudi j premaknejo, tako, da bo prišel j -ti bit v $i+1$ minor zanko ($j-1$ v $i+2$, itd. in prvi bit v $n+1$, to je prvo redundančno zanko). Če je slaba zgolj i -ta minor zanka, bodo biti $j+1, j+2, \dots, n$ prišli v prej namenjene minor zanke. V trenutku t_1 se bo n -bitni blok paralelno prenesel v minor zanke $1, 2, \dots, i-1, i+1, \dots, n+1$. V času od t_1 do $t_1 + 2T_R$ pa se kontrolna zanka zavrti za dva koraka in ponovno pride v začetno stanje. S tem je vpisovanje prvega bloka končano, vse nadaljne bloke pa vpisujemo na povsem enak način. Branje poteka na podoben način, le da kontrolna zanka "krmili" detektor.

Vrstični in stolpični marker vsebujeta le po dva mehurčka, ki se nahajata na sosednjih lokacijah. V vrstičnem markerju obideta mehurčka zanko v času sT_R (to je čas, ki je potreben za vpis - branje enega bloka), dočim je ta obhod v stolpičnem markerju daljši za $2T_R$ (to je čas, ki je potreben kontrolni zanki, da se postavi v začetno stanje). S pomočjo obeh markerjev ugotavljamo začetne pozicije vsake minor zanke in mesto bitov znotraj vsakega. Kontrolna zanka nam s pomočjo dodatnega generatorja G_{CL} omogoča spreminjanje dolžine bloka l , $1 \leq n$. To storimo tako, da v kontrolno zanko vnesemo l magnetnih mehurčkov.

Koincidenčna izbira podatkovnega bloka (coincidence selection of data block). Klasična major/minor zanka organizacija bi pri velikih kapacitetah zahtevala zelo dolge minor zanke in s tem dolge čase dostopa. Z razbitjem pomnilnika na več manjših enot, ki so tudi major/minor zanke organizirane s krajšimi minor zankami, smo skrajšali čase dostopa, mrtve čase in zmanjšali možnost pojava defektnih minor zank [2]. Pomnilniške enote razvrstimo v

matrično obliko, tako, da so vsi elementi v stolpcu povezani s funkcijo prenos (transfer), elementi v vrstici pa s funkcijami: branje, brisanje in vpis (slika 10). Za vpis (branje, oz. brisanje) je sposobna le pomnilniška enota, ki je v presečišču aktivirane prenosne in vpisno-



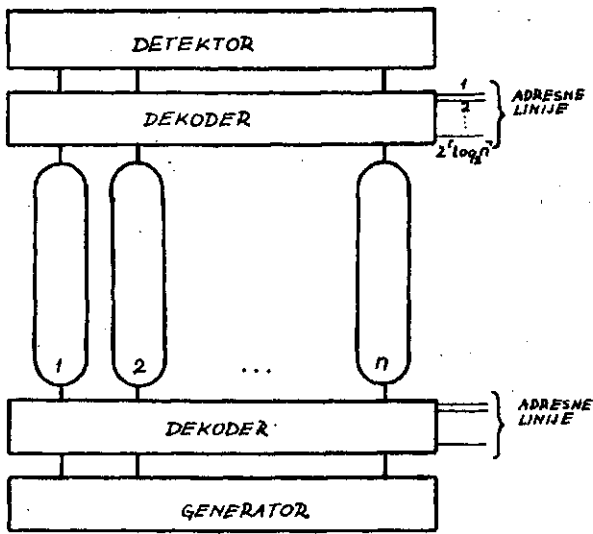
slika 10.

valne (bralne oziroma brisalne) linije, vse ostale pomnilniške enote pa niso aktivne. Takšna organizacija je nekoliko neugodna, zavrloj relativno velikega števila povezav med pomnilnikom in okolico ter bogatejše podporne opreme, kar pa vseeno ne omejuje njene uporabnosti. Zmislimo si 268 M-bitni mehurččni pomnilnik. Če bi bil organiziran kot enovita major/minor zanka struktura in bi vseboval npr. 16384 minor zank s po 16384 biti bi bil pri propagacijski frekvenci 100 KHz, čas dostopa približno 160 ms, potrebnih bi bilo 6 povezav in 4 podpora vezja. Pri organizaciji s koincidenčno izbiro podatkovnega bloka, kjer razdelimo pomnilnik na matriko 16×16 , ki vsebuje 256 pomnilniških enot od katerih ima vsaka enota 10^{24} minor zank s po 10^{24} biti; bi bil čas dostopa le 10 ms, potrebnih bi bilo 81 povezav in 64 podpornih vezij.

Dekoderji (decoders). Tudi pri tej organizaciji je informacija shranjena v množici premikalnih registrov - pomnilniških zank, dodana pa je še potrebna struktura imenovana dekodek (prav tako kot pomnilniška zanka izvedena s pomočjo permalojnih vzorcev in s fotolitografskimi postopki nanešenimi prevodnimi zankami), ki je tej organizaciji dala ime (slika 11) [4, 5].

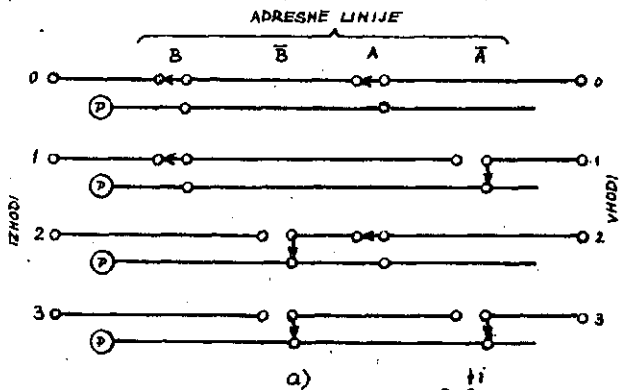
Blok informacije je shranjen v eni od s -bitnih pomnilniških zank, ki jo izberemo s pomočjo $2^{\lceil \log_2 n \rceil}$ tih adresnih linij. Pomnilniške zanke omogočajo nedestruktivno branje, saj vsebujejo elemente, ki magnetne mehurčke po dvajajo.

Če si pogloblje ogledamo sam dekodek, katerega shematični prikaz in pravilnostno tabelo podaja slika 12, vidimo, da ga sestavlja n dvojnih propagacijskih poti. Prva (zgornja) pot vodi mehurčke preko $\lceil \log_2 n \rceil$ (od skupno $2^{\lceil \log_2 n \rceil}$) stikal in jih v primeru, da so vsa stikala sklenjena pripelje od generatorja k pomnilniški zanki (vpis) ali od



slika 11.

pomnilniške zanke k detektorju (branje). Druga (spodnja) pot, pa služi kot ponor mehurčkov v primeru, da zgornja propagacijska pot ni sklenjena. Stikala so izvedena tako kot prikazuje slika 12 c. Tokovni impulz i v zanki, ki se nahaja v zgornji propagacijski poti povzroči, da magnetni mehurček preide v spodnjo propagacijsko pot kjer "izgine".



VHOD	ADRESNE LINIJE			
	B	\bar{B}	A	\bar{A}
0	0	1	0	1
1	0	1	1	0
2	1	0	0	1
3	1	0	1	0

slika 12.

Prednost opisane organizacije je ta, da je čas dostopa t_a , ki je enak:

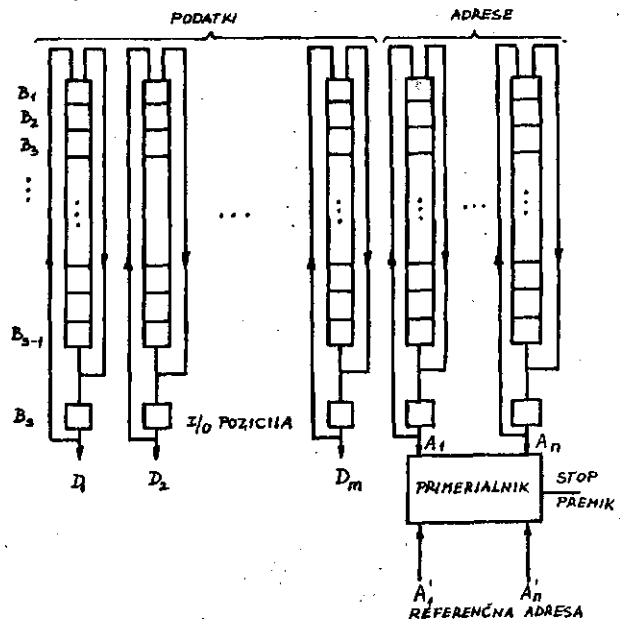
$$t_a = \left(\frac{3}{2} + 2^{\log_2 n}\right) T_R \quad (10)$$

mного krajši kot pri vseh do sedaj opisanih organizacijah; kot slabo stran pa navajamo razmeroma kompleksno

podporno opremo in veliko število priključkov.

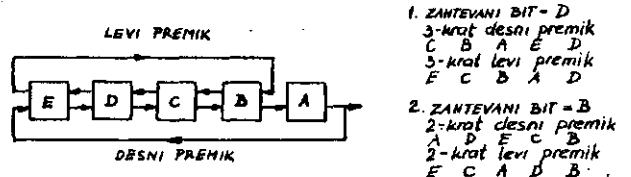
Dinamično urejeni podatkovni bloki. Za to organizacijo je značilno, da se vrstni red podatkovnih blokov, ki jih sestavlja m podatkovnih in n adresnih bitov in, ki so razporejeni vzdolž $(m + n)$ posebno oblikovanih premikalnih registrov neprestano spreminja (slika 13). Do podatkovnih blokov, ki kot rečeno, nosijo svojo addresso, pridemo po naslednjem algoritmu [3]:

1. Adreso željenega podatkovnega bloka (referenčna adresa) pripeljemo na primerjalnik.
2. Vse premikalne registre, tako podatkovne kot adresne premikamo v desno dokler ni izpolnjen pogoj $A_i = A'_i$ za vsak $i = 1, 2, \dots, n$ (referenčna adresa enaka adresi na I/o poziciji). Istočasno poseben števec beleži število premikov x .
3. Izbrani blok zadržimo na I/o pozicijah; in ga lahko preberemo ali prepisemo, vse ostale bloke pa premikamo za x korakov v levo.



slika 13.

Izvajanje algoritma za primer 5 bitnega registra je prikazan na sliki 14. Začetno stanje ABCD preide po izbranih bitih D in B v stanje BDACE. Premikalni registri so izvedeni tako kot pri ostalih organizacijah; magnetni mehurček pa je mogoče ujeti na I/o poziciji bodisi s posebno oblikovanimi permalojnimi vzorci in spreminjanjem smeri rotirajočega magnetnega polja \vec{H}_R (slika 15 a); ali



1. ZANTEVANI BIT - D
3-krat desni premik
C B A E D
5-krat levi premik
E C B A D
2. ZANTEVANI BIT - B
2-krat desni premik
A D E C B
2-krat levi premik
E C A D B

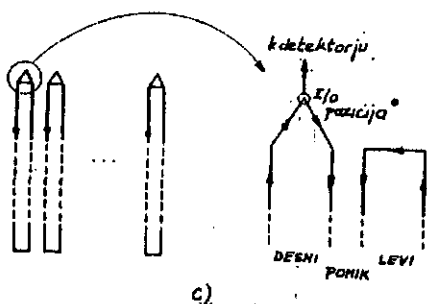
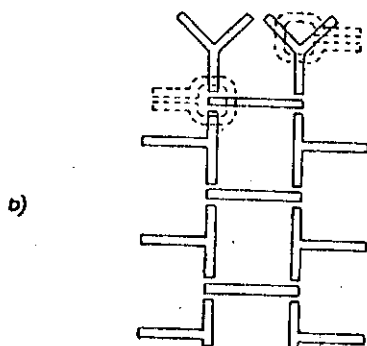
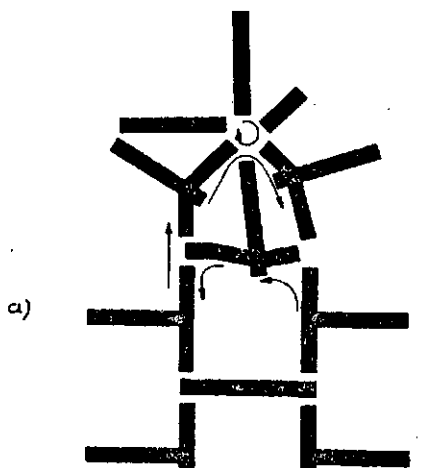
slika 14.

s prav tako posebno oblikovanimi permalojnimi vzorci in tokovnimi zankami (smer polja \vec{H}_R je konstantna) (slika 15 b).

Z ustrežno strukturo te organizacije, to je velikim številom kratkih premikalnih registrov se dosežejo povprečni časi dostopa, ki so enaki:

$$\bar{t}_a \approx T_R \quad (11)$$

kar bi pomenilo pri propagacijski frekvenci 100 KHz povprečni čas dostopa 0,01 mS.



slika 15

elementa in njegove organizacijske strukture, so primerljivi z diskovnimi sistemi, vendar nudijo mnogo krajše čase dostopa do podatkov, to pa že vnaša kvalitativne posega v načrtovanju celotnega računalniškega sistema. Če je mehurčni pomnilnik pridružen ostalim, od njega počasnejšim perifernim pomnilnikom, že sama razlika v časih dostopa vnaša v sistemu dodatne enote, potrebne hitre kanale, kompleksne algoritme, s tem pa je postavljena pod vprašaj prenosna funkcija sistema. Spričo tega je toliko bolj opravičljiva uporaba mehurčnih pomnilnikov in zamenjava diskovnih in ostalih počasnejših perifernih pomnilnikov z njimi; in sicer v:

- Sistemih z dodeljevanjem časa in multiprocesorskih sistemih.
Če je centralni pomnilnik organiziran z dinamično urejenimi podatkovnimi bloki lahko hrani sistemske programe kot so: prevajalniki, zbirniki, nalagalniki in uporabniški podatki. Dodeljevanje opravil je bistveno poenostavljeno, če se opravila dodeljujejo procesorjem v centralnem pomnilniku. Manjkrat pride do konfliktov med procesorji, operacijski sistem je enostavnejši, večja je izoliranost in manjša možnost nastopa napak v podatkih.
- Samostojnih miniračunalniških sistemih z velikim pomnilnikom.
V to skupino sodijo tako imenovani poslovni računalniki, ki ni nujno, da so hitri, pač pa zmožni izvajati veliko število programov.
- V sistemih z vmesnimi pomnilniki, virtualnimi pomnilniki in pomnilniki za shranjevanje datotek.
Mehurčni pomnilnik je dovolj hiter (obenem pa lahko hrani veliko količino podatkov), da ga uporabimo kot vmesni pomnilnik med računalnikom in počasnejšo periferyljo. Pri multi-programiranih računalnikih pa imamo opravka z dinamičnim dodeljevanjem pomnilnika in s tem se kaže potreba po načrtovanju virtualnega pomnilnika. Mehurčni pomnilniki v tej aplikaciji uspešno zadostijo potrebnim zahtevam. Kapaciteta virtualnega dela pomnilnika je lahko dovolj velika pri čemer pa so časi dostopa tako kratki, da poteka preslikava množice virtualnih adres v množico fizikalnih adres kar najhitreje. Možnost oblikovanja strukture znotraj samega pomnilniškega elementa predstavlja važno postavko pri načrtovanju arhitekture virtualnega pomnilnika, saj bistveno skrajša tabelo strani (page table) in poenostvi paginiranje. Posebno velike hitrosti dostopa pa dosežemo, če so v mehurčnih pomnilniških elementih shranjene še majhne ali srednje velike datoteke.

Načrtovalca programske opreme. V običajnih aplikacijah tretirajo načrtovalci programske opreme mehurčne pomnilnike enako kot diskovne in tračne enote. To je tudi razumljivo, saj je možno potegniti paralelo med shranjevanjem podatkov pri major/minor zračni organizaciji in shranjevanjem podatkov na diskih; kakor tudi pri organizaciji z

3. RAZMISLJANJE

Načrtovalca računalniških sistemov. Mehurčni pomnilniki kot masovni pomnilniki, gledano na nivoju pomnilniškega

dolgim premikalnim registrom in magnetnimi trakovi. Vendar pa sama organizacijska struktura mehurčnih pomnilnikov nudi mnogo več, saj omogoča programirano časovno krmiljenje pomnilniškega sistema. Tako se pojavijo dodatne kvalitete, kot so: dvo ali več hitrostno delovanje pomnilnika, trenutni start-stop in dvosmerni dostop do podatkov, ki nam odpirajo možnosti snovanja specifičnih algoritmov in programov.

- Dvo (ali več) hitrostno delovanje pomnilnika (dual speed memory).

Programirano krmiljenje hitrosti delovanja pomnilnika je kvaliteta, ki jo lahko s pridom uporabimo pri snovanju algoritmov za hitro Fourierjevo in Hadamardovo transformacijo, sortiranje podatkov, transponiranje matrik, itd.

- Trenutni start-stop (instantaneous start-stop memory). Trenutni start-stop je kvaliteta mehurčnega pomnilnika, ki omogoča simulacijo delovanja magnetnih trakov in s tem uporabe algoritmov, ki so specifični za magnetne trakove. Takšen je npr. Pavkovičev algoritem za reševanje velikih sistemov linearnih enačb. Seveda pa je izvajanje algoritmov mnogo hitrejšo. Takoimenovan ciklični start-stop pomnilnik je izveden v obliki velikega števila neodvisnih premikalnih registrov (ločeni časovni nadzori) in omogoča konstruiranje zelo učinkovitih algoritmov.

- Dvosmerni dostop do podatkov (bidirectional memory).

Smer pretoka podatkov v mehurčnem pomnilniku spremenimo tako, da obrnemo smer vrtenja magnetnega polja H_R . Dvosmerni start-stop mehurčni pomnilniki predstavljajo pomnilniške sklade kot so: odlagalni sklad, FIFO in FILO sklad.

4. ZAKLJUČEK

Pričujoči članek je poizkušal predstaviti nekatere osnovne organizacijske strukture magnetnih mehurčnih pomnilnikov in njihov nadaljni razvoj. Razvoj organizacijskih struktur širi tudi spekter njihovih aplikacij, tako da le-ti niso uporabljivi le kot enostavni masovni pomnilniki, temveč se je področje uporabe razširilo; npr. na področje upravljalnih sistemov podatkovne baze (data base management systems), kjer mehurčni pomnilniki služijo kot nosilci podatkovne osnove (intelligent magnetic bubble memories) [7,8].

5. LITERATURA

- [1] D.C.MARKHAM: Electronic Engineering, pp. 85-99, June 1979
- [2] H.CHANG: Transaction on Mag., pp. 564-569, September 1972 (1972 INTERMAG Conference, Kyoto, Japan)
- [3] P.I.BONYHARD, T.J.NELSON: The Bell System Tech. Jour. Vol.52, No.3, pp. 307-317, March 1973
- [4] H.CHANG, J.FOX, D.LU, L.L.ROSIER: IEEE Trans. on Mag. Vol. MAG-8, No.2, pp. 214-222, June 1972
- [5] T.T.CHEN, T.R.OEFFINGER, I.S.GERGIS: IEEE Trans. on Mag., Vol. MAG-12, No.6, pp. 630-632, Nov. 1976
- [6] H.CHANG: "Magnetic-Bubble Memory Technology", Electrical Engineering and Electronics/6, Marcel Dekker, inc., New York - Basel, 1978
- [7] H.CHANG, A.NIGAM: IEEE Trans. on Mag., Vol. MAG-14, No.6, pp. 1123-1128, November 1978
- [8] J.W.S.LIU, M.JINO: IEEE Trans. on Computers, Vol.C-28, No.12, pp.888-906, December 1979
- [9] J.ŠILC, B.MIHOVILOVIČ, P.KOLBEZEN: Informatica, letnik 4, 1980 - št. 4
- [10] A.H.BOBECK, E.DELLA TORE: "Magnetic Bubble", North-Holland Publishing Company - Amsterdam, 1975
- [11] C.K.WONG, P.C.YUE: IBM J. Res- Dev. 20, pp. 576-581, Nov. 1976



Simpozij in seminarji Informatica '82
Ljubljana, 10.—14. maja 1982

Simpozij
16. jugoslovanski mednarodni simpozij za računalniško tehnologijo in probleme informatike
Ljubljana, 10.—14. maja 1982

Seminarji
izbrana poglavja računalniških znanosti
Ljubljana, 10.—14. maja 1982

Razstava
mednarodna razstava računalniške tehnologije in literature
Ljubljana, 10.—14. maja 1982

Roki

1. avgust 1981	zadnji rok za sprejem formularja s prijavo in 2 izvodov razširjenega povzetka
1. oktober 1981	pošiljanje rezultatov recenzije in avtorskega kompleta
1. februar 1982	zadnji rok za sprejem končnega teksta prispevka

Symposium and Seminars Informatica '82
Ljubljana, May 10—14, 1982

Symposium
16th Yugoslav International Symposium on Computer Technology and Problems of Informatics
Ljubljana, May 10—14, 1982

Seminars
Selected Topics in Computer Science
Ljubljana, May 10—14, 1982

Exhibition
International Exhibition of Computer Technology and Literature
Ljubljana, May 10—14, 1982

Deadlines

August 1, 1981	submission of the application form and 2 copies of the extended summary.
October 1, 1981	mailing out of the summary reviews and author kits.
February 1, 1982	submission of the full text of contribution