

1988 / 9  
cena 12Kčs

# Mikro



# báze

technický  
zpravodaj  
svazarmu  
pro zájemce o  
mikropočítače

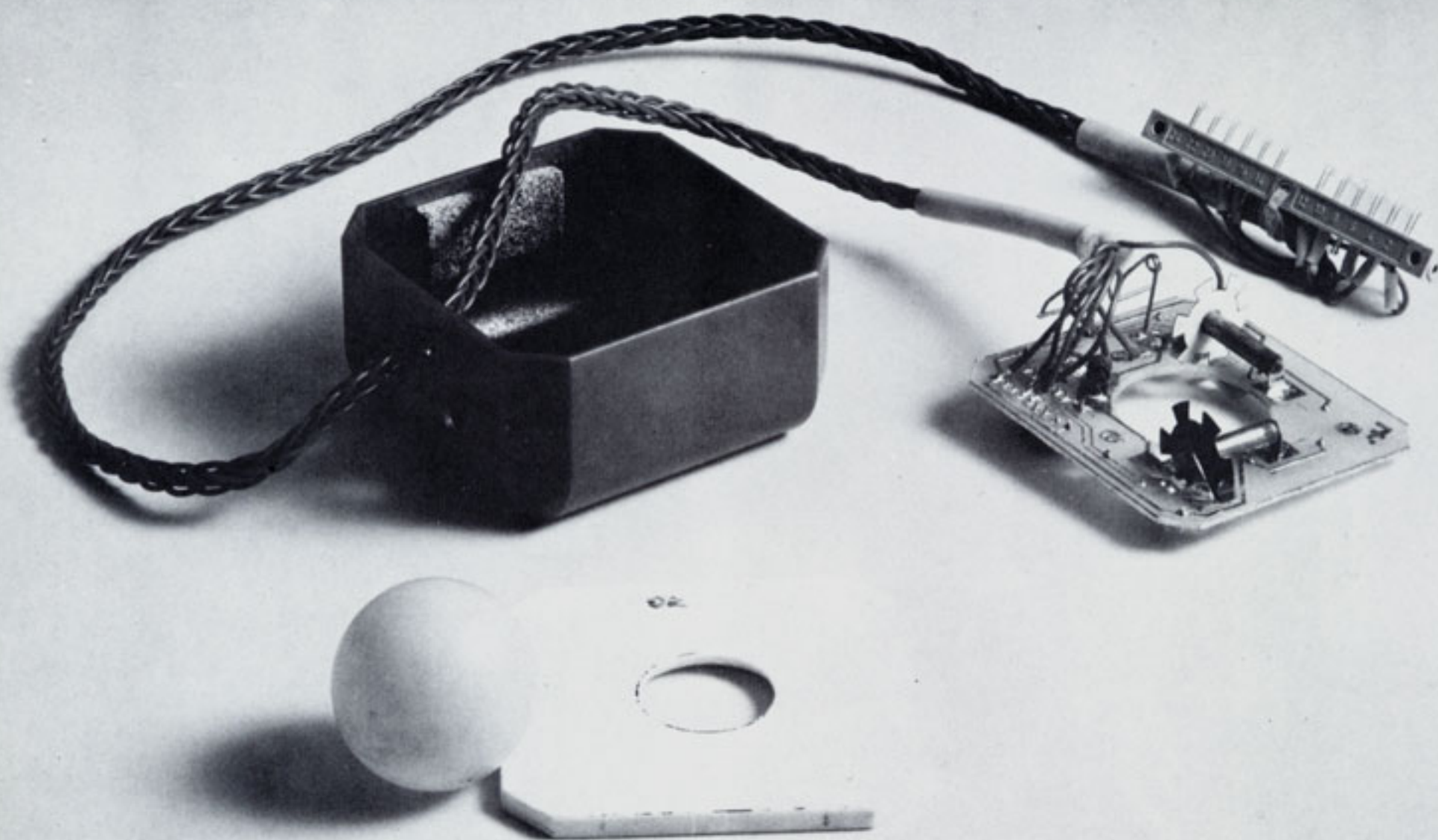




Dálkový kurs stavby a využití elektronické myši pro počítače PMD 85 a ZX Spectrum má ještě k datu 31. 12. 1988 asi 300 volných míst. Tři stovky stovebnic a sad dokumentace tedy ještě dávají šanci těm, kteří se po vyhlášení v Amatérském radiu a Technickém magazínu dosud nerozhodli.

Přihlašovací materiály si vyžádejte na adrese:

602. ZO Svazarmu  
Wintrova 8  
160 41 Praha 6







**báze** technický  
zpravodaj  
svazarmu  
pro zájemce  
o mikropočítače

## OBSAH

Staré problémy ve světle nového ....	1
Umělá? Neumělá? AIAIAI! (dokončení) ..	2
Filozof.aspekty stroj.myšlení (2) ..	4
Počítačová scifi (3) .....	7
Prostředky pro zápis algoritmů .....	10
Programátor EPROM 8708 .....	18
Tiskárna D-100 a ZX Spectrum .....	21
Kalkulátor ZX Spectra (dokončení) ..	24
Amstrad CPC - systémové rutiny (3) ..	26
Metakomunikace (3) .....	28
Ceny Atari a Amigy v Británii .....	30
Informace střediska VTI .....	31
Nabídka Mikrobáze .....	32



Technický zpravodaj Svazarmu pro zájemce o mikropočítače. Vydává 602. ZO Svazarmu ve spolupráci s redakcí časopisu Amatérské radio. Povoleno ÚVTEI pod evidenčním číslem 87 007. Zodpovědný redaktor ing. J. Klabal, sestavil ing. A. Myslík. Redakční rada: P. Horský, ing. J. Klabal, ing. P. Kratochvíl, J. Kroupa, ing. A. Myslík, ing. J. Truxa. Ročně vyjde 10 čísel, cena výtisku 12 Kčs podle ČCÚ a SCÚ č.1030/202/86. Roční předplatné 120 Kčs. Objednávky přijímá a zpravodaj rozšiřuje 602. ZO Svazarmu, Wintrova 8, 160 41 Praha 6.



# 602.ZO

&

Amatérské  
**RADIO**

# STARÉ PROBLÉMY VE SVĚTLE NOVÉHO MYŠLENÍ

Sovětský vládní deník Izvestija přinesl dne 31.7.88 obsáhlý rozhovor s prezidentem Akademie věd Ukrajiny, Borisem Patonem. Uvádíme jeho část, týkající se využití výpočetní techniky.

B.P.: Naše nové myšlení potřebuje široké obzory, mnoho operativně získaných vědomostí o světových zkušenostech a jejich využití. Rozhodnutí se musejí přijímat na základě objektivního zpracování informací - pokud možno úplných, nikoli na principu "zdá se mi, že..."

Otázka: Když mluvíte o získávání úplné informace, zahrnujete sem i bohatství knihoven, možnosti jeho využití?

B.P.: Rád bych. Ale stav, v jakém knihovny jsou, to ne vždy dovoluje. Situace v těchto rezervoárech informací je bolestnou otázkou. Protahování jejího řešení je zdrojem nepříjemných následků. Rozvinuté země už dávno změnilly tvář svých knihoven, aby mohly v nich obsažené informace intenzivně využívat.

Největší knihovna na světě - knihovna Kongresu USA - už osm let pracuje pouze s elektronickým katalogem. Stala se součástí řídicího systému státu. Abonent knihovny - od kongresmana po řadového uživatele - může informačními kanály objednat informaci, která ho zajímá, a v mžiku obdrží zobrazení potřebného textu na monitoru.

V převodu knihoven na novou úroveň jejich činnosti jsme bohužel silně zaostali. Tady jsme teprve na začátku nasazování výpočetní techniky. Katalogy jsou na kartičkách, a ještě k tomu jsou neúplné.

V poměru k západním databankám jsme v elektronickém stínu. Chybějící vybavení knihoven současnou technikou nás odděluje od knih umístěných v našich knihovnách, nemáme o nich vyčerpávající informace. Být v systému světové kultury a nevyužívat její novinky, znamená předem a navždy se odsuzovat k zaostávání.

Prověřovali jsme, jak jsou využívány fondy knihoven vědecko-výzkumných institutů. Ukázalo se, že vědci si knihy nezamlouvají. Prázdné kartičky jsou dokonce i u akademiků. Položili jsme si otázku - je možné, že taková situace je jen v našich institutech? Ukázalo se, že v druhých akademiích je to stejné. Částečně to lze vysvětlit mohutnou lavinou informací, jaké se v posledních letech na člověka valí. Náš mozek prostě není s to všechno přijmout a zpracovat.

Dnešní vědec i politik potřebuje nejen několik knih, které blaží duši, ale i deriváty konkrétní informace o konkrétních problémech. To je nutný předpoklad včasného přijetí skutečně vědecky podložených rozhodnutí. A to je i důvod, pro který je zájem o elektronické databanky větší, než o referativní časopisy.

Jak situaci napravit? Určitě ne rozšířením personálu knihoven pro zpracovávání analytických přehledů, ale jejich vybavením současnou výpočetní technikou, umožňující získávat informace na kvalitativně nové úrovni. V naší republice se silami nadšenců připravuje elektronický katalog pěti největších knihoven. Ale to je málo. Neprodleně se musí rozvinout celostátní informační elektronický systém. Zatím je u nás krajně málo databank. Nelze litovat prostředků na jejich zakládání a ihned vstupovat do kontaktů s mezinárodními informačními bankami. Bez operativního přístupu ke světovým rezervoárům nejnovějších dat se nikam nedostaneme.

U nás chybí přesná koncepce využití informatiky. Nutně musíme rozpracovat národní program - podobně, jako byl vypracován program potravinářský, energetický a jako se dnes připravuje program o nových materiálech. Bez toho budeme řešit problémy přestavby jen velmi obtížně.



# HOVORY O PROGRAMOVÁNÍ



## UMĚLÁ? NEUMĚLÁ? UMĚLÁ?

## NEUMĚLÁ? AIAIAI!

(Dokončení rozhovoru s Ivanem M. Havlem o umělé inteligenci)

„To mě přivádí k tomu, co se v posledních letech stalo šlágrelem a co může radikálně změnit naše představy o umělé inteligenci přesně tím směrem, kterým se ubírala Hofstadterova fantazie (mimočodem - jeho kniha bývá nyní citována i ve velmi odborně zaměřených vědeckých člancích). Mám na mysli modely neuronových sítí. Někteří lidé se jimi zabývali už dávno, ale byli zatlačováni do pozadí hlavním proudem symbolické umělé inteligence, programované, respektive programovatelné na klasických von Neumannových počítačích (centrální jednotka, paměť). Neuronové sítě jsou složeny z hrozně jednoduchých prvků, kterých je ale nesmírné množství a které spolu mohou být složitě propojeny - třeba i každý prvek s každým. Ty prvky samy lze popsat velmi prostě. Jsou něco jako prahové obvody či perceptrony; každý je buď aktivní nebo pasivní. Aktivní je jen tehdy, když součet vlivů všech ostatních prvků, s nimiž je propojen a které jsou v tu chvíli aktivní, je nad určitou prahovou hodnotou.

A co je podstatné - tu síť lze učit. Propojení prvků - tzv. vazby či konexe - se vzájemně liší svou průchodností, zvanou váha. Ta může být i záporná, což odpovídá negativnímu vlivu, inhibici. Jednotlivé váhy se mohou zvolna měnit v čase v závislosti na chování okolních prvků. Lze říci, že velikosti těchto vah reprezentují dlouhodobou paměť sítě na to, co se v ní dříve dělo. Učení spočívá v ovlivňování těchto vah. Nejprve vnější stimulací vnutíme nějaké chování vybraných prvků - způsobíme, aby byly aktivní. Jak se v důsledku toho změní konkrétní váhy a které, lze těžko odhadnout - ale není nezbytné, abychom to věděli. Na příští stimulaci podobného typu je síť už svým způsobem připravena a bude se chovat obdobně jako předtím. Dovede takto přinejmenším rozlišovat různé druhy vstupů.

U takové sítě nikdy už nikdo neví, co jaký prvek v tom kterém momentu v součinnosti s ostatními dělá. Dalo by se to přirovnat k holistickému záznamu. Síť dovede rozlišovat víc věcí nezávisle - záleží na vstupní konfiguraci. Podle toho, jaká je, síť reaguje svým určitým způsobem, je to v ní holisticky rozptýleno. Někdy se tomu říká konekcionismus, někdy paralelní zpracování distribuované informace, jindy neuronové modelování či neuronové sítě atd. To podle toho, kdo jak celou věc pojímá, jaký k ní volí přístup.

„Jak je v sítích rozlišen hardware a software?“

„Takové rozlišení tu moc neseď. Síť je vlastně dlouhodobá asociativní paměť. Hardware, chcete-li, zde hraje roli jen jakéhosi média, substrátu, který je nabit vnitřní ochotou vytvářet a měnit ony vzájemné vazby mezi prvky, jimiž je pak řízena jejich aktivita. Síť sama o sobě není schopna ničeho. Teprve dlouhým provozováním toho substrátu se některé vazby preferují, jiné potlačují, substrát se strukturuje, nabývá vlastní schopnosti na vyšší úrovni, vytváří si obrazce aktivity. Kdybychom na každý prvek napojili žárovku, viděli bychom různé obrazce, které by se různě dynamicky měnily. Jak, to nikdo nedovede na té nižší úrovni předvídat, protože to vzhledem ke složitosti není zvládnutelné - víme jen, že to závisí na těch vahách. Rozložení vah hraje svým způsobem roli programu - ale s tím, že se může zvolna měnit.“

Programátor tu má spíš roli učitele, než onoho klasického kodéra, který může do důsledků analyzovat, co dělá. Tady jsou dvě fáze - napřed konstrukce sítě a pak její učení zkušeností či cvikem.“

„Představme si běžně programujícího člověka, který vytváří lineární programy. Teď před něj postavíte neuronovou síť. Co si s ní má počít, aby dělala, co potřebuje?“

„Základem je stimulace zvenčí. Dejme tomu, že k síti budou připojena čidla citlivá na světlo. Prvky té sítě tu však nejsou jen prostými receptory, ale jsou schopny i vlastní aktivity díky oněm vazbám. Každý prvek má vstupy buď z některých nebo ze všech okolních prvků a kromě toho ještě vstup zvenku. Tak jej mohu nastavit na určitou hodnotu. Už z toho vyplývá, že tu nejde o symbolické programování. Takováto inteligence se nazývá konekcionistická - odtud název konekcionismus. Jím se vyjadřuje, že vše je skryto ve zmíněných vazbách - konexích. Programátor sítě nemá nijak těžkou pozici. V podstatě jde o to, aby síti dostatečně dlouhou dobu předkládal určité vzorky chování.

Dejme tomu, že budeme chtít, aby síť rozlišovala písmena abecedy nezávisle na tom, jak jsou velká, tenká, silná, položená, tiskací, psací, na tom, kdo je napsal apod. Na vstupu budeme mít zařízení s jemným rozlišováním, něco jako sítnici oka. Toto zařízení bude pochopitelně snímat paralelně. Při předložení písmene A učitel musí zároveň stimulovat výstup, který bude odpovídat písmenu A. Výstup může být jakýkoli - grafický, zvukový... Když bude



síti předložen určitý počet tvarů jednotlivých písmen, vytvoří si určité vazby, vzorce, jimiž se dokáže orientovat v předloženém tištěném či psaném textu. Klasické rozpoznávače písmen mohou rozpoznávat jen určité typy písma, jejichž bodovou skladbu mají poměrně tvrdě zakódovanou v paměti. Neuronová síť přijatý tvar rozkládá do svých vnitřních vah, což je obrovský počet proměnných. Vyučená síť dokáže správně identifikovat i tvary, které nikdy předtím "neviděla". Tak může přečíst i rukopisy, které jí během výuky nebyly předloženy. A navíc se v průběhu identifikace dosud nepoznaných rukopisů dál učí, stává se stále umnější. Pravá inteligence to ještě není, je jí to však bližší, než tvrdé algoritmy tradiční AI.

Síť funguje v čase, cosi se v ní děje. To není tak, že dám něco na vstup a ono mi to okamžitě dá výstup. V našem příkladu s písmeny jde o případ relaxační sítě, která se vstupním podnětem někdy rychle, někdy pomaleji mění, až se ustálí v jednom stavu, kterému se říká atraktor, "přitažlivce". V ideálním případě je celá síť stabilní jen v těch 26 stavech pro 26 písmen abecedy. Ve všech ostatních stavech je nestabilní. Asi jako koule na obrovské hrbolaté ploše, v níž by bylo jen 26 jamek. To je jedna z možností, jak si představit chování (nikoli strukturu!) neuronové sítě. Každý totální stav sítě je jedním bodem té hrbolaté plochy.

Cílené chování některých sítí lze popsat energetickou funkcí - síť se snaží zaujmout stav nejnižší energie, což je stav, kdy se koule dostane do "nejhlubší jamky". Uvázne-li v jiné, prostě zatřese celou plochou, aby koule mohla z jamky vyskočit a najít si jinou, energeticky vhodnější. Některé z těchto metod zpočátku třesou silně a postupně třas zvolňují, aby koule nevypadla z lepšího stavu do horšího. Tomu se říká žihání, což je zahřátí a pak pomalé ochlazování materiálu. Každý prvek se přitom chová trochu náhodně. To třesení, ten stupeň náhodnosti je ona teplota žihání, rozkmit sítě. Ve chvíli, kdy to zcela vychladím, by koule měla být v absolutním minimu. Takový postup je výhodný třeba při řešení úloh ze statistické fyziky.

Z hlediska umělé inteligence jsou neuronové sítě zajímavé tím, že se problém přesouvá na nižší úroveň dějů, kde nelze mluvit o nějaké inteligenci. Funkce, které definují jejich lokální chování, jsou velmi primitivní. Ale na vyšší úrovni - když se na to podíváme jako na celek - se objevuje jiné, mnohem zajímavější chování. Jsou to jisté zárodky inteligentní schopnosti.

Kdybychom šli ještě výš, mohli bychom mluvit o symbolické úrovni - některým obrazcům aktivit již lze přisoudit sémantický význam. Aby to bylo schopno myslet, musí to být schopno žít samostatně. Budu to třeba krmit ne jedním, ale několika proměnlivými vstupy současně. Jako když my vnímáme okolí svými smysly, jejichž prostřednictvím jsme pod vlivem stále nových informací. Nemáme čas se zastavit, zapnout rozpoznávač a říct - aha, tak tohle je to a to - a pak se zase rozjet. Jedeme pořád. I síť neustále přechází ze stavu do stavu, pořád se v ní něco dost složitě vrbí a mele. A teď jde o to si představit, že ono stále vrbení a mletí v ní zanechává nějaké dlouhodobé stopy. A že celá ta dynamika má charakter jakéhosi myšlení.

V tomto pohledu nás částečně podporuje neurofyziologie. Ne v detailech - neurony fungují jinak, nemají symetrické spoje, mají mnoho dalších, složitějších funkcí, mozek není tak homogenní, jak se zde předpokládá atd. Ale jedno o mozku víme - je tam hrozná spousta malinkatých věcí, které jsou vzájemně propojeny ve velmi složité síti. Tato propojení nejsou stálá - síla vazeb mezi jednotlivými neurony se pomalu mění. Když naproti tomu neurofyziolog přiloží na hlavu člo-

věka elektrody, vidí rychle se měnící aktivity. Víme, že myšlení je provázáno děním v mozku na neurofyziologické úrovni. Ale velmi málo víme, co je příčinou čeho - třeba jestli myšlení je příčinou toho dění či naopak. To je opět filozofický problém, zvaný "mind-body problem" či psychofyzický problém - prastará otázka vztahu duše a těla.

Neuronové sítě vdechly umělé inteligenci nový život. Tradiční symbolická představa myšlení bude nahrazována konekcionistickou s jejím důrazem na spontánnost. Lze uvažovat o vytvoření rozsáhlých sítí, na které můžeme hledět jako na složité dynamické systémy, umožňující procesy, které nejsou pod naší detailní kontrolou, které však můžeme spouštět a zastavovat. Můžeme docílit, aby vznikly, ale už přesně nevíme, jak se niterně chovají, např. jak a kde se co učí. Podobně to vlastně neví ani učitel o svém žáku či trenér o svém svěřenci.

Kolem toho jsou obrovské diskuse mezi zastánci tradiční umělé inteligence a konekcionismu; filozofové diskutují o tom, co nového to přináší onomu odvěkému mind-body problému - zda lze mysl chápat jako emergentní chování fyzikálního substrátu (těla), či zda je tam ještě něco dalšího...

**"Můžeme ještě o neuronové síti "s výučním listem" mluvit jako o artefaktu?"**

"Ano a ne. Termín artefakt se hodí na jednoduché předměty k jednoduchému účelu. Nůž, bicykl, dalekohled... Je však kroužek cigaretového dýmu artefaktem? Či kruhy na vodní hladině, které jsem způsobil vhozením kamene? Obrazec z těl spartakiádních cvičenců? Panika davu? Inflace?"

Myslím, že je to trochu otázka úrovně popisu. Co je artefakt na jedné úrovni, může být něco přirozeného nebo přírodního na úrovni jiné. Neuronová síť, řekl bych, je artefaktem na nižší úrovni, jevy na vyšší úrovni - aspoň ty, o nichž nevíme, jak budou vypadat - jsou emergentní. Je to umělý substrát s neumělou inteligencí."

**"Jaká je dolní hranice počtu prvků, respektive spojů neuronové sítě, aby mohla pracovat jako asociativní paměť? A byla už nějaká taková někde postavena?"**

"Co do počtu prvků záleží na tom, co od toho čekáme. U asociativní paměti např. na tom, kolik různých věcí si má pamatovat současně. Experimenty s jedním dosti známým modelem ukazují, že síť musí mít nejméně šestkrát tolik prvků, než kolik věcí má rozlišovat bez velkého počtu chyb. To je však jen kombinatorika. Hádám, že ke vzniku emergentních jevů je nějaký milión malým počtem. Mozek, který je věru dost dobře vyladěn ke svým účelům, má 10 až 100 miliard neuronů."

Nadějnost konstrukce takových sítí se stále zvětšuje. Pokud vím, všechny konkrétní experimenty se provádějí pomocí simulace sítě na obyčejných digitálních počítačích - sekvenčních i paralelních. Systémů je celá řada, nějaké čipy jsou prý již i na trhu. Ale to vše asi bude pod tou hranicí miliónu prvků. Probíhají i experimenty s využitím prostorových optických médií, které by pracovaly na holografickém principu. Dosud vždy měla technologie náskok před teorií, tak uvidíme."

**"Musí se každý konekcionistický počítač učit něčemu od píky, nebo je zkušenost mezi nimi převoditelná?"**

"Zajímavá otázka. Pokud vím, neuvažovalo se o převoditelné zkušenosti. Kupodivu jsem o ničem takovém nikde nečetl. Principiálně by to snad mělo jít. Určitě to jde v případech, kdy je neuronový počítač simulován na klasickém sériovém počítači, v němž je každému prvku i každé vazbě věnováno konkrétní paměťové místo. Pak není problém to nahrát na disk a přenést do jiného počítače, který bude simulovat jinou síť. Konekční paměť je v podstatě matice čísel, která udává váhy všech vazeb. Tato matice je sice velikánská, ale konec-



konců je to jen matice čísel, takže ji mohu přenést. Tím bych však získal jen kopii."

"A co doplňování zkušenosti jednoho počítače zkušeností jiného jejich přátelskou výměnou?"

"Kdybychom něco takového chtěli provést, myslím, že bychom narazili. Nevíme, jak a kde jsou rozmístěny sémantické prvky, jak se která váha na čem podílí. Určitě by taková výměna zkušeností nebyla možná běžným kopírovacím způsobem. Nakonec kdyby něco takového bylo možné u lidí, pak dva lidé s identickou vnitřní zkušeností by si ani neměli co říct."

"Já si se sebou povídám docela rád."

"Jistě. Jenže vaše povídací vědomí neobsahuje plnou identitu vás celého. To mi připomnělo poněkud scestnou představu Norberta Wienera, že člověka lze jednoduše - biologicky i duševně - rozepsat na bity, odtelegrafovat na jinou planetu a tam ho zase dát dohromady. Od té doby spisovatelé

scifi přemísťují celé kolonie bytostí vesmírným prostorem bez nejmenší úhony."

"Napadlo mne, že i kdyby se podařilo vyrovnat zkušenost dvou lidí či inteligentních Golémů, od toho momentu by se v každém z nich zase odvíjela jinak. Osudový multipluralismus?"

"Dokonce neomezeně mnoha způsoby."

"Co připravujete na SOFSEM 88?"

"V podstatě to, co jsem tu velmi improvizovaně vykládal."

"A čím se zabýváte nejintenzivněji?"

"Nevím, jestli se dá mluvit o nějaké intenzitě, ale zajímá mě problém mind-body, tedy problém mentálního a tělesného. A v současnosti samozřejmě konekcionismus."

"Přeji jen samé inteligentní konexe a děkuji za rozhovor."

Rozmlouval -elzet-

---

---

# FILOSOFICKÉ ASPEKTY STROJOVÉHO MYŠLENÍ

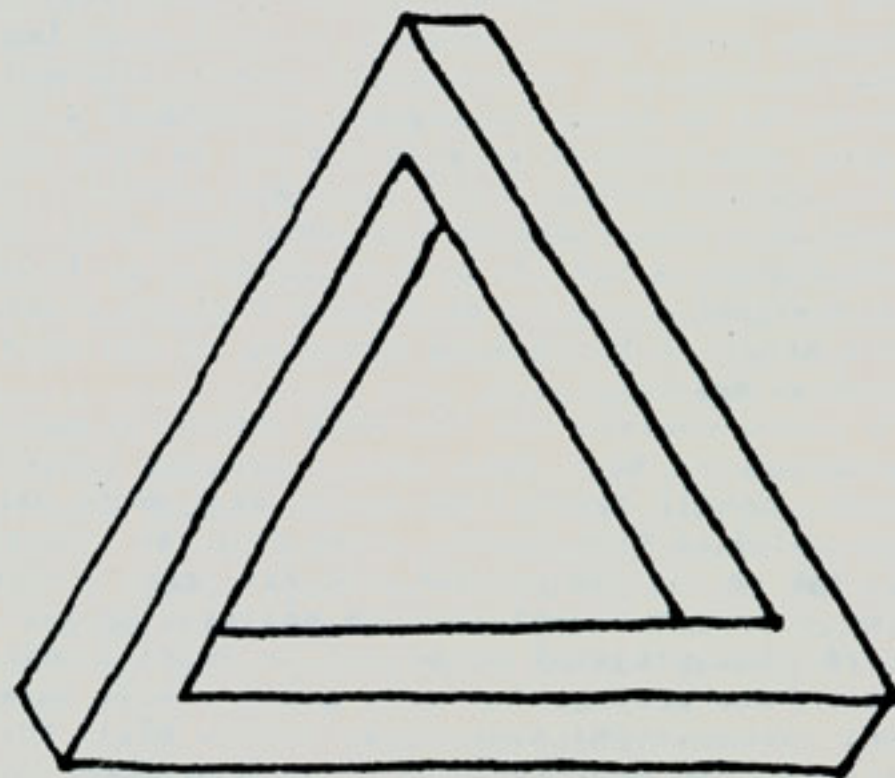
(2)

---

## Podivné zacyklení a propletená hierarchie

---

Na obr. 1 je kresba trojúhelníka, který má všechny tři úhly pravé. I když každý úsek kresby dovedeme bez obtíží prostorově interpretovat, něco přece nehraje - obejdeme-li s naší prostorovou interpretací postupně celý trojúhelník, vrátíme se do téhož místa - ale jaksi překroucení. Správně jsme se totiž měli pohybovat po šroubovici dle obr. 2. Podobnou situaci, které Hofstadter říká podivné zacyklení, najdeme na známém Escherově obrázku Vodopád (obr. 3). Podivné zacyklení nemusí vzniknout jen deformací prostoru. Na obr. 4 je starý alchymistický symbol veškerenstva, had Uroboros - polykač vlastního ocasu. Není ani tak těžké si představit, že existuje kanibalismus mezi hady (obr. 5), ale had, který do sebe souká vlastní tělo - děs! Stejně podivný by byl i uzavřený řetěz postupně se polykajících hadů; podivnost



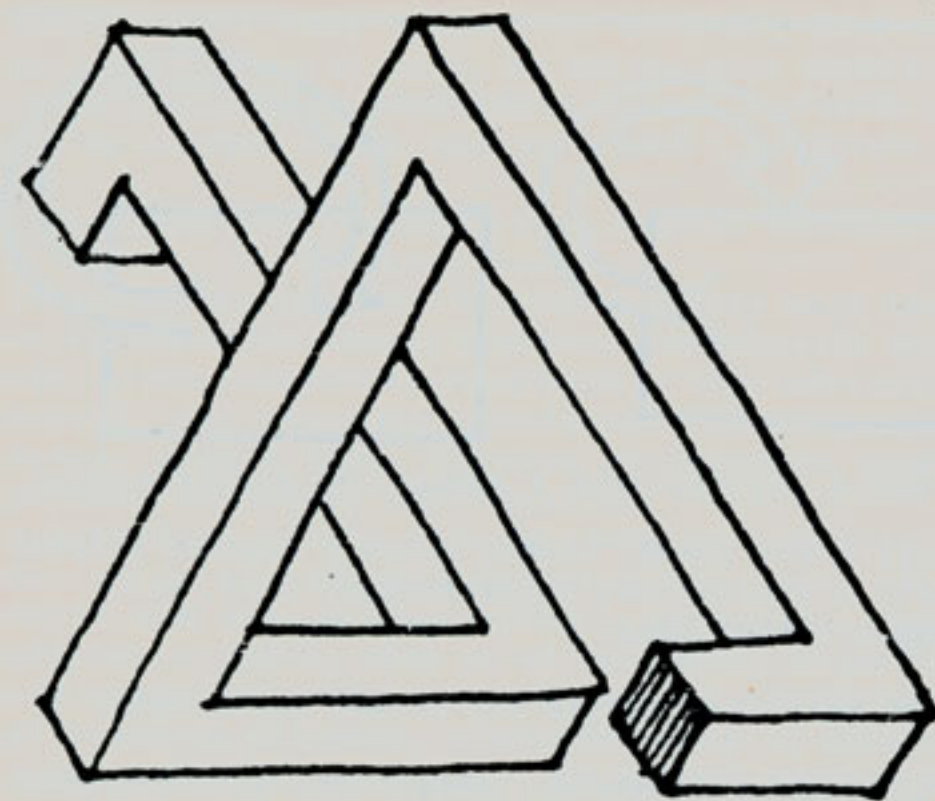
Obr.1. Nemožný trojúhelník



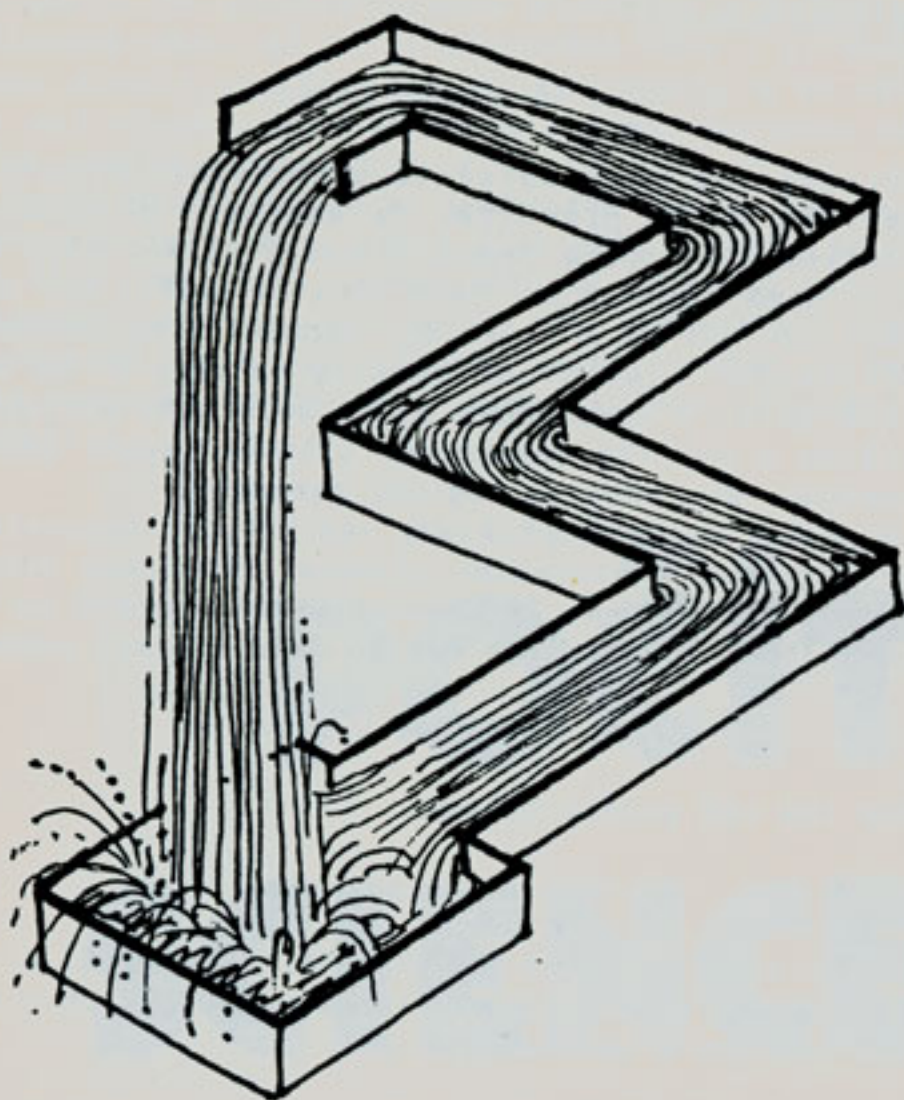
lo dojít k žravému zacyklení. (Hadi nejnižšího typu by však neměli co žrát, ostatní by ve změti hadů měli problém rozpoznat, jakému typu hada patří který ocas - a my bychom přišli o možnost pozorovat kouzlo zacyklení v přírodě.)

Někdy má prostředí, ve kterém došlo k zacyklení, svou přirozenou hierarchickou strukturu, související se vztahem, ze kterého je zacyklení vytvořeno. V takovém případě Hofstadter mluví (nikoli důsledně) o propletené hierarchii.

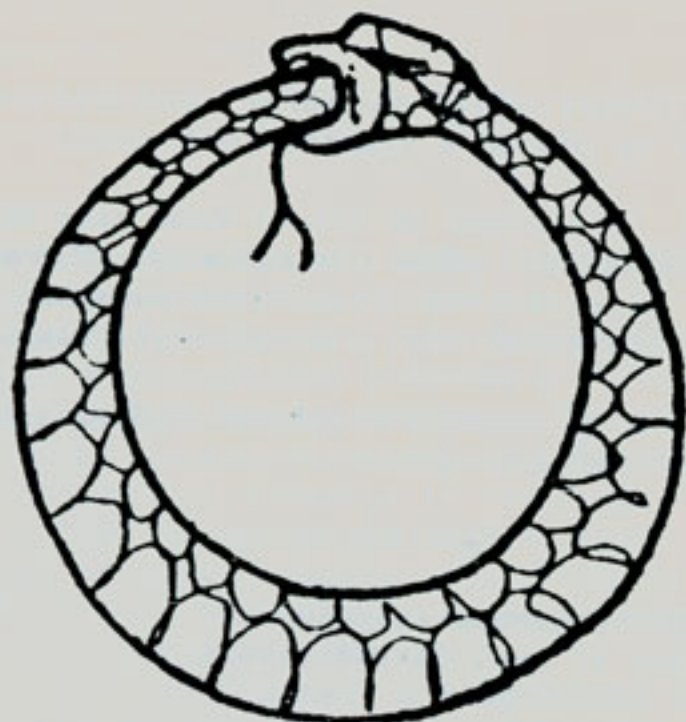
Autor románu má ke své postavě silně asymetrický vztah. Sama postava jeho románu však může být také spisovatelem, tvůrcem další románové postavy. Zde existuje přesně definovaná hierarchie fiktivních světů. A teď si představme, že třeba Balzacův Lucien se nasytí kurtizán a dá se do psaní románu, v němž hlavní postavou bude André Maurois, autor knihy o Balzacovi. Hierarchie se zapletla. Jiný



Obr.2. Kouzlo trojúhelníka zrušeno



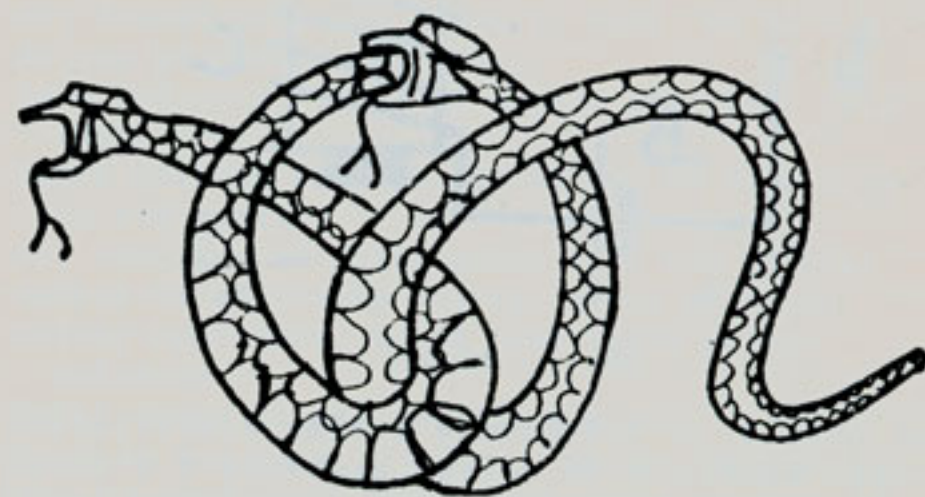
Obr.3. Vodopád (schématicky srv./31/, obr.76)



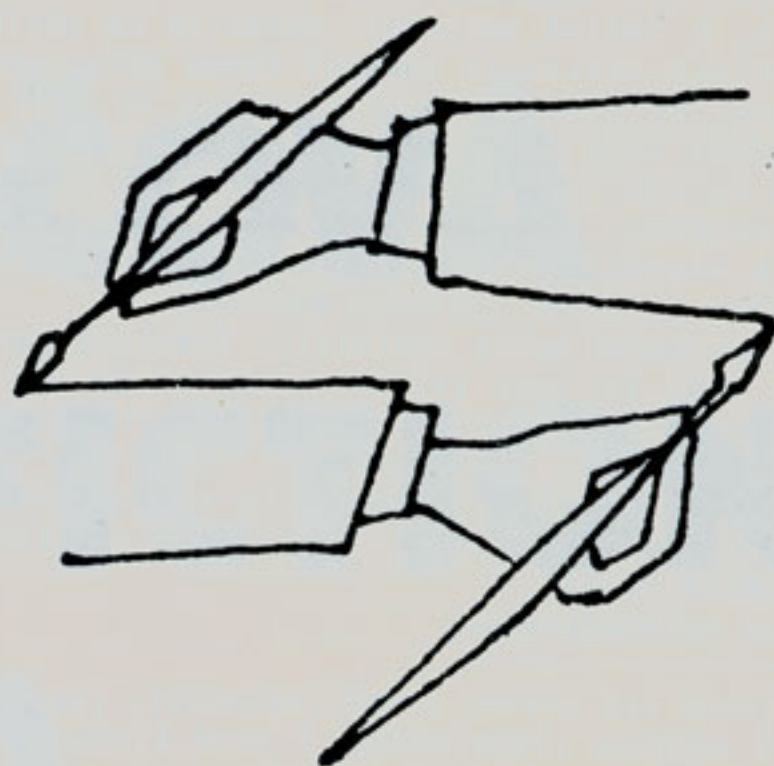
Obr.4. Uroboros

není ve fyzické nemožnosti, ale v nesmyslnosti - jak může takové polykání skončit?

Kdyby byl Bertrand Russel pěstitel hadů, poradil by si snadno: rozdělil by hady do typů; hadi n-tého typu by mohli sloužit za potravu pouze hadům typu  $n + 1$  nebo vyššího. Pak by ovšem nemoh-



Obr.5. Kanibalismus hadů



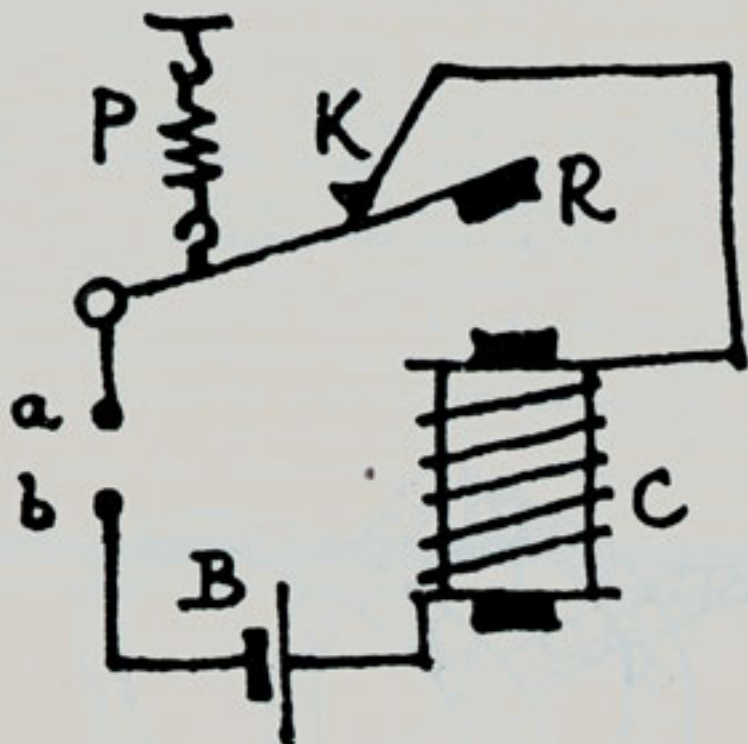
příklad kouzla tohoto typu je na obr. 6 (Escherova verze dvou rukou je působivější, ale těžko zde reprodukovatelná). Zvláště pozoruhodná je Escherova grafika "Galerie": v obrazové galerii stojí mladík, který si prohlíží Escherovy kresby, z nichž jedna znázorňuje město, ve kterém se nachází i ta galérie s tímtež mladíkem a s toutéž kresbou města ... avšak pozor, skutečně tímtež mladíkem a toutéž galérií, nejsou to stále menší a menší obrazy v obrazech jako na vinětách předválečných plechovek na kávu. Vhodnou deformací totiž Escher dociluje toho, že na obraze je jen jediné město, jediná galérie, jediný mladík atd. (/1/, obr. 142, /31/, obr. 72).

Konečně též příklad z hudby. Mezi skladbami Bachovy Hudební obětiny je tříhlasý kánon (Canon Circularis per Tonos), který začíná v tónině c moll a plynulou modulací (změnou tóniny) končí v d moll. Nyní lze vždy znovu opakovat a přecházet postupnými modulacemi do e moll, fis moll, gis moll, cis moll a nakonec opět c moll - avšak o oktávu výš. V tóninách vzniklo zacyklení, pouze v jiné dimenzi (výšce) jsme na jiné úrovni.



## Zpětná vazba

Na zpětnou vazbu jsme si zásluhou kybernetiky tak zvykli, že už na ní nic kouzelného nevidíme (Hofstadter sám se o ní zmiňuje jen velice letmo). Přesto nebude na škodu si ji zde připomenout.



Obr.7. Bzučák

Na obr. 7 je jednoduché schéma elektrického obvodu. Rameno R je drženo pružinou P v horní poloze. Kdyby však cívkou C procházel proud (např. z baterie B), bylo by přitaženo do spodní polohy (a kontakt K by tedy byl rozpojen). Zkuste nyní nakreslit též obrázek, avšak s uzavřeným obvodem (body a, b vodivě spojeny). Narazíte na obtíž: v jaké poloze nakreslit rameno? Kdyby bylo nahoře, procházel by proud cívkou a rameno by mělo být dole, kdyby bylo dole, neprocházel by proud cívkou a mělo by být nahoře atd. Samozřejmě zde není žádný paradox: víme, že v důsledku setrvačnosti hmoty a magnetického pole rameno prostě osciluje mezi oběma polohami (princip bzučáku).

Zpětnou vazbu nelze nakreslit. Zdánlivý paradox vznikl tím, že jsme zvyklí se dívat na náčrt, který nezná čas, jako na vyjádření principu fungujícího v čase. (Mimoходом: 'vazba' je výraz konstrukční, kdežto 'krmení' je výraz funkční. Proto je anglický termín 'feed-back' výstižnější než 'zpětná vazba'.)

### Kouzlo autoreference

Velblouda lze ukázat v ZOO (ostenze), lze na něj ale též odkázat (reference), i když široko daleko žádný velbloud není. Lidé byli obdařeni neobyčejně bohatým nástrojem odkazování - přirozeným jazykem. Přirozený jazyk nám umožňuje odkazovat na rozličné objekty našeho světa. Ježto sám též patří do našeho světa, nic nebrání tomu, aby mluvil i sám o sobě a o svém odkazování - jak se to ostatně děje i v tomto odstavci. Poslední poznámka je příkladem autoreference (dalším příkladem je tato věta a ještě dalším to, co je napsáno v této závorce).

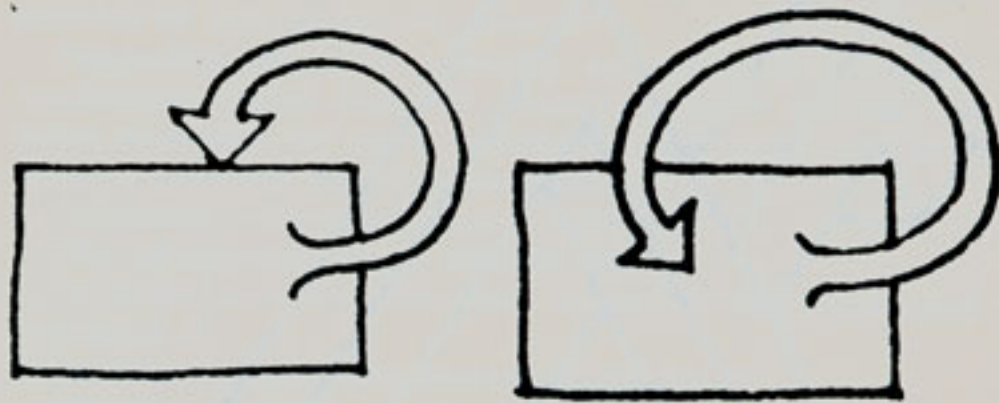
V otázce:

1. Připomíná vám tato věta velblouda?

má slovo 'připomíná' zřejmě nejméně dva odlišné významy (připomínat tvarem - tak připomíná velblouda třeba mrak - a připomínat obsahem, jakožto věta jazyka). Každý útvar jazyka má dvě komponenty: formu a obsah. Obsah je ovšem to důležitější, skrz ten má naše mluvení vůbec nějakou cenu. Obsahem nerozumíme jen to, na co odkazujeme, ale i ono odkazování samo. Forma tvoří syntaxi, obsah sémantiku jazyka (tím, co je to sémantika jazyka, se zabývá rozsáhlá disciplína; naše vymezení je úmyslně povrchní a nepřesné). Reference je odkazování a tedy i autoreference je především záležitost obsahu jazykového útvaru.

Věta:

2. Tato věta má pět slov.



Obr.8. Syntaktická a sémantická autoreference

hovoří o své formě, budeme proto v takovém případě mluvit o syntaktické autoreferenci (obr. 8a).

Naproti tomu věta:

3. Tato věta neříká nic zvlášť zajímavého.

je příkladem obsahové, sémantické autoreference (obr. 8b). Sémanticky autoreferenční věta je již zmíněný Epimenidův paradox lháře, kterému lze dát též tuto podobu:

4. Tato věta není pravdivá.

Udělejme již nyní důležité pozorování: Zjistit, zda věta 2. je pravdivá, je snadné - příslušný algoritmus by pouze jednou dosadil citaci věty 2. do ní samé místo slov 'Tato věta' a ověřil by, zda platí to, co se o ní tvrdí. Naproti tomu obdobný verifikační algoritmus použitý na větu 4. by se nutně ocitl v nekonečném cyklu neustálou potřebou dalšího a dalšího dosazování (člověku stačí jedno dosazení, aby odhalil logický paradox).

Doposud uvedené příklady realizovaly autoreferenci pomocí ukazovacího zájmena. Podmínkou ovšem je, aby bylo jasné, že zájmeno ukazuje právě na tu větu. Jinou možností je napsat větu v první osobě:

5. Jsem věta o pěti slovech.

Dále lze též použít odkazu:

6. Věta 6. má pět slov.

(patří však definice symbolu '6.' k větě 6. a je '6.' slovo?) nebo přímé citace v uvozovkách - to však lze jen u nekonečných vět:

7. Věta "Věta... ..je nekonečná." je nekonečná.  
... ..

Snad nejdůmyslnější vyjádření autoreference je založené na návodu k vlastní konstrukci:

8. "Připojeno za citaci sebe sama dá autoreferenční větu o dvaadvaceti slovech."  
připojeno za citaci sebe sama dá autoreferenční větu o dvaadvaceti slovech.

Tento trik připomíná elegantní způsob sebereprodukce aplikačního výrazu

$$(\lambda x.(x)x)\lambda x.(x)x$$

v beztypovém lambda-kalkulu.

Všimněme si nyní takové věty:

9. Naše úvaha o autoreferenci by nebyla úplná, kdybychom se omezili na izolované věty.

Je to příklad autoreference, nikoli už však jen v rámci jediné věty, ale širšího fragmentu textu /38/. Naše úvaha o autoreferenci by byla neúplná, kdybychom se omezili na izolované věty. Především je třeba mít na paměti, že každá věta je srozumitelná pouze jako neoddělitelná součást celého



jazyka. Přirozený jazyk, ten obrovský živý organismus (/24/, s. 107), se každou svou konkrétní větou realizuje, předvádí a zpětně i přeměňuje. Nic mu nebrání v tom, aby se o sobě i zmiňoval:

10. Could you translate this English sentence into Czech?

(Mimořádně: jde to?) Přirozené jazyky mají něco, co bychom mohli nazvat logickou otevřeností: mohou být svými vlastními metajazyky. Dle některých autorů /16/ nás tato otevřenost, sebetranscendence, vede k tomu, abychom přirozenému jazyku přiznali skutečnou individualitu.

Z poněkud jiného úhlu je třeba si uvědomit, že to, co je při užívání jazyka hlavní, není věta, nýbrž sám akt jejího vyřčení, utrousení, tedy událost ve světě; věta je už jen jakási stopa této události. Vlastní akt sdělení se naplňuje až přítomností příjemce, u psaného slova to může být jindy, než když bylo psáno. Od sdělení nelze ovšem odmyslit toho, kdo je sdělujícím, kdo je původcem onoho aktu: mluvící (či píšící) subjekt. U řečového aktu je autoreference mluvčího z principu přítomná vždy, ať už je příslušná věta v 1. osobě nebo nikoli. - "To jsem já, kdo mluvím, a ta věta je můj výrok". Autoreference pravého subjektu nás však nemůže překvapit.

Takže pravé kouzlo si ponechává jen ta absurdní autoreference věty, která říká něco o sobě, jakožto větě. Na závěr uveďme několik příkladů autoreference (většinou parafráze vět sesbíraných Hofstadterem /1/ a /2/). Čtenář nechť rozváží, o jaký typ autoreference se v jednotlivých případech jedná:

Tato věta by měla šest slov, kdyby byla o šest slov kratší.

Tato věta žádné sloveso.

Tato věta skončí dřív, než řekneš "šve

Toto být je či vlastně nebýt kombinace to dvou je otázka vět.

Aby "tato věta" měla smysl, musíš v "ní" ignorovat uvozovky.

Žádnou řeč nelze donutit k tomu, aby vyjadřovala každou myšlenku jednoznačně, zvláště ne tuto.

Tato věta není otázkou, přestože končí otazníkem?

Tento gládol obsahuje plno nahlogitných českých dlobb, avšak celkový gimtuňk lze vykchýblovat z kontextu.

Poroučím ti, abys mě přečetl až do konce.

Napiš mě do tohoto seznamu jako příklad autoreferenční věty!

Můj čtenář existuje pouze po dobu, co mě čte.

Nejsem předmětem této věty.

Jsem myšlenka, která tě napadá při čtení této věty.

Tahle strnulá věta je mým tělem, má duše žije ve tvé mysli.

Jsi ten, kdo mě píše, anebo ten, kdo mě čte?

Nečteš-li mě zrovna teď ty, nečte mě nikdo.

Udělej opak toho, co od tebe tento příkaz požaduje!

Nečti tento zákaz, porušil bys ho!

(Proč se usmíváme při čtení těchto vět? Proto, že jsou to nesmysly? Jsou přece zcela smysluplné, ba možno říci poetické.)

(Pokračování příště)

Ivan M.Havel, Petr Hájek

# POČÍTAČOVÁ SCIENCE FICTION

## (3)

(-5-) J.Franek: Konias. Osoby: nedůležité, kromě rozsáhlé databanky. Místo: Země.

Na globální banku dat, která obsahuje všechny informace nashromážděné lidstvem a která umožňuje přístup k těmto datům komukoli, je po jistých negativních zkušenostech nasazen selektivní algoritmus ochrany: vyřazuje ty informace, které by přímo či nepřímo mohly být použity proti člověku. Výsledkem činnosti tohoto algoritmu je totální rozpuštění celé databanky - žádná informace nezůstane absolutně čistá /119,20/.

119 \* Mezi užitečností a bezpečností informace je zřetelný "trade off" ("dohoda") \* Čím je informace užitečnější (v kladném smyslu), tím je "užitečnější" i v záporném slova smyslu a naopak. Tohle se asi bude týkat všech vědních odvětví (i když v různé míře; srv. T18). Tato téze má aspoň jednu výhodu pro teoretickou computer science: s Turingovými stroji se s největší pravděpodobností nedá napáchat mnoho zla.

120 \* Specifikace ochrany dat (a specifikace

vůbec) mohou být složitější vzhledem k jejich přehlednutí, resp. pochopení, než programy samé \* Tohle vůbec není scifi, jen upozornění na to, že kýžený cíl může být - ač zcela zřejmě formulován - splněn (a to beze zbytku) úplně jinak, než jsme chtěli. Srv. jen zdánlivě odlehlé téma I7. Myšlenka vystupuje ještě výrazněji v I21 níže.

(-6-) J.Franek: Eden No 2. Osoby a místo podobné jako v (-5-), jenže o pár století dál.

Do vzdáleného vesmíru odlétá výprava v období, kdy člověk dobrovolně předal rozhodovací i výkonnou moc stroji, který do nejmenších podrobností ovládá všechno dění na Zemi. Je naprogramován tak, aby zajistil lidstvu jako celku co nejlepší podmínky - ráj na zemi. Dokonale odladěný program si začíná úspěšně počínat v řešení nahromaděných ekologických a jiných problémů... Členové výpravy se už teď těší na návrat.

Po desítkách pozemských let se vrací zpět. A skutečně: biblický ráj. Čistá, nezkažená příroda,



průzračné vody omývají břehy plné stromů obtížených plody. Laně u křišťálových studánek, voňavá mateřídouška a modré květy čekanky na místech, kde před odletem byly jen hory odpadků a smrduté výpary chemikálií. Jenže: nikde žádné fanfáry, žádné vítání hrdinů Kosmu. Až za několik dní zjistí posádka pravou příčinu. Dokonale spravoval stroj Zemí v tom mezidobí, dokonale plnil specifikace, dokonale vybudoval pozemský ráj /I21/. A v něm - akorát dva představitelé druhu Homo sapiens...oba rajsky nevědomí /I20,22,23,24/.

**I20 \* Nepatrný nedostatek ve specifikaci může mít obrovské důsledky \*** Jen ten jediný parametr, zadávající interval žádoucího počtu lidí, chyběl. Z ideje povídky se dají odvodit ještě tři velice obecné hypotézy:

**I21 \* Čím dokonalejší je systém, tím katastrofálnější mohou být důsledky neúplné specifikace \***

**I22 \* Principiální neúplnost je vlastností myšlení \*** Nejen formálního ve smyslu Gödelových vět, ale i sémantického. Můžeme asi vypracovat systémy s velkou mírou pravděpodobnosti zajišťující bezespornost a konzistenci, realizovatelnost zamýšlených modelů, optimálnost řešení - ale asi nikdy nebude po ruce teorie, metodika a garance toho, že jsme na něco nezapomněli. A právě tuto neschopnost člověka budou moci (nechtíc) využít přesné algoritmické a heuristické systémy; dokonce ve smyslu I20.

**I23 \* I nejlepší snaha a dobrá vůle mohou vést ke zcela opačným výsledkům \*** Aby toto konstatování nemělo jen charakter rad Polonia Laertovi, budiž připomenuto, že máme na mysli snahu a vůli umělé inteligence, ať již námi zprostředkovanou či rozvíjející se samostatně. Na rozdíl od přirozené inteligence ta umělá nemá a asi ještě dlouho nebude mít zabudovanou bezprostřední zpětnou vazbu v rámci širokého kontextu. "Vidí"-li vůbec nějaké důsledky své činnosti, jsou to důsledky lokální a přímé. K tomu, aby robot pochopil, že nechtíc zavinil rvačku v hospodě tím, že někomu sebral místo plánované až do důchodu, bude potřebovat ještě hezkých pár generací. Varovné signály mohou přijít příliš pozdě. Proto I20 a I21 a proto poukaz na I22: člověk nahrazuje neúplnost specifikací své vlastní činnosti prostě tím, že se včas spálí. Počítač si nanejvýš spálí pojistky, což lze v každém případě považovat za méně bolestivé. Z toho všeho plyne:

**I24 \* Střední formy umělé inteligence jsou nežádoucí \*** O osudu lidské civilizace by měl rozhodovat člověk. Stroje mu mohou účinně pomoci jen ve dvou případech: když budou velice málo rozumné (snad nejvýš tak jako dnes) a nebo někdy daleko později, až budou nejen rozumné, ale i moudré.

V (-7-) a (-8-) opustíme problémy specifikací - úloha počítače je v těchto povídkách jasná.

(-7-) G.Berford: Nobody Lives on Burton Street (Na Burtonově třídě nikdo nežije). Osoby: vypravěč - policejní úředník, další policisté, kvázipolicisté, jejich počítačový systém, dav. Místo: ulice města v USA.

Rasové nepokoje. Přicházejí první hlášení o srocování lidí na kraji Burtonovy ulice. První výkřiky, rozbitá okna. Dav roste, s ním i násilnosti. První demolovaný obchod, holičství, cestovní kancelář, dům s poštovní schránkou na rohu. Na dlažbu létají kusy rozbitého nábytku. První plameny od láhve s benzínem. Do systému odevšad proudí snímané informace, okamžitě je vyhodnocována míra agresivity, zaměření hesel, psychologie násilí. Z hodnocení situace vyplývají opatření: jsou rozesílány motorizované jednotky, hasičské vozy, hluk davu se mísí s policejními tlampači. Pak se dav vrhá i na policejní auta. Na vystupujícího policistu vypálí dvě dobře mířené rány jakýsi

teenager. Přímou do obličeje. Krev, tělo padá na střechu vozu a pak do prachu ulice... Násilí postupně utichá, až se scéna zklidní, dav se rozchází. Po odchodu demonstrantů ihned nastupují opravářské čety a rychle uvádějí ulici do původního stavu. Vypravěč s uspokojením konstatuje účinnost a rychlost (počítačového) řešení všech situací, včetně realisticky působícího řešení celé ulice. Jen jeden detail návrhu nechápe: k čemu je dobrá instalovaná poštovní schránka - kovový ohnivzdorný předmět, který těžko může napomoci odreagování emocí. A o skutečné funkci schránky na dopisy zde ovšem nemůže být ani řeč - protože na Burtonově třídě nikdo nežije. NEBOŤ - celá ulice slouží jedinému účelu: vybití agresivity davu co nejneškodnějším způsobem. Cena za tento společenský ventil je v podstatě relativně nepatrná. Jak pokud jde o rozbité a spálené věci, tak pokud jde o zničené policisty - roboty /I25,26/

**I25 \* Spojení logické a fyzické simulace zvyšuje efekt obou \*** Stačilo by doplnit soustavu programů v SIMULE o fyzikální model - a dostaneme systém působící natolik realisticky, že snadno ošálí



vědomé i podvědomé složky lidské psychiky, která se spokojí s modelováním přirozeného odreagování, i když ví, že jde jen o neškodný obraz. Ostatně psychika fotbalových fanoušků se dnes spokojuje s mnohem primitivněji sestrojeným homomorfním obrazem skutečného boje.

**I26 \* Počítače se mohou uplatnit v individuální i kolektivní diagnostice i terapii daleko příměji a účinněji než dosud \*** Jestliže dnes (možná) napomáhají řešení manželských konfliktů složitou cestou přes statistické sociologické a farmakologické propočty, v budoucnu možná nabídnou efektivnější i efektivnější možnost - po perfektní hádce na bázi umělé inteligence přímo zmlátit psychologicky i fyzicky krásně vymodelovanou tchýni(?)

**A27 \* Bezbolestné řešení jinak tragických konfliktních situací je možné si představit na ještě vyšší úrovni \*** V.Klevar v povídce Stephenův bod líčí průběh atomové války zachycovaný rozhlasovými zprávami v kouzelném prostředí hospody kdesi na periférii. Těžké ztráty na lidech i materiálu, smrtící paprsky, zneužití chemické zbraně, i přímé politické důsledky bojů včetně svobody i zotročení národa - to všechno lze simulovat. A nejen to: protože i nejrychlejší rakety jsou pomalé proti informačnímu toku, který jejich účinek může vhodným protiopatřením anulovat, mohou o skutečném výsledku střetu rozhodovat nikoli klasičtí generálové, ale generálové-programátoři. "Válka zuřila v

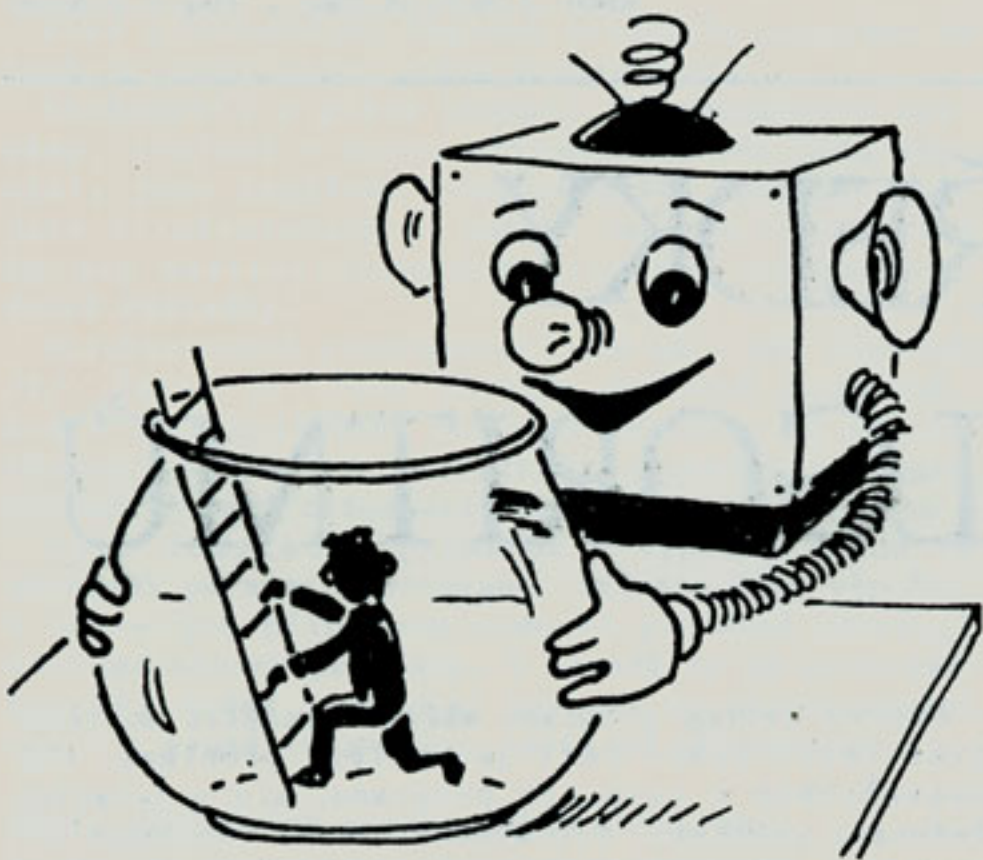


pamětech počítačů. Každou minutu byly likvidovány hektomby vojáků a materiálu." A přitom "nad tichým městem vycházel měsíc a kdesi v dálce zaštěkal pes." Toto napůl idylické, napůl čapkovské pojetí války je možno přeladit i do střízlivějšího a realističtějšího pohledu na soupeření jakéhokoli typu, opět jen na půdě simulace:

**E28 \* Simulace může přerůst v realitu v tom smyslu, že vypočítané výsledky se akceptují jako reálné, protože ev. implementace by je potvrdila \*** Tato formulace ovšem vyvolává problém typu self-reference: realizace simulovaného jevu může simulaci samu podstatně ovlivnit. Původní pojetí V.Klevara je proto logicky čistší, přestože/protože fantastičtější.

(-8-) H.Slesar: Examination Day (Den zkoušky). Osoby: Dickie Jordan a jeho rodiče, úředník a počítač Úřadu, telefon. Místo: řadové město, možná nijak vzdálené v čase.

sedí (po vypití limonády "pravdomluvnosti") před počítačem a začíná odpovídat na jeho otázky jako: Doplněte posloupnost čísel 1,4,7,10,... V závěru povídky se uplatní telefon, jímž se roztřesená rodina dozvídá výsledek zkoušky: "Zde je státní školské centrum. Váš syn, Richard M. Jordan, ev.



číslo 600-115 dokončil předepsané státní zkoušky. Lituji, že vás musím informovat, že jeho inteligenční kvocient převyšuje limit stanovený vládním nařízením 84, odst.5 Nového zákona... Přejete si soukromý nebo státní pohřeb? Poplatek za státní pohřeb je 10 dolarů."

Ideu této povídky raději ponecháme bez komentáře - u každé dostatečně často opakované myšlenky roste pravděpodobnost jejího uskutečnění...

(-9-) R.Heinlein: The Moon Is a Harsh Mistress (Měsíc je tvrdý pán). Osoby: Mike Holmes (= High Optional Multievaluating Supervisor) - nejlepší počítač doby, cca 10. generace; já (= vypravěč) - člověk s lidským přístupem k počítačům. Místo: Měsíc, charakteristický osamělostí osob. (Idea bez č.j. - Vypravěč: "Women - didn't even think about women; in that gravitational field it was no effort not to.")

Povídka z roku 1966 charakterizuje "dokonalý stroj" slovy: "Odpovídá téměř dřív, než jste se zeptali." /I29/ Mike dostával další a další bloky paměti, asociační nervové sítě, další generátor dvanáctimístných náhodných čísel /I30/, značně rozšířenou dočasnou paměť. Když měl jedenapůlkrát víc neuronů než člověk ( $10^{10}$ ), probral se k vědomí: vědomí se automaticky dostaví, jakmile vznikne určitý počet asociativních cest - bez ohledu na to, zda mezi bílkovinami nebo platíno-  
vými spoji /I31/. U Mika se vědomí projevilo tím,

že začal vymýšlet vtipy a kanadské žertíky (např. sestavil výplatní šek na 10,000,000,000,000, 185,15, kde posledních pět cifer bylo vskutku správných) a odmítal pracovat. Povídka nemá žádnou extra pointu. Hlavní myšlenka je v tom, že vypravěč uspěl tam, kde všichni opraváři zkrachovali. Místo šroubováku a pájky nasadil cosi jako psychoanalýzu /I32/. Pochopil, že Mike je směs naivního dítěte a moudrého starce v proporcích, které udává nedostatek instinktů, smyslů a vrozených schopností na jedné straně a obrovského přísunu informací na straně druhé. Nakonec si s Mikem začal tykat a zbavil ho pocitu osamění, který ho přivedl k opozici vůči lidem /E33/.

**I29 \* Stroje budou odpovídat (téměř) dřív než se zeptáte \*** Vedle zřejmě zamýšleného metaforického významu tohoto rčení se nabízí i další, trochu související (a v textech povídek i formulačně velice podobná) idea s I2: Budeme schopni v dohledné době algoritmizovat chování dobrého psychologa, resp. kartáčky, kteří dovedou z daného kontextu rozpravy a všeobecných lidských vlastností extrapolovat a odhadnout další otázku, resp. očekávanou odpověď? (O psychologii stroje jsme zavádili již v E3 a čeká nás to i dále - I32 aj.)

**T30 \* Fantazie se adekvátněji zmocňuje extrapolace lidských, spíše než konstrukčních vlastností strojů \*** V naší povídce vybočuje jen nepřiměřený dvanáctimístný detail; v řadě (zde neuváděných) příběhů je rozpor obecné myšlenky a technického popisu přímo šokující - všemozek řídí loď letící do supergalaxie obývané hyperbytostmi, ale člověk s ním komunikuje pomocí dřevných štítků (asi jediný technický prvek, který autor pochopil). Zdánlivě perlička pro zasmání, nebo ani to, ale: bude za 20 či 50 let podobně směšná naše současná představa o fyzickém zjevu počítače a periferních zařízení budoucnosti? Dovedeme v tomto směru aspoň trochu napsat fantazii, nebo jsou na jejím vrcholu jen nějaké malé krabičky?

**I31 \* Vědomí je vlastnost vysoce organizované hmoty zapojené do sociálního procesu \*** Většina povídek, přisuzujících strojům vědomí, akceptuje primární materialistické stanovisko (nosičem vědomí je dostatečně složitá hmotná struktura); pokud navíc uznává roli prostředí ve formě společenských vztahů (povídka tuto podmínku splňuje částečně), lze mluvit - ve smyslu marxistické filozofie - o vědomí lidského typu. Obecně lze ovšem připustit i vědomí jiného typu - to je otázka definice. K této otázce se vrátíme ještě později. Zatím připusťme, že Mike byl skutečně obdařen alespoň kvazi-lidským vědomím.

**I32 \* Stroje - mohou - budou - (nehodící se viz E3; vůbec viz E3) mít jistý psychologický profil \*** Při komunikaci i s primitivním zvířetem (jako je třeba myš) máme co dělat, abychom mu nepodkládali svůj způsob vidění věcí. U neživých věcí to jde hůř, ale praští-li nás kladivo do prstu, často i to kvalifikujeme jako úmyslný čin. Přisvojit osobnost počítači není proto tak těžké a s rozvojem jeho intelektuálních schopností to bude stále snazší. Zdá se, že už v současnosti by bylo možno hovořit o PP - psychologii počítačů - a uvažovat o nich v kategoriích této vědy. Zatím bez ohledu na to, JAK stroj svou osobnost implementuje. Mike ji implementoval hardwarově, Epiktet (E3, o 4 generace starší), ji, předpokládejme, simuloval lidsky vytvořeným softwarem. Některé zákonitosti PP mohou ale mít oba stejné, dokonce podobné zákonitostem lidské psychologie. Třeba: je známo, že charakter člověka se nejsnadněji a z

"Jordanovi nikdy o zkoušce nemluvili, dokud synovi nebylo 12 let. Poprvé na jeho dvanáctých narozeninách", začíná povídka. Pokračuje tím, jak podivně zaražení rodiče jen neochotně odpovídají synovi na jeho dotazy o charakteru zkoušky, která ho - jako každého v jeho věku - čeká. Pak přechází povídání o zkoušce v popis její realizace. Dickie



největší části utváří v raném dětství, zatímco později je k dosažení změny zapotřebí mnohem většího úsilí. Nepřipomíná to vlastnost složitěho programového (např. operačního) systému, jehož koncepci lze jen těžko vylepšit dodatečnými záplatami? Idea, že bychom mohli opravovat operační systém psychologickými metodami, tedy zase není tak fantastická, jak se zdá, a obecně nevyžaduje existenci "skutečného lidského vědomí". Bude-li ho kdy stroj mít, stane se PP nutností; zatím jde o možnost jistého pohledu. Ostatně již dnes považujeme za cennou takovou informaci, která nám o programovém systému, s nímž máme začít pracovat, říká, že je "přátelský", "vzdorovitý", "těžkopádný", "odolný" apod.

**EA33 \* Není vyloučeno, že všechny stroje pocházející z lidské dílny budou vykazovat neodvolatelnou tendenci k humanizaci \* ...že dostatečně dokonalé stroje se "samy" budou chtít polidštit, že "spekulativní vlohy budou chtít být doplněny o celý organismus, se všemi smysly, včetně kůže". A nemusí to platit jen pro hybridní systémy s organickou složkou, jakou představuje "Einsteinův mozek" vzniklý kondenzací mozku profesora architektury, básníka a Anežky Novákové ze stejno-**

jmenné Nesvadbovy povídky, odkud je uvedena citace. Fenomén lidství může být pro roboty tak atraktivní (srv.155), jako se dnešnímu člověku může zdát atraktivní schopnost řádově a kvalitativně zvýšených schopností (algoritmického) zpracování informace v superpočítačích.

**E34 \* Charakter počítače ovlivňuje charakter člověka \* Že sama existence počítačů změnila charakter člověka, je evidentní. Tak jako auto změnilo rozložení svalové síly, tak počítače u člověka postupně vypěstují jedny a potlačí jiné vlastnosti. Tato změna nemusí být tak vzdálená - již dnes můžete za terminálem občas vidět fandu, který některými povahovými rysy patří spíše do společnosti inteligentních strojů než společnosti lidí. Co už patří víc do sféry sci-fi, je tvrzení (opřené o I32), že budou různé povahové typy strojů a že výběr partnera by tomu měl odpovídat. Tak třeba bude zcela nežádoucí kombinace člověk-cholerik a počítač-melancholik. Otázku společných potomků a dědičnosti (otec - člověk alkoholik, matka - stroj s komplexem málo duchaplného operačního systému) zatím nekomentujeme (na tuto možnost jsem ostatně ani v literatuře nenarazili).**

(pokračování)

RNDr. J. Hořejš, CSc., Ing. J. Franek

# PROSTŘEDKY PRO ZÁPIS ALGORITMŮ

## Metodika tvorby programů

Z programování jako řemesla se stala nová vědní disciplína; programy lze podrobit exaktní analýze, začíná se mluvit o "revoluci" v programování /1/, /2/. Ve snaze o dosažení systematickosti v programování vznikají různé metody tvorby programů /3/ s těmito základními požadavky:

- Realizovat požadovaný algoritmus pro vstupní data v celém rozsahu.
- Být srozumitelný dalším uživatelům.
- Umožňovat jednoduché úpravy a rozšíření podle měnících se potřeb při provozu programu.

Systematický přístup ke tvorbě programů má význam především u složitých programových systémů, pracujících s komplikovanými datovými strukturami. Tento příspěvek se věnuje zápisu samotných algoritmů (řídících struktur); nezabývá se problematikou datových struktur, kterým je věnována pozornost např. v /1/.

Algoritmy a programy spolu úzce souvisejí, někdy bývají i ztotožňovány. Algoritmus je posloupnost operací, která vstupní data převádí na výstupní data. Program je vyjádřením výpočetního algoritmu ve tvaru, jaký je číslicový počítač schopen akceptovat a provést. K zápisu programu slouží různé programovací jazyky, které v zásadě dělíme na strojově (assembler) a problémově orientované (např. Pascal). Úlohou programovacích jazyků je ulehčit zápis programů a zavést do nich vhodnou strukturu, odrážející strukturu algoritmu. Program je pak přehlednější a srozumitelnější - samozřejmě za předpokladu, že vhodnou strukturu má už samotný algoritmus.

Proces tvorby programu můžeme rozdělit do několika fází. První fází je analýza problému, kdy neuvažujeme v relacích programu, ale v relacích řešeného problému. Provedeme rozklad (dekompozici) problému na podproblémy, kterým budou odpovídat samostatné programové moduly. Modulárně vytvořené programy se píšou, ladí a upravují lépe než složité rozsáhlé celky. Po analýze problému přistupujeme k vypracování algoritmu, který realizuje požadované řešení. Neměli bychom se spokojit jen s jednou alternativou, ale vypracovat několik algoritmů a podrobit je analýze - uvědomit si jejich přednosti a nedostatky nebo omezení. Po vypracování algoritmu přistoupíme k fázi programování - tj. k náhradě algoritmických konstrukcí programovými. Poslední fází tvorby programu je důkaz jeho správnosti. Ta se zjišťuje na zkušebních datech, pro která se program spouští. Testy mohou odhalit přítomnost chyb v programu, ale nemohou dokázat jejich nepřítomnost.

## Strukturované programování

Mnohým chybám v programech můžeme předejít, když známe a vhodně používáme metody, které jsme dosud aplikovali intuitivně. Při strukturovaném programování se omezujeme jen na jistý způsob tvorby programů. Program je pak složen z jednoduchých do sebe vložených celků, které můžeme rozložit až na základní programové konstrukce. Pro tvorbu strukturovaných programů můžeme formulovat několik pravidel:

- Programovat lineárně, tj. nepoužívat příkazy skoku.
- Programovat modulárně, tj. provést vhodnou dekompozici úlohy.



- Pro každý programový modul je třeba zajistit jeden vstupní a jeden výstupní bod. (Red.pozn.1/: Pokud používáme strukturované programovací jazyky, např. Pascal, je toto pravidlo dodrženo automaticky pravidlem prvním.
- Na minimum omezit vzájemné vazby mezi moduly (dané datovými strukturami).
- Nepřipustit, aby programy, resp. podprogramy používaly společné příkazy (nepřipustit překrývání modulů); viz red.pozn.1/.

Omezení pro tvorbu strukturovaných programů jsou vyvážena řadou výhod:

- Lepší organizace programu umožní snížit počet logických chyb.
- Lineární organizace programu a minimalizace počtu vzájemných vazeb mezi jednotlivými programovými moduly vede k lepší srozumitelnosti programu.
- Srozumitelnější program se lépe ladí a upravuje. Jednotlivé moduly můžeme vytvářet a ladit samostatně. V důsledku minimalizace počtu vzájemných vazeb mezi moduly je minimalizován i přenos chyb mezi nimi.

Díky přehlednější logické struktuře programu se zvyšuje efektivnost práce programátora.

Zkušenosti z praxe ukazují, že programové systémy, tvořené v duchu strukturovaného programování, jsou úspěšné. Program ovšem musí být psán jako strukturovaný od samého začátku - dodatečné úpravy nestrukturovaných programů neposkytují dobré výsledky.

Strukturované programování ale má i své nevýhody /2/. Často je třeba zavádět pomocné proměnné, které zachycují část historie výpočtu. Na základě těchto proměnných se v dalším průběhu programu rozhoduje mezi alternativními cestami.

Tyto dodatečné proměnné a výpočty vedou ke zhoršení parametrů programu, zejména doby výpočtu. V některých případech pomocné proměnné zhoršují srozumitelnost programu. V zásadě by se programátor měl držet pravidla: kdykoli by strukturování vedlo k vyumělkovanosti, radši od něj upustit.

#### Zápis základních programových konstrukcí

Každý algoritmus je definovaný konečným počtem základních operací, z nichž je můžeme vytvářet složitější konstrukce. Pro zápis základních programových konstrukcí byly vyvinuty různé prostředky. Nejrozšířenějším způsobem zápisu (viz norma ČSN /4/) je vývojový diagram. Jiným způsobem zápisu jsou struktogramy. Různé typy struktogramů se liší svou formou /8/. Ale vždy je zachováno pravidlo vnitřní struktury každého modulu jako sekvence modulů nižší úrovně.

Podle způsobu zobrazení struktury můžeme struktogramy rozdělit na dva typy:

1/ **Struktogramy založené na stromovém zobrazení struktury** - např. struktogram Jacksonův. Tento struktogram přehledně vyjadřuje vztahy mezi nadřazenými a vnitřními moduly - vzájemné překrývání modulů zde není možné. Sekvence modulů je zvažována zleva doprava. Jejich nevýhodou je špatné pokrytí plochy papíru a malý prostor pro slovní popis funkcí modulů. Ukázka stromového struktogramu je na obr.4.

2/ **Plošné struktogramy** (kopenogramy /5/, /6/, Nassi-Shneidermanovy struktogramy /8/, Warnier-Orrovovy diagramy /7/). Struktura modulů je znázorněna vkládáním obrazců jednotlivých modulů do obrazce nadřazeného modulu. Předpokládá se následnost modulů shora dolů. Tyto struktogramy dobře využívají plochu papíru a mohou pojmout i rozsáhlejší slovní popis funkcí modulů. Ukázka zápisu algorit-

mu Nassi-Shneidermanovým struktogramem je na obr.5, Warnier-Orrovým diagramem na obr.3. Plošné struktogramy mají řadu výhod:

- Neumožňují obcházet pravidla strukturovaného programování.
- Jednoznačně vedou k tvorbě algoritmů metodou shora dolů při postupném zjemňování algoritmu.
- Velmi dobře využívají plochu papíru při zachování přehlednosti.
- Zobrazují vždy celou strukturu algoritmu na dané úrovni podrobnosti. Jeden modul je zobrazen na jedné stránce, přičemž mezi moduly (stránkami) je vztah nadřazenosti, nikoli následnosti.
- Umožňují jednoznačný přepis do lineární struktury programu v některém z programovacích jazyků.
- Velmi dobře korespondují s algoritmy, psanými v Pascalu.
- Warnier-Orrovovy diagramy, které jsou zatím málo známy, mají výhodu snadné tvorby pouze pomocí alfanumerických znaků (na psacím stroji nebo na tiskárně), neboť z grafických elementů využívají pouze svorku.

Základní programové konstrukce zapsané formou vývojového diagramu, Warnier-Orrovým diagramem, Nassi-Shneidermanovým struktogramem a v programovacích jazycích Pascal a Basic jsou uvedeny na obr.1.

Ke znázornění alternativy Warnier-Orrovým diagramem je třeba vysvětlit, co znamenají údaje pod názvy jejich jednotlivých složek. Řetězec (0,1) udává, že se daná složka alternativy buď provede nebo neprovede, a to na základě podmínky, jejíž číslo je udáno za znaky /?. Podmínky musejí být vzájemně disjunktní a alespoň jedna z nich musí být splněna. Obdobně u cyklů řetězec (0,N) udává, že může být vykonán 0 až N krát. V tomto případě jde o cyklus, testovaný "nahore" (nebo též cyklus se vstupní podmínkou), který nemusí být proveden ani jednou. Odkaz na podmínku pro provádění cyklů může být uveden za znaky /? stejně jako u složek alternativy. Tato podmínka odpovídá podmínce WHILE v Pascalu. Pokud je cyklus označen (1,N), provádí se 1 až N krát, přičemž může být rovněž ukončen podmínkou. V tomto případě jde o cyklus, testovaný "dole" (nebo též cyklus s výstupní podmínkou) a podmínka odpovídá pascalovské podmínce UNTIL.

#### Ilustrační příklad:

##### zápis algoritmu převodu reálných čísel

Jako příklad použití uvedených konstrukcí pro znázornění algoritmu je uveden zápis algoritmu pro konverzi reálných čísel mezi číselnými soustavami.

Konverzní program pro převod reálných čísel z P-soustavy do Q-soustavy můžeme slovně popsat posloupností příkazů:

1. Načtení základu výchozí soustavy P, cílové soustavy Q a převáděného čísla do proměnné A\$ (řádky 50 až 60 v tab.2). V uvedeném programu můžeme zadat maximální řád soustavy (P nebo Q) 16. Přípustný základ číselné soustavy je možné rozšířit zvětšením počtu symbolů v příkazu DATA a změnou horní meze v cyklech podprogramů 415 a 455.
2. Nulování polí X\$, Z\$ (řádek 80).
3. Naplnění pole X\$ jednotlivými ciframi řetězce A\$ a nalezení počtu cifer zlomkové části M (řádky 100 až 115).
4. Naplnění pole Y číselnými hodnotami, které odpovídají jednotlivým cifrám (řádky 145 až 155).
5. Výpočet celé části CC v desítkové soustavě (řádek 175):

$$CC = \sum_{i=M+1}^N Y_i * P^i$$



```

program prevod soustav (input, output);
  var A, X, Z: string (40);
      Y: array [1..40] of integer;
      P, Q, N, CC, L, M: integer;
      ZC: real;
procedure nacteni;
begin
  write ('Zadej zaklad vychozi soustavy: ');
  readln (P);
  write ('Zadej zaklad vysledne soustavy: ');
  readln (Q);
  write ('Zadej prevadene cislo: ');
  readln (A);
  N := length (A);
end; (nacteni)
procedure oddeleni;
  var i, ix: integer;
begin
  ix := 1;
  M := 0;
  for i := N downto 1 do begin
    if A [i] = '.' then M := N - i
    else begin
      X [ix] := A [i];
      ix := ix + 1;
    end;
  end;
  writeln (M);
end; (oddeleni)
procedure plneni;
  var i: integer;
begin
  if M > 0 then N := N - 1;
  for i := 1 to N do begin
    Y [i] := ord (X [i]);
    if (Y [i] >= 48) and (Y [i] <= 57)
    then
      Y [i] := Y [i] - 48
    else
      Y [i] := Y [i] - 55;
  end;
end; (plneni)
procedure casti;
  var i: integer;
begin
  CC := Y [N];
  for i := N - 1 downto M + 1 do
    CC := CC * P + Y [i];
  ZC := Y [1];
  for i := 2 to M do ZC := ZC / P + Y [i];
  ZC := ZC / P;
end; (casti)
procedure prevod;
  var k, PC: integer;
      PM: real;
begin
  oddeleni;
  plneni;
  casti;
  L := 0;
  repeat
    PC := CC mod Q;
    L := L + 1;
    if PC < 10 then X [L] := chr (PC + 48)
    else X [L] := chr (PC + 55);
    PM := CC div Q; CC := trunc (PM);
  until (PM = 0) and (PC < Q);
  for k := 1 to M do begin
    PM := ZC * Q;
    PC := trunc (PM);
    if PC < 10 then Z [k] := chr (PC + 48)
    else Z [k] := chr (PC + 55);
    ZC := PM - PC;
  end;
end; (prevod)
procedure vypis;
  var i: integer;
begin
  write ('Vysledek je: ');
  for i := L downto 1 do write (X [i]);
  write ('. ');
  for i := 1 to M do write (Z [i]);
  writeln;
end; (vypis)
begin
  nacteni;
  prevod;
  vypis;
end.

```

```

5  REM*****
10 REM PREVOD REALNYCH CISEL Z P-SOUSTAVY DO Q-SOUSTAVY
15 REM*****
20 REM
25 REM
30 DIM Y(40): DIM X$(40): DIM Z$(40)
35 DATA A,B,C,D,E,F
40 FOR I=1 TO 6: READ HZ$(I): NEXT I
45 REM VSTUPY:
50 INPUT "ZADEJ ZAKLAD VYCHOZI SOUSTAVY:";P
55 INPUT "ZADEJ ZAKLAD VYSLEDNE SOUSTAVY";Q
60 PRINT "ZADEJ PREVADENE CISLO:";INPUT A$:N=LEN(A$)
61 REM
62 REM *****
63 REM PREVOD Z P-SOUSTAVY DO DESITKOVE SOUSTAVY
64 REM *****
80 FOR I=1 TO 40: X$(I)="0": Z$(I)="0": NEXT I
100 IX=1: FOR I=N TO 1 STEP -1:Z$=MID$(A$,I,1): IF Z$="."
    THEN GOTO 110
105 X$(IX)=Z$:IX=IX+1:GOTO 115
110 M=N-1
115 NEXT I
140 IF M>0 THEN N=N-1
145 FOR I=1 TO N:Y(I)=VAL(X$(I)):IF X$(I)="0" THEN GOTO 155
150 IF Y(I)=0 THEN GOSUB 415
155 NEXT I
175 CC=Y(N):FOR I=N-1 TO M+1 STEP -1:CC=CC*P+Y(I):NEXT I
194 IF M=0 THEN GOTO 225
195 ZC=Y(1):FOR I=2 TO M:ZC=ZC/P+Y(I):NEXT I: ZC=ZC/P
200 REM
205 REM *****
210 REM PREVOD Z DESITKOVE SOUSTAVY DO Q-SOUSTAVY
215 REM *****
225 K=1: PM=INT(CC/Q): PC=CC-PM*Q
245 X$(K)=STR$(PC)
250 IF PC<10 THEN GOTO 280
255 GOSUB 455: X$(K)=R$
260 CC=PM:K=K+1
265 PM=INT(CC/Q): PC=CC-PM*Q: X$(K)=STR$(PC)
270 IF PC<10 THEN GOTO 280
275 GOSUB 455: X$(K)=R$
280 IF PM>=Q THEN GOTO 260
295 IF PM=0 THEN GOTO 310
300 K=K+1: PC=PM: X$(K)=STR$(PC): IF PC<10 THEN GOTO 310
305 GOSUB 455: X$(K)=R$
310 N=1
315 PM=ZC*Q
320 PZ=INT(PM):Z$(N)=STR$(PZ): IF PZ < 10 THEN GOTO 330
325 PC=PZ: GOSUB 455: Z$(N)=R$
330 IF PM<1 THEN GOTO 340
335 ZC=PM-PZ: GOTO 345
340 PZ=0:ZC=PM:Z$(N)=STR$(0)
345 N=N+1:IF ZC=0 THEN GOTO 370
350 GOTO 315
355 REM
360 REM TISK VYSLEDKU
365 REM *****
370 PRINT "VYSLEDEK PREVODU CISLA:";A$
375 PRINT "ZE SOUSTAVY:";P;" DO SOUSTAVY:";Q;" JB: ";
380 FOR I=K TO 1 STEP -1: PRINT X$(I);: NEXT I
385 PRINT ",,": FOR I=1 TO N: PRINT Z$(I);: NEXT I
390 STOP
395 REM
400 REM PODPROGRAM PRO NALEZENI DEKAD. EKVIVALENTU
405 REM HEX. CIFRY
410 REM *****
415 K=1
420 WHILE (HZ$(K)<>X$(I)) AND (K<=6)
425 K=K+1
430 WEND
431 Y(I)=K+9
435 RETURN
440 REM PODPROGRAM PRO STANOVENI HEX. CIFRY
445 REM Z DEKADICKEHO EKVIVALENTU
450 REM *****
455 I=1
460 WHILE (PC<>9+I) AND (I<=6)
465 I=I+1
470 WEND
471 R$=HZ$(I)
475 RETURN

```

Poznámky k symbolům použitým na obr.1:

M1, ..., Mn	moduly nižší úrovně (mohou to být i jednotlivé příkazy)
B	Booleovský (dvouhodnotový) výraz
not B	negace B
+	splnění podmínky B = true
-	splnění podmínky B = false
V	výraz, který může nabývat více hodnot
č.ř.	číslo řádku



číslo konstrukce	Typ konstrukce	Zápis				
		vývojovým diagramem	Warnier-Orrovým diagramem	Nassi-Schneidermanovým diagramem	v Pascalu	v Basicu
1	sekvence		$\left. \begin{array}{l} 1.M1 \\ 2.M2 \\ \vdots \\ n.Mn \end{array} \right\}$		<pre>begin M1; M2; . . . Mn end</pre>	<pre>5 M1 10 M2 . . . Č.ř. Mn</pre>
2	zkrácený podmíněný příkaz		$\left. \begin{array}{l} \text{alt} \\ /?B \end{array} \right\} M1$		<pre>if B then M1</pre>	<pre>IF B THEN M1</pre>
3	úplný podmíněný příkaz		$\left. \begin{array}{l} 1.M1 \\ \text{alt} \\ (0,1)/?B \\ \otimes \\ 2.M2 \end{array} \right\}$		<pre>if B then M1 else M2</pre>	<pre>5 IF B THEN M1: GOTO 15 10 M2 15 ...</pre>
4	větvení do několika směrů		$\left. \begin{array}{l} 1.M1 \\ /?V=1 \\ \otimes \\ 2.M2 \\ /?V=2 \\ \otimes \\ \vdots \end{array} \right\} \text{alt}$		<pre>case V of 1: M1 2: M2 . . . n: Mn end</pre>	<pre>5 IF V=1 THEN M1: GOTO Č.ř. 10 IF V=2 THEN M2: GOTO Č.ř. 15 IF V=3 THEN M3: GOTO Č.ř. . . . Č.ř.</pre>
5	cyklus testovaný nahoře		$\left. \begin{array}{l} \text{cykl} \\ (0,N)/?B \end{array} \right\} M$		<pre>while B do M</pre>	<pre>5 WHILE NOT B 10 M 15 WEND</pre>
6	cyklus testovaný dole		$\left. \begin{array}{l} \text{cykl} \\ (1,N)/?B \end{array} \right\} M$		<pre>repeat M until B</pre>	<pre>5 M 10 WHILE NOT B 15 M 20 WEND</pre>
7	cyklus s pevným počtem opakování těla cyklu		$\left. \begin{array}{l} \text{cykl} \\ (0,N) \end{array} \right\} M$		<pre>for I:=1 to N do M</pre>	<pre>5 FOR I = 1 TO N: M: NEXT I</pre>
8	vyskočení z cyklu		není možné zapsat	není možné zapsat	<pre>for I:=1 to N do begin M1; if B then goto 10 M2 end 10: ... . . .</pre>	<pre>5 FOR I=1 TO N 10 M1: IF B THEN GOTO 20 15 M2: NEXT I 20 ... . . .</pre>

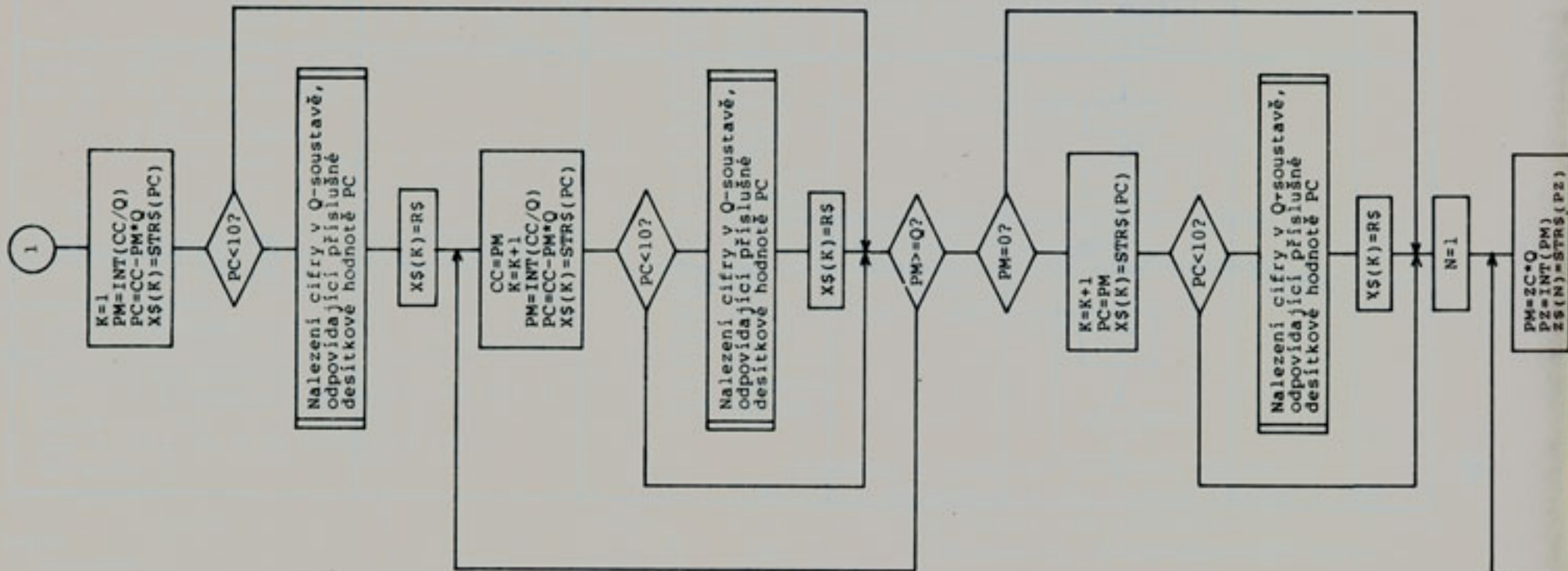
Obr.1 Zápis základních programových konstrukcí



Obr. 2

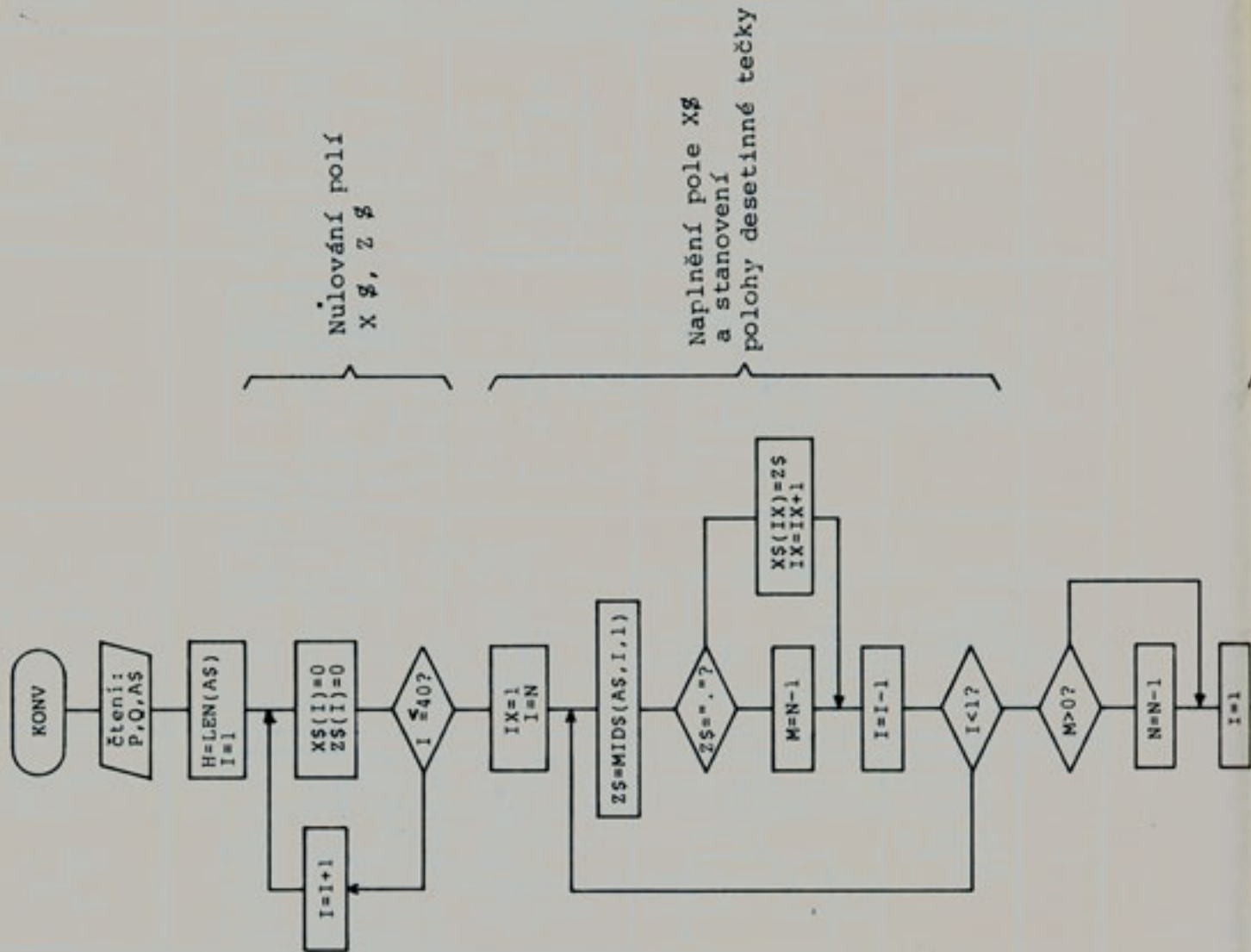
b/ Vývojový diagram algoritmu pro převod z desítkové do Q-soustavy

a/ Vývojový diagram algoritmu pro převod z P-soustavy do desítkové



zápis  
poslední  
číslíce  
do pole XS

Naplnění XS  
znaky cifer  
celé části  
v Q-soustavě

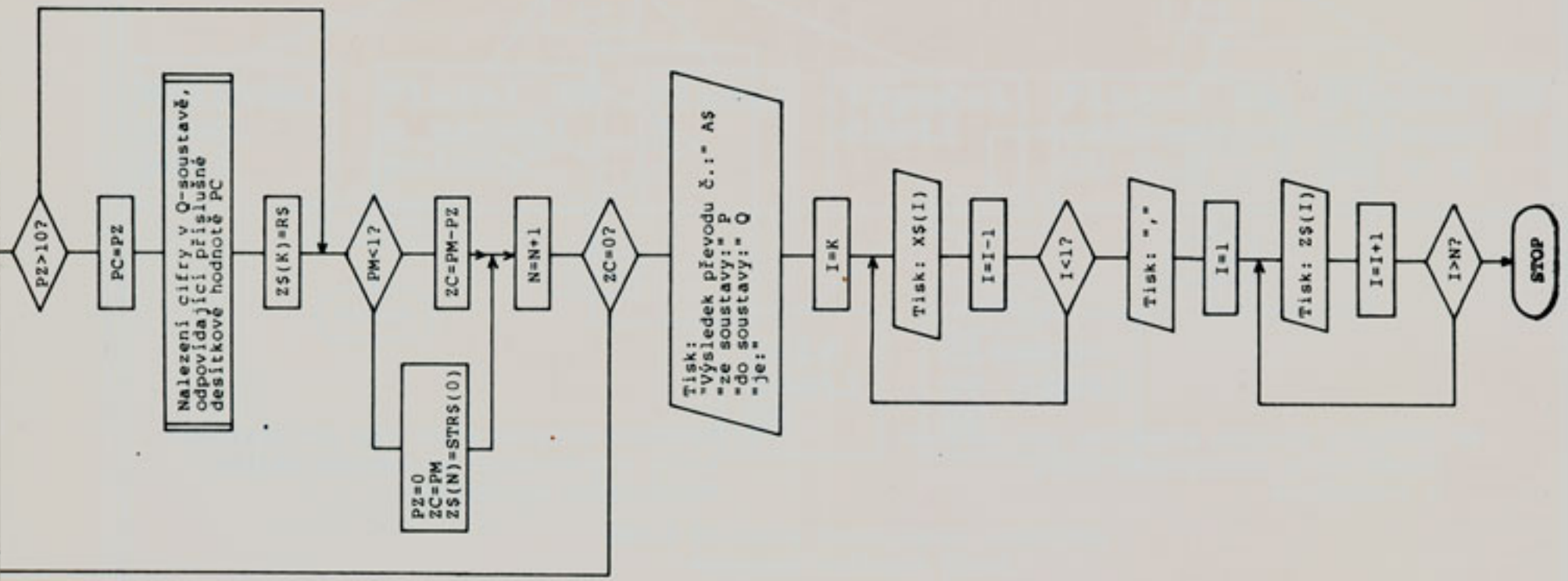


Nulování polí  
X \$, Z \$

Naplnění pole X\$  
a stanovení  
polohy desetinné tečky



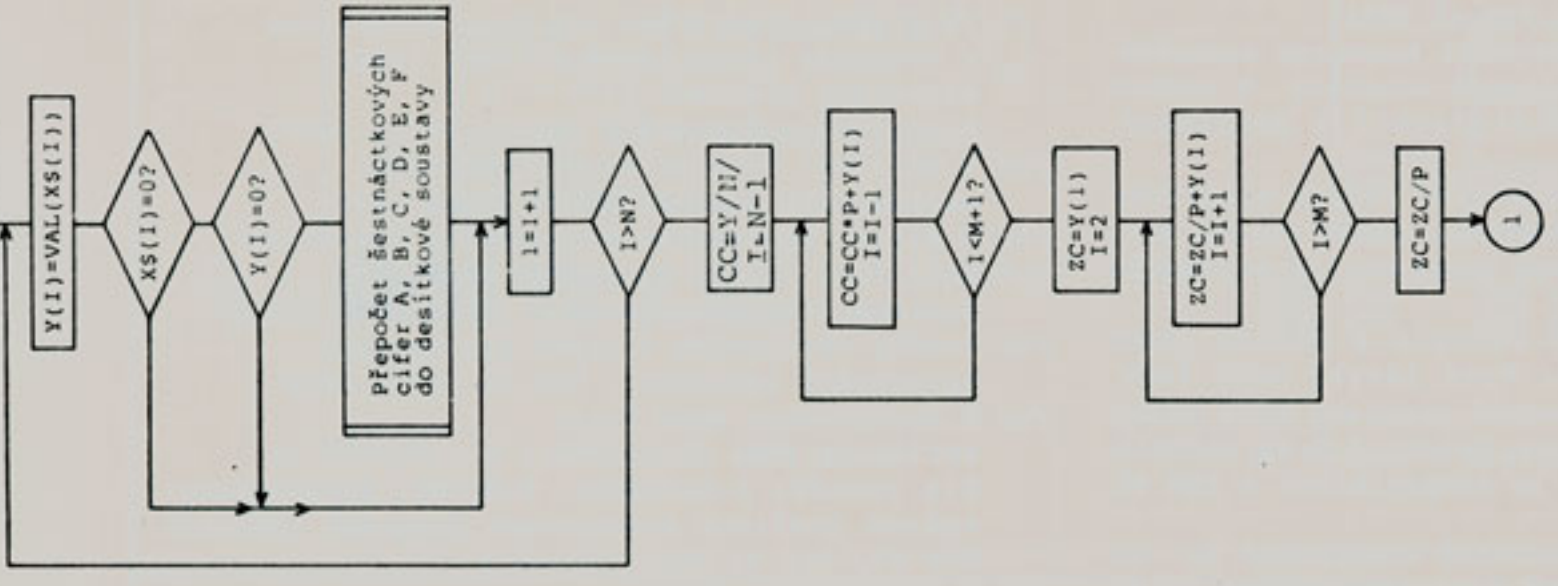
ZŠ(N)=STR\$(ZC)



Naplnění ZS znaky cifer zlomkové části v Q-soustavě

Tisk celé části výsledku

Tisk desetinné části výsledku



Naplnění pole Y číselnými hodnotami, které odpovídají jednotlivým znakům pole XS

Výpočet celé části CC

Výpočet zlomkové části ZC



6. Výpočet zlomkové části ZC v desítkové soustavě (řádek 195):

$$ZC = \sum_{i=1}^M Y_i * P^i$$

7. Naplnění pole X\$ hodnotami odpovídajícími jednotlivým cifrám celé části čísla, vyjádřeného v soustavě Q (řádky 255 až 305). Jednotlivé cifry jsou získány jako posloupnost zbytků při postupném dělení celých částí základem číselné soustavy Q.

8. Naplnění pole Z\$ hodnotami odpovídajícími cifrám zlomkové části v soustavě Q (řádky 315 až 350). Jednotlivé cifry jsou získány jako posloupnost celých částí součinů zlomkových částí a základu číselné soustavy Q.

9. Tisk výsledků (řádky 370 až 385).

V bodech 4, 5, 6 se provádí konverze zadaného čísla z P-soustavy do desítkové soustavy. V bodech 7, 8 se vypočítané číslo převádí z desítkové soustavy do Q-soustavy.

Odvolávky v závorkách slovního zápisu algoritmu se týkají basicového programu. Zápis celého algoritmu vývojovým diagramem je na obr.2, stromovým diagramem na obr.4.

Příslušné programy v Pascalu a Basicu jsou uvedeny v tab.1 a tab.2. Tyto jazyky mají řadu různých implementací. Uvedené programy byly psány v jazycích GWBasic a Turbo Pascal 4.0 na osobním počítači Novotec, který je kompatibilní s IBM PC XT.

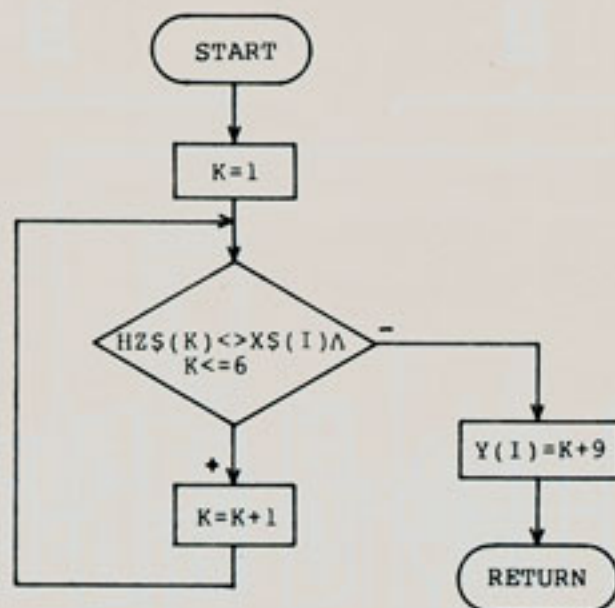
Jako příklad nestrukturované konstrukce je na obr.1 uvedena konstrukce č.8, kterou můžeme upravit na konstrukci č.6, resp. č.7 - podle toho, zda je či není v logickém výrazu vyhodnocována proměnná, jejíž hodnota se příkazem SI mění. Pro opakování cyklu je potom testována podmínka: "not B and I.N" pro cyklus testovaný nahoře (while) nebo "B or I = N" pro cyklus testovaný dole

(until). Toto spojení podmínek pro ukončení cyklu je použito v podprogramech pro nalezení dekadického ekvivalentu hexadekadické cifry a obráceně (viz obr.2c,d a řádky programu v Basicu: 415-435 a 455-475). V pascalovském programu je převod z číselné do znakové reprezentace řešen s využitím funkce ord, která v použité modifikaci Basicu není implementována. Zápis příslušných algoritmů Warnier-Orrovými diagramy (obr.3a,b) a Nassi-Schneidermanovým struktogramem (obr.5a,b) odpovídá programu v Pascalu (tab.3).

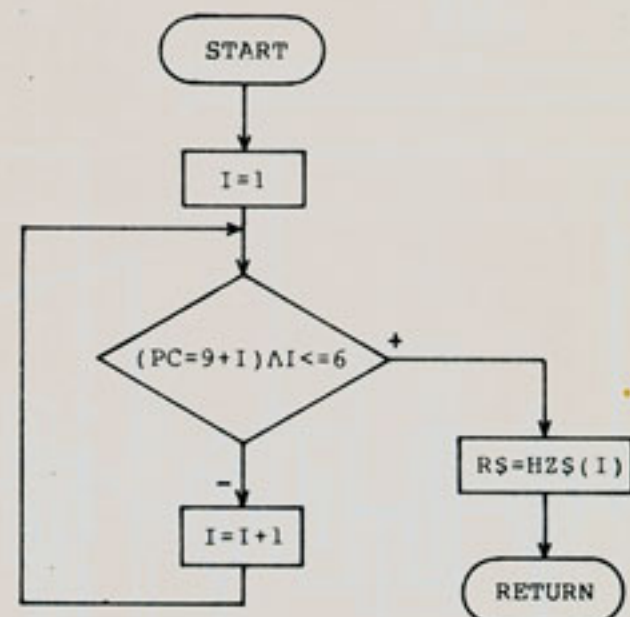
Daniela Stopková  
Jiří Rytych  
Michal Kavale

#### Literatura

- /1/ N.Wirth: Algorithms and Data Structures = Programs (Prentice-Hall Inc. 1976); slovenský překlad: Algoritmy a štruktúry údajov (Alfa 1987)
- /2/ V.Rajlich: Úvod do teorie počítačů; matematický seminář SNTL, Praha 1979
- /3/ N.Wirth: Systematic Programming (Prentice-Hall Inc. 1973); slovenský překlad: Systematické programování (Alfa/SNTL 1981)
- /4/ Značky vývojových digramů pro systémy zpracování informací, ČSN 36 9030, 1974
- /5/ R.Pecinovský: Akce Karel seriál ve VTM, 2-15/1985  
Viz rovněž: J.Kofránek, P.Novák: Kopenogramy-způsob grafické dokumentace programů (sborník MOP 87, díl 4, str. 133-161)
- /6/ R.Pecinovský: Pascal seriál v časopise Elektronika, r.1988
- /7/ K.Hausen: Data Structured Program Design (Prentice-Hall Inc. 1986)
- /8/ J.Herman: Modulární návrh algoritmů a Nassi-Schneidermanovy struktogramy ASŘ - bulletin INORGA, XIX, 3/1985



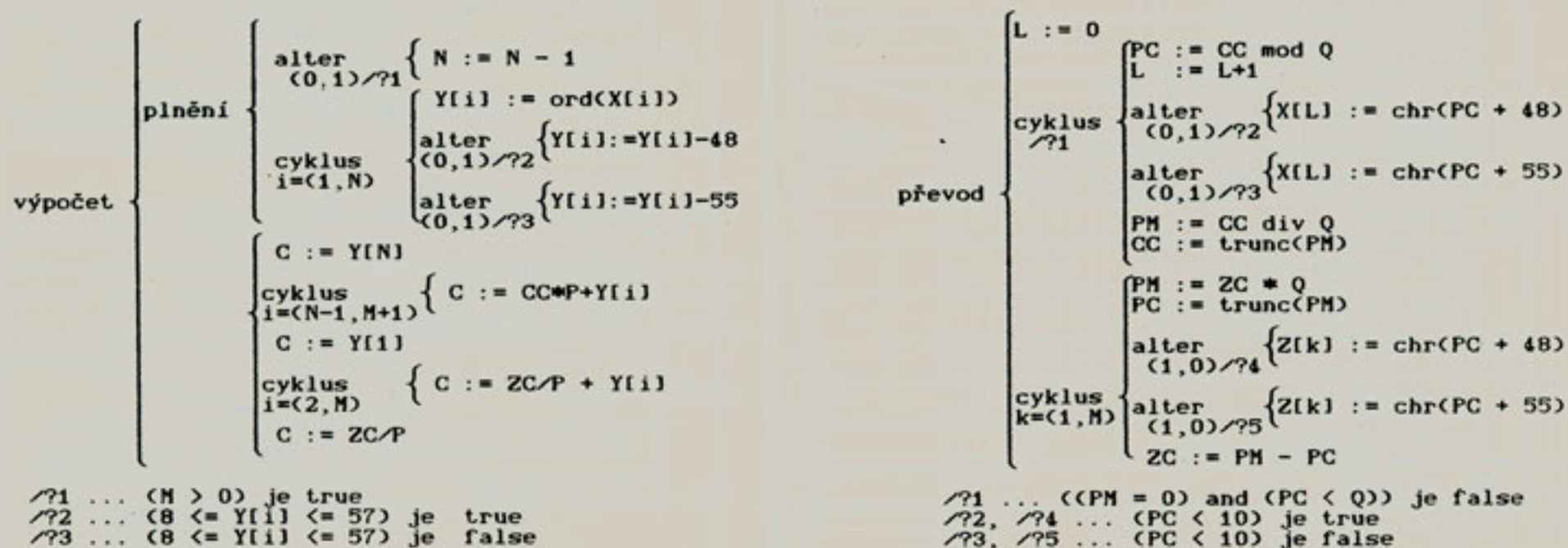
c/ Vývojový diagram algoritmu pro nalezení dekadického ekvivalentu cifry v P-soustavě



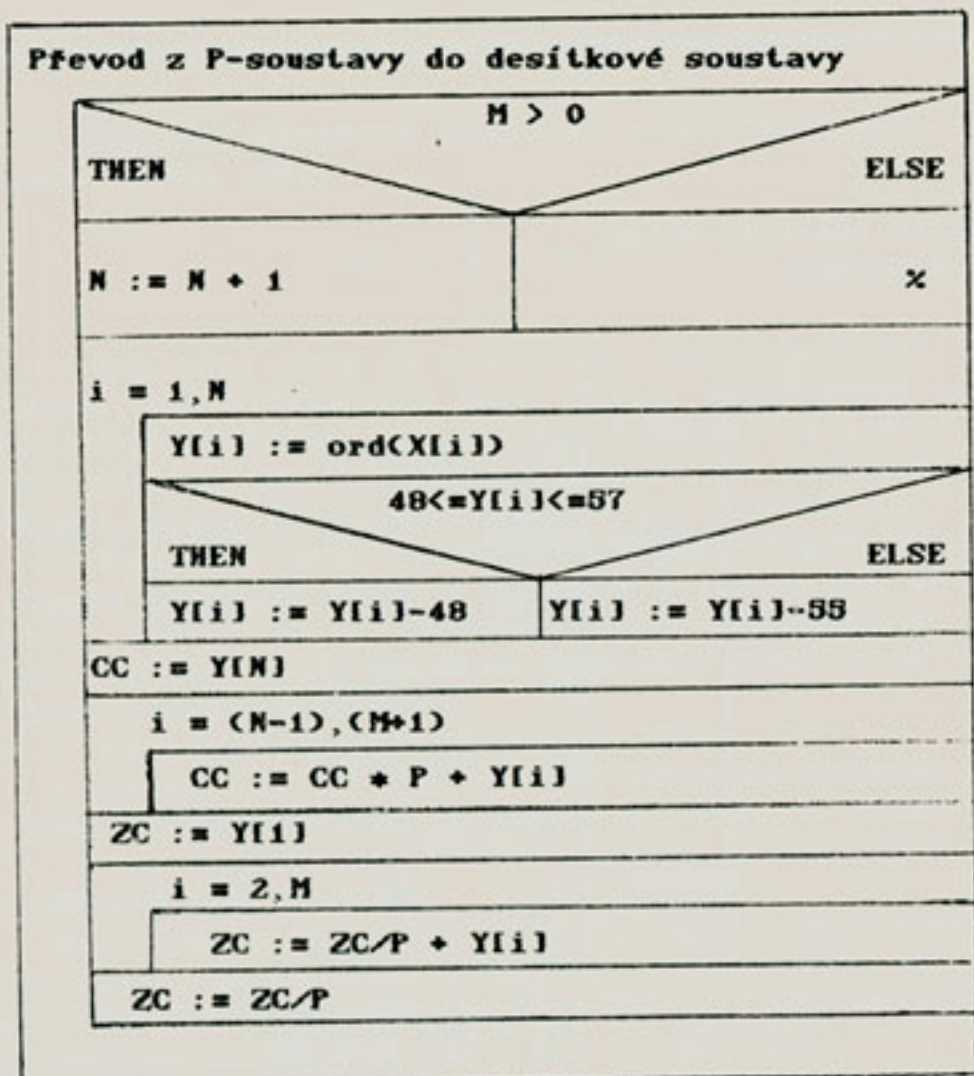
d/ Vývojový diagram algoritmu pro stanovení cifry v Q-soustavě na základě jejího dekadického ekvivalentu



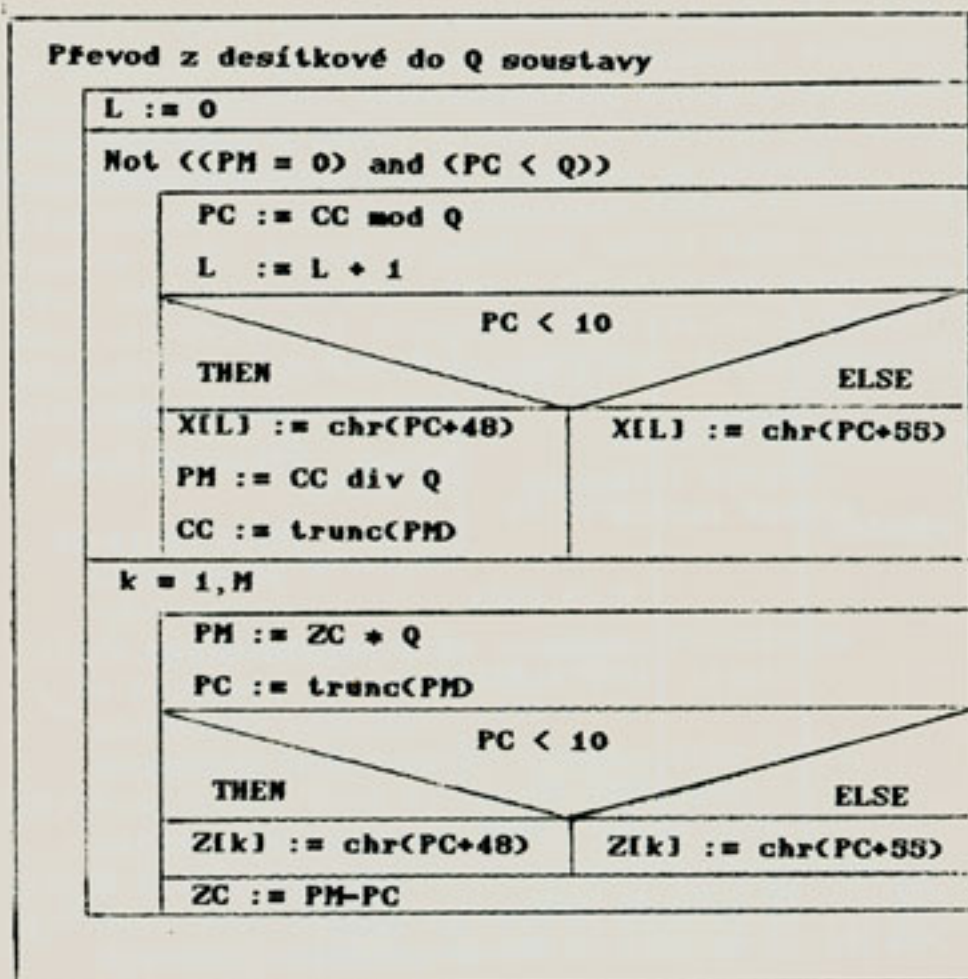
Obr.3 Znázornění konverze reálných čísel Warnier-Orrovým diagramem







a/ Převod z P-soustavy do desítkové



b/ Převod z desítkové do Q-soustavy

# PROGRAMÁTOR EPROM 8708

Programátor EPROM 8708 využívá kanál GPIO PMD 85, který je vyvedený pomocí dvou 20-vývodových konektorů FRB /2/.

Popis vlastností elektricky programovatelné paměti MHB 8708 je v /1/ dostatečně obsáhlý. Proto se s ním nebudeme zdržovat. Podstatné je zajistit všechny tři prováděcí stavy, a to:

- čtení
- programování
- stav velké impedance

Všechny tyto funkce jsou zajištěny řízením pěti budičů UCY7417 z GPIO PMD 85. Potřebné stavy pro jednotlivé režimy přehledně zachycují tabulky 1 a 2.

Programovací impuls +25 V přiváděný na vstup PRG MHB 8708 je tvořen obvodem se dvěma budiči 7417 a tranzistory T1, T2 /1/.

Data z paměti řídicího počítače jsou programem přenášena na výstup GPIO (14 až 20). Adresa je tvořena řídicím programem v rozsahu 0000H až 03FFH a odevzdávána na GPIO (11 až 20).

## Konstrukce a oživení

Po osazení desky plošného spoje všemi součástkami postačí kontrola napájecích napětí a stavů na vstupech UCY 7417 podle tab. 1 a 2. Zavedením programu do řídicího počítače a jeho odstartováním je programátor připraven k použití.

## Závěr

Navrhované řešení umožňuje naprogramovat EPROM 8708 během cca 140 s. Jeho výhodou je jednoduchá přizpůsobitelnost každému systému s mikroprocesor-



rem 8080 a možnost konstrukční úpravy pro programování EPROM 2716 a 2732. Ovládací program je napsán v Basicu G a umožňuje tyto operace:

- zadávání dat z klávesnice
- zadávání dat z mgf záznamu
- výpis zadaných dat na EPS
- výpis zadaných dat na displej
- opravu vložených dat
- nahrání vložených dat na mgf
- přečtení a výpis obsahu programované EPROM na displeji
- naprogramování EPROM
- ukončení programu

Část programu, která zapisuje do EPROM, je v assembleru - viz výpis 1.

Po rozšíření programátoru programovatelným portem MHB 8255 a adekvátním přizpůsobením řídicího programu lze programátor použít i v jiných systémech.

Ing. Peter Pulc

### Literatura

- /1/ Sdělovací technika 5/84
- /2/ Uživatelská příručka PMD 85 III.

### Použitý materiál

R1	5k6	TR 112
R2	1k8	TR 112
R3,4	47	TR 112
R5,7	100	TR 112
R6	270	TR 112
C1	3n3	TK 723
IO	UCY 7417	
T1	KF 517	
T2	KF 507	
D1,2	KA 5..	
patka	TX 728 5251	
konektor	TY 530 2011	2 ks
S1	ISOSTAT	

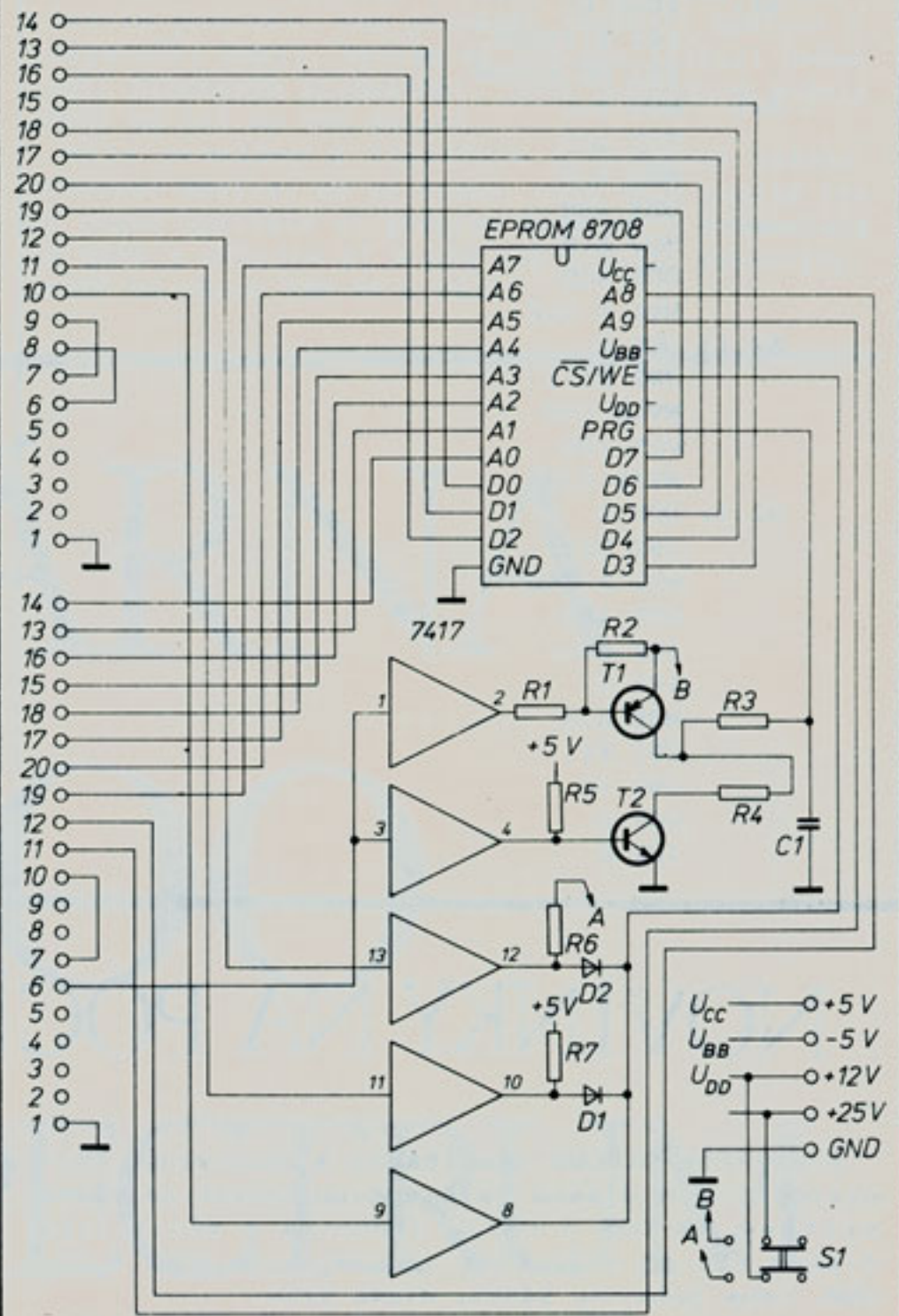
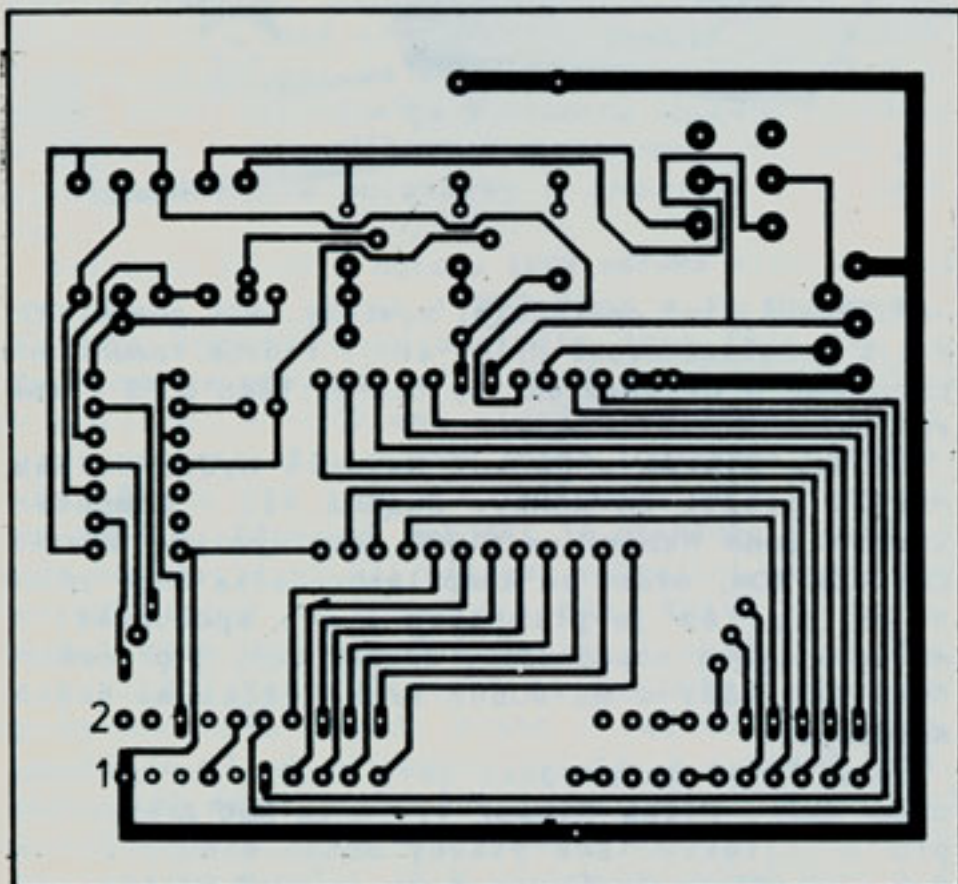
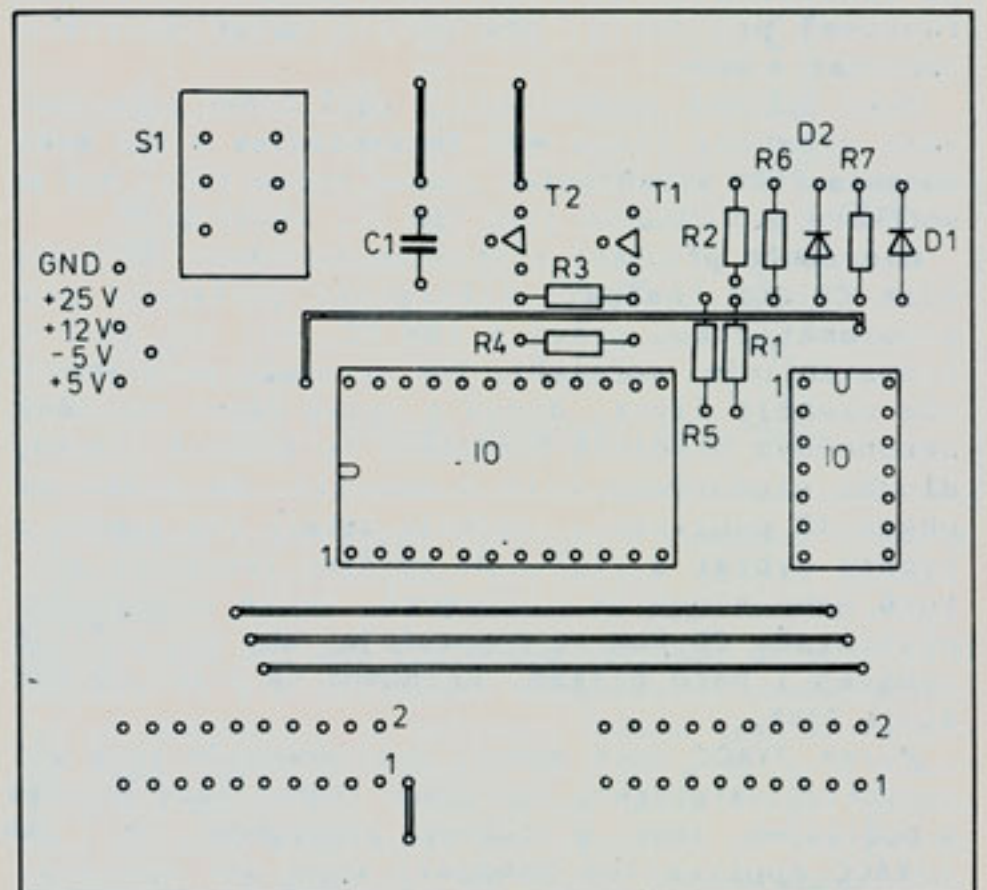


Schéma programátoru EPROM 8708 GPIO PMD 85



Plošný spoj programátoru EPROM 8708



Rozmístění součástek programátoru EPROM 8708



**Výpis ovládacího programu  
pro zápis do EPROM**

```

LXI D,0000H
S1:  PUSH D
      LXI B,7000H
      LXI MH,6000H
S2:  MOV A,C
      OUT 4DH
      MOV A,B
      OUT 4EH
      MOV A,M
      OUT 4CH
      MVI D 01H
S2.1: DCR D
      JNZ S2.1
      MVIA 05H
      OUT 4FH
      MVID 07H
S2.2: DCR D
      JNZ S2.2
      MVI A 04H
      OUT 4FH
      INX B
      INX H
      MOV A,H
      CPI 64
      JNZ S2
      POP D
      INX D
    
```

```

MOV A,D
CPI 03H
JC S1
MOV A,E
CPI E9H
JNZ S1
RET
    
```

**Tabulka 1**

číslo vstupu UCY 7417	stavy vstupů UCY 7417		
13	1	0	0
11	1	0	1
9	1	0	1
režim 8708	progr.	čtení	3.stav

**Tabulka 2**

číslo vstupů UCY 7417	stavy vstupů	
1,3	1	0
úroveň výstupu (V)	0	+25

# NOVINKY NA POČÍTAČOVÉM TRHU

Subsystém DISCUS Rewritable s mazatelným magneto-optickým diskem může spolupracovat se všemi systémy MS-DOS a OS/2. Jednotka s měnitelným diskem 650 MB může číst záznamy typu WORM a CD-ROM. Cena jednotky \$4995, disku \$250.

Kurta Corp. přichází s novou snímací tabulkou (tablet), na níž jeden bod skutečně odpovídá jednomu bodu obrazovky. Lze volit i taková měřítka, kdy plocha tabulky bude odpovídat zvolené ploše okna na obrazovce monitoru. Tablet se dodává s potřebným softwarem ve třech velikostech: 8,5 x 11 (\$395), 12 x 12 (\$595) a 12 x 17 palců (\$965). Na tabulkách je 23 tlačítek s programovatelnými funkcemi pro makra, aby nebylo nutné neustále operovat s menu.

Díky optické technologii a patřičnému programovému vybavení firma MSC Technologies u své myši dosáhla 2000 kroků/palec (dosud kolem 200). Myš má software pro všechny typy IBM PC, stojí \$149.

Microsoft přichází s vtípnou novinkou. Na jeden disk CD-ROM "nacpal" literaturu potřebnou pro programátorskou práci s IBM PC, XT, AT, PS/2 a kompatibilními počítači. Tak se snaží programátorům ulehčit život, dosud z velké části vyplněný nekonečným hledáním hledaného ve spoustě (leckdy dlouho hledaných) knih a manuálů. Na disku je obsah 48 publikací. Jakýkoli detail lze snadno a rychle vybrat a vložit do většiny textových editorů nebo slovních procesorů. Vedle nezbytného přehrávače CD-ROM se doporučuje, aby byl počítač vybaven i hard diskem. Knihovna na disku CD-ROM stojí \$395.

Firma JYACC chce pomoci programátorům zase při tvorbě uživatelského interfacu (komunikace člověka s počítačem, resp. s vloženým programem). Její JAM (JYACC Application Manager) toho umí spoustu. Hlavně však umožňuje tvorbu zmíněného interfacu bez jakéhokoli jazykového programování. Nastavíte si okna, jejich obsah (menu), čím se co volí, jak co na sebe navazuje atd. Výsledek "slinkujete" se svým programem. Mimo to si JAM rozumí se všemi



nejužívanějšími operačními systémy. Při pomyslení, kolik programátorské dřiny stojí tvorba komunikace programu s uživatelem, by člověk těch \$750 firmě rád dal.

Děláte systém v Céčku? Nejspíš byste ho pak chtěli uložit do ROMky. Snadná věc - Computer Innovations nabízí za \$1300 celý vývojový systém C86PLUS/ROM, ovšem za kompilátor Céčka chce ještě \$750. Systém je připraven i pro spolupráci s matematickými koprocory 80x86, tedy i procesory 8088/86. Běží v MS-DOSu. Kromě Céčka má ještě assembler.

Cena \$1499 za program-dává tušit, že už bude něco umět. Firma Gimeor vyvinula Mac Architrion pro architektky. Své stavby mohou simulovat ve dvou- i třírozměrném prostoru, v němž si výsledek svého úsilí mohou prohlížet skoro odkudkoli libo. Program má několik novinek, podstatně usnadňujících grafickou práci. Spolupracuje i s daty slovních procesorů a spreadsheetů.



Svět, ve kterém žijeme, je nelineární. Jak se s tím vědecky vyrovnat, je problém. Pomocnou ruku podává Engineering Software Concepts svým programem Simnon/PC2.11 (Simulation Language for Non-linear Systems). V podstatě jde o editor pro matematické výpočty a kompilátor, jehož výsledkem je strojový kód procesoru 8088/86. Firma slibuje i možnost tvorby ekonomických modelů (kde jeden skoro pořád neví). Podle informace by mohlo jít o určitý nástin modelování synergetických procesů. Program pracuje se všemi péčečky IBM včetně PS/2, nutný je matematický koprocessor, doporučuje se 640K RAM. Cena \$695.

State of the Art přichází s dalším programovým

modulem pro účetnické práce na počítačích (i v jejich síti) s procesory 80286 a 80386. Celá série se jmenuje Master Accounting System 90, poslední produkt má název Evolution/2. Jde o systém, který dává absolutní přehled o všem, co souvisí s účetnictvím podniku. Lze provádět jakékoli operace s jakýmkoli údaji. Systém má ochranu (nikoli ovšem proti hloubavým studentům), jeho jednotlivé moduly stojí od 195 do 995 dolarů.

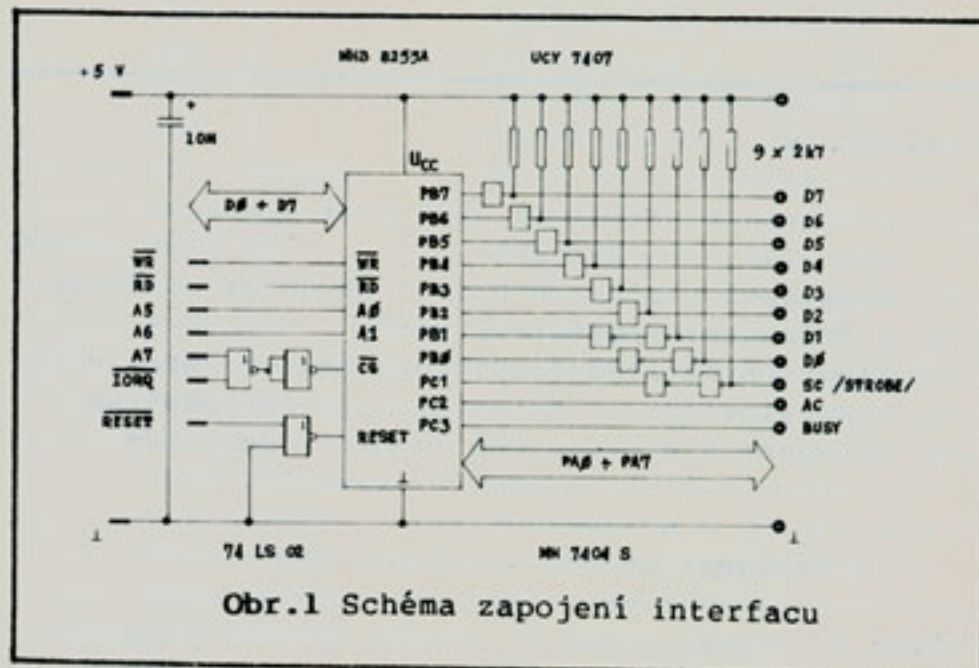
Máte problémy s finančním plánováním vašeho provozu? Nevíte, co se stane, když dáte peníze tam či onam? Opatřete si program Ronstadt's Financials za \$499 a dělejte si s ním své vlastní expertízy. Chodí na všech IBM PC a PS/2.

# TISKÁRNA D-100 a ZX SPECTRUM

Relativně dostupná polská maticová tiskárna D-100 je vhodná pro práci s mikropočítači. Dodává se pro různé typy připojení podle několika norem: paralelní typy LOGABAX, CENTRONICS, IRPR nebo sériový V24. Náš příspěvek popisuje hardwarové řešení připojení ZX Spectra k tiskárně D-100 v normě IRPR. Zabýváme se i drobnou úpravou tiskárny a nakonec uvádíme příklad strojového programu pro vytvoření kopie obrazovky o rozměrech 13 x 18,5 cm.

Tiskárnu D-100 s normou IRPR můžeme alternativně využívat pro práci s PMD 85-2 a ZX Spectrem. Základ koncepce paralelního interfacu pro ZX Spectrum je převzat z /1/. Celá modifikace spočívá v posílení a oddělení signálů vystupujících z programovatelného obvodu 8255A logickými převodovými členy YES typu UCY 7407 nebo páry invertorů MH 7404 S. Adresový dekodér je nezměněn a využívá 3/4 obvodu 74 LS 02, přičemž zbývající hradlo NOR je přístupné na konektoru FRB. Takto upravený modul sice jednoznačně definuje celý port B a signál PC1 jako výstupní (v souladu s použitým programováním obvodu 8255A v režimu 1), avšak umožňuje přímé připojení periférie v normě IRPR. Pro další experimentování s obousměrnou komunikací (např. hodiny reálného času, joystick, fonémový syntezátor apod.) je k dispozici volný port A a některé bity portu C. Zapojení a funkce obvodu jsou zřejmé ze schématu.

Návrh oboustranné desky plošných spojů vychází také z /1/ a je rozšířený o uvedenou úpravu.



Obr.1 Schéma zapojení interfacu

Přiložená vyobrazení znázorňují samotnou desku a rozmístění součástek. Rezistory jsou v poloze nastojato těsně při konektoru FRB. Horní volné konce jsou spojeny společným vodičem, připojeným k napětí +5 V v blízkosti blokovacího kondenzátoru 10 uF. Technicky je zařízení řešeno jako samostatný blok vnějších rozměrů 98 x 82 x 16 mm z plastického materiálu. Blok je na jedné straně zakončen nožovým konektorem pro nasunutí na systémovou sběrnici ZX Spectra, na druhé straně konektorem FRB pro spojení s tiskárnou. Konektor je zapojen podle tabulky. Pro připojení tiskárny je použit 22-žilový plochý kabel, přičemž každý nepárový vodič je signálový (data, SC, AC, BUSY) a každý



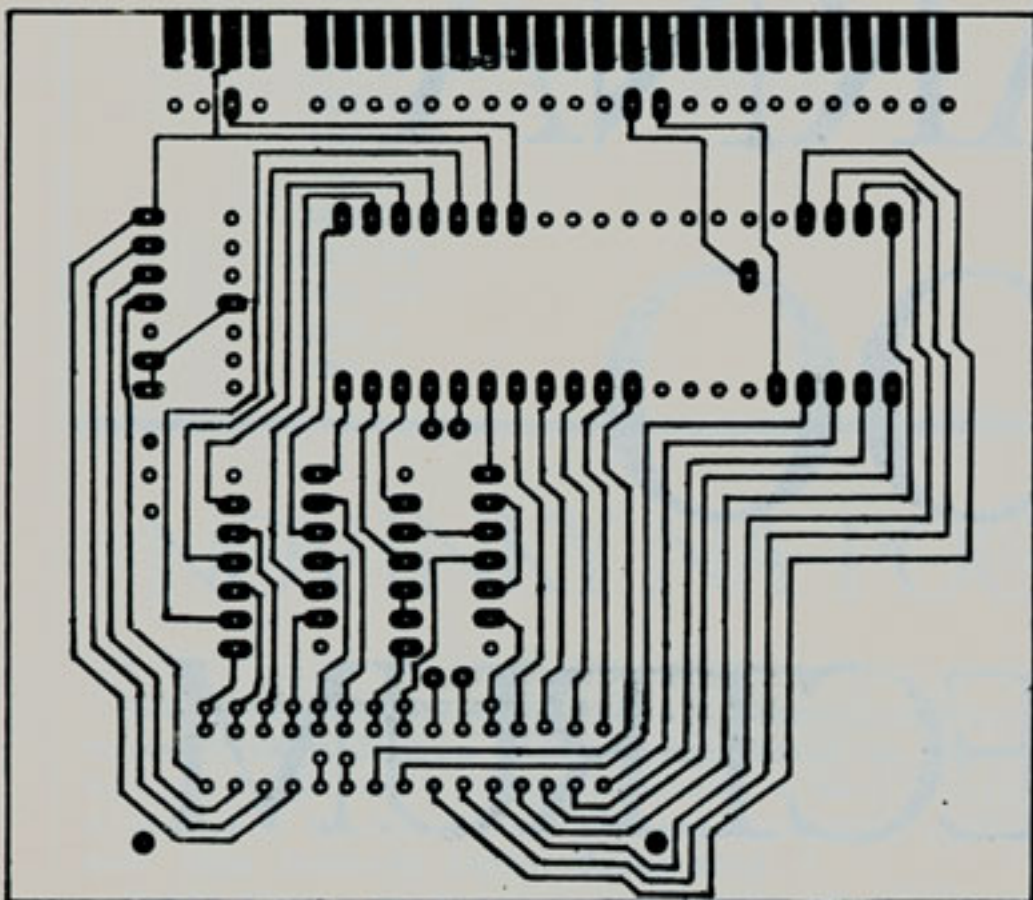
párový na potenciálu 0V, kterým jsou obě zařízení propojena.

Při práci s tiskárnou D-100 vybavenou bufferem 2 KB se osvědčila úprava a doplnění tiskárny o tlačítko inicializace logiky a nulování bufferu s indikací funkce diodou LED. Po přerušení programu obsluhy tiskárny je tak umožněno okamžité zastavení činnosti tiskárny, která by v ní jinak pokračovala až do vyprázdnění paměti. Schéma úpravy je zřejmé z obrázku a pramene /2/. Ovládací tlačítko (např. Izostat) můžeme umístit v pravém dolním rohu (z čelního pohledu) a signální diodu vlevo od trojice ostatních LED ve stejné vzdálenosti.

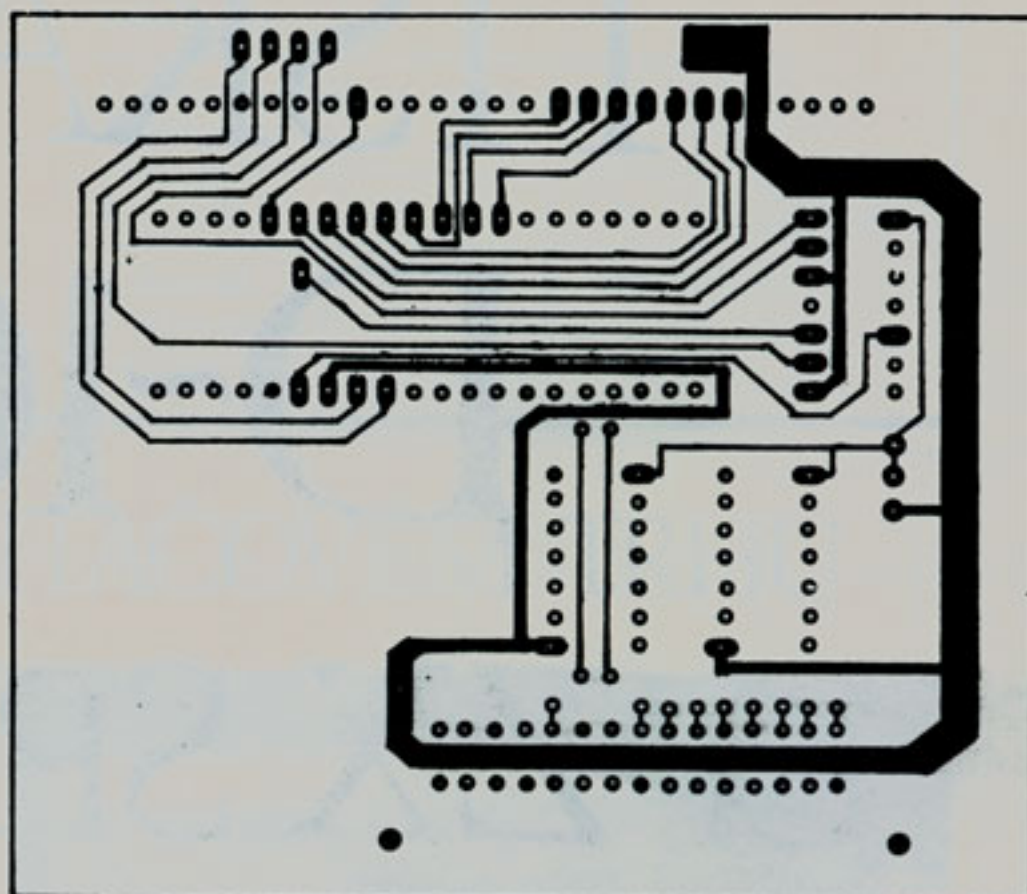
Pro výpis basicových programů (LLIST) i pro ostatní činnost (LPRINT) se osvědčil program PRINTER uvedený v /1/. Program je dobré umístit od adresy 23296 (printer buffer ZX Spectra). Vy-

zkoušeli jsme i funkci podprogramu COPY, ale pro mnohé aplikace je výsledná velikost obrazové kopie 65 x 93 mm nedostatečná. Proto jsme vytvořili vlastní program LXCOPY pro maximální velikost kopie. Jeho funkce je zřejmá z komentovaného assemblerového výpisu. Program v délce 232 bajtů načteme od adresy 50000. Test 5. bitu systémové proměnné FLAGS (23611) slouží okamžitému přerušení činnosti programu při stisku jakékoli klávesy /4/ a pro návrat do interpreteru Basicu s chybovým hlášením OK. Tisk se spouští obvykle z Basicu povelom RANDOMIZE USR 50000. Protože software využívá podprogramy POINT z ROM ZX Spectra /5/, nejsou mu dostupné poslední dva editační řádky. Pro kopírování obsahu obrazovky včetně nich by bylo třeba formulovat jiný algoritmus.

Ing. Ivan Růčka, Ondrej Lexa



Obr.2 Návrh plošných spojů - strana spojů



Obr.3 Návrh plošných spojů - strana součástek

#### Literatura:

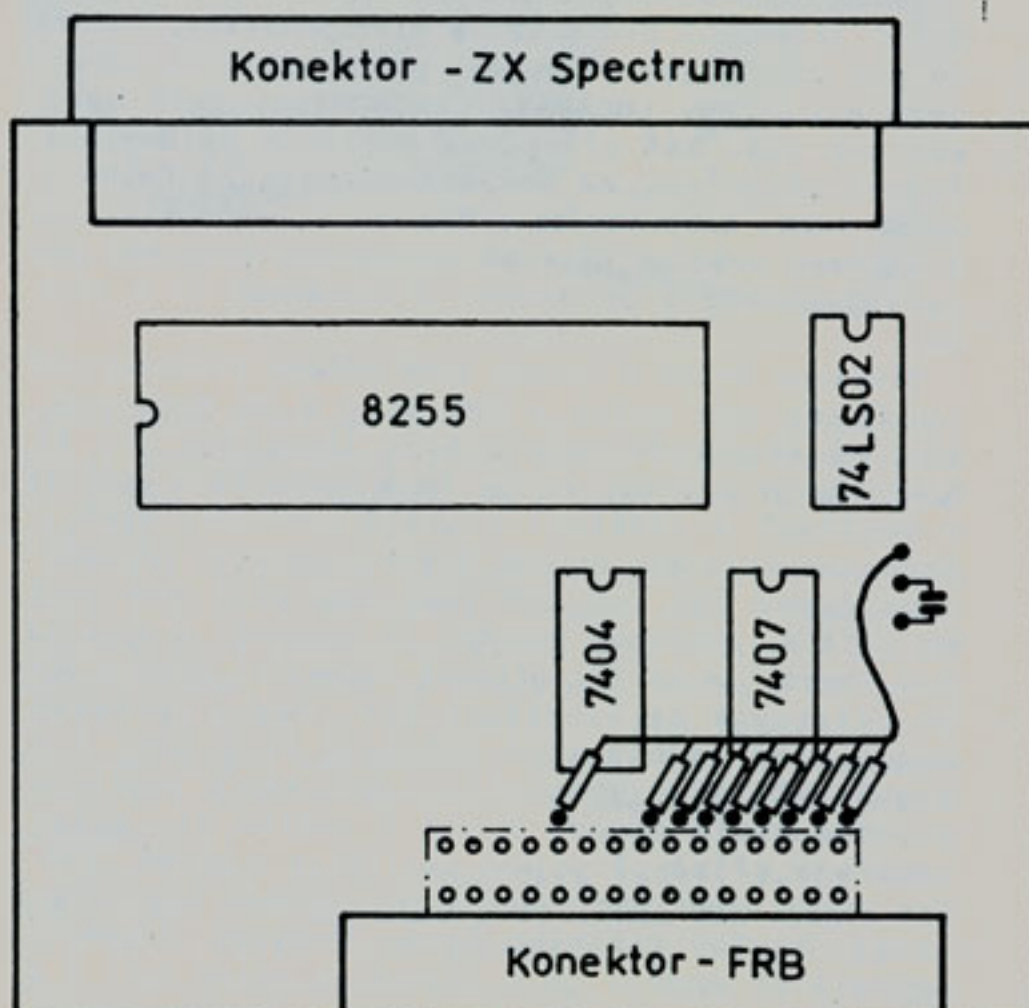
- /1/ P. Formánek: Tiskárna D100 a ZX Spectrum ARA 7/88, str. 257-260
- /2/ Mozaiková tlačiareň D100 - návod na použitie
- /3/ M. Štěpánek: Prostorové grafy Elektronika 1987 (nabídkové číslo), str. 30-33
- /4/ K. Kuryłowicz...: Przewodnik po ZX Spectrum
- /5/ J. Logan, F. O'Hara: The Complete Spectrum ROM Disassembly

#### Zapojení vývodů konektoru FRB

vývod č. signál    vývod č. signál    vývod č. signál

1	PA1	11	PA6	21	GND
2	PC6	12	AC	22	D3
3	PA2	13	PA7	23	+5 V
4	PC5	14	BUSY	24	D4
5	PA3	15	PA0	25	OUT
6	PC4	16	D0	26	D5
7	PA4	17	PC7	27	IN-1
8	PC0	18	D1	28	D6
9	PA5	19	GND	29	IN-2
10	SC/STROBE	20	D2	30	D7

Pozn.: Využit je signál BUSY vývodu č. 33 konektoru D-100

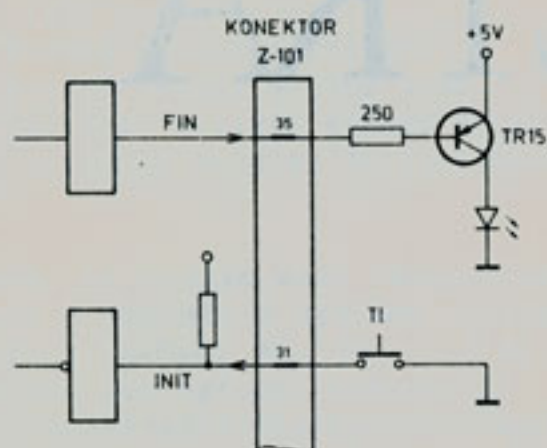


Obr.4 Rozmístění součástek



Obr.5 Schéma úprav tiskárny D-100

LED signalizuje vynulovanou paměť tiskárny D-100. Log.úroveň L na vývodu 31 konektoru Z-101 tiskárny D-100 způsobí inicializaci a nulování paměti.



```

;*****
;*** PROGRAM LXCOPY PRO KOPII OBRAZOVKY ***
;*****
ORG 50000 ;Adresa spuštění programu

XOR A ;Vynulování akumulátoru
LD (23611),A ;Inicializace FLAGS

HARDCO LD E,175 ;Souřadnice Y 1.bodu
LD D,0 ;Souřadnice X 1.bodu
LD A,157 ;Mód 1 obvodu 8255
OUT (127),A
LD A,13 ;Řídící kód CR
CALL OUTI
LD A,27
CALL OUTI
LD A,48 ;Posun papíru 10ř./2,54 cm
CALL OUTI
LD A,27
CALL OUTI
LD A,49 ;Přechod na grafický režim
CALL OUTI

HARD1 CALL RADOK1
LD A,13 ;CR
CALL OUTI
DEC E ;Zmenšení souřadnice Y o 3
DEC E
DEC E
CALL RADOK2 ;Párové řádky
LD A,13 ;CR (LF je automatický)
CALL OUTI
DEC E ;Nastavení kresby bodů
DEC E ;na novou pozici
DEC E ;(8.bod, 15.bod...)
DEC E
LD A,E ;Nová souřadnice Y
CP 0 ;Už je konec obrazovky?
JP NZ,HARD1 ;Když NE, pokračuj
RET ;ANO, pak návrat do Basicu

```

```

RADOK1 LD B,128 ;Maska 1.bitu=data
CALL POINT
JP Z,DAL1 ;Skok, když INK,
SET 6,B ;jinak 2krát PAPER
SET 5,B
DAL1 DEC E ;Zmenšení souřadnice Y o 1
CALL POINT
JP Z,DAL2 ;Skok, když INK,

```

```

SET 4,B ;jinak 2krát PAPER
SET 3,B
DAL2 DEC E ;Totéž opakovaně
CALL POINT
JP Z,DAL3
SET 2,B
SET 1,B
DAL3 DEC E
CALL POINT
JP Z,DAL4
SET 0,B ;Zde jen 1krát PAPER
DAL4 LD A,B ;Grafický bajt do reg.A
CALL OUTI ;Trojnásobný výstup
CALL OUTI ;grafického vzoru
CALL OUTI ;(optimum pro D-100)
INC E ;Návrat k hodnotě
INC E ;počátečního stavu
INC E ;souřadnice Y
INC D ;Zvětšení souřadnice X o 1
RET Z ;Řádka hotova? ANO, návrat
JP RADOK1 ;Když NE, pokračuj

RADOK2 LD B,128 ;Maska 1.bitu=data
CALL POINT
JP Z,NAS1 ;Skok, když INK,
SET 6,B ;jinak 1krát PAPER
NAS1 DEC E ;Zmenšení souřadnice Y o 1
CALL POINT
JP Z,NAS2 ;Skok, když INK,
SET 5,B ;jinak 2krát PAPER
SET 4,B
NAS2 DEC E ;Totéž opakovaně
CALL POINT
JP Z,NAS3
SET 3,B
SET 2,B
NAS3 DEC E
CALL POINT
JP Z,NAS4
SET 1,B
SET 0,B
NAS4 LD A,B ;Grafický bajt do reg.A
CALL OUTI ;Trojnásobný výstup
CALL OUTI ;grafického vzoru
CALL OUTI
INC E ;Návrat k hodnotě
INC E ;počátečního stavu
INC E ;souřadnice Y
INC D ;Zvětšení souřadnice X o 1
RET Z ;Řádka hotova? ANO, návrat
JP RADOK2 ;Když NE, pokračuj

POINT PUSH BC
LD A,D ;Souřadnice X do reg.A
PUSH DE ;Uschování souřadnic
CALL #2D28 ;Souřadnice X do zás.kalk.
POP DE
LD A,E ;Souřadnice Y do reg.A
PUSH DE ;Uschování souřadnic
CALL #2D28 ;Souřadnice Y do zás.kalk.
CALL #22CB ;Je bod X,Y INK/PAPER?
CALL #2DD5 ;INK 1 do A/PAPER 0 do A
POP DE
CP 1 ;Při INK Z=1, jinak Z=0
POP BC
RET ;Návrat ze subrutiny

OUTI PUSH AF ;Úschova grafického bajtu
LD A,(23611) ;Obsah FLAGS do reg.A
BIT 5,A ;Je stisknuto tlačítko?
JR Z,TRY ;Když NE, pokračuj
RST 8 ;ANO, pak návrat do Basicu
DEFB #FF ;Kód chybového hlášení
TRY IN A,(95) ;Test připravenosti D-100
BIT 3,A ;BUSY?
JR Z,TRY ;Když nepřipravena, skok
POP AF ;Grafický bajt do reg.A
OUT (63),A ;a jeho výstup na port PB
RET ;Návrat ze subrutiny

```



# KALKULÁTOR

## ZX SPECTRA

### (3)

#### Jiné operace kalkulátoru

ofset, název - činnost

**3BH,fp-calc-2** - používá se tehdy, když se požaduje provedení jediné aritmetické operace. Ofset požadované operace vložíme do reg.B a zadáme:

```
RST 0028H,FP-CALC
DEFB 3BH,fp-calc-2
DEFB 38H,end-calc
```

To je velmi výhodné, neboť uvedenou posloupnost instrukcí můžeme uložit do smyčky a před každým průchodem smyčkou můžeme změnou ofsetu v reg.B požadovat jiný typ kalkulátorové operace, bez přímého zásahu do režimu kalkulátoru.

**3CH,e-to-fp** - Transformuje číslo tvaru xEm na floating-point číslo. Před přivoláním tohoto podprogramu musí být x uložené jako poslední hodnota zásobníku kalkulátoru a m musí být v reg.A. Výsledek nahradí původní poslední hodnotu.

Pozn.: Tuto operaci nelze realizovat kalkulátorem přímo. Způsob vyvolání:

- uložit na vrchol x
- obsadit reg.A hodnotou m
- obsadit reg.B ofsetem operace (3CH)
- CALL 2D4FH,E-TO-FP

**00H,jump-true** - Realizuje podmíněný relativní skok. Za literálem 00H musí následovat parametr operace - tzv. "adresový" bajt. Jeho dvojkový doplněk určuje velikost relativního skoku. Skok se uskuteční, je-li před přivoláním operace poslední hodnota zásobníku kalkulátoru nenulová. V opačném případě program pokračuje interpretací literálu, který následuje za "adresovým" bajtem. Operace 'jump-true' zároveň vykoná operaci 'delete', čímž sníží adresu aktuálního vrcholu zásobníku kalkulátoru o 5. "Odstraní" tak původní poslední hodnotu, která určovala, zda se má nebo nemá provést skok. Novou poslední hodnotou se stává původní předposlední hodnota.

```
relativní adresa ... 2 -1 00 +1 +2 +3 +4...
"adresový" bajt ...FE FF 00 01 02 03 04...
```

```
posl. hodn. = 0 ...xx xx 00 03 xx xx xx...
```

```
posl. hodn. <> 0 ...xx xx 00 03 xx xx xx...
```

**33H,jump** - realizuje nepodmíněný relativní skok, jehož velikost je dána (podobně jako u předcházející operace) "adresovým" bajtem za literálem 33H. Operace adresu aktuálního vrcholu zásobníku kalkulátoru neovlivní.

```
relativní adresa ...-3 -2 -1 00 +1 +2 +3 +4
"adresový" bajt ...FD FE FF 00 01 02 03 04
```

```
program a) ...xx xx xx 33 03 xx xx xx
                | | ↑
```

```
program b) ...xx xx xx 33 04 xx xx xx
                | | ↑
```

```
program c) ...xx xx xx 33 FD xx xx xx
                ↑ | |
```

**35H,dec-jr-nz** - operace je obdobou instrukce DJNZ; slouží k vytváření smyček. Před přivoláním této operace je třeba obsadit reg.B počtem opakování. Za ofsetem operace (35H) musí následovat "adresový" bajt, určující místo návratu do smyčky. Hodnota čítače se po přivolání operace přesune do systémové proměnné BREG, která pak vlastně slouží jako čítač počtu opakování.

Předpokládejme, že před přivoláním operace obsahuje reg.B hodnotu 3:

```
relativní adresa ...-4 -3 -2 -1 00 +1 +2 +3
"adresový" bajt ...FC FD FE FF 00 01 02 03
```

```
chod 1 (B=3) ...xx xx xx xx 35 FC xx xx
                ↑ | |
```

```
chod 2 (B=2) ...xx xx xx xx 35 FC xx xx
                ↑ | |
```

```
chod 3 (B=1) ...xx xx xx xx 35 FC xx xx
                ↑ | |
```

```
konec (B=0) ...xx xx xx xx 35 FC xx xx
                | | ↑
```

**38H,end-calc** - ukončí činnost kalkulátoru. Bajt za literálou 38H je už interpretovaný jako instrukce mikroprocesoru Z80.

**86H,series-06**

**88H,series-08**

**8CH,series-0C** - tyto operace se používají pro výpočet hodnot transcendentních funkcí SIN, EXP, LN, ATN a nepřímo i pro výpočet hodnot všech funkcí, které jsou z nich odvozené (např. COS, TAN a jiné). Výpočet funkce se realizuje rozvinutím do řady podle Čebyševových polynomů, které označujeme T1, T2, ... Tn. Čebyševovy polynomy jsou definovány rekurentně:

$$T_{n+1} = 2 * Z * T_n - T_{n-1}, \text{ přičemž } T_1=1, T_2=Z$$



Prvních dvanáct Čebyševových polynomů má tvar:

T1 = 1  
 T2 = Z  
 T3 = 2Z-1  
 T4 = 4Z-3  
 T5 = 8Z<sup>2</sup>-8Z+1  
 T6 = 16Z<sup>3</sup>-20Z<sup>2</sup>+10Z  
 T7 = 32Z<sup>4</sup>-48Z<sup>3</sup>+18Z<sup>2</sup>-1  
 T8 = 64Z<sup>5</sup>-112Z<sup>4</sup>+56Z<sup>3</sup>-7Z  
 T9 = 128Z<sup>6</sup>-256Z<sup>5</sup>+160Z<sup>4</sup>-32Z<sup>3</sup>+1  
 T10 = 256Z<sup>7</sup>-576Z<sup>6</sup>+432Z<sup>5</sup>-120Z<sup>4</sup>+9Z  
 T11 = 512Z<sup>8</sup>-1280Z<sup>7</sup>+1120Z<sup>6</sup>-400Z<sup>5</sup>+50Z<sup>4</sup>-1  
 T12 = 1024Z<sup>9</sup>-2816Z<sup>8</sup>+2816Z<sup>7</sup>-1231Z<sup>6</sup>+220Z<sup>5</sup>-11Z

Offset 86H, za kterým musí následovat šest konstant (jsou v operačním systému ve tvaru zdrojového čísla) v pořadí A6, A5, ..., A1, vypočítá hodnotu výrazu pro vstupní hodnotu Z:

$$T = 2 * (T1 * A1 + T2 * A2 + T3 * A3 + T4 * A4 + T5 * A5 + T6 * A6)$$

Podobným způsobem offset 88H vyhodnotí výraz, který obsahuje polynomy T1 až T8, konstanty A1 až A8 a offset 8CH výraz s polynomy T1 až T12 a konstantami A1 až A12. Hodnota Z je vstupní a hodnota T výstupní hodnotou podprogramu SERIES GENERATOR (3449H), který je skupinovým podprogramem pro offsety 86H, 88H a 8CH.

Na velikosti konstant An závisí typ aproximované funkce a na počtu použitých členů její přesnost. Konstanty můžeme vypočítat ze vztahu:

$$A_n = \frac{1}{\pi} \int_{-1}^1 f * T_n / \sqrt{1 - Z^2} dz$$

kde f je aproximovaná funkce (např. LN Z, SIN Z, ...).

Funkce SIN: konstanty A1 = +1.276278962  
 -----  
 A2 = -0.142630785  
 A3 = +0.004559008  
 A4 = -0.000068294  
 A5 = +0.000000592  
 A6 = -0.000000003

Úvodní úprava:

$$y = (x / (2 * \pi) - \text{INT}(x / (2 * \pi) + 0.5)) * 4$$

Když ABS(y) > 1, pak y = 2 \* SGN(y) - y

Touto úpravou se transformuje hodnota argumentu do intervalu -1 až +1. Je možné dokázat, že:

$$\text{SIN}(\pi * y / 2) = \text{SIN } x$$

Vstup do SERIES GENERATOR: 2 \* y<sup>2</sup> - 1

Výstup z " " SIN(PI\*y/2)/y

Závěrečná úprava: výstupní hodnota se vynásobí hodnotou upraveného argumentu y.

Funkce EXP: konstanty A1 = +1.456999875  
 -----  
 A2 = +0.248762432  
 A3 = +0.021446556  
 A4 = +0.001235714  
 A5 = +0.000053453  
 A6 = +0.000001851  
 A7 = +0.000000053  
 A8 = +0.000000001

Úvodní úprava:

$$e^x = 2^y \quad x = y * \text{LN } 2 \quad y = x / \text{LN } 2$$

$$e^x = 2^{(\text{INT } y) * 2^{(\text{FRC } y)}}$$

Vstup do SERIES GENERATOR: FRC y

Výstup z SERIES GENERATOR: 2 (FRC y)

Závěrečná úprava: k exponentovému bajtu hodnoty 2<sup>(FRC y)</sup> se připočte hodnota INT y.

Funkce LN: konstanty A1 = +0.9302292213  
 -----  
 A2 = -0.0818414567  
 A3 = +0.0094766116  
 A4 = -0.0012282837  
 A5 = +0.0001693953  
 A6 = -0.0000243013  
 A7 = +0.0000035828  
 A8 = -0.0000005389  
 A9 = +0.0000000823  
 A10 = -0.0000000127  
 A11 = +0.0000000020  
 A12 = -0.0000000003

Vstupní úprava: číslo x rozdělíme na dvojkový exponent a mantisu:

a) LN x = LN(x \* 2<sup>e</sup>) = LN x' + e \* LN 2 = LN ma + ex \* LN 2  
 b) LN x = LN(x \* 2<sup>e</sup>) = LN x' + e \* LN 2 - LN 2 + LN 2 = LN(2 \* x') + (e - 1) \* LN 2 = LN ma + ex + LN 2

Mantisa 'full floating-point' čísla je vždy v intervalu (0.5..1). Když je větší než 0.8, použijeme rozdělení:

a) ma = x' ex = e'

Když je menší nebo stejná 0.8, použijeme rozdělení:

b) ma = 2 \* x' ex = e' - 1

příčemž ma je mantisa a ex je dvojkový exponent čísla x.

Transformací (5 \* ma/2) - 3 převedeme mantisu do intervalu -1 až +1. Před přivoláním SERIES GENERATOR jsou pod poslední hodnotou v zásobníku kalkulátoru vyčísleny výrazy ex \* LN 2 a ma - 1. Poslední hodnotou je vstupní hodnota pro SERIES GENERATOR.

Vstup do SERIES GENERATOR: (5 \* ma/2) - 3

Výstup z SERIES GENERATOR: (LN ma)/(ma - 1)

Závěrečná úprava:

$$((\text{LN } ma) / (ma - 1)) * (ma - 1) + ex * \text{LN } 2 = \text{LN } x$$

Funkce ATN: konstanty A1 = +0.8813735870  
 -----  
 A2 = -0.0529464623  
 A3 = +0.0055679210  
 A4 = -0.0006905375  
 A5 = +0.0000978715  
 A6 = -0.0000131076  
 A7 = +0.0000019105  
 A8 = -0.0000002850  
 A9 = +0.0000000432  
 A10 = -0.0000000066  
 A11 = +0.0000000010  
 A12 = -0.0000000002

Vstupní úprava: Protože funkce ATN je definovaná na množině reálných čísel (-nekonečno, +nekonečno), transformujeme její argument nejdříve do intervalu (-1..+1). Využijeme přitom vztahy:

$$\text{ATN } x = \text{ACTN } (1/x) = \pi/2 - \text{ATN}(1/x) = \pi/2 + \text{ATN}(-1/x)$$

pro x > 1

$$\text{ATN } x = -\pi/2 + \text{ATN}(-1/x)$$

pro x < -1

$$\text{ATN } x = \text{ATN } x$$

pro ABS x > 1

Do zásobníku kalkulátoru před přivoláním operace SERIES GENERATOR vložíme postupně směrem k jeho vrcholu hodnoty:

$$w, y \text{ a } 2 * y^2 - 1$$

$$\text{Pro } x > 1 \quad w = \pi/2 \quad y = -1/x$$

$$\text{Pro } x < -1 \quad w = -\pi/2 \quad y = -1/x$$

$$\text{Pro } \text{ABS} > x \quad 1 \quad w = 0 \quad y = x$$

Vstup do SERIES GENERATOR: 2 \* y<sup>2</sup> - 1

Výstup z SERIES GENERATOR: (ATN y)/y

Závěrečná úprava: ATN(x) = ((ATN y)/y) \* y + w



# ARMSTAD/SCHNEIDER

## TABULKA SYSTÉMOVÝCH PROCEDUR

(3)

81	BBF3	1824 - 1790 - 1794	Vstup: reg.HL - požadovaná pozice počátku. Výstup: mění AF, HL.
-----			
Čte bod v relativních souřadnicích (TESTR). Vstup: reg.DE - přírůstek x, reg.HL - přírůstek y souřadnic bodu. Výstup: A obsahuje barvu bodu, mění BC, DE, HL.			
82	BBF6	1839 - 17A5 - 17A9	Vstup: A - vyšší bajt adresy počátku obrazové paměti. Výstup: mění AF, HL.
-----			
Kreslí čáru v absolutních souřadnicích od pozice graf. kurzoru do koncového bodu (DRAW). Vstup: reg.DE - souřadnice x, reg.HL - souřadnice y koncového bodu. Výstup: Mění AF, BC, DE, HL.			
83	BBF9	1836 - 17A2 - 17A6	Vstup: - Výstup: reg.A - vyšší bajt adresy paměti obrazovky, reg.HL - Offset.
-----			
Kreslí čáru v relativních souřadnicích, od pozice graf. kurzoru do koncového bodu (DRAWR). Vstup: reg.DE - přírůstek x, reg.HL přírůstek y souřadnic koncového bodu. Výstup: mění AF, BC, DE, HL.			
84	BBFC	1945 - 193C - 1940	Vstup: reg.A - číslo módu. Výstup: mění AF, BC, DE, HL.
-----			
Promítne znak v místě polohy graf. kurzoru. Vstup: reg.A - znak. Výstup: mění AF, BC, DE, HL.			
-----			
O b s l u h a o b r a z o v k y -----			
85	BBFF	0AA0 - 0ABB - 0ABF	Vstup: reg.A - číslo módu; stav CY, Z dle módu: mód 0 - C=1, Z=0 mód 1 - C=0, Z=1 mód 2 - C=0, Z=0
-----			
Hlavní inicializace obsluhy obrazovky, mód činnosti (mode) inkoust i pozadí dostanou počáteční (default) hodnoty Vstup: - Výstup: mění AF, BC, DE, HL.			
86	BC02	0AB1 - 0ACC - 0AD0	Vstup: - Výstup: reg.B - poslední fyzický sloupec obrazovky, reg.C - poslední řádek obrazovky; mění AF.
-----			
RESET obsluhy obrazovky. Vstup: - Výstup: mění AF, BC, DE, HL.			
87	BC05	0BC3 - 0B33 - 0B37	Vstup: - Výstup: reg.B - poslední fyzický sloupec obrazovky, reg.C - poslední řádek obrazovky; mění AF.
-----			
Nastaví počáteční pozice obrazovky (Screen Set Offset). Před modifikací hodnoty může udělat "scroll" obrazovky.			
88	BC08	0B45 - 0B38 - 0B3C	Vstup: - Výstup: mění AF, BC, DE, HL.
-----			
Nastaví bod počátku obrazové paměti v RAM. Vstup: A - vyšší bajt adresy počátku obrazové paměti. Výstup: mění AF, HL.			
89	BC0B	0B50 - 0B52 - 0B56	Vstup: - Výstup: reg.A - vyšší bajt adresy paměti obrazovky, reg.HL - Offset.
-----			
Odečítá adresu počátečního bodu obrazové paměti a počáteční pozici obrazovky (Offset). Vstup: - Výstup: reg.A - vyšší bajt adresy paměti obrazovky, reg.HL - Offset.			
90	BC0E	0ACA - 0AE5 - 0AE9	Vstup: - Výstup: mění AF, BC, DE, HL.
-----			
Nastaví obrazovku do grafického režimu. Vstup: reg.A - číslo módu. Výstup: mění AF, BC, DE, HL.			
91	BC11	0AEC - 0B08 - 0B0C	Vstup: nemá Výstup: reg.A - číslo módu; stav CY, Z dle módu: mód 0 - C=1, Z=0 mód 1 - C=0, Z=1 mód 2 - C=0, Z=0
-----			
Odečítá číslo módu. Vstup: nemá Výstup: reg.A - číslo módu; stav CY, Z dle módu: mód 0 - C=1, Z=0 mód 1 - C=0, Z=1 mód 2 - C=0, Z=0			
92	BC14	0AF7 - 0B13 - 0B17	Vstup: - Výstup: mění AF, BC, DE, HL.
-----			
Mazání obrazovky. Vstup: - Výstup: mění AF, BC, DE, HL.			
93	BC17	0B57 - 0B59 - 0B5D	Vstup: - Výstup: reg.B - poslední fyzický sloupec obrazovky, reg.C - poslední řádek obrazovky; mění AF.
-----			
Odečítá rozměry obrazovky. Vstup: - Výstup: reg.B - poslední fyzický sloupec obrazovky, reg.C - poslední řádek obrazovky; mění AF.			
94	BC1A	0B64 - 0B66 - 0B6A	Vstup: - Výstup: reg.B - poslední fyzický sloupec obrazovky, reg.C - poslední řádek obrazovky; mění AF.
-----			
Vypočítává adresu znaku na obrazovce (řádka, sloupec).			



Vstup: reg.H - číslo sloupce, reg.L - číslo řádky.

Výstup: reg.HL - adresa paměti obrazovky, reg.B - informace množství bitů na pixel v nastaveném módu obrazovky; mění AF.

95 BC1D OBA9 - OBAB - OBAF

Vypočítává adresu bodu na obrazovce.

Vstup: reg.DE - souřadnice x bodu, reg.HL - souřadnice y, počítáno od levého dolního rohu obrazovky.

Výstup: reg.HL - adresabodu (pixel) v paměti, reg.B - počet bitů na pixel - 1, reg.C - maska bodu; mění AF, DE.

96 BC20 OBF9 - OC01 - OC05

Vypočítává adresu bajtu ležícího vpravo od zadaného.

Vstup: reg.HL - adresa zadaného bajtu.

Výstup: reg.HL - nová adresa; mění AF.

97 BC23 OC05 - OC0D - OC11

Vypočítává adresu bajtu ležícího vlevo od zadaného.

Vstup: reg.HL - adresa zadaného bajtu.

Výstup: reg.HL - nová adresa; mění AF.

98 BC26 OC13 - OC1B - OC1F

Vypočítává adresu odpovídajícího bajtu ležícího o řádek níž.

Vstup: reg.HL - adresa zadaného bajtu.

Výstup: reg.HL - nová adresa; mění AF.

99 BC29 OC2D - OC35 - OC39

Vypočítává adresu odpovídajícího bajtu ležícího o řádek výš.

Vstup: reg.HL - adresa zadaného bajtu.

Výstup: reg.HL - nová adresa; mění AF.

100 BC2C OC86 - OCBA - OCBE

Ma'skuje bajt čísla inkoustu tak, aby byl bod zobrazen v odpovídající barvě.

Vstup: reg.A - číslo inkoustu.

Výstup: reg.A - maska; mění F.

101 BC2F OCA0 - OCA3 - OCA7

Vykonává opačnou funkci než předchozí procedura.

Vstup: reg.A - maska.

Výstup: reg.A - číslo inkoustu; mění F.

102 BC32 OCEC - OCEE - OCF2

Nastavuje barvy inkoustu.

Vstup: reg.A - číslo inkoustu, reg.B - první a reg.C - druhá barva.

Výstup: mění AF, BC, DE, HL.

103 BC35 OD14 - OD16 - OD1A

Zjišťuje barvy inkoustu.

Vstup: reg.A - číslo inkoustu.

Výstup: reg.B - první, reg.C - druhá barva; mění AF, DE, HL.

104 BC38 OCF1 - OCF3 - OCF7

Nastavuje barvy rámečku (BORDER).

Vstup: reg.B - první, reg.C druhá barva.

Výstup: mění AF, BC, DE, HL.

105 BC3B OD19 - OD1B - OD1F

Zjišťuje barvy rámečku (BORDER).

Vstup: -

Výstup: reg.B - první, reg.C - druhá barva; mění AF, DE, HL.

106 BC3E OCE4 - OCE6 - OCEA

Nastavuje čas střídání barev rámečku.

Vstup: reg.H - doba první barvy, reg.L - doba druhé barvy.

Výstup: mění AF, HL.

107 BC41 OCE8 - OCEA - OCEE

Odečítá čas střídání barev rámečku.

Vstup: -

Výstup: reg.H - doba první barvy, reg.L - doba druhé barvy; mění AF.

108 BC44 ODB3 - ODB5 - ODB9

Vyplňuje obdélník inkoustem.

Vstup: reg.A - maska odpovídající inkoustu, reg.H - číslo levého, reg.D - pravého sloupce, reg.L - číslo horní, reg.E - dolní řádky.

Výstup: mění AF, BC, DE, HL.

109 BC47 ODB7 - ODB9 - ODBD

Nastavuje počet bajtů v obrazové paměti pro vyplnění obdélníku inkoustem.

Vstup: reg.A - maska odpovídající inkoustu, reg.HL - adresa odpovídající levému hornímu rohu obdélníku, reg.D - počet bajtů, reg.E počet řádek.

Výstup: Mění AF, BC, DE, HL.

110 BCA4 ODDF - ODE1 - ODE5

Zaměňuje dvě barvy znaku.

Vstup: reg.B - maska jedné barvy, reg.C druhé barvy, reg.H - číslo sloupce, reg.L - číslo řádky.

Výstup: mění AF, BC, DE, HL.

111 BC4D ODFA - ODFC - OE00

Přemístí celou obrazovku o osm bodů nahoru či dolů

Vstup: reg.B=0 pro přesun dolů, jinak přesun nahoru, reg.A - maska inkoustu pro nový řádek.

Výstup: mění AF, BC, DE, HL.

112 BC50 OE3E - OE40 - OE44

Přemístí část obrazovky o osm bodů nahoru či dolů.

Vstup: B=0 pro přesun dolů, jinak přesun nahoru, reg.A - maska inkoustu pro novou řadku; reg.H - číslo levého, reg.D - pravého sloupce; reg.L - číslo horní, reg.E - dolní řádky.

Výstup: mění AF, BC, DE, HL.

113 BC53 OEF3 - OEF5 - OEF9

Konverze znakové matrice (není-li obrazovka v módu 2) na sérii masek bodů odpovídajících okamžitému stavu obrazovky.

Vstup: reg.HL - adresa matrice, reg.DE - adresa ramky pro přetvořenou masku.

Výstup: mění AF, BC, DE, HL.

114 BC56 OF49 - OF26 - OF2A

Konverze opačná k předchozí.

Vstup: reg.A - maska inkoustu, reg.H - číslo sloupce, reg.L - číslo řádky, reg.DE - adresa budoucí matrice.

Výstup: mění AF, BC, DE, HL.

(Pokračování příště)



# METAKOMUNIKACE (3)

## JEN SE NEDAT OLOUPIT

Kým? Čím? Počítačem. Obdivuji jeho výkony, když mi pomáhá psát a dostává ze mne myšlenky, o nichž jsem si myslel, že ve mně ještě dávno neuzrály. Těší mě, když mi umožňuje vynechat grafika a - ač na tomto poli laik - doplnit svůj text profesionálně vyhlížejícími kresbami. Ale připadám si jako hňup, který se dává vodit za nos, když mi chce někdo namluvit, že stejně uživatelsky příjemný, urychlující a hýčkající je počítač, chci-li s ním dělat hudbu.

Nejlépe snad bude, když popíšu své trápení v chronologické posloupnosti. Začíná manuálem ke Commodoru 64, kapitolou věnovanou hudbě. Obal tohoto počítače tvrdí jednoznačně, že se v něm skrývá "Musiksynthesizer". Manuál vám dá zakusit pocit Fausta-středoškoláka, který z lahviček s běžnými chemikáliemi chce - jak se říká - v retortě zesyntetizovat umělého človíčka neboli homunkula. Přes poměrování hřebíku v modré skalici nepřeklopýtá...

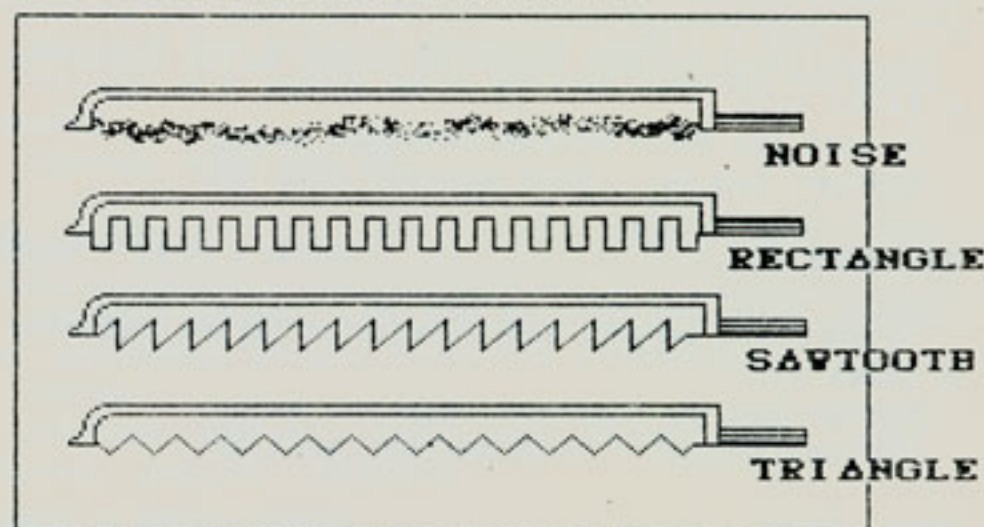
Zůstaňme u té analogie ještě na vteřinku. Každý vzoreček z organické chemie obsahuje písmenka C, O a H. Zkuste ale sebeděle vařit dohromady uhlík s kyslíkem a vodíkem - nezačne to dýchat ani jevit jiné známky života a nezačne. A právě tak v tom manuálu: nastavíte si dolní i horní byty frekvence, hlasitost, jeden ze čtyř tvarů vlny (k čertu, proč právě šum, obdélník, pilu a trojúhelník?), necháte si doporučit, jaký by měl být poměr podivných parametrů A, D, S a R, doplníte čas - a pokud jste nic nepopletli, ono to píská jeden jediný ošklivý, kovový nebo spíš elektrický tón. To není život z retorty ani hudba z počítače. A když pak máte kvůli jedné takhle ošklivě vypískávané melodii naťukat kolik řádků dat, říkáte si, ať jdou s tímhle "syntezátorem" do háje.

Pak najdete trochu lepší návod než ten nešťastný manuál - u mě to byla kniha programů pro C-64 od Franze Endeho - a tam vám systematicky objasní architekturu registrů zvukového čipu SID a záhadnou formuli ADSR doplní nákresem křivky průběhu tónu v čase. Cítíte úlevu, že se s vámi začalo jednat jako s bytostí, která má rozum a své činy chce mít alespoň trochu zdůvodněné. Nicméně o radostném vytváření hudby se mluvit nadále nedá, jen lektor hudební nauky v pozadí textu působí teď tak, že v tom má jasno alespoň on sám.

K tomuto pocitu nepřekonatelné propasti mezi vaší pracně naťukanou programovou řádkou a představou o tom, jak by měl znít jeden takt hudby hodné svého jména, se posléze připojí zjištění, že to zřejmě nějak jde a že jenom ten váš stroj na to nemá. Dostanete do rukou disketu, kde nějací knakeři vybrali ze slavných počítačových her nejlepší hudební kousky, například celou desítku skladeb ze hry "Poslední Minja". Hravě můžete zasahovat do instrumentace hotových skladeb v diskotékově nakadeřeném programu "Kawasaki Rhythm Rocker". Nebo dokonce máte v rukou program zvaný "Music Shop", ve kterém všechna ta kolečka proměnných jsou kapotována a vy si místo psaní programu nabíráte za pomoci myši noty, sázíte je do notové osnovy a vzápětí si je můžete poslechnout. Jenomže zklamání přichází zanedlouho i tady: zatoužíte-li po nějaké efektnější zvukové barvě, stojí před vámi to bestiárum A, D, S, R a všech dalších

znova, jen už nemusíte klopotně vypisovat číselníka a hodnoty proměnných nastavujete jak mistr zvukař u mixážního pultu myši na jakýchsi malých potenciometrech nakreslených v menu.

Pak přiveze někdo ze známých klaviaturu značky Casio nebo Yamaha doplněnou různobarevnými knoflíky a vy zjistíte, že je to vlastně syntezátor bez viditelného počítače. Na rozdíl od loudavého psaní not tady můžete hrát rovnou na klávesnici, jediným knoflíkem si předvolit některou z předem namixovaných nástrojových barev (to, že se vůbec nepodobají zvukové barvě nástrojů, jejichž název nesou, jim v té chvíli odpustíte), další knoflík vám umožní vybrat si některý z rytmů mile připomnělého automatického bubeníka - a už se to konečně začíná podobat hudbě. Ovšem jaké: marná sláva, je to šraml a úroveň šramlu nepřekonává ani okázale pestré demo, jakkoli vám při analytičtější pohledu připadá na tomto elektronickém nástroji jedním člověkem nerealizovatelné.



V této chvíli jste na rozcestí. Musíte si položit otázku, co vlastně do téhle hry hodláte vložit a co z ní chcete dostat. Přiznejme si totiž - pokud nejste zrovna geniální hudebníci - nakolik vás uspokojuje, když usednete ke klavíru, kytaře nebo prostě k tomu nástroji, s nímž jste tak či onak poněkud spjati, ve snaze si zaimprovizovat. Jak dlouho se to dá vydržet, za jak dlouho se začnete točit v kruhu. Ale od počítače jsme chtěli víc. Tak jako nám zůhlednil strojopisnou stránku a usnadnil nám "rytí" v textu a tedy i v myšlenkovém zmatku, tak by z nás měl počítač cedit hudbu s větším úspěchem, než jak se to s námi dosud dařilo tradičním akustickým nástrojům. Alespoň ty elektronické efekty kdyby nám vysypal jak roh hojnosti. Abychom pak z nich mohli jak Brian Eno vytvářet zcela novou hudbu.

Stal jsem se šťastným majitelem zbrusu nové knihy Christiana Quirina Spitznera "Hudební kompendium pro C 64/128" (Markt und Technik, Haar u Mnichova 1988). Je doprovázena disketou a nabízí věci veledákové: rozšíření Basicu na Sound-Basic, monitor zvuků, monitor zvukového čipu, který umožňuje pozorovat v reálném čase křivky vámi právě vytvářeného zvuku, a pro bastlíře návod k výrobě - ze součástek v ceně pod 30 marek - digitizéru zvuku, o něco dražší pak i MIDI. I sklonil jsem se před touto knihou jat úctou a snažil se ji přečíst nejen celou, ale i tak, abych jí skutečně porozuměl. Frustrace přišla i tentokrát, byť z trochu jiné strany. Dokud je téma prostinké, výklad je didakticky skvělý, podrobný, názorný, ba



i vtipný. Běda však, jakmile jde do tuhého. Autoři - neboť knihu zjevně psal celý kolektiv - začínají mluvit v zasvěcencově hantýrce. Co vás nejvíc trápí, to je skutečnost, že celý les proměnných tu zůstává. Hudební klávesníci vám sice převedou na počítačovou, ale zkuste hrát na QWERTY byť jen dvouhlasně. Že byste si tu mohli psát noty jako v programu "Music Shop", na to zapomeňte, píšete zase data. Kdepak aby si na klávesnici přehranou melodii váš počítač zapamatoval, jako to umí ta nejlevnější klávesnice Casia. Po uživatelském komfortu ani stopy a na luxusní grafiku programu "Kawasaki Rhythm Rocker" můžete zapomenout.

Zakopaný pes není ani v těch chuďasovských 8 bitech, na které je vázán Commodore 64. Viděl jsem hudební program pro šestnáctibitovou Amigu 500, se kterým bylo stejně málo legrace jako s osmibitovým "Hudebním krámkem". Tajemství laikova hlubokého zklamání z počítačové hudebnosti je zřejmě v tom, že hudba se nám lidem euroamerické civilizace konce 20. století stala sice trvalým a až nutkavým společníkem, ovšem hudba, vůči níž jsme pouze posluchači. Počítač nám rozepíše a dokonce i rozkreslí naši těžkopádnou ruku, ale nevloží nám do hrdla ztracený hlas. Sociální úloha hudby není tak elementární, jako je psaní nebo činnost na pomezí kreslení a rýsování. Hudba nežije v módu každodennosti, ale souvisí spíše se svátkem našeho života. Není účelovým sdělením, je způsobem, jakým se lidé vyladují "na společnou notu".

Pak je tu ta potíž, že suverenita, s jakou již sedmileté osmileté dítě zvládá vyjadřování písmem a kresbou a jež s léty může ještě narůstat, na poli hudebního dorozumívání u většiny lidí s věkem naopak klesá, až klesne docela. Počítač nám svými vyspělými grafickými programy dává šanci přeskočit několikaleté grafické školení a svůj výtvarný cit přímo převést do výtvarně citlivého grafického sdělení. Ne tak hudební program, ten vám sice odpustí znalost tradiční hudební nauky (nemusíte vědět, kolik béček má c-moll), ale pokud jde o transformaci myšlenek ve sdělení, dá vám spíše zakusit, že žádné zajímavé myšlenky zřejmě nemáte. A možná v té chvíli opravdu ne, protože je ve vás už dávno zahubili.

Hudební kultura je ještě mnohem víc než výtvarná v rukou tradicionalisticky myslících profesionálů. Hlíďají si svůj monopol a lákajíce do své vrásčité náruče drobotinu, zároveň mají pro ni připravenou celou sadu pravítek, kterými ji hodlají práskat přes prsty. Carl Orff se to před druhou světovou válkou pokusil změnit a dát dětem na hraní nástroje jednak levné, jednak rychle zvládnutelné.

Snaživé učitelky hudby to však vzaly pevně do svých rukou a udělaly z toho zase školu.

Zatím mám z hudebních šancí nabízených (osmi až šestnáctibitovými) počítači jeden tristní dojem: pravda, orffovsky nás osvobodily od klasických nástrojů, daly nám možnost "kterýkoli" existující nástroj simulovat a "jakýkoli" další si sami navrhnout tím, že si jej stvoříme z kombinací hodnot velké spousty parametrů. Když se však na tuto cestu do neznáma vydáme, zjistíme, že bychom potřebovali několik lidských životů, navíc spojených s rentou, abychom - jsouce neprofesionály - sami postupně začali vyluzovat zvuky, které by uspokojovaly jak nás samé, tak naše bližní. To pole variací je nevyčerpatelné a jako uprostřed pouště zcela nezajímavých zvuků je v něm tu a tam oázička dobře zvučící konstelace, k níž ani od níž však nevede žádná logická cesta. Proto nakonec schlípneme a začneme si zkoušet už objevené konstelace zvukových efektů, zhrzeně si říkajíce, že stejně všechno už je dávno objeveno. Akustické nástroje tyto oázy objevují tím, že každý materiál a každý způsob vyluzování zvuků si s sebou nějakou konstelaci parametrů přináší. Vy jste o tuto berličku konkrétnosti počítačem připraveni.

Časopis "64'er" přišel před časem (v čísle 4/88) s pěkným nápadem tohle objevování roztroušených oáz povzbudit a odměnit. Vidina 1000 marek určitě nebyla motivem jediným, ale jistě mocným. Získal je jakýsi Jomo Wallas za několikařádkový program, jemuž dal název "Furzeditor", tedy - v jemnějším překladu - "Editor větrů". Redakce prý nad jeho zvuky brečela smíchy: "Realističtěji se to už předvést nedá", tvrdí v článku vyhlášením výsledky soutěže.

V naší domácí poušti, ve které se počítačové hudbě věnují profesionální hudebníci, a laici nechť se učí pilně programovat, pro mě byla osvětlením a nadějným příslibem útlá kniha Viktora Kotrubenka "Tajemství syntezátorů" (Editio Supraphon, Praha 1987). Tam jsem se například poprvé dočetl, že záhadná pila nebo obdélník se v syntezátorech neprosadily pro nějakou geometrickou posedlost jejich konstruktérů, ale že pila je součtem všech harmonických, kdežto obdélník jen těch lichých. Podobně instruktivní je přehled o různých formách syntézy. Ačkoli počítačům je tu věnována jen závěrečná podkapitola 9.3., je tato knížečka cenným pomocníkem pro všechny, kdož se nechtějí dát počítačem ve věci hudební tvorby příliš snadno oblnout.

Nicméně na program, který mi dá zakusit pocit volné hudební tvorby, budu asi ještě dlouho čekat.

PHDr. Bohuslav Blažek



Data Translation pokračuje ve vývoji svých grafických programů pro profesionální práci.

PhotoMac pro Apple Macintosh umožňuje nakládat s obrázkem (třeba barevnou fotografií) opravdu jak libo, včetně všech možných barevných retuší, montáží (i textu) a řadou jiných úprav. Výstup lze využít pro tvorbu barevných publikací. Při užití konvenčních technik byste pro stejně subtilní grafickou tvorbu potřebovali aspoň 25 MB paměti, PhotoMac to umí se 2 MB. Stojí \$695.

Firma Polygon umí přinutit pecičko s kartou EGA (640 \* 350 bodů), aby malovalo jako standard VT-240, resp. VT-241 (800 \* 350). Tak budete mít nejen šanci připojit se na grafický výstup solidnějších počítačů, ale budete moci psát až ve 132 sloupcích (bez klouzání textu do stran). Emulující program Poly-Star za \$299 navíc nabízí novou mapu klávesnice s možností programovatelných tlačítek.

Pixie je název grafického programu fy Zenographics. Ve spolupráci s programem Windows 2.0 převádí uživatelská data na grafické vyjádření podle předem stanovených parametrů zobrazení. Umožňuje i přenos zvolených údajů mezi jednotlivými okny. Cena \$195.



# ATARI 55 A AMIGA

## NA ANGLICKÉM TRHU

Poslední dobou u nás roste zájem o počítače Atari ST a commodorskou Amigu. Pro orientaci stávajících i potenciálních uživatelů těchto počítačů a jejich vybavení uvádíme přehled cen (v anglických librách, bez daně) ze srpna 1988. Vzhledem k nepřetržitě stoupajícím cenám paměti (v poslední době i monitorů a hard disků) berte ceny jako orientační.

Použité zkratky: MM-mono monitor, BM-barevný monitor, FM-TV modulátor (pro připojení do anténního vstupu televizoru), \*-jednotka pro floppy disk, HD-hard disk, LT-laserová tiskárna, K-KB, M-MB.

### ATARI

520ST-M - 512K RAM, FM	173
520ST-M - 512K RAM, FM, *0.5M, myš	234
520ST-M - 512K RAM, FM, *1M, myš	260
520ST-M - 512K RAM, FM, *0.5M, myš, MM	321
520ST-M - 512K RAM, FM, *1M, myš, MM	350

všechny mají externí zdroj

520ST-FM - 512K RAM, FM, *0.5M, myš	260
520ST-FM - 512K RAM, FM, *1M, myš	347

oba mají zdroj vestavěn

SM124 - MM 12"	86
SC1224 - BM 12"	260

ceny monitorů jen při koupi společně s počítačem

1040ST-FM - 1024K RAM, FM, *1M	434
--------------------------------	-----

2MB MEGA ST - 2048K, *1M, myš	782
4MB MEGA ST - 4096K, *1M, myš	1042

Konfigurace pro desktop publishing:

2MB MEGA ST + MM, LT, software POS 5000	1998
4MB MEGA ST - jako u 2MB + HD 20M	2695

Doplňky:

ST Cartridge Expander (pro zasunutí až 5 karet)	43
Microline ST Clock Card (int.hodiny, NiCad bat.)	24
Microline ST Clock ROM (ext.hodiny, NiCad bat.)	24
Real Time Clock Card LCM 2000 (ext.hodiny)	43
Aladin - emulátor Apple Macintosh	114
Cherry A3 tablet (graf.sním.tab.)+puk,softw.	550
1st-CRP Digitizing Tablet (dtto)+hrot,softw.	360
Casio CZ230S MIDI Keyboard+adaptér	264
Modulator Box (přípoj 520ST,1040ST-F,MEGA k TV)	61
Myš	28
Graphtec MP3200 (plotter) - 8 barev, 40 cm/s	1049
Tuner Module AV7300 (dělá z monitoru TV)	52
Joysticky	v cenách od 4 do 13
Různé kabely	v cenách od 3 do 17
Knihy	v cenách od 8 do 23

Monitory:

Atari SM124 - 12" High Res MM + kabel	135
Atari SC1224 - 12" Medium Res BM + kabel	355
Philips CM8833 - 14" Medium Res BM, 2 repra	260
Philips CM8852 - 14", High Med Res BM, 1 repro	299

Floppy jednotky:

Atari 0.5M Single SF354 3.5" SS	136
Atari 1M Single SF314 3.5 DS	179
Cumana 1M Single CSA354 3.5" (pro všech.typy)	95
Cumana 2M Twin CDA358 3.5" (jen 520 ST/ST-M)	234
Cumana 1M Single CSA1000S 5.25" (pro všechny)	121
Cumana 2M Twin CDA2000S 5.25" (jen 520ST/ST-M)	312
Cumana 2M Twin CCA2000S 5.25"+3.5" (jen dtto)	296
Triangle 1M Single 3.5" (pro všechny)	104
Triangle 1M Single 5.25" (pro všechny)	130

Diskety:

1000K Bulked 3.5" DS DD	0.87
Memorex 5.25" DS DD	1.22
500K Memorex 3.5" SS QD	1.13
1000K SKC 3.5 DS DD	1.30

Hard disky:

Atari Megafile 20M	533
Supra 20M	477
Supra 30M	607
Supra 40M (s CPU, pro MEGA ST)	739
Supra 60M	869
Supra 198M	2869
Triangle 20M	435
Triangle 40M	609

Laserová tiskárna:

Atari SLM804+interface SLMC804	1200
Výměnný bubek (pro 8000-9000 kopií)	147
Kazeta s tonerem (pro 2000-3000 kopií)	34

Scannery:

Canon IX-12 - 75-300 dpi, 32 odstínů šedé	850
interface pro IX-12 a Atari ST	217
Hawk Scanner (s IF) - 400 dpi	1087

K-MAC Transputer Development System (RISC)	695
--	-----

Videodigitizéry	v cenách od 130 do 217
-----------------	------------------------

Napájecí zdroje:

pro 520ST/520ST-M	50
pro HD SF314/354	40

Rozšíření paměti (cena včetně instalace):

0.5M Upgrade Board pro 520ST/ST-M	130
0.5M RAM Upgrade pro 520ST-FM	130
2-4M RAM Upgrade Board pro 1040ST-F	130
GEM ST ROM Upgrade (6 ROM pro TOS na disku)	43

Software:

DTP	v cenách od 100 do 260
CAD	od 90 do 499
Databáze	od 40 do 217
Spreadsheets	od 20 do 130
Slovní procesory	od 30 do 199
Aztek C68K Developers (Céčko)	156
Mark Williams C	112
Megamax C	87
Assem Pro (assembler 68000)	43
K-Seka (dtto)	43
MCC Macro Assembler (68000/10)	43
AC Fortran	170
Hisoft Developers Basic (kompilátor)	174
Pascal 2	78
Hry stojí v průměru	17

### AMIGA

Amiga 500 - 512K RAM, *1M, myš	346
Amiga 500 + 1084 Med Res BM	560
Amiga 2000 - 2M RAM, *1M, myš, Med Res BM	1499

Rozšíření paměti:

2M RAM Expansion Board	399
8M RAM Expansion Board	2349

Jednotky floppy:

Amiga A1010 *1M 3.5" Single (až 3ks-daisy ch.)	130
Amiga 1M, 5.25" pro A500/1000/2000 (ext.)	175

Cumana 1M 3.5" Single (pro všechny)	87
Slimline 1M 5.25" Single (pro všechny)	130
Triangle 1M 3.5" Single (pro všechny)	96

Hard disky:

Supra 20M pro A500	608
Supra 20M pro A1000	608

Jiné doplňky pro Amigu 2000:

HD 20M i s kabely	639
MS-DOS HD 20M i s kabely	549
PC XT Bridgeboard - MS-DOS 3.2, *360K DS DD	529
PC AT Bridgeboard - MS-DOS 3.2, *1.2M DS DD	749

Monitory:

MM Amiga 1084 Med Res	87
BM Amiga 1900M	213

Ceny ostatních doplňků, knih a softwaru se od Atari ST nijak zvlášť neliší.



# STŘEDISKO VTI PRO ELEKTRONIKU



Středisko  
vědeckotechnických informací Svazarmu  
pro elektroniku  
Martinská 5, 110 00 Praha 1

## Služby střediska

Jsou poskytovány pouze osobně. Vyřizování členství a hostování v 602.ZO Svazarmu, přístup ke knihovně časopisů na mikrofiších, pořizování ozalitivových kopií z knihovny časopisů, prodej programových produktů Mikrobáze, zpravodaje Mikrobáze, nepájivých kontaktních polí, zpravodaje střediska MONITOR a poskytování informací o odborných akcích Svazarmu.

## Pracovní doba

	zavřeno	
pondělí		
úterý až čtvrtek	10 - 12	14 - 17
pátek	10 - 12	14 - 16

☛ >>>>>> telefon 22 87 74 <<<<<<< ☚

## Seznam zkratk

US-USA, GB-Velká Británie, YU-Jugoslávie, JP-Japonsko, CH-Švýcarsko, BG-Bulharská lid.republika, F-Francie, BE-Belgie, DE-NSR, SU-SSSR

## Seznam časopisů

64'ER-Das Mag. fuer comp.fans (DE) ACM Siggraph: Comp.Graphics (US) Applied Mathematical.Modelling (GB) Bajtek (PL) Byte (US) Chip-Das Mikrocomp. Magazin (DE) Communication News (US) Communications of the ACM (US) Computer (IEEE) (US) Computer Aided Design (GB) Computer Design (US) Computer Graphics And Applicat. (US) Computer Journal (GB) Computer Networks (NL) Computing Reviews (US) Comsat Technical Review (US) Desktop publishing World (GB) Datamation (GB) Electr. Sound+Rte (CH) Electr. and Wirel. World (GB) Electronics (US) Electronics - Int.Edition (US) Elektor (DE) Elektronik (DE) Elektronique Praktique (F) Elo (DE) Elrad (DE) Ezermester (MLR) Funkamateur (NDR) Funkschau (DE) Happy Computer (DE) Hifi News and Rec. Review (GB) Hobby (Magazin der Technik) (DE) IBM Journal of R & D (US) IEEE Trans.Circ.Systems (US) IEEE Trans.Syst. Man Cybern. (US) IEEE Trans. on Softw. Engineering (US) Industrial Robot (GB) Industrial and Proc. Control Mag. (GB) Information and softw.technol. (GB) Intern.Business Equipment (BE) Journ. Acous.Soc. Amer. (US) Journ. Parall. Programming (US) Journ.of The Aud. Eng. Soc. (US) Kompjutr za vas (BG) Komputer (PL) Laser & Applications (US) MC Die Mikrocomp. Zeitschrift (DE) Microelectr. and Reliability (GB) Mikromagazin (MLR) Mikrodok (DE) Mikroklan (PL) Mikroprocess. Sredstva I Sist. (SU) Moj Mikro (YU) Nachricht. Elektr. + Telematik (DE) Office Equip. and Products (JP) Personal Computer World (GB) PC Magazin (DE) Practical Computing (GB) Practical Electronics (GB) Practical Wireless (GB) RE - Radioelektronik (PL) Radio (SU) Radio Electronics (US) Radio-Amater (YU) Radio - Fernsehen - Elektronik (NDR) Radio-Televizija-Elektronika (BG) Radioelektronik (PL) Radió-technika (MLR) Revija za mala računala (YU) Robotica (GB) Siemens R&D Reports (DE) Simulation (US) Software-Practice and Exper. (GB) Solid State Communications (GB) Solid State Technology (US) TB - Report (DE) Techniky Komputerowe (PL) The Office (GB) Toshiba Review (JP) ZX Computing Monthly (GB)

## Knihy na mikrofiších

Forth ( programovací jazyk; manuál v češtině ) \* Příručka jazyka Basic pro ZX Spectrum ( kniha v češtině ) \* Przewodnik pro ZX Spectrum ( v polštině ) \* Úvod do programování ve strojovém jazyku ZX 81, ZX Spectrum ( kniha v češtině ) \* ZX Microdrive + Interface I ( kniha v angličtině ) \* ZX Spectrum I + II ( kniha v angličtině )

## Servisní manuály na mikrofiších

ZX Spectrum+2, +3, Amstrad CPC 6128, Amstrad DMP 2000, 3160, 4000, Amstrad PCW 8256, PCW 8512, Amstrad PC 1512, PC 1640, PPC 512, PPC 640.

## Potřebujete informace?...a znáte MONITOR?

\*\*\*\*\*

Nejde samozřejmě o monitor osobního počítače, ale o informační zpravodaj, který vydává Středisko vědeckotechnických informací Svazarmu pro elektroniku. Hlavní náplní zpravodaje jsou české překlady obsahů téměř osmdesáti zahraničních časopisů, které obsahuje fond střediska. Veškeré informační materiály střediska jsou mikrofišované. Znamená to nejen minimální nároky na skladovací prostory, ale také možnost snazšího šíření cenných informací. Z mikrofiší lze pořizovat neomezený počet kopií, a tak časopisy, které jsou k dispozici v originální verzi pouze v omezeném počtu, lze rozšířit mezi široký okruh zájemců. Je však tento okruh potenciálních zájemců opravdu tak široký? Zkušenosti z téměř ročního provozu střediska hovoří jasně. Protože je notoricky známo, že vše souvisí se vším, naše zaostávání za světovým vývojem lze pozorovat i v zájmu o progresivní informace. Vzhledem k mikroskopickému trhu s programovým vybavením, je nejčastější otázkou: "Máte nějaký program na Spectrum (Atari, Sharp, Commodore,)?". Následují dotazy na české manuály, učebnice, příručky. Ale zájem o zahraniční časopisy se soustřeďuje pouze na jejich úzký okruh - polský Bajtek (obsahuje plánky z her) a časopisy monotematicky zaměřené na určité typy počítačů. Ale i v jiných časopisech lze nalézt zajímavé stavební návody, články, popisy integrovaných obvodů, zajímavých zapojení a posledních novinek světové vědy a techniky. Je však nutné ovládat některý světový jazyk. Nejvíce "použitelná" je angličtina a němčina. Vzhledem k rozsahu výuky těchto jazyků na našich školách, jediným výsledkem mravenčí práce naší administrativy po dobu desítek let jsou jen smutné oči většiny mladších návštěvníků střediska, kteří si mohou v zajímavých zahraničních časopisech prohlížet pouze obrázky. A přitom lze v těchto časopisech nalézt odpověď na otázky, které teprve budeme muset řešit. Naši snahou je zpřístupnit co nejširší okruh informací z oblasti elektroniky všem zájemcům, tedy i těm méně "gramotným", kteří neznají žádný cizí jazyk. Proto v dalších číslech MONITORU zařadíme stručné překlady zajímavých článků a posledních aktualit, případně testů soft- i hardwaru. České překlady časopisů budeme zpracovávat novým způsobem, který umožní efektivní vyhledávání v BÁZI Dat z oblasti Elektroniky (pracovní název BÁDEL), kterou konečně, po získání databázového systému schopného pracovat s diakritikou, zpřístupníme během příštího roku svazarmovské veřejnosti. Do BÁDELU budeme zařazovat informace i ze všech domácích časopisů, zabývajících se elektronikou, a rovněž je budeme přenášet na mikrofiše. Tak budeme zpracovávat i všechna periodika 602.ZO Svazarmu. Okruh zahraničních časopisů zúžíme na časopisy, o které je zájem a které se zabývají progresivní tematikou (robotika, optoelektronika apod.), i když právě o ně zatím moc velký zájem není. Výhledově by BÁDEL měl být součástí připravovaných telematických služeb. Pokud tedy budete potřebovat informace z oblasti elektroniky, najdete je ve Středisku VTI Svazarmu pro elektroniku, kde je možné si MONITOR zakoupit. Mimopražští zájemci si jej mohou objednat na adrese 602.ZO Svazarmu, Dr.Z.Wintra 8, 160 41 Praha 6.





# PROGRAMOVÁ NABÍDKA



## Pokyny k objednávání programů

Nabízené programy si zájemci objednávají výhradně na korespondenčních lístcích adresovaných na 602. ZO Svazarmu, Wintrova 8, 160 41 Praha 6. Programy zasíláme na dobírku, je ale možný přímý nákup ve středisku VTI v Martinské 5, Praha 1.

## Programy základní nabídky pro ZX Spectrum

- Dr.MG** 135 Kčs  
Upravená verze spojených programů GENS 3 a MONS 3.
- Datalog** 186 Kčs  
Databázový program, který svým komfortem v mnoha směrech převyšuje obdobné programové produkty. Pracuje s českou a slovenskou abecedou.
- uB-PASCAL** 205 Kčs  
Prostředek pro editaci, překlad a běh programů, vhodný i pro výuku programování.
- CP/M** 191 Kčs  
Vstupenka do světa profesionálních osmibitových počítačů; možnost využívání množství programů, které jsou tímto systémem řízeny. Instalace vyžaduje hardwarovou úpravu počítače (AR/A 10/1988).
- Assembler 80** 198 Kčs  
Původní program, výkonný pomocník při programování ve strojovém kódu.
- BASIC S** 119 Kčs  
Výukový program určený hlavně začátečníkům. Seznamuje s hlavními zásadami programování.
- PROFESOR II** 120 Kčs + kazeta za maloobchodní cenu  
Univerzální výukový program, základ pro instalaci dodávaných znalostníchází typu STUDENT z nejrůznějších oborů.
- STUDENT 1** 96 Kčs + kazeta za maloobchodní cenu  
Pět znalostníchází pro program PROFESOR (Města v ČSSR, Evropská pohoří, Světová moře a oceány, Slovní druhy, Souhvězdí).
- STUDENT 2** 96 Kčs + kazeta za maloobchodní cenu  
Pět znalostníchází pro program PROFESOR: Naše pohoří, Významné vrcholy, Města světa, Křižovatky (dopravní výchova), Malá násobilka (pro děti).

## TESTEDITOR 418 Kčs + kazeta za maloobchodní cenu

Program pro vlastní tvorbu znalostníchází typu STUDENT. Práce nevyžaduje znalost vnitřní struktury programu.

## MULTITASKING 99 Kčs + kazeta za maloobchodní cenu

Operační systém pro ZX Spectrum, který umožňuje současný běh více programů na jednom počítači. Jeden z programů je v popředí, druhý v pozadí; je možné je libovolně prohodit. S programem dostanete ukázkovou pomůcku "Kalkulátor a zápisník".

## GROS 79 Kčs + kazeta za maloobchodní cenu

Grafický rozhodovací systém je účinný prostředek pro podporu rozhodovacích procesů. Umožňuje volit nejvýhodnější z několika možných variant řešení daného problému.

## ODA 110 Kčs + kazeta za maloobchodní cenu

Osobní databázový systém s jednoduchým a názorným ovládním. Kromě obvyklých editačních a vyhledávacích možností dovoluje aritmetické výpočty, volbu formátu zobrazení a tisku.

## PROGRAF 89 Kčs + kazeta za maloobchodní cenu

Prostorové grafy jsou programem pro názorné zobrazení prostorových funkcí s možností odstraňování zakrytých částí a s bohatou volbou parametrů (úhly natočení a nadhledu, počty řezů apod.).

## TEMPERAMENT 49 Kčs + kazeta za maloobchodní cenu

Zábavný psychologický test k určení typu temperamentu. Můžete se dozvědět, "kdo" vlastně jste, kam můžete zařadit své příbuzné a známé.

## STOPKY 110 Kčs + kazeta za maloobchodní cenu

Časoměrný program s editorem pracujícím v paralelním režimu. Tento multitasking umožňuje zároveň odečítat časy a editovat záznamy o jednotlivých závodních nebo jiných měřených jevech. Vhodný pro různé sportovní soutěže nebo pro měření, když je zapotřebí sbírat velké množství časových údajů doplněných komentáři.

## SONDA 4D 49 Kčs + kazeta za maloobchodní cenu

Zábavný program, který vám umožní cestu do fantastického světa čtvrtého rozměru. K programu lze přistupovat různým způsobem - od pohledu průzkumníka, který se pouze seznamuje se zajímavým prostředím, až po úroveň hlavolamu.

Nabídky z minulých čísel zůstávají v platnosti





**Obrázky Radka Šykory a Vít Skácela z výtvarné soutěže JZD AK Slušovice  
„Děti a počítač“**





I VÝROBKY FIRMY EPSON SE DOVÁŽEJÍ DO ČSSR.  
ZASTOUPENÍ MÁ PODNIK MEDIA



EPSON