

**Knihy došlé do redakce  
(Books received)**

Microcomputing. Tagung III/1979 des German Chapter of the ACM am 24. und 25. 10. 1979 in München (*Werner Remmele, Heinz Schecher, Hrsg.*). (German Chapter of the ACM 3.) B. G. Teubner, Stuttgart 1979. 280 Seiten; DM 40,—.

UNIX — Konzepte und Anwendungen. Fachseminar des German Chapter of the ACM am 9. und 10. 12. 1982 in Bonn (2., erweiterte Auflage; *Thomas Kreifelts, Peter Schnupp, Hrsg.*). (German Chapter of the ACM 12.) B. G. Teubner, Stuttgart 1983. 184 Seiten; DM 36,—.

Intelligenztechnologie — Konzepte, Sprachen, Praktische Anwendungsmöglichkeiten. Fachseminar des German Chapter of the ACM am 3. und 4. 5. 1983 in Berlin (*Wolfgang Giloi, Max Schulze-Vorberg jr., Hrsg.*). (German Chapter of the ACM 16.) B. G. Teubner, Stuttgart 1983. 182 Seiten; DM 42,—.

*Pietro Schicker*: Datenübertragung und Rechnernetze. (Leitfäden der angewandten Informatik.) B. G. Teubner, Stuttgart 1983. 224 Seiten; zahlreiche Figuren und Tabellen; DM 25,80.

*Floyd J. Gould, Jon W. Tolle*: Complementary Pivoting on a Pseudomanifold Structure with Applications in the Decision Sciences. (Sigma Series in Applied Mathematics 2.) Heldermann Verlag, Berlin 1983. iv + 204 pages; numerous figures; DM 58,—.

*Karl Hainer*: Numerik mit BASIC-Tischrechnern. (MikroComputer-Praxis.) B. G. Teubner, Stuttgart 1983. 252 Seiten; 51 Algorithmen und Programmen zur Numerischen Mathematik; DM 26,80.

*Thomas Ottmann, Michael Schrapp, Peter Widmayer*: PASCAL in 100 Beispielen. (MikroComputer-Praxis.) B. G. Teubner, Stuttgart 1983. 258 Seiten; DM 24,80.

Microcomputing II. Tagung III/1983 des German Chapter of the ACM vom 25. bis 27. 10. 1983 in München (*Werner Remmele, Heinz Schecher, Hrsg.*). (German Chapter

of the ACM 17.) B. G. Teubner, Stuttgart 1983. 358 Seiten; DM 63,—.

*Werner Schoenborn, Klaus Fritzsche, Gerd Stanke*: Lernverfahren für technische Systeme. (Wissenschaftliche Taschenbücher Band 138, Mathematik-Physik.) Akademie-Verlag, Berlin 1983. 143 Seiten; 63 Abbildungen, 5 Tabellen; DM 12,50.

M. J. FLYNN, N. R. HARRIS,  
E. P. McCARATHY (Eds.)

**Microcomputer System Design  
An Advanced Course, Trinity College  
Dublin, June 1981**

Lecture Notes in Computer Sciences 126. Springer - Verlag, Berlin—Heidelberg—New York 1982.

Stran vii + 397; cena DM 38,—.

Zaměření publikace je dáno cílem postgraduálního kursu, pro který byly přednášky určeny. Vychází z toho, že dosavadní tradiční výchova odborníků ve výpočetní technice je v evropských zemích zaměřena buď na programové nebo na technické prostředky. S příchodem mikropočítačů, obecněji integrovaných obvodů velké (LSI) a velmi velké (VLSI) integrace se však ukázalo, že jednostranné zaměření neumožňuje dostatečně efektivní práci programátorům ani inženýrům navrhujícím výpočetní systémy s těmito obvody. Návrhář musí zvládnout problematiku: technických i programových prostředků, propojování subsystémů a prostředků návrhu, včetně automatizace navrhování.

Kurs byl zřejmě určen pro programátory i inženýry a proto zahrnuje témata z celé naznačené problematiky na úrovni základních kursů s odkazy na literaturu, která umožňuje proniknout do problémů hlouběji. Koncepce kursu je velice aktuální a užitečná mj. již proto, že vznikla na základě zkušenosti řady předních odborníků z oblasti výzkumu, výroby i provozu.

V úvodním článku „Perspektivy mikropočítačů“ uvádí M. J. Flynn oblasti jimiž je nutno se při návrhu počítače zabývat a na struč-

ném vývoji výpočetní techniky naznačuje, jak se uvedené oblasti vyvíjely a jak se vzájemně ovlivňovaly.

Z rozboru požadavků na návrh obvodů VLSI vycházejí problémy, kterým je třeba věnovat zvýšenou pozornost:

1. *prostředky automatizace návrhu* (ověřené a testovatelné)
2. *návrhy architektury* optimalizovaných z hlediska konkrétních aplikací (jazyků),
3. *operační systémy* poskytující účinné místní služby a spolehlivé spojení s centrálními prostředky,
4. *jazyky funkčně/aplikačně orientované*,
5. *sítě přenosu údajů* spolehlivé, umožňující pružně přizpůsobovat/rozmištovat distribuované a centrální prostředky.

To jsou také tematické celky rozvedené v dalších částech knihy, které jsou posuzovány dále.

#### **Integrated Circuit Physics and Technology**

J. F. Gibbons and J. D. Shott: Fyzika a technologie integrovaných obvodů (str. 9—64).

Přehledný článek o fyzikálních základech a technologii integrovaných obvodů, převážně MOS, krátká zmínka je věnována bipolárním obvodům ECL.

Základní pojmy fyzikální elektroniky jsou vysvětlovány bez toho, že by se předpokládaly předchozí znalosti. Nosiče proudu a jejich generace v polovodičích, příměsově polovodiče, vyprázdňená vrstva, kapacitor MOS, MOSFET v režimu ochuzování a obohacování, charakteristiky tranzistoru MOSFET, inventory s tranzistory MOS jsou popisovány s minimem matem. vztahů, nicméně bez nepřipustných zjednodušení. Návrh složitějších čipů je ilustrován na obvodu PLA. Zmínka je též o hradlových polích CMOS.

V dalším jsou probrány základy technologie obvodů MOS, metody optické litografie, dosažení soukrytu, příprava masek. Diskutují se technologická pravidla a konečně výhodnost pravidelných struktur (hradlová pole, PLA, RAM a ROM) z hlediska využití plochy a rychlosti.

V závěru jsou odvozeny spínací časy invertoru N-MOS a uvedeny alternativy obvodů MOS pro vysoké rychlosti: obvody ECL.

Článek neprobírá celé téma integrovaných obvodů, chybí např. obvody  $I^2L$ ; ostatním bipolárním obvodům je věnováno velmi málo. Připomínky lze mít i k řazení jednotlivých partií za sebou. Vhodné snad pro pracovníky, kteří se zabývají systémovým návrhem a programovým vybavením, aby získali základní přehled o těchto otázkách a aby si uvědomili jak může např. systémový návrh ovlivnit velikost plochy křemíku, použitelnost prostředků automatizace návrhu, kompromisy mezi programovými a technickými prostředky, atd.

#### **Computer Aided Design for Microcomputer Systems**

D. Lewin: Automatizace návrhu (CAD) mikropočítačových systémů (str. 65—137).

V úvodní části autor uvádí základní problémy a metody návrhu číslicových systémů, přičemž se soustřeďuje na systémy, tvořené obvody LSI a VLSI. Vysvětluje nové problémy a kritéria návrhu, která s sebou nese rostoucí hustota integrace a rostoucí složitost funkcí, vykonávaných základními stavebními prvky a ukazuje důsledky, které se projevují v oblasti automatizace návrhu takových systémů.

V další části ukazuje autor metody specifikace a hodnocení číslicových systémů. Jsou zde podrobněji rozebrány 3 základní přístupy:

- a) využití programovacích jazyků pro funkční popis,
- b) graficko-teoretické metody;
- c) konečné automaty.

U každého z přístupů jsou charakterizovány nejnámější metody. Značná pozornost je věnována jazykům pro popis počítačů.

Další kapitola je věnována problematice simulace a testování. Jsou uvedeny základní úrovně, na nichž lze číslicové systémy simulovat a jsou charakterizovány nejdůležitější rysy simulačních systémů a to jak z hlediska uživatelského tak z hlediska jejich implementace.

Poslední kapitola je zaměřena na prostředky syntézy, které v současné době počítačový návrh poskytuje. Autor ukazuje, že v oblasti logického návrhu prostředky používané při návrhu obvodů malé a střední integrace mají pro návrh čipů LSI a VLSI omezené použití. Blíže si všímá návrhu řadiče na bázi PLA.

Celkově lze konstatovat, že daná problematika je zpracována vhodným způsobem a že poskytuje dostatečný přehled o prostředcích automatizace návrhu v oblasti mikropočítačových systémech.

#### Properties of Instruction Set Processor

D. Aspinall: Vlastnosti procesoru instrukčního souboru (ISP) (str. 138 – 181).

V úvodní a zároveň teoretické části příspěvku je procesor rozdělen na 3 funkční bloky – blok řízení procesoru, prováděcí blok a blok mapování adresy – a prováděcí blok je dále rozčleněn na části: zpracování adresy instrukce, zpracování adresy operandu a vlastního provádění operace.

Zavádění zmíněných bloků a částí je stručně dokumentováno na vývoji architektury počítače.

V rozsáhlé prakticky zaměřené části příspěvku autor analyzuje a vzájemně srovnává nové rysy instrukčních souborů procesoru PDP 11 a 16-bitových mikroprocesorů 2. generace (I 8086, Z 8000, MC 68000). V návaznosti na úvodní analýzu struktury ISP věnuje pozornost způsobům adresace, modelování složitějších způsobů pomocí jednoduchých, zásobníkovým operacím, segmentací paměti, skokovým instrukcím, režimům činnosti procesoru, druhům a obsluze přerušení.

V závěrečné části si všímá programových prostředků pro zabezpečení paralelní činnosti spolupracujících procesorů a zdůrazňuje význam a možnosti použití nových instrukcí zmíněných procesorů.

Metodickým přínosem je systematický a vyčerpávající popis způsobů adresace paměti, od přímé adresy až po složitější techniku stránkování paměti. Praktickým přínosem je jejich doložení na konkrétních instrukcích 16-bitových mikroprocesorů 2. generace a instrukčních procesorů PDP 11. Záměru seznámit techniky s programovými prostředky počítače vyhovuje.

#### Customized Microcomputers

M. J. Flynn: Zákaznické mikropočítače (str. 182 – 222).

Teoreticky laděný příspěvek zkoumající proces návrhu mikropočítače, který by svou architek-

turou co nejvíce vyhovoval požadavkům uživatele a analyzující vliv některých faktorů (vývojových prostředků, technologie a jazykových prostředků návrháře) na pružnost, rychlost a cenu výsledného řešení.

Nejdříve definuje vlastní proces návrhu jako hierarchickou posloupnost specifikací a interpretací výchozího abstraktního stroje odpovídajícími prostředky – od specifikace problému, přes algoritizaci řešení, vyjádření v programovacím jazyku až po definici cílové architektury. Zdůrazňuje význam úrovně jazyka první programové interpretace výchozího abstraktního stroje a zdůvodňuje volbu jazyka vysoké úrovně. Analyzuje parametry kompilace a interpretace jazyků podílejících se na různých strategiích transformace výchozí, koncepční (image) architektury na architekturu cílovou (host); tyto úvahy dále rozvádí ve 3 progresivních koncepcích návrhu cílového stroje – v návrhu univerzálního interpretačního stroje (UHM – universal interpretative host machine), univerzálního prováděcího stroje (UEM – universal executing host machine) a v návrhu jazykově orientované architektury DEL (directly executed languages); ta je doložena konkrétním příkladem pro jazyk FORTRAN. Diskutuje vztah technologie a architektury a vyvozuje příslušné závěry.

Výsledkem je závěr, že v jazykově orientované architektuře se požadavky uživatele vyjadřují stručněji a efektivněji než při tradičním přístupu.

#### High Level Sequential and Concurrent Programming

R. H. Perrott: Programování sekvenční a paralelní na vyšší úrovni (str. 223 – 272).

V sérii dvou článků R. H. Perrotta (The Queen's University, Belfast) je diskutován současný stav vývoje filosofických a jazykových prostředků pro programování sekvenčních a souběžných (paralelních) procesů na strojově nezávislé úrovni.

V první části „Sekvenční programování“ (Sequential Programming) autor vysvětluje hlavní přínosy programovacího jazyka PASCAL v oblasti údajových a řídicích struktur, význam funkcí, procedur a blokové

struktury a úlohu rekurse v programování. V závěru poukazuje na vzájemný vztah mezi řídicími a údajovými strukturami, jenž je důležitý pro systematické vytváření dobře strukturovaných programů.

Druhá část „Programování souběžných procesů“ (Concurrent Programming) je motivována problémem konstrukce algoritmů a programů v paralelním prostředí kde několik mikroprocesorů buď sdílí společnou paměť, nebo mají lokální paměť, avšak komunikují vzájemně mezi sebou asynchronním způsobem. Řešení problému vzájemné vylučnosti přístupu souběžných procesů ke společným zdrojům a žádoucí kooperace těchto procesů je vysvětleno z hlediska aplikace dvou nejdůležitějších technik: monitorů s podmíněnými proměnnými a primitivních prostředků pro předávání zpráv. Poslední celek je věnován dvěma programovacím jazykům, pro programování souběžných procesů — jazyku PASCAL PLUS a ADA. Specializované prostředky těchto jazyků pro programování souběžných procesů oběma technikami ilustruje autor na netriviálním příkladě (Bounded Buffer Problem).

Význam obou článků lze vidět v popularizaci metod a prostředků moderních metod programování na strojově nezávislé úrovni a jejich propagaci v oblasti mikropočítačů.

#### Microcomputer Operating Systems

N. R. Harris: Operační systémy mikropočítačů (str. 273—302).

Příspěvek seznamuje se základními pojmy, problémy a koncepcemi výstavby operačních systémů počítačů a naznačuje odlišnost operačních systémů moderních mikropočítačů.

Po stručném srovnání vývoje operačních systémů klasických počítačů a mikropočítačů je detailněji charakterizován OS s technikou souběžných periferních operací (spooling). Základní funkce operačního systému jsou dále rozvedeny v popisu vícepřístupového výpočetního systému s variantami dávkového zpracování uživatelských programů, zpracování interaktivního příp. v reálném času. Čtenář získá představu o složkách OS a jejich vzájemné interakci, seznámí se s problémem a

prostředky ochrany paměti a komunikace mezi jednotlivými procesy. Značná pozornost je věnována správě prostředků, přesněji paměti; jsou popsány techniky segmentace a stránkování paměti a je analyzován jejich vliv na výkonost počítačového systému. Závěr příspěvku je věnován funkci a prostředkům operačního systému pro ovládání souborů a je zdůrazněn jejich význam jak u klasických počítačů tak i u mikropočítačů.

Článek velmi hutným, ale zároveň srozumitelným jazykem seznamuje na 30 stránkách se základními pojmy a širokou problematikou operačních systémů. Je vhodným úvodem do jejich podrobnějšího studia. Má *obecný* charakter, *odlišnosti* či spíše trendy odlišující operační systémy *mikropočítačů* od operačních systémů jiných tříd počítačů jsou skutečně *pouze naznačeny*.

#### Notes on Distributed Systems of Microprocessors

G. J. Popcek: Distribuované systémy mikroprocesorů (str. 303—348).

V článku jsou naznačeny některé otázky z problematiky distribuovaných systémů a možnosti jejich řešení na úrovni operačních systémů. V rozsahu jedné kapitoly jsou zde specifikovány vlastnosti distribuovaných systémů (dále jen DS) s dálkovým přenosem dat a DS s lokálním přenosem dat (lokální síť). Vzhledem k tomu, že základní parametry komunikačních medií těchto systémů (chybovost, doba odezvy, komunikační rychlost, atd.) se mohou lišit i o několik řádů, představují oba systémy kvalitativně odlišné prostředky spojení mezi výpočetními prostředky. V článku jsou probrány různé aspekty výstavby databázových operačních systémů (resp. úpravy stávajících pro distribuovaný systém) pro mikroprocesorové lokální sítě.

Mezi diskutované otázky patří např. transparentnost síťových operací, její výhody pro uživatele a programátora a problémy s její implementací (vytvoření účinného systému jmen objektů, invariantních od umístění v síti, možnosti zásahu do síťových operací z programu, řešení konfliktů mezi síťovým přístupem k prostředkům a jejich vlastnictvím.

Krátká zmínka je věnována problémově orientovaným protokolům (nejnižší vrstva v síti) protože v prostředí lokálních sítí jsou velmi jednoduché. Těžištěm článku je tvorba a optimalizace dálkového přístupu do databázi.

Jsou zde definovány tzv. transakce tj. množina operací nad objektem a prostředky pro jejich ovládní pomocí tzv. commit protokolu (podrobnosti již spadají do problematiky databázi).

V krátkosti je pojednáno o problémech synchronizace a blokování.

Zbývající část článku tvoří stručný popis operačního systému LOCUS, kde jsou implementovány již dříve uvedené mechanismy. Zhodnocení článku:

- Článek je orientován směrem k operačním systémům a k databázím.
- Pro souhrnou informaci o distr. systémech se nehodí. (Je nutné znát OS PDP, databáze a mít s nimi jisté předběžné zkušenosti).
- Příklad operačního systému LOCUS je založen na OS PDP (např. Unix) a je třeba znát jeho podrobnosti.
- Článek má cenu zvláště pro zájemce o některé detaily a nové pohledy z dané problematiky.

#### **Lilith: A Personal Computer for the Software Engineer**

N. Wirth: Osobní počítač systémového inženýra-programátora (str. 349—381).

V článku je popsáno technické a programové vybavení osobního počítače LILITH, jenž byl navržen a realizován pro aplikace z oblasti systémového inženýrství. Dokumentuje, jak požadavky na aplikace počítačů ovlivňují návrh programovacího jazyka a jak prostředky tohoto jazyka určují architekturu počítače a strukturu jeho technického vybavení. Skutečnost, že technické vybavení je navrhováno podle programovacího jazyka a nikoliv naopak, je významná zvláště vzhledem k současným možnostem technologie VLSI.

Počítač LILITH se skládá z CPU realizované z jednobitových mikroprocesorů Am 2901, paměti o kapacitě 128 K 16-bitových slov, PROM paměti pro mikroinstrukce (2 K), řadičů pro zobrazovací jednotku a diskovou

jednotku (10 MB) a rozhraní pro klávesnici, světelné pero a tiskárnu. Uváděná cena počítače včetně práce a přídatných zařízení, zvláště 10 MB disku, je 20 000 SFr. Na základě dvou prototypů bylo realizováno 20 těchto počítačů, které jsou nyní používány na ETH v Zürichu.

Jediným programovacím jazykem počítače LILITH je jazyk MODULA-2 jenž je určen jak pro programování na vyšší, strojově nezávislé úrovni, tak pro programování strojově specifických oblastí (ovládání V/V zařízení, přidělování paměti). Počítač LILITH tak nemá ve svém programovém vybavení JSI (assembler); strojový kód je navržen tak, aby přímo interpretoval příkazy speciálního mezijazyka, jenž je generován překladačem jazyka MODULA. Ve srovnání s implementací tohoto jazyka na počítači PDP 11 se zvýšila hustota generovaného kódu na 4-násobek

Vznik osobního počítače LILITH představuje velmi zdařilý příklad aplikace systematického návrhu shora dolů, jenž není omezen hranicí programového a technického vybavení. Tento postup přinese žádoucí efekt — levný počítač značné výpočetní mocnosti.

*Jan Blatný a kolektiv*

IMMO KIESSLING, MARTIN LOWES

#### **Programmierung mit FORTRAN 77**

Teubner Studienskripten — Datenverarbeitung/Informatik.

B. G. Teubner, Stuttgart 1982.

Stran 184; cena DM 12,80.

FORTRAN je nejstarší z vyšších programovacích jazyků, avšak stále patří k těm nejrozšířenějším. Přestože řada teoretiku má vůči jeho celkové koncepci značné výhrady, jeho jednoduchost, rychlost a v neposlední řadě i široká paleta hotových programů jej zejména mezi techniky stále udržuje v popředí zájmu.

FORTRAN se od svého vzniku v roce 1957 vyvíjel dost živelně. Jeho standardní definice byla přijata teprve v roce 1966. Brzy se však začaly ozývat různé kritické připomínky, a

tak byla po téměř desetileté práci v roce 1978 přijata úřadem ANSI norma X 3.9-1978, která definuje jazyk FORTRAN 77.

Hlavní zlepšení oproti jazyku FORTRAN 66 jsou, že jazyk FORTRAN 77

1. umožňuje používat konstrukce IF THEN a IF THEN ELSE,
2. povoluje v podprogramu několik vstupních bodů,
3. umožňuje pracovat s pojmenovanými konstantami,
4. rozšiřuje logické operace o ekvivalenci a nonekvivalenci,
5. kromě aritmetických a logických hodnot dokáže pracovat i s řetězcovými konstantami a proměnnými. Pro práci s nimi je k dispozici aparát, který se může honosit přídomek "user friendly" (v modifikované podobě jej do svého BASICu převzala firma Sinclair), a který je pro uživatele jistě daleko příjemnější, než aparát, který poskytuje např. jazyk PL/1 nebo většina verzí BASICu,
6. poskytuje rozsáhlé možnosti práce s datovými soubory, prakticky ekvivalentní možnostem poskytovaným např. jazykem PL/1,
7. umožňuje deklarovat proměnné s atributem

SAVE, který je obdobou atributu STATIC v jazyce PL/1.

Norma definuje jazyk ve dvou úrovních, a to jako

- a) plnou verzi,
- b) podmnožinu.

Knihy popisuje úplnou verzi jazyka. V prvé kapitole jsou vysvětleny základní pojmy a použitý formalismus. Druhá kapitola se zabývá aritmetickými daty, jejich deklarací a některými základními příkazy. Třetí kapitola popisuje použití logických dat a výrazů a na ně navazujících programových konstrukcí. Čtvrtá kapitola je věnována podprogramům a otázkám s nimi spojeným. Pátá kapitola pojednává o textových proměnných a konstantách, jejich deklaraci a operacích na ně aplikovatelných. V šesté kapitole je popsán způsob práce se soubory. Sedmá kapitola vysvětluje možnosti formátování textů a osmá kapitola je věnována některým speciálním otázkám organizace paměti (bloky COMMON, příkaz EQUIVALENCE a DATA apod.).

Knihy je napsána velice srozumitelně a výklad je doplněn mnoha názornými příklady. Jistě ji uvítají nejen uživatelé jazyka FORTRAN 77.

*Rudolf Pecinovský*