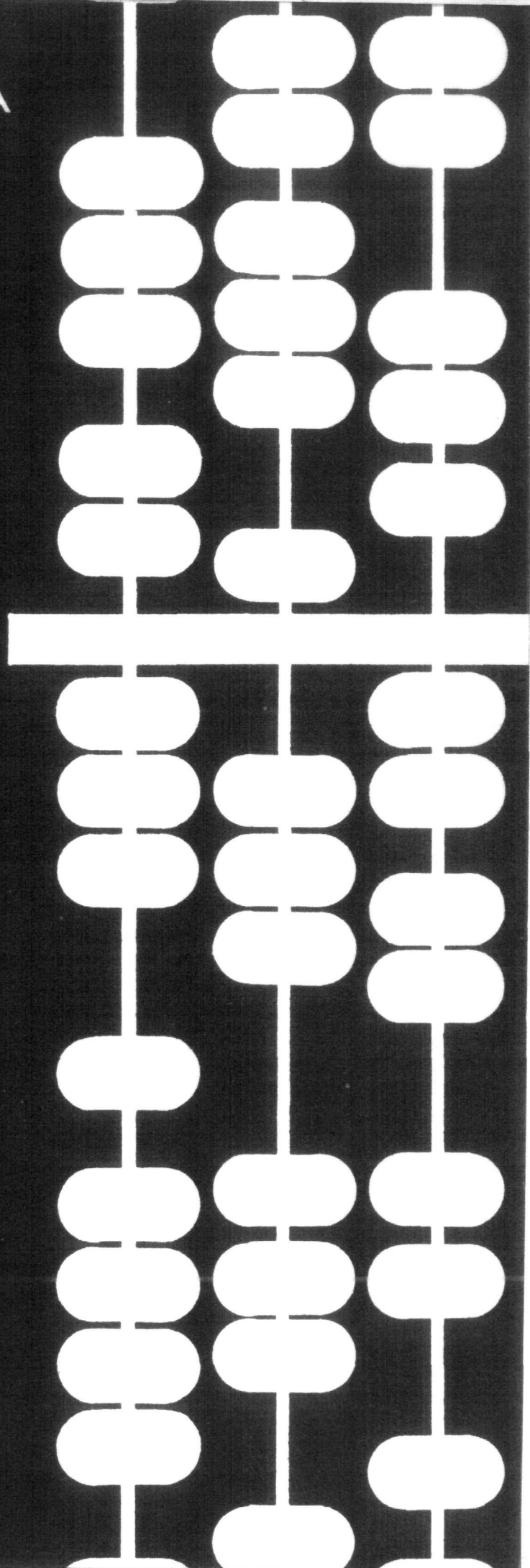


INFORMATICA



3

1977

INFORMATICA

časopis za tehnologijo računalništva in
probleme informatike
časopis za računarsku tehnologijo i pro-
bleme informatike
spisane za tehnologija na smetanjetu i
problemi od oblasti na informatikata

YU ISSN 0350-5596

Časopis izdaja Slovensko društvo INFORMATIKA,
61000 Ljubljana, Jamova 39, Jugoslavija

UREDNIŠKI ODBOR:

Člani: T. Aleksić, Beograd, D. Bltrakov, Skopje, P. Dra-
göjlović, Reka, S. Hodžar, Ljubljana, B. Horvat,
Maribor, A. Mandžić, Sarajevo, S. Mihaljčić, Va-
raždin, S. Turk, Zagreb.

Glavni in odgovorni urednik : A.P. Železnikar

TEHNIČNI ODBOR:

Uredniki področij:

- V. Batagelj - programiranje
- I. Bratko - umetna inteligenca
- D. Čeček-Kecmanović - informacijski sistemi
- M. Exel - operacijski sistemi
- A. Jerman-Blažič - novice založništva
- B. Jerman-Blažič-Džonova - literatura in srečanja
- L. Lenart - procesna informatika
- D. Novak - mikro računalniki
- N. Papić - študentska vprašanja
- L. Pipan - terminologija
- B. Popovič - novice in zanimivosti
- V. Rajković - vzgoja in izobraževanje
- M. Špegel, M. Vukobratović - robotika
- P. Tancig - računalništvo v humanističnih in družbe-
nih vedah
- S. Turk - materialna oprema

Tehnični urednik : R. Murn

ZALOŽNIŠKI SVET

- T. Banovec, Zavod SR Slovenije za družbeno planiranje,
Ljubljana
- A. Jerman-Blažič, Slovensko društvo Informatika,
Ljubljana
- B. Klemenčič, ISKRA, Elektromehanika, Kranj
- S. Sakšida, Institut za sociologijo in filozofijo pri
Univerzi v Ljubljani
- J. Virant, Fakulteta za elektrotehniko, Univerza v
Ljubljani

Uredništvo in uprava: 61000 Ljubljana, Institut "Jožef
Stefan", Jamova 39, telefon (061) 63 261, telegram:
JOSTIN, telex: 31 269 YU JOSTIN.

Letna naročnina za delovne organizacije je 300,00 din,
za posameznika 100,00 din, prodaja posamezne številke
50,00 din.

Žiro račun: 50101-678-51841

Štališče uredništva se lahko razlikuje od mnenja avtorjev

Na podlagi mnenja Republiškega sekretariata za prosveto
in kulturo št. 4210-151/77 z dne 4.5.1977, je časopis
INFORMATICA strokovni časopis, ki je oproščen temelj-
nega davka od prometa proizvodov.

Tisk: Tiskarna Kresija, Ljubljana

Grafična oprema: Tihomir Simončič

letnik I. 1977 - št. 3

VSEBINA

- | | | |
|---|----|---|
| D. Michie | 5 | New Face of Artificial Intelligence |
| V. Doberlet
M. Mrak
B. Popovič
M. Gazvoda | 12 | Problematika izvedbe in vključevanja velikih programskih sistemov realnega časa |
| T. Kalin
I. Lajovic
V. Rajković
D. Viličić | 15 | Osnovni parametri in usmeritev načrtovanja računalniške mreže v Sloveniji |
| D. Novak
M. Kovačević
A.P. Železnikar | 20 | Priključitev hitrega čitalnika luknjanega traku na mikro računalnik |
| S. Tripković
M. Aškračić | 23 | Kros-assembly i simulator za mikro/minil računare |
| B. Kette | 27 | Specifičnosti mikroprocesorske izvedbe merila broja okretaja |
| M. Kovačević
D. Novak
J. Novak
A.P. Železnikar | 32 | Preizkušanje mikror računalniških modulov |
| D. Gojanović | 35 | Graphics Machine Independence with Help of Network Graphics Protocol |
| J. Kalan | 40 | Primer koncepta računalniškega omrežja |
| D. Glušac | 46 | Pouzdanost prenosa podatka izmedju magnetnih medijuma i CRC-Kontrola |
| J. Tasić
L. Lenart
B. Blatnik | 50 | Ceneni mikro računalnik s procesorjem SC/MP |
| | 56 | Moč združenih mikroprocesora |
| | 56 | Študentska vprašanja |
| | 57 | Novice in zanimivosti |
| | 58 | Iz časopisov |
| | 59 | Literatura in srečanja |

INFORMATICA

Journal of Computing and Informatics

Published by INFORMATIKA, Slovene Society for Informatics, 61000 Ljubljana, Jamova 39, Yugoslavia

volume I. 1977 - N° 3

EDITORIAL BOARD:

T. Aleksić, Beograd, D. Bltrakov, Skopje, P. Dragojlović, Reka, S. Hođžar, Ljubljana, B. Horvat, Maribor, A. Mandžić, Sarajevo, S. Milčič, Varaždin, S. Turk, Zagreb.

EDITOR-IN-CHIEF:

A. P. Železnikar

TECHNICAL DEPARTMENTS EDITORS:

V. Batagelj - Programming
I. Bratko - Artificial Intelligence
D. Čučez-Kecmanović - Information Systems
M. Exel - Operating Systems
A. Jerman-Blažič - Publishers News
B. Jerman-Blažič-Džonova - Literature and Meetings
L. Lenart - Process Informatics
D. Novak - Microcomputers
N. Papčič - Student Matters
L. Pipan - Terminology
B. Popovič - News
V. Rajkovič - Education
M. Špegel, M. Vukobratović - Robotics
P. Tancig - Computing in Humanities and Social Sciences
S. Turk - Hardware

EXECUTIVE EDITOR:

R. Murn

PUBLISHING COUNCIL:

T. Banovec, Zavod SR Slovenije za družbeno planiranje, Ljubljana
A. Jerman-Blažič, Slovensko društvo Informatika, Ljubljana
B. Klomenčič, ISKRA Elektromehanika, Kranj
S. Sakaida, Institut za sociologijo in filozofijo pri Univerzi v Ljubljani
J. Virant, Fakulteta za elektrotehniko, Univerza v Ljubljani

Headquarters: 61000 Ljubljana, Institut "Jožef Stefan", Jamova 39, Phone: (061)63 261, Cable: JUSTIN Ljubljana, Telex: 31 269 YU JOSTIN

Annual subscription rate for abroad is US \$ 18 for companies, and US \$ 6 for individuals.

Opinions expressed in the contributions are not necessarily shared by the Editorial Board.

Printed by: Tiskarna Kresija, Ljubljana

DESIGN: Tihomir Simončič

CONTENTS

D. Michie	5	New Face of Artificial Intelligence
V. Doberlet M. Mrak B. Popovič M. Gazvoda	12	Problems of Implementation and Installation of Large Real Time Systems
T. Kalin I. Lajovic V. Rajkovič D. Viličič	15	Basic Parameters and Guidelines for Planning Computer Network in Slovenia
D. Novak M. Kovačević A.P. Železnikar	20	Connection of the Paper Tape Reader to the Microcomputer
S. Tripkovič M. Aškrabič	23	Cross-Assembler and Simulator for Micro/Mini Computers
B. Kette	27	The Construction of the Tachometer Using Microprocessor 4040
M. Kovačević D. Novak J. Novak A.P. Železnikar	32	Microcomputer Module Testing
D. Gojanovič	35	Graphics Machine Independence with Help of Network Graphics Protocol
J. Kalan	40	Systems Network Architecture (SNA)
D. Glušac	46	Reliability of Transfer Data between Magnetic Media and CRC-Control
J. Tasič L. Lenart B. Blatnik	50	Low Cost Microcomputer with the Processor SC/MP
	56	Stacked Microprocessors
	56	Student Matters
	57	News
	58	From Current Literature
	59	Literature and Meetings

navodilo za pripravo članka

Avtor je prosimo, da pošljejo uredništvu naslov in kratek povzetek članka ter navedejo približen obseg članka (število strani A 4 formata). Uredništvo bo nato poslalo avtorjem ustrezno število formularjev z navodilom.

Članek tipkajte na priložene dvokolonske formularje. Če potrebujete dodatne formularje, lahko uporabite bel papir istih dimenzij. Pri tem pa se morate držati predpisanega formata, vendar pa ga ne vrišite na papir.

Bodite natančni pri tipkanju in temeljiti pri korigiranju. Vaš članek bo s foto postopkom pomanjšan in pripravljen za tisk brez kakršnihkoli dodatnih korektur.

Uporabljajte kvaliteten pisalni stroj. Če le tekst dopušča uporabljajte enojni presledek. Črni trak je obvezen.

Članek tipkajte v prostor obrobljen z modrimi črtami. Tipkajte do črt - ne preko njih. Odstavek ločite z dvojnimi presledkom in brez zamikanja prve vrstice novega odstavka.

Prva stran članka

- a) v sredino zgornjega okvira na prvi strani napišite naslov članka z velikimi črkami;
- b) v sredino pod naslov članka napišite imena avtorjev, ime podjetja, mesto, državo;
- c) na označenem mestu čez oba stolpca napišite povzetek članka v jeziku, v katerem je napisan članek. Povzetek naj ne bo daljši od 10 vrst.
- d) Če članek ni v angleščini, ampak v katerem od jugoslovanskih jezikov izpustite 2 cm in napišite povzetek tudi v angleščini. Pred povzetkom napišite angleški naslov članka z velikimi črkami. Povzetek naj ne bo daljši od 10 vrst. Če je članek v tujem jeziku najpišite povzetek tudi v enem od jugoslovanskih jezikov;
- e) izpustite 2 cm in prične v levo kolono pisati članek.

Druga in naslednje strani članka

Kot je označeno na formularju začnite tipkati tekst druge in naslednjih strani v zgornjem levem kotu.

Naslovi poglavij

naslove ločuje od ostalega teksta dvojni presledek.

Če nekaterih znakov ne morete vpisati s strojem jih čitljivo vpisite s črnim črnilom ali svinčnikom. Ne uporabljajte modrega črnila, ker se z njim napisani znaki ne bodo preslikali.

Ilustracije morajo biti ostre, jasne in črno bele. Če jih vključite v tekst, se morajo skladati s predpisanim formatom. Lahko pa jih vstavite tudi na konec članka, vendar morajo v tem primeru ostati v mejah skupnega dvokolonskega formata. Vse ilustracije morate (nalepiti) vstaviti sami na ustrezno mesto.

Napake pri tipkanju se lahko popravljajo s korekcijsko

folijo ali belim tušem. Napačne besede, stavke ali odstavke pa lahko ponovno natipkate na neprozoren papir in ga pazljivo nalepite na mesto napake.

V zgornjem desnem kotu izven modro označenega roba oštevilčite strani članka s svinčnikom, tako da jih je mogoče zbrisati.

Casopis INFORMATICA
Uredništvo, Institut Jozef Stefan, Jamova 39, Ljubljana

Naročam se na časopis INFORMATICA za leto 1977 (štiri številke). Predplačilo bom izvršil po prejemu vaše položnice.
Cenik: letna naročnina za delovne organizacije 300,00 din, za posameznika 100,00 din.

Casopis mi pošiljajte na naslov: stanovanja delovne organizacije.

Priimek.....

Ime.....

Naslov stanovanja

Ulica.....

Poštna številka _____ Kraj.....

Naslov delovne organizacije

Delovna organizacija.....

Ulica.....

Poštna številka _____ Kraj.....

Datum..... Podpis:.....

Instructions for preparation of a manuscript

Authors are invited to send in the address and short summary of their articles and indicate the approximate size of their contributions (in terms of A 4 paper). Subsequently they will receive the author's kits.

Type your manuscript on the enclosed two-column-format manuscript paper. If you require additional manuscript paper you can use similar-size white paper and keep the proposed format but in that case please do not draw the format limits on the paper.

Be accurate in your typing and thorough in your proof reading. This manuscript will be photographically reduced for reproduction without any proof reading or corrections before printing.

Use a good typewriter. If the text allows it, use single spacing. Use a black ribbon only.

Keep your copy within the blue margin lines on the paper, typing to the lines, but not beyond them. Double space between paragraphs.

First page manuscript:

- a) Give title of the paper in the upper box on the first page. Use block letters.
- b) Under the title give author's names, company name, city and state - all centered.
- c) As it is marked, begin the abstract of the paper. Type over both the columns. The abstract should be written in the language of the paper and should not exceed 10 lines.
- d) If the paper is not in English, drop 2 cm after having written the abstract in the language of the paper and write the abstract in English as well. In front of the abstract put the English title of the paper. Use block letters for the title. The length of the abstract should not be greater than 10 lines.
- e) Drop 2 cm and begin the text of the paper in the left column.

Second and succeeding pages of the manuscript:

As it is marked on the paper, begin the text of the second and succeeding pages in the left upper corner.

Format of the subject headings:

Headings are separated from text by double spacing.

If some characters are not available on your typewriter write them legibly in black ink or with a pencil. Do not use blue ink, because it shows poorly.

Illustrations must be black and white, sharp and clear. If you incorporate your illustrations into the text keep the proposed format. Illustration can also be placed at the end of all text material provided, however, that they are kept within the margin lines of the full size two-column format. All illustrations must be placed into appropriate positions in the text by the author.

Typing errors may be corrected by using white correction paint or by retyping the word, sentence or paragraph on a piece of opaque, white paper and pasting it nearly over errors.

Use pencil to number each page on the upper-right-hand corner of the manuscript, outside the blue margin lines so that the numbers may be erased.

Casopis INFORMATICA

Uredništvo, Institut Jožef Stefan, Jamova 39, Ljubljana

Please enter my subscription to INFORMATICA for the volume 1977 (four issues), and send me the bill.

Annual subscription price: companies 300,00 din (for abroad US \$ 18), individuals 100,00 din (for abroad US \$ 6)

Send journal to my home address company's address.

Surname.....

Name.....

Home address

Street.....

Postal code _____ City.....

Company address

Company.....

.....

Street.....

Postal code _____ City.....

Date..... Signature.....

new face of artificial intelligence

donald michie

UDK 519.1:681.3

Machine Intelligence Research Unit
University of Edinburgh

NOVA PODOBA UMETNE INTELIGENCE - Področje umetne inteligence v računalništvu je eden izmed pomembnih faktorjev pri prehodu družbe v stanje, ki bi mu lahko rekli informacijska doba. Članek analizira vzroke zaradi katerih nekatere raziskave v umetni inteligenci v preteklosti niso dale pričakovanih rezultatov: predvsem zaradi neustrezno sestavljenih ambicij, iz katerih so izhajali neustrezni kriteriji za vrednostenje teh rezultatov. Definirana so nova, trdnjša izhodišča za raziskave v umetni inteligenci ter tisti ključni cilji teh raziskav, ki bodo po avtorjevem mnenju igrali pomembno vlogo v razvoju kompleksnih informacijskih sistemov, ki jih avtor označuje s "knowledge enhancement" in "conceptual interface". Pri prvem gre za povečevanje oziroma sistematiziranje človekovega znanja ob razvoju "inteligentnega" informacijskega sistema, pri drugem pa za "konceptualni vmesnik" med sistemom in uporabnikom, tako da je ta vmesnik sposoben ne samo posredovati svoje odločitve uporabniku, temveč tudi pojasniti vzroke za te odločitve.

Artificial intelligence has a role to play in the transition to the information society. A definition of aims and approaches is advanced which lays emphasis on knowledge enhancement and conceptual interface.

INTRODUCTION

The present moment is one of upheaval in the information sciences. The microprocessor revolution continues, and new main-frames enter service with processors capable of executing 100 million instructions per second (e.g. Cray computer installed at the European Medium-range Weather Forecasting Centre) and memories giving immediate access to 10^{12} bits of stored information (operational at the Lawrence Radiation Laboratories, Livermore). Table 1 shows some numerical aspects of the trend. At the same time new programming techniques are under development for representing human accumulations of knowledge in problem-solving systems. In step with

these technical developments we see the reduction of working hours through factory automation, and from this increase of leisure the rise of new social need for "knowledge automation". Institutions such as Britain's Open University are tooling up to supply consumers of all ages with intellectual and cultural goods, - a pointer to tomorrow's information society.

Artificial intelligence (AI) is capable of playing a key role in the social transition, but is still the prisoner of its own past. I shall advance a definition of aims and approaches for AI in terms somewhat more humble than has been customary, throwing the emphasis where it belongs, - namely on the "conceptual interface" between people and machines.

THE PERFORMANCE CRITERION

AI scientists have tended in the past to use as their measuring stick an information system's unaided ability to solve problems in competition with the trained human expert. Yet AI methods have recently been outmanoeuvred by orthodox computing technology in a number of complex domains. The cumulative effect is to discredit the pretensions of artificial intelligence to be the only way to machine solution for all complex problems. Such facts do not, however, discredit the field itself. On the contrary, only after unsound pretensions have been stripped away can AI's true role be recognised in the next phase of our society's development. Some pertinent facts from the Devil's advocate (see Table 2):

EPOCH	FAST MEMORY	PROCESSING
early 1970's	10^8 -bit random-access	10 m.i.p.s.
late 1970's	10^{12} bits (Precision Data Corp.) (Cray Research Corp.)	100 m.i.p.s.
early 1980's	10^{16} bits (Lawrence Radiation Laboratory tender)	1000 m.i.p.s. (NASA Ames tender)

Table 1. Advance in "brute force" hardware capability. Processing speeds are shown in millions of instructions per second (m.i.p.s.). "Fast memory" denotes anything with random access in milliseconds.

- COMPUTATIONAL CHEMISTRY
- AUTOMATIC GUIDANCE
- SPEECH UNDERSTANDING
- ROBOTICS
- COMPUTER CHESS

Complex domains in which conventional computing methods have scored notable successes. Add a sixth:

- WEATHER PREDICTION

Table 2. Some complex task environments for computing (see text).

Computational chemistry. Stanford University's DENDRAL system⁴ for organic chemistry has an "intelligent" part. This is the component which proposes molecular structures from data on unknown compounds, and it represents a noteworthy advance in artificial intelligence. I have in mind the suite of programs which interpret mass spectrogram and nuclear magnetic resonance data, together with the Meta-DENDRAL module which proposes modifications and extensions of the identification rules. These are not, however, the programs which most excite interest among chemists, nor will they deserve their interest until the width of DENDRAL's built-in chemistry knowledge has been expanded by at least a factor of ten. The chemists want the CONGEN module, which allows them interactively to generate possible molecular structures within the constraints set by the data and by their own intuitions. As for being intelligent about CONGEN's output, the chemist prefers to do this for himself. CONGEN was, of course, written by AI people, but primarily in their capacity as gifted computer scientists rather than as AI specialists. Conclusion: orthodox computing methods have already demonstrated high performance in practical chemistry, but AI has not.

Automatic guidance. The computer programs which guided the Apollo space vehicles, and with which ground control interfaced during the desperate days of the Apollo 13 mission, were developed by Draper Laboratories, Massachusetts. They perform work comparable in many respects to the intellectual tasks of a trained space navigator. These programs, however, contained nothing corresponding to the broad and flexible conceptual structures characteristic of human thought. Instead they embodied a complete Newtonian model of the solar system together with an elaborate hierarchy of provisions for everything which could possibly be foreseen as going wrong. The one big thing which did go wrong was in the event not foreseen, - the explosion which disabled the space-craft's main engine. But so excellent was the software engineering for man-machine communication that although large parts of the computing system had been rendered irrelevant by the disaster, other parts were able to mediate emergency human intervention. Conclusion: orthodox computing methods have already demonstrated high performance in practical astronautics, but AI has not.

Speech understanding. A difficult and important challenge to intelligent systems; machine understanding of connected speech, has recently been subjected to an intensive

6-year study at Carnegie-Mellon University with notable success.⁴ Two computer programs were completed, HEARSAY-II and HARPY. HEARSAY-II embodies machine intelligence techniques. But HARPY, which has little AI in its design, is the one which more convincingly met the exacting performance criteria of the University's research contract from the US Defense Department. Conclusion: orthodox methods have already demonstrated high performance in practical speech understanding, but AI has not.

Robotics. Edinburgh's FREDDY system represented a step forward in the study of how a computing machine can acquire and manipulate descriptive concepts about a "hand-eye" world, and General Motors have been able to incorporate spin-off from this project into their robotics R and D programme. But for those primarily interested in cost-effective automatic assembly, I would recommend a visit to Draper Laboratories in Cambridge, Mass. There you may see Jim Nevins' impressive combination of high-calibre software engineering with clever mechanical design in a context of factory economics.³ Conclusion: orthodox computing methods have already demonstrated high performance in practical automation, but AI has not. It should be added that the Edinburgh project was motivated by the knowledge-representation issue rather than by the goal of cost-effective shop-floor automation.

Computer chess. In chess we have seen two or three programs for unaided play of the complete game based on the AI approach, i.e. with emphasis on conceptualisation and casual reasoning. The earliest program of this type was Newell, Shaw and Simon's.⁴ More recent examples are Berliner's CAPS⁵ and Pitrat's program⁶ for finding tactical combinations. But the programs which have won the respect of the chess world by defeating expert players are different. They are clever programming exploitations of the gift which hardware technology has bestowed upon computing, - a million-fold advantage in raw computing power over what the human brain is physically equipped to do (see Table 3). Conclusion: orthodox computing methods have already demonstrated high performance in practical chess, but AI has not.

Does this mean, then, that artificial intelligence has been squeezed out into the cold, with nothing left to do for its living. Quite the opposite. It means that AI scientists can switch attention from what should never be

1. Rate of information transmission along any input or output channel	30 bits per sec.
2. Maximum amount of information explicitly storable by the age of 50	10^{10} bits
3. Number of mental discriminations per second during intellectual work	18
4. Number of addresses which can be held in short-term memory	7
5. Time to access an addressable "chunk" in long-term memory	2 seconds
6. Rate transfer from long-term to short-term memory of successive elements of one "chunk"	3 elements per second

Table 3. Some information-processing parameters of the brain.

their central aim and concentrate instead on the things which only they can do, and which the world is going to need very badly: (1) AI problem-solvers for domains not reducible by brute force, (2) conceptual interfaces for brute-force problem-solvers and (3) knowledge enhancement for human problem-solvers.

DOMAINS NOT REDUCIBLE BY BRUTE FORCE

Some task environments are not reducible by brute force computing because they are messy: no compact algorithmic representation exists. The Stanford program MYCIN identifies bacterial organisms from the results of blood and urine tests and from other data about the patient, and prescribes antibiotic regimes. It is unlikely, to put it mildly, that an algorithmic theory exists analogous to the topological and group-theoretical basis for chemical structure-generation or the game-theoretical basis of chess.

Other non-reducible tasks are enlarged versions of reducible tasks. The chess end-game King, Queen and Pawn against King and Queen is reducible, although not fully fathomable by Grandmasters. In fact it has been reduced, by Arlazarov and Futer⁸, to a computer look-up table of some hundreds of millions of entries capable of faultless play within tournament time constraints. The full game chess is, however, not reducible to a look-up table in any feasible sense. Grandmaster-level play machine will almost certainly necessitate AI treatment of the kind which Pitrat, working in Paris, recently brought to bear upon the problem of finding deep combinations. This program, incidentally, scores higher than many Grandmaster would be expected to do on this narrowly defined task. Indeed it is not yet clear whether the program can be matched by human skill.

For completeness, this section should strictly have been "Domains thought to be not reducible by brute force". It can never be excluded that new mathematical tools might be developed for simplifying into algorithmic form for the description of a given messy-seeming domain. In some cases a shrewd application of existing tools may transform the situation: perhaps an example is Barrow and Burstall's attack on the description-matching problem in robot vision.⁹ Previously people had fought against the combinatorial explosion with collections of *ad hoc* heuristics concerning local features of description-graphs. Without the mathematics of maximal cliques they would have continued to hoe a very hard row. Another case is the recent reformulation by Ray Carhart, working in our laboratory on leave from Stanford, of the topological problem of generating chemical structures, leading to a computationally more efficient CONGEN program¹⁰. Its outmoded predecessor, however, much richer in chemical knowledge, played an unexpected role in improving the minds of the computational chemists! I shall return to this later in connection with the theme of "knowledge-enhancement".

THE CONCEPTUAL INTERFACE

As earlier stated, conventionally programmed systems can now outperform human experts in a variety of complex intellectual tasks, namely those which can be reduced by brute force. The list given above did not include weather prediction, the subject of an early AI attempt by Widrow and Smith¹¹. But it might reasonably have done so. Contemporary number-crunching system perform the task at ever-increasing levels of effectiveness, but they do not conceptualize the problem-domain in terms resembling those in which the human meteorologist

represents it. At first glance it would seem that machine representations of human knowledge are therefore not required in such a case as meteorology. A second glance makes clear that on the contrary the construction of such representations acquires a new urgency. How else will the human user retain any comprehension of what is going on? This indicates the first big task of "knowledge automation", the design and construction of conceptual interface to allow people (who are still much more intelligent than machines) and machines (which are already cleverer than people) to understand each other.

The point can be dramatised by reference to a computer chess occurrence at the recent IFIP meeting at Toronto. Kenneth Thompson of Bell Laboratories has computed out the entire problem space for King and Queen against King and Rook, and had represented the optimal strategy for both sides in the form of a look-up table in machine memory. The table comprises about 4 million entries corresponding to the total number of legal King-queen-Rook positions after excluding symmetrical cases. From all but a few special starting positions this ending is a theoretical win for queen's side. It is also known that chess-masters can ordinarily guarantee to execute the win against any human opponent. At Toronto two International Masters, former World Correspondence Chess Champion Hans Berliner and Canadian Champion Lawrence Day, were invited to demonstrate the winning strategy in play against the optimal defence embedded in the database. To their embarrassment they discovered that they could not win against the machine in spite of many renewed attempts.

Naturally they then wanted to ask the program what were the key ideas underlying its defensive strategy. Of course, neither the program nor its author could answer such a question, since the program's knowledge, although comprehensive, had not been condensed into a form which humans can mentally handle. Since it does not contain things like ideas, concepts, themes, goals and the like, it is in no position to give answers like: "At this stage White must drive the enemy King into the edge of the board". The program has no conceptual interface.

It may be said that chess is just a game. But let the reader generalise a little. Let him imagine a comprehensive but opaque strategy not in the world of chess but in the world of air-traffic control, urban design, economic planning or even in factory automation. A full-scale investigation of Royal Dutch Steel's Hoogovens-plant was provoked by unexpected and untoward developments in their automatically controlled strip-rolling mill. These centred around the operators' problem of judging when to step in to take over from the computer. The following account is from Hadley Voysey (*New Scientist* 18 August, 1977, pp. 416-7):

"The operators became so unsure of themselves that, on some occasions, they actually left the pulpits used for control unmanned... The operators also failed fully to understand the control theory of the programs used in the controlling computer, and this reinforced their attitude of "standing well back" from the operation—except when things were very clearly going awry. By intervening late, the operators let the productivity drop below that of plants using traditional control methods. So automation had led to lower productivity and operator alienation simultaneously."

The same article also mentions a University study of air traffic control which expresses analogous anxieties. The US Federal Aviation Administration has plans to increase the scope of automation in such a way as to

displace human control interaction with individual aircraft altogether. Voysey comments: "Because of the vulnerability of human lives in the air traffic control scheme, the certainty and uncertainties of particular combinations of man and computers attract acute attention". There are thus already grounds for worry that unless the study of conceptual interfaces is given priority our technological society may flounder in a sea of informational complexity, in which machines swim but people sink.

A number of academic AI laboratories have practical plans. An example is the CGHELP proposal¹² of the DENDRAL group which will constitute an understanding buffer between the user and the new and conceptually opaque version of CONGEN.

The main interfacing vehicles between existing knowledge sources take the form of language and pictures. I am thinking of communication between one human being and another, and between humans and inanimate devices such as books and films. Clearly AI studies devoted to language-understanding and computer vision have immediate relevance, and these studies have begun to harvest their first fruits in application. In language understanding, Harris' ROBOT system¹³ for database interrogation is commercially available by dial-up access to the Artificial Intelligence Corporation, operating from Washington D.C. In computer vision, as earlier noted, General Motors¹⁴ are applying techniques in part derived from Edinburgh's FREDDY work to the identification and manipulation of engineering parts; but from the stand-point of the conceptual interface, even more significance attaches to studies in progress elsewhere on machine understanding of diagrams, drawings and cartoon strips.¹⁵

KNOWLEDGE ENHANCEMENT

I mentioned earlier the chemists, although greatly interested in using the CONGEN part of Stanford's DENDRAL system, are less by the part which makes intelligent interpretations of mass spectrogram patterns. I should qualify this statement. Although they do not so often want to run the program, the chemists write to the Stanford group asking for the rules in which the program's knowledge of spectroscopy is represented. These rules have been constructed by chemists working interactively with DENDRAL on a trial-and-error basis. Moreover, a recent paper¹⁶ in the chemical literature consists essentially of machine-discovered rules output by the Meta-DENDRAL part of the program and subsequently accredited by a committee of chemists. So the DENDRAL project has achieved something important for mass spectroscopy which was never envisaged, namely a reformulation of text-book material, including the addition of new knowledge. Running the rules on the forms part of a process of accrediting a knowledge-source intended for humans, - an improved rule-book on "how to do it". Likewise, the initial phase of the Stanford work on the CONGEN structure-generator achieved a conceptual clarification of many strictly chemical issues to do with symmetry properties of molecules. This gain in knowledge-representation for human consumption would not have come about without the AI heuristic programming approach. Although Ray Carhart's mathematical treatment is now leading to a stripped-down algorithmic version which runs fast, the earlier knowledge-rich representations made a contribution all of their own to chemical understanding.

In Edinburgh we witnessed a similar phenomenon during a recent visit from Ivan Bratko of the Josef Stefan Institute, Ljubljana. He used our local knowledge-representa-

1. Look for a way to mate opponent's King in two moves.
2. If the above is not possible, then look for a way to further constrain the area on the chess-board to which the opponent's King is confined by our Rook.
3. If the above is not possible, then look for a way to move our King closer to opponent's King.
4. If none of the above pieces of advice 1,2,3, works, then look for a way of maintaining the present achievements in the sense of 2 and 3 (i.e. make a waiting move).
5. If none of 1, 2, 3, or 4 is attainable then look for way of obtaining a position in which our Rook divides the two Kings either vertically or horizontally.

Table 4. How to mate with King and Rook against King (see text).

tional language AL1 to formulate an elementary end-game in chess, which he tested on the machine. He also developed a formal proof of correctness¹⁷. Then he translated the AL1 expression back into English. The result (Table 4) was found to be a text on this ending superior to any of the accounts in the chess-books: more compact, more complete, certifiably correct, and easy to memorise.

Similar ideas are suggested by the experience of Stanford Medical School with the MYCIN knowledge-based program for bacterial identification and drug prescription. The range of diseases and organisms over which MYCIN's knowledge presently extends is too limited for it to be widely used by pathologists as yet. But for knowledge-transfer to students (i.e. as a teaching aid) it has established a solid position.

From such shoots could grow a branch of the technical publishing industry devoted to the machine clarification and enhancement of existing instructional texts.

INTERACTIVE KNOWLEDGE-BASES

In social terms the new direction is the planning and incremental construction of interactive knowledge-bases for selected areas of science. A facility is needed to act as the computer-based analogue of today's scientific and technical publishing industry, but destined to develop on a large scale. Prototype systems exist for mass spectrometry¹, clinical bacteriology⁷, internal medicine¹⁸, molecular genetics¹⁹ and X-ray crystallography²⁰, to name some of the best known. In the USA industrial and other applied scientists have begun to interact with these facilities as paying users, and academic scientists both as non-paying users and as royalty-earning contributors. In the case of chemistry a national computational resource has recently been established by the US Government, on lines which (defensibly, according to the argument developed earlier in the paper) place the immediate emphasis on brute force methods and heavy-duty numerical computing. But the participation of the Stanford group ensures that the "conceptual interface" and "knowledge enhancement" themes will be built in from the start.

As stated, the main "knowledge industry" in the past has been the publication of text-books. We should now be preparing for a future in which home TV and dial-up systems of the "CEFAX" and "VIEWDATA" type, enhanced with inferential retrieval capabilities, will increasingly

supplement the book as the typical knowledge source, not forgetting the contribution also of the "home computer" development. Working scientists and engineers will not be sorry. More than one survey²¹ has established that they almost never turn to books and papers for answers to technical questions. They call up other professionals on the telephone, using each other, in effect, as interactive text-books.

THE COMPLEXITY EXPLOSION

A drawback to existing text-book technology is that scientific knowledge is now proliferating so fast as to create obstacles to its own growth. In 1974 the Science Citation Index covered about 400,000 articles and communications citing about three million different publications (*Nature*, Dec. 16, p. 609). Workers at the frontiers of some research areas, for example molecular genetics and organic chemistry, are finding it hard to be simultaneously aware of enough of the relevant facts to plan the next experiment intelligently. According to Joshua Lederberg, whose Nobel Prize was for work in the genetics of micro-organisms and who played an initiating role in the DENDRAL developments described earlier, vitally needed experiments in genetics are being missed just for this reason.

Scientists continue to combat the complexity explosion with centralised library and inter-library facilities, private circulation of technical notes and personal interactions and meetings of all kinds. These measures offer multiple access to the knowledge-base distributed through the brains of the participating scientists. But they also throw additional demands on these same brains, deflecting some of their activity into administrative and social functions. A break-even point comes beyond which further elaboration of conventional knowledge aids is unproductive, or even counter-productive. With so much to absorb, how can a person get any work done? New directions are needed.

NEW PERSPECTIVE

Now return to the list of specimen complex domains in Table 1. In calling the roll for a second time we shall note the place of AI according to the new perspective. This paper opened with a recital of facts from the point of view of the prosecuting counsel. The peroration belongs to the defence.

Computational chemistry. The lack of immediate market for the "intelligent" part of DENDRAL has no more significance for the long term than the fact that the Wright brothers did not open an air line. It is a matter of time. In the present case time is needed to expand DENDRAL's base of useful knowledge by methods already tried and proved. In the meantime we have seen from this project:

- (1) The first original contribution by a machine system to scientific knowledge;
- (2) The first concrete experience in support of the "knowledge enhancement" theme;
- (3) A well-documented requirement, in the aftermath of successful reduction of an apparently "messy" problem to efficient algorithmic form, for construction of a conceptual interface, - the CGHELP facility.

Automatic guidance. Beyond the guidance tasks of interplanetary flight lies a computational challenge of an entirely different order: the Mars Rover project. The need

is for an unmanned robot vehicle to navigate Mars' irregular terrain, conduct scientific measurements, and condense and communicate the results.

The problem is dominated by a constraint not encountered in the general run of terrestrial undertakings. Remote man-machine interaction is limited to the transit time of radio signals between Earth and Mars. This time is of the order of half an hour, depending on the relative positions of the two planets in their orbits round the Sun. A signal delay of this magnitude means an extension of the machine's responsibilities from the tactical towards the strategic end of the scale, leaving to mission control only "superstrategic" judgments and decisions.

The NASA-supported group at the Jet Propulsion Laboratory, Pasadena, is now some five years into the Mars Rover project²², with another five before the first interplanetary trial. These space roboticists see the problem as primarily one of artificial intelligence. Their domain is par excellence one in which a special need for stand-alone performance exists, unattainable by conventional computing methods: either the robot has a "cognitive map" of its task environment or it fails.

Speech understanding. To forestall premature rejoicing by HARPY over her more intelligent stable-mate HEARSAY-II, we have to raise the stakes a little, so enlarging the universe of discourse as to render indispensable a thorough grasp of its semantics and causality. Once a world is large enough to contain both "nitrate" and "night rate", both "fizzy cysts" and "physicists", both "abominable" and "a bomb in a bull", then not all the theories of acoustics, phonetics and linguistics will salvage sense from the waveform. The use of stored semantic and pragmatic models, first indicated by Winograd's pioneer study, is now accepted as a necessary and growing ingredient for further language-understanding developments. It is of interest that Winograd is currently collaborating with Bobrow in an office automation project, since this is an application area within which cost-effective machine stenography will one day find a welcoming niche. That day is, however, rather remote, even for restricted and stereotyped dictation chores. Air traffic control, mentioned earlier, is likely to be an earlier beneficiary of significant advance. Stockmarket transaction and a variety of booking-clerk and ticketing applications are already the target of prototype commercial speech-input systems, but the limited repertoire required may well be satisfiable for a long time to come by clever but knowledge-poor systems of the HARPY type.

Robotics. Nevins' project³, having reached first-base success by exploiting conventional methods to the limit, is now reaching towards the incorporation of machine vision, in collaboration with Stanford Research Institute. SRI in turn have enjoyed uptake for the machine vision studies done at Edinburgh and other centres of academic AI. As for the "conceptual interface", the case study quoted earlier from Royal Dutch Steel underlines the need. Among moves towards filling it by AI techniques it is legitimate to cite the man-machine communication facilities of Edinburgh robot's description matching and description-learning module.

Computer chess. Brute force computing has achieved expert play for the games as a whole. The author's laboratory is engaged in studies from a slightly different point of view. Here the aim is to implement master-level, and where possible error-free, play in selected sub-games of chess, enlarging step by step the size and complexity of the sub-games attempted. The basis of this approach is a special "advice-taking" computer language, ALI²³, used

as a vehicle for transferring knowledge of how to handle the giving sub-game from chess book or from direct interaction with chess-masters. Representation of knowledge is in terms of pattern-based rules, comprising ordered lists of constraints and goals, in something like the same form that the Master is taught to represent his knowledge. Low-level detail and calculational lookahead is kept behind the scenes, freeing the human "teacher" to converse with the machine "pupil" purely at the level of concepts.

Early indications are that by this AI approach a rate and effectiveness of knowledge-transfer is achievable which outruns conventional programming methods.

Weather prediction. Mention was made of the European Centre for Medium-range Weather Forecasting, whose accomplishments rest solidly on mathematical modelling and large-scale number-crunching. The meteorologists, however, see themselves at some future stage facing diminishing returns, and are accordingly including methodological diversification in their new planning proposals. These include use of statistical and Monte Carlo approximations and in addition investigation of the value of incorporating heuristic knowledge.

In summary the interplay between conventional (algorithmic) and AI (knowledge-base) methods and representations is complex and varies from case to case. Even on strict performance, we see cases where algorithmic execution of fixed mathematical models (as in interplanetary navigation) has succeeded so well that AI will only be required for conceptual mediation between the high-performance system and the technical personnel. We see other cases where the domain is so ill-structured and subject to change (MYCIN; Mars Rover project) as to demand that performance be based on large stores of machine knowledge about the objects, relations and heuristic and gross causal laws of the task environment. Between these extremes lies an intriguing scatter of mixed cases: problems initially only tractable through machine use of heuristic knowledge (chemical structure generation which improved mathematical treatment has subsequently laid open to a more efficient, knowledge-poor, algorithmic approach; problems where conventional methods deliver results up to a certain level of acceptability (speech-understanding; chess; weather prediction), but which will be obliged to recruit AI resources should radically higher performance goals be set, as demonstrated in a preliminary way by Pitrat's chess program; and finally problems whose impressively complete algorithmic solutions (factory automation; air traffic control) are vitiated or rendered suspect by the human factor. Far from concluding that AI's place is in limbo, a more energetic development of the art seems indicated, in active liaison with related branches of information science. In this connection note should be taken of the development usually known as automatic programming.

AUTOMATIC PROGRAMMING

It is common ground in every branch of computer science that significant extension of present achievement in automating the skills of programming would constitute the greatest leap forward of all. The common goal of making programming successively easier, more self-verifying, and more explainable to the user of the resulting code unites (whether they like it or not) theoretical purists, such as Dijkstra and Hoare, and practical software engineers such as Baltzer. Intervening territory includes projects such as C.C. Green's²⁴ which combines the criteria and man-machine approaches of experimental AI with a well-ordered intellectual background in

computation theory and logic. This project is building a "how to do it" knowledge-base about LISP programming for task which include sorting and description-building. Interface facilities guide the user intelligently in the step-by-step construction of his program.

NEGLECT OF EXPERIMENTAL AI

This example reminds us that in common with other branches of computer science, AI needs theory and it also needs experiment. So long as authoritative opinion in Britain remained aware of this it was possible to achieve a specially close co-operation between computation theory and experimental programming. Recent years have seen a hiatus. A re-evaluation of the field for the Science Research Council was instigated by physicists in 1972-3. Their conclusions were that certain theoretical and biologically oriented aspects of the field are healthy, as also immediate industrial application, but that the academic study of the knowledge-representation issue by experimental means is not.

This decision may ultimately be seen as hasty, and in cultural and long-term economic terms costly. Analogy with another complex information science is not out of place. Recent events in Britain (which find no echo in scientific opinion anywhere else in the world) are as though at an early stage of twentieth century genetics it were said: "You have the mathematical theory of Mendel to play with and a theoretical model of chromosomal linkage. You have breeding work to do for the community's good in improving crops and farm animals. You are also free, and we will even fund this modestly, to investigate the broader matrix of biological processes in which the genetical phenomena are embedded. But frankly we see no need to be breeding fruit flies."

Fortunately for everybody, including ultimately for the farmers too, Morgan and his colleagues were enabled to get ahead with the work. I hope that enough has been indicated of accomplished fact and new opportunity to establish at least a prima facie case that the same should be done for AI. Opposition can be expected, and in Britain has already shown itself. The offending discipline, high-energy physics, has long enjoyed a paramount position in the public confidence. It is not surprising, therefore, if physicists find unwelcome the possibility that their discipline is no longer the sole inheritor of the medieval Church's role as chief explainer of the Universe. Let us hope that they will learn to live in peace with upstart sciences such as ours, and to accept that knowledge-based computation can offer to all sciences new explanatory tools.

REFERENCES

1. R.E. Carhart, S.M. Johnson, D.H. Smith, B.G. Buchanan, R.G. Dromey and J. Lederberg. Networking and a collaborative research community: a case study using the DENDRAL programs. ACS Symp. Ser. No. 19: Computer networking and Chemistry (ed. P. Lykos), 192-217, Amer. Chem. Society, 1975.
2. F. Hayes-Roth and V.R. Lesser. Focus of attention in the Hearsay-II speech understanding system. Proc. 5th Intern. Joint Conf. Art. Intell., 27-35, 1977. Speech understanding systems: report of a steering committee. SIGART Newsletter No.62, 4-8, 1977.

3. J.L. Nevins, D.E. Whitney et al. Research Issues for Automatic Assembly. Cambridge: Charles Stark Draper Laboratory, 1977.
4. A. Newell, J. Shaw and H. Simon. Chess-playing programs and the problem of complexity. IBM J. Res. Develop., 2, 320-35, 1958.
5. H. Berliner. Chess as problem solving: the development of a tactics analyzer. Ph. D. Thesis, Carnegie-Mellon University, 1974.
6. J. Pitrat. A chess combination program which uses plans. Artificial Intelligence, 8, 275-321, 1977.
7. E.H. Shortliffe. Computer-based medical consultations: MYCIN. New York, Amsterdam, Oxford: Elsevier.
8. V.L. Arlazarov and A.V. Futer. Computer analysis of a Rook end-game, in Machine Intelligence 9 (eds. J.E. Hayes, D. Michie and L.T. Mikulich), in press.
9. H.G. Barrow and R.M. Burstall. Subgraph isomorphism, matching, relational structures and maximal cliques. Information Processing Letters, 4, 83-84, 1976.
10. R. Carhart. Reprogramming DENDRAL. AISB European Newsletter, 28, October 1977.
11. B. Widrow and F.W. Smith. Pattern recognising control systems in Computer and Information Sciences (eds. J.T. Tou and R.G. Wilcox). Clever Hume Press, 1964.
12. Unpublished design study by R. Carhart and others of the DENDRAL group, 1976.
13. L. Harris. ROBOT: a higher performance natural language data-base query system. Proc. 5th Intern. Joint Conf. Art. Intell., 903-4, 1977.
14. S.W. Holland. A programmable computer vision system based on spatial relationships. Research Publication GMR-2078, Michigan: General Motors Corporation, 1976. W.A. Perkins. A model-based vision system for industrial parts. Research Publication GMR-2410, Michigan: General Motors Corporation, 1977.
15. see, for example, papers by A.K. Mackworth, A.H. Dixon, M.R. Adler and F. Tsuji and colleagues on pp. 598-610 of Proc. 5th Intern. Joint Conf. Art. Intell., 1977.
16. B.G. Buchanan, D.H. Smith, W.C. White, R.J. Gritter, E.A. Feigenbaum, J. Lederberg and C. Djerassi. Applications of artificial intelligence for chemical inference, 22, Automatic rule formation in mass spectrometry by means of the Meta-DENDRAL program. J. Amer. Chem. Soc., 98, 6168-6178, 1976.
17. I. Bratko. A KRK Advice Table and a proof of its correctness. Personal communication, 1977.
18. H. Pople. The formation of composite hypotheses in diagnostic problem solving: an exercise in synthetic reasoning. Proc. 5th Intern. Joint Conf. Art. Intell., 1030-37, 1977.
19. N. Martin, P. Friedland, J. King and M. Stefik. Knowledge base management for experimental planning in molecular genetics. Proc. 5th Intern. Joint Conf. Art. Intell., 882-887, 1977.
20. R.S. Engelmores and H.P. Nil. A knowledge-based system for the interpretation of protein X-ray crystallographic data. Memo HPP-77-2 and Report No. STAN-CS-77-589, Stanford University, 1977.
21. see C. Scott. The science of science: what scientists read and why. Discovery, 20, 110-114, March 1959.
22. A.M. Thompson. The navigation system of the JPL robot. Proc. 5th Intern. Joint Conf. Art. Intell., 759-760, 1977.
23. I. Bratko, D. Kopec and D. Michie. Pattern-based representation of chess end-game knowledge. Computer Journal (in press).
24. C.C. Green. A summary of the PSI program synthesis system. Proc. 5th Intern. Joint Confer. Art. Intell., 380-81, 1977. D. Barstow. A knowledge-based system for automatic program construction. Proc. 5th Intern. Joint Conf. Art. Intell., 382-88, 1977.

problematika izvedbe in vključevanja velikih programskih sistemov realnega časa

v. doberlet
m. mrak
b. popovič
m. gazvoda

UDK 621.395.345:681.3

Iskra Telekomunikacije, Kranj

V priloženem članku podajamo nekaj specifičnih problemov, s katerimi se srečujemo pri gradnji in vključevanju velikih sistemov realnega časa z zelo kratkimi odzivnimi časi (reda 10ms). Kot tipičen primer smo vzeli programsko vodene telefonske centrale. Navedemo izkušnje, ki smo jih pridobili pri izvedbi in vključevanju tranzitnih central M10C v Jugoslovansko telefonsko omrežje. Poseben poudarek smo dali težavam, s katerimi smo se srečali ob postopnem povečevanju števila uporabnikov sistema do nekaj tisoč, ne da bi pri tem motili že priključene naročnike.

PROBLEMS OF IMPLEMENTATION AND INSTALLATION OF LARGE REAL TIME SYSTEMS. In the paper there are outlined some specific problems we usually come upon at the implementation and installation of large real time systems with very short response time (10 ms). As a typical example we are giving SPC telephone exchanges. Some experiences got with toll exchanges M10C in Yugoslav telephone network are presented. We underline troubles we met at the step-by-step increasing number of users up to some thousands without disturbing already connected subscribers.

V naglo razvijajoči se družbi, se paralelno s celotnim razvojem razširja tudi telekomunikacijsko omrežje. Preobremenjenost telefonskega omrežja je pred petimi leti prisilila J. PTT, da v praktično že popolnoma zasedeno telefonsko omrežje vključi nove, večje centrale. Večina PTT podjetij se je odločila za moderne SPC (Stored Program Control) centrale tipa Metaconta 10C, ki jih proizvaja ISKRA v sodelovanju z belgijsko tovarno "Bell Telephone Mfg. Co". Trenutno je v jugoslovanskem omrežju vključenih že osem tranzitnih central tega tipa.

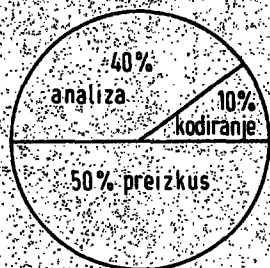
Vse telefonske operacije M10C centrale nadzorujeta dva ali trije procesorji s svojo programsko opremo. Uporabljajo se procesorji ITT 3200 z 32 bitno besedo, ki imajo možnost adresiranja do 512 k besed hitrega pomnilnika. Ker za telefonske probleme še nimamo dovolj učinkovitih prevajalnikov, so vsi programi pisani v zbirnem ("assemblerskem") jeziku. Ta je optimalen za procesiranje telefonskega prometa, vendar zahteva večji napor pri programiranju. Programski sistem posluhuje tudi več tisoč telefonskih linij in odzivni čas sistema mora biti reda 10 m sek.

Sposobnost centrale je odvisna od programske opreme, dimenzij multipleksa in sposobnosti delovnih organov (naprav za krmiljenje spojin in signalnih enot ter za

vzpostavljanje signalnih in govornih poti). S pomočjo programske opreme odkrivamo in analiziramo telefonske signale ter krmilimo delovne organe. Dodatna programska oprema nadzoruje delovanje same centrale, omogoča komuniciranje vzdrževalnega osebja s sistemom in posreduje v primeru izpada procesorja ali pomnilniškega bloka. Pomožna programska oprema za diagnosticiranje služi za preizkus posameznih enot centrale, za ugotavljanje statističnih podatkov o telefonskem prometu, za ažuriranje parametrov centrale ter za merjenje kvalitete telefonskih linij. Celotna programska oprema obsega cca 110 k. Vse informacije o telefonskih pozivih in stanju sistema morajo biti dostopne v časih reda 10 sekund, zato je celotna podatkovna baza v hitrem pomnilniku in obsega do 160 k spomina.

Nova tehnologija je prinesla poenostavitev večine telefonskih vezij v centrali. To je prineslo enostavnejšo in hitrejšo proizvodnjo ter učinkovitejšo kontrolo. Zapletene telefonske funkcije so se prenesle na programsko opremo centrale. Tako je postala programska oprema bistveni sestavni del centrale. Programsko vodene centrale lahko nudijo dodatne funkcije, npr. administrativne, ki jih ne moremo vnesti v elektromehanske, to pa poveča kompleksnost programske opreme. Pisanje programov je tako moralo prerasti individualne okvire in

oblikovala se je timska proizvodnja programske opreme. Izkušnje so pokazale, da so bili doseženi najboljši rezultati ob neprekinjenem delu kvalitetnih kadrov. To pomeni, da so bili isti ljudje prisotni pri vseh fazah proizvodnje programske opreme od analiz, kodiranja in končnega preizkusa na centralah.



Slika 1: Poraba časa pri posameznih fazah proizvodnje programske opreme.

Najširši nivo strokovnosti je bil potreben pri samem preizkusu, oziroma vključevanju central v obratovanje. Samo poznavanje programske opreme ni bilo zadostno, potrebno je bilo še detaljno poznavanje telefonskih signalizacij in načini delovanja posameznih vezij.

Vključevanje M10C central v jugoslovansko omrežje je bilo postopno, tako kot se je postopno čistila programska oprema. Ob začetku preizkušanja je bila verjetnost programske napake zelo velika. Ne moremo še govoriti o slabi kvaliteti programske opreme, ker je bilo število odpravljenih napak do trenutka, ko so bile realizirane prve testne zveze relativno majhno.

Če upoštevamo, da je verjetnost programske napake / funkcija vhodnih pogojev (2) in število odpravljenih napak, potem lahko rečemo, zakaj je bilo potrebno postopno vključevanje prometa. S tem se je prostor vhodnih pogojev postopno večal, ne da bi se problemi kopičili. Vsaki novi stopnji vključevanja so skoraj redno sledili izpadi računalniške konfiguracije in programske blokade različnih vezij. Toda upoštevati moramo, da je bilo to še vedno obdobje preizkušanja programske opreme. Jasno je, da se ni nadaljevalo z vključevanjem dokler ni bil ponujen promet zadovoljivo realiziran. Vključevanje novih linij ob prekinitvah delovanja, gotovo ni govorilo v prid sistema. Toda, ko je bil ponujen promet spet zadovoljivo realiziran, je nezaupljivost uporabnika usahnila. Poleg tega se je programska oprema z vsako naslednjo centralo tako izboljševala, da so bili koraki pri postopnem vključevanju vedno večji. Kasneje vključene centrale so v enem samem koraku prevzele celoten promet starih tranzitnih central.

Izboljšanje kvalitete programske opreme in možnost upo-

rabe različnih "programov na zahtevo", ki omogočajo preizkus posameznih enot centrale, merjenje kvalitete telefonskih linij in opazovanje telefonskega prometa, vse to je postavilo nov koncept vzdrževanja central. Posledica tega pa je, da je postalo vzdrževalno delo bolj kreativno in zato tudi bolj zanimivo. Z uporabo statističnih podatkov o telefonskem prometu na različnih centralah, dobiva PTT pomembne podatke za nadaljnje načrtovanje telefonskega omrežja.

Zaradi velike kompleksnosti programske opreme telefonskih central, ni smotno poskušati odpraviti vse napake. To bi bil zelo dolgotrajen postopek, ki ni več ekonomsko upravičen. Za vse sisteme se zahteva zanesljivost do določene meje. V času eksploatacije centrale se še vedno pojavljajo delne ali totalne odpovedi računalniške konfiguracije. To ne imenujemo izpad centrale, ampak je to le motnja v njenem delovanju. Centrala še ni prenehala opravljati svojih funkcij. Ena izmed funkcij centrale je namreč tudi držanje obstoječih telefonskih zvez. Ker se ob izpadu in takojšnjem ponovnem startu računalniške konfiguracije te zveze ne porušijo, smatramo izpad računalniške konfiguracije le kot motnjo. Za normalne vhodne pogoje lahko rečemo, da napak v programski opremi ni več, še vedno pa obstajajo v stranskih vejah programov, kamor zaidemo šele ob pojavu raznih neregularnosti. Take neregularnosti so signali, sprejeti iz drugih central, ki niso v skladu s specifikacijami. Neregularnosti lahko izvirajo tudi iz nepravilnega delovanja organov v sami centrali. Te nepravilnosti lahko tvorijo kombinacijo vhodnih pogojev, ki jih diagnostični programi ne znajo obdelati, ker take kombinacije sploh niso bile predvidene, ali pa ustrezne veje programa niso bile dovolj dobro preizkušene.

Vse te napake pa ne dosegajo več takega nivoja, da bi lahko blokirale centralo za dobo daljšo od nekaj minut.

Razpoložljivost sistema se zaradi takih napak zmanjša. Odprava napak, ki se pojavljajo zelo naključno, pa je ravno zaradi svoje naključnosti zelo težavna. Za to bi bili potrebni na vseh centralah strokovnjaki najvišjega nivoja ali pa centralni vzdrževalni center, kjer bi se zbirali vsi podatki iz vseh central (4). V centralnem vzdrževalnem centru, bi poleg analiz napak, tudi distribuirali vse intervencije in izvajali distribucijo korekcij ter pripravo sistemskih trakov za vse centrale (5).

Na koncu lahko ugotovimo, da so se SPC centrale tudi v jugoslovanskem prostoru uspešno uveljavile, povpraševanje se večja, uporabniki pa vedno več izkoriščajo možnosti, ki jih nudijo programske vodene sistemi.

Literatura:

- (1) M. Mekinda: "Novi prijemci razvoja programske opreme programske vodene telefonskih central"

10. Jugoslovanski simpozij o telekomunikacijah:
Ljubljana oktober 1976
- (2) M. Mekinda, S. D. Daniel, V. Doberlet, B. Popović,
V. Šila: "Zanesljivost, modularizacija in optimizaci-
ja programske opreme M10C".
Naloga za SBK 1977
- (3) V. Doberlet: "Avtomatsko obnavljanje pomnilniške
konfiguracije v multiprocesorskem sistemu".
Informatica, Bled 4-9 oktober 1976
- (4) B. Popović: "Avtomatizirani administrativni in vzdr-
zevalni centri".
10. Jugoslovanski simpozij o telekomunikacijah,
Ljubljana, oktober 1976
- (5) R. K. Ballard, D. W. Goble, M. K. Ward:
"Software Production for the Metaconta 10C System
In the Australian Telecommunications Commission",
Software Engineering for Telecommunication Switching
Systems, Conference Publication nr. 135
Salzburg February 1976

osnovni parametri in usmeritev načrtovanja računalniške mreže v Sloveniji

UDK: 621.395.74:681.3(497.12)

t. kalin
lajovic
v. rajkovic
d. viličič

Evropéan Informatics Network
Readington, in Institut J. Stefan, Ljubljana
Téxtil Commerce, Ljubljana
Institut J. Stefan, Ljubljana
Républiški računski center, Ljubljana

Članek poskuša podati grobo usmeritev za načrtovanje bodoče računalniške mreže v Sloveniji. Najprej prikaže obstoječe in planirane javne mreže za prenos podatkov in primerja sisteme s preklapljanjem paketov z omrežji s preklapljanjem vodov. Nato sledi komentar dosedaj opravljenih anket o potrebah po prenosu podatkov v Sloveniji in predlog vzpostavitve komunikacijskega podsistema za preklapljanje paketov v Sloveniji. Kot poseben primer so prikazane možnosti za komunikacije v prvi fazi Republiške računalniške mreže. Po kratkem prikazu prizadevanj drugod po Jugoslaviji je posebej poudarjena pomembnost skupnih akcij v celi Jugoslaviji, kjer bodo važno vlogo igrate PTT organizacije.

BASIC PARAMETERS AND GUIDELINES FOR PLANNING COMPUTER NETWORK IN SLOVENIA - The article attempts to propose basic guidelines for the planning of the future computer networks in Slovenia. Some operational and planned public computer communication networks are described and basic differences between line and packet switching are given. Two recent polls on data communication in Slovenia are examined and a proposal for a packet switching network given as the result. Data communication plans for the first stage of the Republic Computer Network are described. A short outline of efforts taking place in other parts of Yugoslavia is given with emphasis on future coordinated work where a very important part has to be fulfilled by the PTT organization.

I. UVOD

V moderni družbi postajajo računalniške komunikacije pokazatelj razvojne stopnje posamezne države. Tako je Simpson (1) pokazal, da bo leta 1985 narodni dohodek premo sorazmeren s količino prenešenih podatkov:

$$ND = 1.8 + 1.7 D$$

kjer je "ND" narodni dohodek na prebivalca v tisoč dolarjev (preračunanih na vrednost leta 1972) in "D" pa število megabitov na dan na zaposlenega.

Zgornja formula je seveda povprečje za več razvitih evropskih držav, od katerega odstopajo nekatere države, posebno Velika Britanija, kjer je več prenosa podatkov kot katerikoli državi razen USA.

To formulo se da razlagati na dva načina:

- dežela, ki bo imela velik nacionalni dohodek na prebivalca, bo visoko informacijsko razvita,
- družba, ki bo hotela dohiti ostale v višini nacionalnega dohodka, bo morala investirati v informacijski sistem, kajti investicije izključno le v proizvodne segmente družbe ne bodo mogle prinesiti zaželenega rezultata.

Različne dežele se razvijajo in se bodo razvijale na področju računalniških komunikacij na svoj način. Tarifna in investicijska politika posameznih PTT organizacij bosta daljši ton razvoju. Tako je določena razvojna smer ekonomska v eni deželi, v drugi pa ne. Vendar je treba predvsem usmeriti razvoj v takšno smer, ki je tehnično in ekonomsko daljnosežna.

Najprej bomo pregledali razvoj javnih mrež za prenos podatkov v svetu, nato pa situacijo na področju prenosa podatkov in končno možnosti izgradnje sistema za prenos podatkov v Sloveniji z oceno razvojnih faz in njihovih stroškov.

II. RAZVOJ JAVNIH OMREŽIJ ZA PRENOS PODATKOV V SVETU

1. Trenutno stanje ob uporabi analognih telefonskih omrežij

Vse PTT organizacije nudijo bolj ali manj zainteresirane svoje klasično telefonsko omrežje kot medij za prenos podatkov s pomočjo digitalno-analognih in analogno-digitalnih konverterjev-modemov. Pri tem so PTT organizacije dokaj konzervativne glede hitrosti, ki jih dopuščajo, čeprav sodobni modemi omogočajo večje hitrosti.

Modemi za večje hitrosti so zelo dragi, tako da lahko dosežejo cene, ki so precej višje kot cene terminalov, ki jih povezujejo. Zaradi močno omejenega frekvenčnega pasu, ki ga smejo uporabljati, so potrebne kompleksne modulacijske tehnike, kot recimo kombinacije fазne in amplitudne modulacije. Vse to seveda močno zviša cenó modemu. Zato gotovo prihodnost ne leži v uporabi analognih komunikacijskih sistemov, ki so standardizirani in prilagojeni za prenos glasu.

2. Digitalni sistemi za prenos podatkov

V severni Ameriki sta tako Kanada kot USA uvedli ali uvajata docela digitalne sisteme za prenos podatkov. To sta Dataroute (2) v Kanadi in sistem najetih digitalnih kanalov DDS, ki ga nudi ATT (3), (4), (5). Pri Bellovem sistemu so računalniki in terminali povezani s centralo s strižničnimi zankami, ki omogočajo prenos od 2400 do 56000 bitov na sekundo. V centrali se več takšnih kanalov multipleksira v T1 nosilni sistem, ki prenaša 1.544 Mbit/s. V tranzitnih centralah se ta tok demultipleksira, preusmeri na posamezne glavne smeri in zopet multipleksira in prenaša po sistemu "podatki pod glasom" s hitrostjo 1.544 Mbit/s do druge tranzitne centrale, kjer se postopek odvija v nasprotni smeri, to je demultipleksira na posamezne zanke.

Konec leta 1975 je bilo že 26 mest povezanih s tem sistemom. Poleg večjih hitrosti in nižjih cen prenosa, je posebno važna prednost novega sistema nizka cena priključnih naprav zaradi njihove enostavnosti v primerjavi z modemom enake hitrosti.

V začetku leta 1977 je DDS pokrival 96 mest. Glavni vzrok za tako hiter razvoj je na eni strani dober sprejem novih uslug, na drugi pa možnost uporabe obstoječih zvez.

V Evropi je v pogonu testni sistem Švedske, Norveške in Finske, ki imajo vsaka po tri stikalne centrale, med katerimi poteka prenos podatkov s hitrostjo 72 kb/s.

Večje število evropskih držav, med njimi Velika Britanija, Francija, Skandinavija, pričakuje, da bodo do leta 1980 lahko nudile uporabnikom najete digitalne kanale do 19,2 b/s. Prav tako namerava ZRN svoj EDS sistem, ki ga je instaliral Siemens in je bil namenjen pretežno manjšim hitrostim, do 200 b/s, prilagoditi digitalnemu prenosu in zato tudi zvišati hitrosti posameznih kanalov.

3. Sistemi s preklapljanjem paketov

Dosedaj opisani sistemi so predpostavljali zvezo med izvorom in ponorom podatkov. Druga možnost je, da se podatki spravljajo blizu izvora in se nato v obliki paketov z veliko večjo hitrostjo pošljejo proti aktivnemu elementu mreže, ki je najbližji ponoru. Ta vrsta prenosa podatkov se imenuje prenos s preklapljanjem paketov (angl. packet-switching). Ta način omogoča prilagoditev hitrosti pošiljatelja in sprejemnika, ki sta lahko različni, en pošiljatelj lahko zaporedno pošilja sporočilo poljubnemu številu prejemnikov, to pomeni, da sistem opravlja funkcijo multipleksiranja - demultipleksiranja.

Nekatera mednarodna gibanja na tem področju, med njimi tudi projekt COST 11, "Evropska računalniška mreža", nadalje EURONET, mreža za diseminacijo tehnične in znanstvene informacije, so doprinesla, da je prišlo do sprejetja mednarodnega standarda za interface med računalniško in terminalno opremo in pa mrežo za preklapljanje paketov (6). X25, kot se imenuje novi standard, v principu zagotavlja kompletno kompatibilnost terminalne opreme, ki bo zmožna pošiljati standardne pakete. X25 predvideva možnost vzpostavitve "virtualnega klica", to je navidezne zveze med pošiljateljem in prejemnikom. Ko je ta zveza vzpostavljena, je mogoča enostavnejša komunikacija. Takšen način "virtualnega klica" je primeren za prenos večjega števila paketov, manj pa je ugoden za prenos posameznih kratkih transakcij, saj zahtevani proces vzpostavitve zveze pomeni zakasnitev pri dostavi paketa in pa dodatni promet. Zato bo zelo verjetno kot dodatno možnost X25 nudil še tako imenovani datagram, kjer je vsak paket neodvisna enota, ki potuje od pošiljatelja k prejemniku ne glede na ostale pakete, ki so v tem času v tranzitu skozi mrežo. Takšen način je nedvomno primeren za krajše transakcije.

4. Pregled nekaterih najvažnejših javnih sistemov s preklapljanjem paketov

Tu ne bodo opisani zelo znani sistemi kot so ARPA - mreža, ki je bila že opisana (7), kot tudi ne EIN Evropska računalniška mreža, ker sta ti obe mreži zaprtega tipa in ne javni mreži, na kateri bi se lahko priključil vsak, ki želi.

Med javnimi sistemi s preklapljanjem paketov so danes najvažnejši:

a) Transpack (8), sistem s preklapljanjem paketov, ki ga bo francoska pošta dala v redno obratovanje konec leta 1978. V svoji osnovni verziji je načrtan tako, da lahko podpira 1500 terminalov, z možnostjo razširitve do 6000

terminalov. Čas prehoda paketa naj bi bil manj kot 0.25 sek in napačen manj kot vsak 10^{10} bit. Sistem je zasnovan tako, da bo uporabljal novi X25 standardni interface in ne nudi možnosti datagramov.

- b) Druga javna mreža s preklapljanjem paketov je EPSS, ki je v preizkusnem pogonu v Veliki Britaniji. Sedaj je nanj lahko priključenih 260 terminalov. Nudi virtualni klic in datagram.
- c) Med ameriškimi sistemi je gotovo najpomembnejši Telenet (9), to je mreža, ki jo je zgradila, in katere lastnik je firma Belt, Beranek, in Newton, na podlagi izkušenj, ki si jih je pridobila z izgradnjo ARPA mreže. Sistem je startal v začetku leta 1976 z desetimi vozlišči v desetih večjih mestih in s šestimi dodatnimi koncentratordi, z načrti za nadaljnjih 25 vozlišč. Kmalu po začetku dela je bilo na Telenet priključenih več kot 50 računalnikov. Sistem omogoča komunikacije med računalniki, kot tudi priključenih asinhronih terminalov, ki prenašajo znak za znakom, ker ima enoto za sestavljanje in razbijanje paketov. Sedaj beleži Telenet 15% prirast števila naročnikov na mesec!
- d) Poleg naštetih tudi Avstralija in Japonska pripravljata in delno že poganjata svoje mreže.

5. Preklapljanje paketov ali vodov

Mnoge od prednosti preklapljanja paketov v primerjavi z analognim prenosom podatkov so prisotne tudi pri preklapljanju vodov. Računalniško kontroliran sistem lahko doseže čase vzpostavitve zveze, ki so krajše kot 0.3 sekunde. Preklapljanje vodov ima nekatere prednosti pred preklapljanjem paketov:

- omrežje za preklapljanje paketov garantira transparentnost glede na vsebino in vrstni red dostavljanja paketov, ne more pa zagotoviti, da so časi dostavljanja v kakršni koli zvezi s časi oddaje paketov, saj so časi prehoda paketov skozi omrežje odvisni od obremenitve. Zato je sistem za preklapljanje paketov neuporaben za prenos podatkov, ki služijo za procesno kontrolo ali podobne aplikacije,
- na omrežju s preklapljanjem vodov je mogoče pošiljati govor in digitalizirani obliki,
- vzpostavitev zveze ima uporabnik zagotovljeno propustnost odgovarjajočo tipu povezave, ves čas zveze, pri sistemu za preklapljanje paketov pa je efektivna propustnost odvisna od prometne situacije v sistemu.

Kljub tem omejitvam v uporabnosti pa ima sistem za preklapljanje paketov celo vrsto prednosti, ki ga postavljajo v splošni uporabnosti daleč pred katerega koli drugega:

- a) Ekonomičnost: Časovna slika kaže kratke intervale ko se podatki prenašajo z velikimi pavzami med njimi. Tipični izkoristek linije je med 1 in 10%. Ta neekonomičnost je moč odpraviti z uporabo stikalnih računalnikov omrežja, ki multipleksira mnogo uporabnikov na iste linije s pomočjo paketne tehnike. Ker stroške računalnikov podaja mnogo hitreje kot stroške prenosa je ekonomično uporabljati računalnike za to funkcijo.
- b) Fleksibilnost: Mreža za preklapljanje paketov je v principu fleksibilna. Sprejme malo ali mnogo paketov, koliko, kdaj, iz katere točke mreže, o tem odloča uporabnik. Uporabnik ima na razpolago medij za prenos podatkov, ki ima spremenljivo hitrost prenosa, ni problemov predvideti asimetrično zvezo, z različnimi hitrostima v obeh smereh. Tudi stroški za prenos so elastični, saj večina omrežij predvideva plačevanje po količini prenesene informacije.
- c) Kvalitetne usluge:
- hitra vzpostavitev zveze 250 ms
 - polno duplexna povezava v primerjavi s polduplexnimi zvezami na komutiranih linijah.
 - zagotovljena nizka pogostnost napak. V mreže vgrajeni mehanizmi za odkrivanje napak zagotavljajo, da je

pogostost napak reda velikosti 10^{-12} , kar je vsaj za 5 redov velikosti boljše kot pri najboljših najetih linijah, zanesljivost javna omrežja za prenos podatkov s pomočjo preklapljanja paketov so načrtana s ciljno zanesljivostjo 1-2 uri izpada posameznega vozlišča na leto.

Glavno vprašanje pa so seveda stroški prenosa podatkov v omrežjih za preklapljanje vodov v primerjavi s preklapljanjem paketov za širšo skupino, kot je recimo nacionalna PTT ali skupnost PTT organizacij, kot je CEPT.

Ta dilema, ki je bila dolgo časa prisotna posebno v krogih PTT, je že razjasnjena (10). Pokaže se, da je cena prenešene informacije pri preklapljanju paketov bistveno nižja kot pri preklapljanju vodov. Tudi, če bi s pomočjo novih tehnik (PCM) za cel velikostni razred znižali ceno samega prenosa po kanalih, bi še vedno preklapljanje paketov ostalo cenejše (11). Edino prenos zelo velikih količin podatkov je cenejši v sistemu s preklapljanjem vodov. Zato so potrebne nove raziskave hibridnih računalniških prenosnih omrežij, ki bi najlažje, najceneje in z najmanj režije obvladovala tako prenos velikih količin podatkov, kot tudi kratke transakcije, za katere je najprimernejše preklapljanje paketov.

III. PRENOS PODATKOV V SLOVENIJI

1. Potrebe po prenosu podatkov v Sloveniji

V zadnjih treh letih sta bili o stanju in potrebah po prenosu podatkov pripravljene dve anketi.

Prvo je napravila poletni 1974 delovna skupina za prenos podatkov komisije za računalništvo in informatiko pri IS SRS, drugo pa poletni 1976 komisija za računalništvo in informacijski sistem pri GZ Slovenije.

Tabela 1. predstavlja povzetek iz obeh anket

Tabela 1. Letni prirastek

Anketa	1974	1975	1976	1977	1978	1979	1980
poletni 1974	13	115	203	83	103	16	7
poletni 1976			39	81	25	15	18

Če primerjamo rezultate obeh anket vidimo, da so bile poletni 1974 ocene po potrebah zelo optimistične in da so korekture v novejši anketi znatne. Te razlike so verjetno delno posledica slabega poznavanja problematike pred tremi leti, ko je bila delana prva anketa, deloma pa rezultati omejevanja uvoza računalnikov v letih 1975/76.

Vsekakor je letni prirastek, kot je predvidevan, izredno majhen. Vedeči pa je treba, da nekatere potrebe niso bile pokrite z anketo. Tu gre predvsem za terminalske mreže negospodarskih organizacij, kot so SDK, TNZ, Univerza, pa tudi mreža elektrogospodarstva. Ne sme pa se pozabiti na dokaj obširno mrežo centra, ki bi delal pretežno za javno upravo in na katerega bi bili priključeni terminali nameščeni v občinah po Sloveniji. Upoštevati je treba tudi bodočo mrežo avtomatskih merilnih postaj meteorološke in ekološke službe v Sloveniji, ki je sedaj v načrtovanju.

Zato bi bila realna predvidevanja lahko bistveno višja, posebej, če bo zaradi delne sprostitve uvoza računalniške opreme prišlo do montaže nekaterih večjih sistemov, ki bodo zmogni podpiranja terminalske mreže.

Če pogledamo razvojne načrte dveh velikih ustanov s področja finančnega poslovanja, to je Ljubljanske banke in Službe družbenega knjigovodstva SRS, vidimo, da imata dokaj slično topologijo s središčem v Ljubljani in vozlišči v večjih krajih.

Podobno sliko bi kazale tudi ostale načrtovane in obstoječe mreže.

Treba je poudariti, da je realizacija in dinamika realizacije teh načrtov odvisna od mnogih zunanjih elementov, vendar je važno, da se zavedamo potreb po prenosu podatkov v Sloveniji v srednjeročnem obdobju, ki so večje, kot bi to sklepali na prvi pogled.

2. Možnosti za vzpostavitev komunikacijskega sistema v Sloveniji

Iz pregleda obstoječih mrež in potreb za prenos podatkov v Sloveniji je mogoče sklepati, da bi bilo že v srednjeročnem obdobju, vsekakor pa pred letom 1985, smotno vzpostaviti sistem za prenos podatkov v SR Sloveniji. Takšen sistem bi seveda morala vzpostaviti PTT organizacija v Sloveniji v povezavi s sorodnimi podjetji v drugih republikah.

Možne oblike komunikacijskega sistema:

- sistem najetih linij - drag in nefleksibilen;
- v študiji "Računalniško omrežje SRS" iz leta 1975 (12) je bila predlagana rešitev z večkanalnimi modemi, ki z vrhunsko tehniko in kombinacijo fázne in amplitudne modulacije omogočajo prenos štirih neodvisnih 2400 b/s kanalov po enem samem telefonskem govornem kanalu. Takšna rešitev je lahko zanimiva za nekatere manj obremenjene relacije, ki zahtevajo prenos večje količine podatkov, vendar je zelo draga in neracionalna, saj jo zlahka nadomesti računalniško multipleksiranje, ki ga samo po sebi nudi omrežje za preklapljanje paketov, ki ima poleg tega še mnogo večjo perspektivo. Če brez povečanja propustnosti povezav med vozlišči komunikacijskega podsistema je mogoče doseči mnogo večje trenutne propustnosti za posamezne kanale, kot pri frekvenčnem multipleksu, kjer je frekvenčni pas rezerviran za posamezen kanal ne glede na to ali je ta v rabi ali ne. Pri sistemu s preklapljanjem paketov pa je na razpolago celokupna hitrost povezav za krajše čase, kar je posebej pomembno za sisteme z večjim številom krajših transakcij.
- Sistem za preklapljanje paketov, z vozlišči v Ljubljani, Celju, Mariboru, Novem mestu, Postojni, Novi Gorici, Kopru, Kranju. Same fizične povezave med vozlišči, ki naj bi bile prav tako kot vozlišča v kompetenci PTT podjetij, bi bile lahko speljane po alternativnih poteh, kjer bi to bilo mogoče (VF, UKV), da bi s tem zagotovili dovolj veliko varnost poslovanja. Za povezavo med komunikacijskim podsistemom in računalniki in terminali bi seveda moral služiti standardni X25 vmesnik, ki bo do realizacije tega načrta doživel vse potrebne korekture in tako postal zrel in uporaben standard, ki se mu bodo pokoravali tudi vsi najvažnejši proizvajalci računalniške in terminalske opreme. Že sedaj so ga namreč podprli številni dobavitelji vključno z IBM.

Na tem mestu bi bilo preuranjeno načrtati komunikacijski podsistem v večje detajle, saj so potrebne temeljite študije možnih tehniških rešitev in izkušenj, ki si jih bodo nabrale v naslednjih nekaj letih javna omrežja za preklapljanje paketov. Poglejmo pa, kakšne bi bile lahko možnosti za preklapljanje sporočil z interaktivnih terminalov v republiški računalniški mreži v njenem prvem obdobju.

3. Možnosti komunikacij v Republiški računalniški mreži v njeni najzgodnejši fazi

V prvem obdobju ob nastajanju in konsolidaciji RRM, ki naj

bi bila v začetku sestavljena iz obeh sistemov v RRC, računalnika za javno upravo v upravljanju Zavoda za statistiko SRS in interaktivne enote v upravljanju Univerze, je vpražanje fizičnih povezav računalnikov samih med seboj nekoliko manj važno, kar pa ne pomeni, da bi definicijo osnovnih parametrov le-tega smeli odlašati. Drugače se bo namreč pojavila nekompatibilnost v terminalski in komunikacijski opremi, ki je nikoli več ne bo mogoče prebroditi, ali pa bo povezava terjala veliko večja sredstva, kot če bi pravočasno načrtali komunikacijski sistem.

Da bi se to preprečilo, se je treba dogovoriti za naslednje:

a) Primarni tip terminala, ki naj bi omogočal doseg do podatkov na vseh računalnikih RRM. Sam po sebi se vsiljuje terminal, ki je TTY kompatibilen, to je, da prenaša podatke v asinhronem načinu s hitrostmi od 110 do 4800 b ali višje. To je edini tip terminala, ki ga bolj ali manj uspešno podpirajo vsi sistemi, ki pridejo v poštev. Eventualni drugi display terminali, ki sicer lahko dajejo več, kot npr. 711 (pri CDC), 3270 (IBM) itd., pa bodo lahko nastopali le v lokalnih mrežah posameznih centrov, za katere bo treba paziti, da ne bi preveč narasle, predno bi si bili na jasnem, kolikšne so potrebe za delo na drugih računalnikih. Pri definiciji vmesnika med aplikacijami in terminali bomo lahko uporabili izkušnje, ki jih bosta dotedaj nabrala EIN in EURONET z implementacijo virtualnega terminalnega protokola.

b) Komunikacijski sistem naj bi omogočal vsem terminalom, ki ne bi pripadali lokalni mreži enega od hostov RRM, da se priveže na katerikoli sistem v mreži. Možnih je več struktur takšne mreže:

- Zvezdasto s centralnim vozlom, na katerega so privezani vsi terminali, ki migrirajo od enega na druge računalnike. Ta vozle služi kot koncentrator in je povezan z dekoncentratorji, ki so pri vsakem od host računalnikov.
- Distribuiran komunikacijski podsistem (KPS). Tu nastopa ob vsakem hostu po en vozle, ki upravlja s terminali za "domač" host in posreduje sporočila drugim vozlom, ki jih predajajo svojemu hostu.

Druga varianta pomeni manjše telefonske stroške kot prva in manjšo koncentracijo funkcij ter s tem večjo zanesljivost.

V obeh primerih lahko vozli komunikacijskega podsistema (KPS), ki so ob posameznem host računalniku, demultipliksirajo informacijo v posamezna asinhrona vrata. To pa pomeni nepotrebno veliko opreme (vrata, modemi ali zero-modemi), pač pa ni potreben noben poseg v komunikacijski softver host računalnikov. Druga racionalnejša možnost je, da se ta funkcija prenese v čelne računalnike host računalnikov.

Ne glede na ta tehnični detajl (a z velikimi finančnimi posledicami) bi vzpostavitev zvez med terminalom in hostom potekala tako, da bi terminal najprej svojemu vozlu KPS dal podatek na kateri host bi se rad priključil. KPS bo na zaželenem hostu poskušal najti prosta fizična ali logična vrata (glede na metodo demultipliksiranja) in če jih bo našel, bo to sporočil terminalu in ga povezal s hostom. Zatem bi moral terminal izvesti standardni razpoznavni postopek, kot ga zahteva dotičen računalnik. V kasnejšem obdobju bi lahko premišljevali o enotnem postopku in enotnih gestih za celo mrežo, kjer bi transformacijo na specifično proceduro izvajal KPS.

V takšnem KPS je mogoče postaviti manjši koncentrator na mesto, kjer je v bližini večja koncentracija terminalov in ga povezati z enim ali dvema vozlova KPS. Če koncentrator stoji v drugem kraju in je nanj privezanih

več terminalov, se njegova cena hitro izplača s prihranki pri telefonskih vodih.

Promet med vozli KPS bi potekal v obliki paketov s fiksnim formatom, kjer bi bila dolžina podatkovnega polja od nič do 132 znakov. Kontrola prenosa od terminala do vozla bi potekala s pomočjo lokalnega odmeva med vozli pa s standardnimi postopki (parnost ali CRC) in med KPS in hostom spet s pomočjo odmeva.

Stroški za KPS bi se gibali okrog 50.000 \$ na vozle, če bi uporabili standarden mini računalnik, oziroma več kot pol manj, če bi se odločili za načrtovan multimikroprocesni sistem (13).

c) Kartični terminali imajo pri vsakem od proizvajalcev svoj komunikacijski protokol in bi bilo njihovo direktno prilaganje neracionalno. Prenosa datotek bi se lotili v okviru povezave host računalnikov.

d) Povezava host računalnikov. Če bi hoteli doseči polno povezavo, bi morali v vseh host računalnikih ali njihovih čelnih procesorjih predvideti softver, s katerim bi vzpostavili vse nivoje potrebnih protokolov. Pri tem bi morali upoštevati izkušnje EIN in se javnati že v skladu z novimi standardi na področju računalniških mrež (X25, EURONET VIP, itd.).

e) Iz povedanega je razvidno, da je spekter komunikacijskih problemov v mreži heterogenih računalnikov zelo širok in da je treba z načrtovanjem začeti dovolj zgodaj, te načrte verificirati in se nato po njih ravnati, da nas dogodki in nesmotrni nakupi ne prehitijo.

IV. PRIZADEVANJA V DRUGIH REPUBLIKAH

Iz podatkov, ki so na razpolago, je načrtovanje računalniških KPS najbolj napredovalo v SR Hrvatski (14, 15). Tam je v okviru republiškega sveta za informatiko in gospodarske zbornice Hrvatske, nastala študija o grobih elementih planiranja računalniške mreže. Predstavljeni so bili nekateri izhodiščni pogoji za izgradnjo mreže:

- mreža mora biti odprta
- mreža mora biti splošna in omogočiti zasnovano skupnih baz podatkov
- mora biti ekonomična in omogočiti izrabljanje prostih kapacitet računskih centrov
- mora biti modularna in fleksibilna
- mora biti hierarhično organizirana in mora omogočiti priključitev tako velikih sistemov na eni kot enostavnih terminalov na drugi strani
- mora omogočiti povezavo z mrežami drugih republik in grupacij (SDK, JLA, EIN, EURONET itd.)
- transparentna
- primerna za eksploatacijo različnih struktur informacijskih sistemov.

Načrt predvideva izgradnjo mreže z glavnimi vozlišči v Zagrebu, Osijeku, Reki in Splitu ter s koncentratorji v številnih drugih krajih in povezavo s Slovenijo, Vojvodino in BiH.

Realizacija načrta naj bi trajala pet let, stroški pa so ocenjeni na 173 milijonov dinarjev.

V. SKUPEK REPUBLIŠKIH MREŽ ALI JUGOSLOVANSKA MREŽA

Nedvomno je, da bi bilo zelo neekonomično načrtovati republiške mreže, ki bi bile le bolj ohlapno povezane med seboj v jugoslovanski sistem. Zato bo treba že v obdobju načrtovanja komunikacijskega podsistema združiti sile in razviti skupne poglede in skupne koncepte prenosa podatkov

v SFRJ. Takšno funkcijo povezovanja pa lahko uspešno vrši le skupnost PTT podjetij Jugoslavije, saj je za to poklicana in kompetentna. Glavni naporji bodo sedaj potrebni na področju osveščanja strokovne javnosti, posebej pa še organov samopravnih interesnih skupnosti, za PTT promet na vseh nivojih.

VI. ZAKLJUČEK

Potrebno je še naprej zasledovati razvoj računalniških komunikacij v svetu in sprti slediti potrebi v naši republici ter še nadalje poskušati predvideti smer v kateri je smiselno načrtati komunikacijski podsystem za računalniško omrežje v SRS.

LITERATURA:

1. K.M. Simpson: Planning for Data Communications, World Telecommunications Forum: ITV (1975) 2.1.2
2. D.J. Horton et al.: An Overview of Dataroute: System and Performance, Int. Conference on Comm. (1974)
3. R.T. James and P.F. Muench, AT & T Facilities and Services, Proc. IEEE, 60, 11 (nov. 1972)
4. B.G. Stueler: The Bell Systems Dataphone Digital Service, Proc. 2nd Int. Conf. on Comm. (1974)
5. C.R. Master, L.R. Pamm: The Digital Data System Lounches A New Era In Data Communications, Bell Lab. Report (1975)
6. CCITT: Proposals for New and Revised Series

T. Kalin et al.: Recommendations - Data Transmission Over Public Data Networks (1976)

7. "Računalniška mreža", Institut "J. Stefan" letno poročilo 1974
8. R. Despres, et d.: The French public packet Switching Service: The Transpack Network, Proc. 3rd ICCS, Toronto (1976) 251-260
9. L.G. Roberts: The Telenet network: The benefits of the public packet service. Proc. 3rd ICCS, Toronto (1976)
10. R.D. Rosner, B. Springer: Circuit and packet switching, A Cost and performance tradeoff study, Computer Networks 1, 1 (1976), 7-26
11. G.J. Conviello, R.D. Rosner: Cost Considerations for a Large Data Network, Proc. of ICCS (1974), Stockholm
12. Poročilo "Računalniško omrežje SRS", Fakulteta za elektrotehniko, Ljubljana (1975)
13. D. Viličić, T. Kalin, I. Lajovic, V. Rajkovič: Razvoj multiprocesorskega računalnika za komunikacijski center heterogenega računalniškega omrežja, Informatica 77, Bled, 1977
14. S. Turk: Determinante i osnovni pravci projektiranja i uspostavljanja kompjuterske mreže u SRH, Savjetovanje centara i korisnika AOP u SRH, Opatija (1976)
15. S. Dobrinčić: Determinante i osnovni pravci projektiranja i izgradnje informatičkog sistema SR Hrvatske, Opatija (1976)

priključitev hitrega čitalnika luknjanega traku na mikro računalnik

d. novak
m. kovačević
a. p. železnikar

UDK 681.3-181.4:681.327.44

Institut "Jožef Stefan"

Članek opisuje priključitev hitrega čitalnika FACIT PE 1000 na mikro računalnika F8-SDB (Software Development Board) in Motorola 6800. Poleg opisa vmesnika sta podana programa za čitanje in način vključitve teh programov v obstoječa monitorja.

CONNECTION OF THE PAPER TAPE READER TO THE MICRO COMPUTER - The following article describes the connection of the fast FACIT PE 1000 reader to the micro computer F8-SDB (Software Development Board) and to Motorola 6800. Besides interface description two programs for reading and the way of including these programs to the existing monitors is given.

1. UVOD

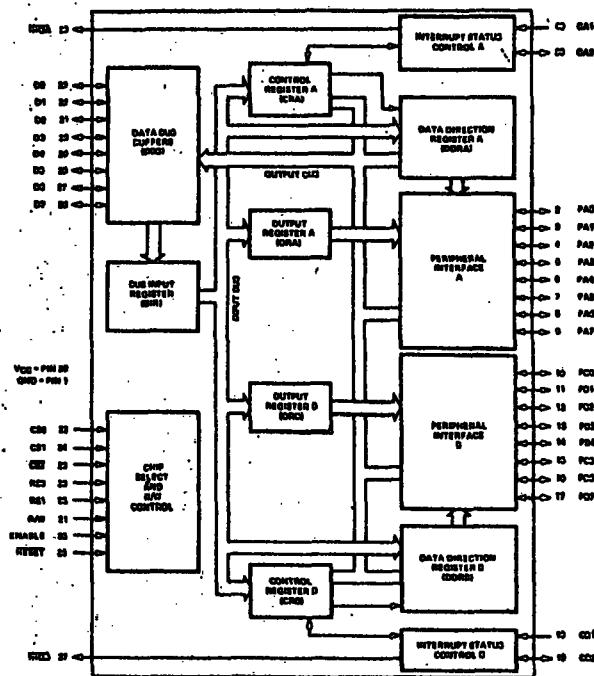
Pri večini manjših mikroprocesorskih sistemov je teleprinter edina periferna naprava, ki služi za vnašanje in izpis podatkov. Razlogov za to je več. Poglavitna pa sta univerzalnost (teleprinter združuje tastaturo, tiskalnik in luknjalnik traku) in nizka cena. Njegovi hibi, počasnost (110 Bodov) in hrupnost, pa sta pogosto vzrok, da ga zamenjamo z zmogljivejšimi perifernimi napravami, kot so: video terminal, hitri čitalnik in luknjalnik ali floppy disk, oz. kaseta.

S priključitvijo hitrega čitalnika traku (1000 ali 500 znakov/sek.) se bistveno skrajša čas, potreben za nalaganje daljših programov. Hitri čitalnik se priključi paralelno (vseh 7, oz. 8 informacijskih bitov se prenaša istočasno), z razliko od teleprinterja, ki se priključi serijsko.

2. OPIS VMESNIKA IN ČASOVNIH DIAGRAMOV KRMILNIH SIGNALOV

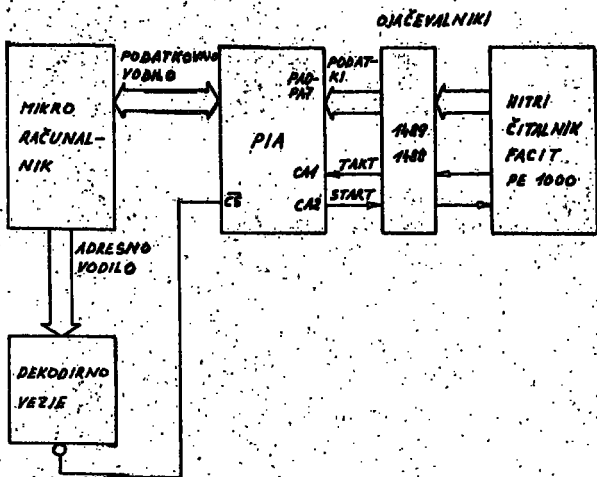
Standardno vmesniško vezje za priključitev perifernih enot na mikro računalnik je PIA (Peripheral Interface Adapter). Bločna shema vezja je prikazana na sliki 1. To vezje ima dve dvosmerni osembitni vodili (PA0-PA7, PB0-PB7) in štiri kontrolne linije (CA1, CA2, CB1, CB2), ki so namenjene za priključitev perifernih naprav. Funkcionalno konfiguracijo modula je mogoče programsko določiti. PIA (6820, 3820) je interno razdeljena na dve neodvisni konfiguraciji registrov. Vsaka polovica ima po tri registre: podatkovni register, register za določitev smeri in kontrolni register. Register za določitev smeri omogoča programsko definiranje smeri vsake izmed osmih vhodno-izhodnih linij. Kontrolni register omogoča programsko definiranje načina delovanja štirih kontrolnih linij.

S pomočjo integriranega vezja PIA je mogoče zelo enostavno priključiti hitri čitalnik luknjanega traku na mikro računalnik. Na sliki 2 je prikazan način priključitve. Podatkovne linije hitrega čitalnika D₀-D₆ so priključeni na V/I linije PA₀ - PA₆. Na linije PA₇, CA₁ in CA₂ pa so

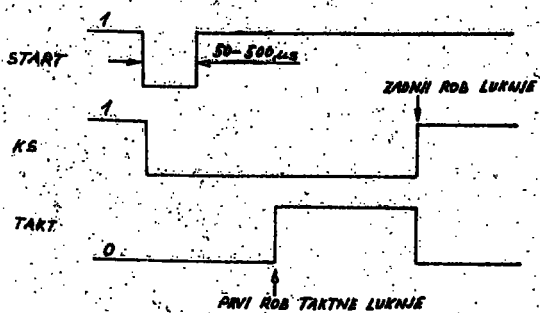


Slika 1.

priključene linije KS, TAKT in START. Delovanje hitrega čitalnika je zelo preprosto. Ko sprejme signal START, pomakne trak za eno kolono naprej. Med pomikanjem traku čitalnik odda signal KS (Klar Signal), ki signalizira, da se trenutno čita nova kolona. Signal TAKT se generira, ko pride taktna luknja pod senzor. Ta signal daje referenco kdaj je treba odčitati podatke, da bomo prečitali pravilne vrednosti. Slika 3 prikazuje časovne diagrame opisanih signalov.



Slika 2.



Slika 3.

3. PROGRAMSKA KRMILNIKA ZA PRIKLJUČITEV HITREGA ČITALNIKA NA MIKRO RAČUNALNIKA S PROCESORJEM 6800 IN F8

Sedaj, ko smo hardwareško priključili hitri čitalnik, potrebujemo še program - krmilnik, ki bo omogočil čitanje posameznih znakov s traku. Tak krmilnik lahko, kot subrutino, potem vključimo v ostale programe, ki čitajo podatke z luknjanege traku. V ta namen mora tak krmilni program generirati v skladu s časovnimi diagrami na sliki 3. startni impulz, če se seveda čitanje že ne izvaja. Ob pojavu taktnege impulza je potrebno prečitati znak iz podatkovnih linij. Pseudo kod takega krmilnika je sledeči:

Subroutine Krmilnik za hitri čitalnik

Inicializacija;

čakaj dokler je KS=0;

Formiraj startni impulz dolžine 50-500 µs;

čakaj dokler se ne pojavi TAKT (TAKT=1);

Prečitaj znak;

Endsubroutine

Poglejmo si sedaj subrutini za čitanje enega znaka za računalnika 6800 in F8 (Software Development Board). Logično sta obe enaki, saj morata obe ustrezati gornjemu

psevdo kodu. Zato bomo opisali samo eno. Listinga obeli sta na sliki 4. Če pogledamo program (za procesor 6800) vidimo, da je na začetku čakalna zanka (FINCH), v kateri čaka program dokler ne zavzame signal KS vrednosti 1. Torej čaka, da se izvrši čitanje, ki je v teku. Nato oblikuje, z vpisom ustreznih vrednosti, v kontrolni register (36₁₆ in 3E₁₆) in z zakasnilno zanko med obema vpisoma (WAIT), startni impulz, ki začne pomikati trak. Zatem se program spet ujame v čakalno zanko (TAKT), v kateri čaka, dokler se ne pojavi takti signal in odčita vrednosti na podatkovnih linijah.

Slika 4.

```

PRC   ORG   H'5000
PRC   EQU   H'19   PIA KONTROLNI REG.
PRD   EQU   H'18   PIA PODATKOVNI REG.
TU    EQU   H'6
TL    EQU   H'7
TK2   EQU   H'719
*
*
BR    FA0   INICIALIZIRAJ
IN    PRD   IF (KS=0) THEN NADALJUU
NI    H'80   ELSE GOTO FA1
BNZ   FA1
XS    TU    IF (TAKOJSNJA VRNITEV)
BP    FA    THEN VRNI SE NAZAJ V
*          GLAVNI PROGRAM
*          ELSE ČAKAJ V ZANKI
FA1   LI    H'36   POSTAVI START NA VR-0
      OUT   PRC
      LI    H'20
      LR    TLA
LOOP  DS    TL    ČAKALNA ZANKA ZA FORMI-
      LR    A:TL  RANJE START IMPULZA
      CI    H'00
      BNZ   LOOP
      LI    H'3E
      OUT   PRC   POSTAVI START NA VR-1
      IN    PRC   ČAKAJ DOKLER NI TAKT=1
      BP    WAIT
      IN    PRD   VCITAJ ZNAK
      NI    H'7F
      LR    TLA
      JMP   TK2
FA0   LIS   H'0   INICIALIZACIJA
      OUT   PRC
      OUT   PRD   DEFINIRANJE SMERI
      LI    H'3E
      OUT   PRC
      POP
      END

```

Krmilni program za hitri čitalnik FACIT PE1000 za mikro računalnik F8-SDB

```

PIAD  EQU   $EFFC   PIA PODATKOVNI REG.
PIAC  EQU   $EFFF   PIA KONTROLNI REG.
*
FINCH LDAA   PIAD   ČAKAMO NA KS=1.
      ASLA
      BCC   FINCH
      LDAA #35     GENERIRAMO START
      STAA PIAC
      LDAA #20
WAIT  DECA
      BNE  WAIT
      LDAA #3E
      STAA PIAC
      LDAA PIAC   ČAKAMO NA PODATEK
      ASLA
      BCC  TAKT
      LDAA PIAD   PREČITAMO PODATEK
      ANDA #7F
      RTS
*

```

Krmilni program za hitri čitalnik FACIT PE1000 za mikro računalnik 6800

4. VKLJUČITEV PROGRAMSKIH KRMILNIKOV V OBSTOJEČE MONITORJE

Opisane programske krmilnike je seveda treba vključiti v obstoječe monitorje. To je doseženo na dva različna načina. Za mikro računalnik 6800 je bilo potrebno napisati še nalagalni program, ki uporablja opisani krmilnik. Ta nalagalni program se potem aktivira z vnaprej definirano direktivo. Pri mikro računalniku F8-SDB (Software Development Board) pa je že obstoječi monitor napisan tako, da je mogoče nove krmilnike enostavno vključiti v obstoječi nalagalni program. Ob inicializaciji sistema definiramo naslov, kjer leži programski krmilnik za novo periferno napravo, mu priredimo simbolično ime (recimo FA) in nato definiramo kateremu ozmed sistemskih kanalov bo dodeljena določena periferna naprava. Obstaja šest kanalov (SI, SO source input/output; OI, OO object input/output; CI, CO console input/output). Če imamo krmilnik za hitri čitalnik na naslovu 5000₁₆ in če želimo čitati izvorne trakove, je potrebno ob inicializaciji storiti naslednje:

```
.L
.M 4072
4072 80 LF      LF pomeni "Load F"
4073 00 IA
4074 00 50      Naslovu 500216 smo priredili
4075 00 02      simbolično ime FA
4076 00 80
4077 00
.M :SI
:SI :TR :FA
:SO :TT
```

Pri tem je zanimiva še naslednja posebnost. Program za krmilnik se začne na naslovu 5000₁₆. Pri definiranju pa

smo navedli naslov 5002₁₆. Vzrok je sledeč: Pri subrutinskem klicanju krmilnika se začne krmilnik izvajati od definirane naslova dalje (tj. od 5002₁₆). Pri inicializaciji, ko prirejamo periferne naprave posameznim sistemskim kanalom, pa se enkrat izvede krmilni program, in sicer od naslova, ki je za 2 manjši od definirane, torej od 5000₁₆. Ta mehanizem omogoča enkratno inicializacijo perifernih naprav, če je ta potrebna. Zato je na prvih dveh lokacijah običajno skok na segment za inicializacijo, ali pa vrnitev v klicujoči program, če inicializacija ni potrebna.

5. SKLEP

Iz opisane priključitve je razvidno, da je mogoče z relativno malo napora (enostaven hardverski vmesnik, zelo enostaven programski krmilnik) priključiti običajne periferne naprave tudi na mikro računalnik. Eden izmed razlogov je dejstvo, da obstojajo že zelo integrirana in kompleksna splošna vmesniška vezja, ki jih je mogoče enostavno programirati. Taka vezja so napr. PIA, UART (Universal Asynchronous Receiver Transmitter), CRTC (CRT Controller), DMAC (DMA Controller), FDC (Floppy Disk Controller) itd.

6. LITERATURA

1. Motorola Semiconductor Products Inc., M6800 Application Manual 1975
2. Mostek, F8 Development System Documentation
3. Facit PE 1000 Tehnična dokumentacija

kros-assembly i simulator za mikro/mini računare

s. tripković
m. aškrabić

UDK 681.3 - 181.4.06

Institut za računarske i informacione
sisteme-IRIS: ENFERGOINVEST, Sarajevo

Poznato da se moć mikroracunala može povećati obezbjeđivanjem veće softverske podrške u smislu instrukcijskog skupa i "viših" prevodioca. Veličina datog programa i neophodnog prevodioca je obično veća od raspoložive memorije mikroracunara. Da bi se razriješio iskrslji problem, koristi se kros-assembly (kros kompajleri). Ovaj rad daje opštu strategiju, uključujući sve relevantne detalje, za razvoj takvih prevodilaca. Dalje, u radu je predložen jedan način razvoja simulatora.

CROSS-ASSEMBLER AND SIMULATOR FOR MICRO/MINI COMPUTERS. It has been realised for a while that the power of a microcomputer could be greatly improved by software support in terms of the powerful instruction set and necessary translating device. The size of a given program and the translating device is usually beyond the size of available microcomputer memory. As a remedy, something called cross-assembler (cross-compiler) is used. This paper gives the general strategy, including all relevant details, for such translating device development. The development is given in a well-structured form in order to be comprehensible as much as possible. Furthermore, we propose a way for a simulator development.

1. UVOD

Smanjenje cijene mikroracunara, povećanje njegove hardverske moći i veličine memorije ne uspijeva da premosti jaz između njegovih mogućnosti i zahtjeva korisnika. Moć mikroracunara se može povećati obezbjeđivanjem veće softverske podrške u smislu instrukcijskog skupa i "viših" prevodilaca. Veličina datog programa i neophodnog prevodioca je obično veća od raspoložive memorije mikroracunara. Proširenje memorije ne predstavlja rješenje, nego je ono nadjeno u razvoju kros-assembly i kros-kompajlera.

Neka su date dvije mašine, mašina X i mašina Y čije su veličine memorija M_x i M_y respektivno. Pretpostavimo da je $M_x \ll M_y$. Neka je veličina programa za mašinu X koji želimo da provedemo u objektni kod i veličina prevodioca veća od M_x i neka je programski jezik u kome je izražen dati program nerazumljiv za mašinu Y. Ovaj rad razmatra i daje algoritamsko rješenje dva problema:

1. napraviti niži prevodilac (assembly) koji će se izvršavati na mašini Y i na osnovu

izvornog programa pisanog za mašinu X generisati objektni kod razumljiv toj mašini;

2. napraviti simulator koji omogućava izvršavanje programa pisanih za mašinu X na mašini Y.

Algoritmi su razvijeni u koracima preciziranja (strukturirani pristup sa vrha ka dnu) i izraženi su u programskom jeziku PASCAL. Ovaj jezik je korišten zbog dobrih kontrolnih struktura, razumljivosti i zbog lakoe prevodjenja algoritama napisanih u ovom jeziku u bilo koji drugi, uključujući i assembly jezike.

2. KROS-ASSEMBLER

Pretpostavimo da mašina X prihvata programe napisane u assembly jeziku čija je sintaksa data u sljedećoj BNF notaciji:

```
(labela) ::= (slovo) | (labela){(slovo)|(cifra)}
(mnemonic) ::= (slovo)(mnemonic)
(crt) ::= /
(dolar) ::= $
```

<operand> ::= <labela> | <cijeli broj>
 <izraz> ::= <operand> <operator> <izraz> | <operand>
 <operator> ::= + | -
 <c> ::= komentár

/ označava početak komentara
 % označava kraj izvornog programa

Izvorni red definišimo kao tekst (na bilo kom ulaznom mediju) dužine do 80 karaktera:

<izvorni red> ::= <instrukcija> | <instrukcija>
 <crta> |
 <instrukcija> <crta> <c> | <crta>
 <c> | <dolar>

<instrukcija> ::= <mnemonik> | <labela> <mnemonik> |
 <mnemonik> <izraz> |
 <labela> <mnemonik> <izraz>

Kros-Assembler koji ćemo razviti pripada klasi dvo prolaznih assemblera i koristi sljedeća tabele:

SYMTAB - tabela simbola za storiranje labela
 MOT - tabela u kojoj su zapamćeni mnemonici instrukcija "niže" mašine
 MTRAK - izlazna tabela koja sadrži objektni kod
 VRED - tabela vrijednosti brojača izvornih redova.

Osnovna struktura apstraktnog algoritma je:

```

brojač := 0;
while ima izvornih redova do
begin
  pročitaj izvorni red;
  izvrši potrebne radnje;      (1)
  brojač := brojač + 1
end
  
```

Mnemonici i operatori su odvojeni posebnim znakom koji može biti blanko, zarez ili bilo koji drugi posebni znak, pa ćemo ga jednostavno nazivati "znak". Ako izvorni red ne sadrži labelu mora početi "znakom". Dužina labela nije ograničena na uobičajenih šest karaktera da bi se omogućila razumljivija imena. (Ova odluka utiče samo na tabelu simbola SYMTAB). Barem jedan blank mora da odvoji mnemonik od adrese (ili izraza) kao i izraz od komentara.

Sadržaj izvornog reda je zapamćen u ulaznom baferu "izvor" i predstavljen je datotekom tipa text. Pretpostavlja se da je sadržaj ulaznog bafera iscrpljen poslije 72 "kolone". Kada nastupi ova situacija, učitava se novi izvorni red. On se učitava poslije normalnog završetka tijela petelje (1) kao i po nastanku bilo koje greške.

Apstraktni iskaz

'izvrši potrebne radnje'

analizira sadržaj ulaznog bafera odnosno izvornog reda, a zadatak mu je da izgradi odgovarajuće simbole (labele i mnemonike) zajedno sa njihovom klasom. Klasa (vrsta) izvornog karaktera je zapisana na promjenljivoj klasi definisanoj sa

var klasa: (cifra, slovo, crta, dolar, blanko, mnemonik)

Simbol koji daje skaner se čuva u globalnoj promjenljivoj simbol deklarisanog kao

var simbol: text

Poslije ovih razmatranja se algoritam (1) može izraziti kao

L:=0; {brojač izvornih redova}

while ima izvornih redova do

begin

var ch: (cifra, slovo, crta, dolar, znak blanko, operator);

var komentar: Boolean; {ima/nema komentara}

var l: integer; {brojač karaktera u baferu}

var lc: integer; {brojač karaktera u labeli}

l:=0;

procedure getchar; {pročitaj karakter}

begin read (izvor, ch);

l:=l+1;

if l>72 then error

end

if ch in ['A'...'Z'] then klasa := slovo else

if ch = ' ' then klasa := mnemonik else

if ch = '/' then klasa := crta else

if ch = '\$' then klasa := dolar

else error;

repeat getchar {traži prvi ne-blank karakter}

until ch # ' '

case klasa of

slovo: begin lc:=0; (2)

repeat write (simbol, ch);

getchar;

if then not in

['A'...'Z']

v ['0'...'9'] v ' '

then error;

lc:=lc+1

until ch = znak;

syntab {poziv procedure syntab}

end;

```

mnemonik: operator; {poziv procedure
                    operator}
crta: komentar:= true;
dolar: begin
        'ima izvornih redova':=
        false;
        komentar:= true
      end
end; {of case}
procedure operator; {čitanje i testiranje
                    mnemonika}

if not komentar then
begin repeat getchar;
        if ch not in ['A'...'Z']
        then error;
        write (simbol, ch)
      until ch = '';
      if simbol not in MOT then error;
      'zapamti simbol u MTRAK'
    end
  else komentar:= false;

L:=L+1
end {glavna petlja}
'stampaj: kraj prvog prolaza'

```

Procedura `syntab` vrši smještanje pročitane labela u tabelu simbola `SYMTAB`, zapisuje njenu adresu (pokazivač) i dužinu.

Te podatke smješta u tabelu `POKAZIVAČI`. Jasno je da "`syntab`" ne mora biti deklarirana kao procedura, nego se može uključiti u složeni iskaz unutar grananja "slovo" čime se ostvaruje veća efikasnost i razumljivost.

Završetak algoritma (?) predstavlja kraj prvog prolaza kros-assemblyera. U njemu je korištena procedura `syntab` koja stvara labelu u tabelu simbola. Pošto su labela promjenljive dužine neophodno je zapamtiti i dužine labela.

Upotrebom pokazivača se problem sortiranja tabele simbola svoji na izmjenu pokazivača. Za pretraživanje uređene tabele pretpostavimo metod binarnog pretraživanja pokazivača u tabeli `POKAZIVAČI`.

```

Drugi prolaz kros-assemblyera je znatno jednostavniji i u apstraktnoj formi je dat sa:
begin 'sortiraj SYMTAB';
      'nadj ista imena i indiciraj grešku';
repeat 'pročitaj izvorni red';
      'pročitaj MTRAK';
if memorijska instrukcija then
begin 'razriješi način adresiranja';
      'sračunar aritmetički izraz'
    end;
if EQU then 'pridruži vrijednost labeli'
  else begin 'pridruži vrijednost instrukciji'
          'upiši instrukciju na izlazni medij'
        end;
      'stampaj greške';
if listing then stampaj
until ch='$';
stampaj SYMTAB;
stampaj broj grešaka
end

```

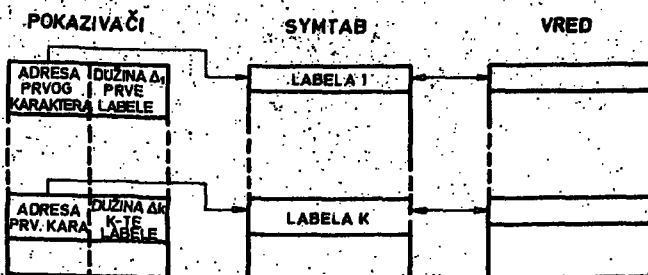
end

Prije definisanja forme izdavanja objektnog koda na perifernom memorijskom mediju (na pr. papirnoj traci) potrebno je definisati loader, koji može biti apsolutni ili relativni. U drugom slučaju njemu se mogu dodati još neke funkcije, kao prevodjenje nekih relativnih adresa u apsolutne, što će opet uticati na formu objektnog koda.

3. SIMULATOR

Simulator omogućava da se dobijeni objektni kod namijenjen mašini X izvršava na mašini Y. Ovo zapravo predstavlja simulaciju jednog računara na drugom. Obilježimo sa W_x i W_y dužine memorijskih riječi mašina X i Y respektivno. Pretpostavimo da je $W_x \leq W_y$.

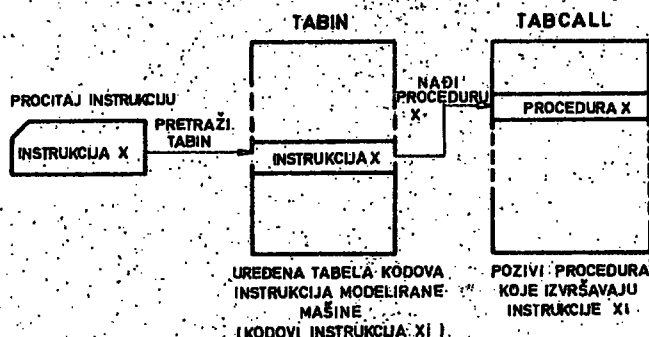
Način preslikavanja memorijskog prostora mašine X u memorijski prostor mašine Y mora biti definisan u loaderu. Kontrolnu jedinicu mašine X predstavlja sam simulator. Radni registri, indikatori i ostali neophodni registri mašine X se modeliraju u određenoj memorijskoj zoni mašine Y. Osnovni uslov koji se nameće simulatoru je da je sadržaj memorije u kojoj je smješten objektni kod mašine X i njenih "radnih registara" poslije izvršenja jedne instrukcije mašine X isti kao da je ta



$$ADR. PRV. KAR. K-TE LABELA = ADR. PRV. KAR. PRVE LABELA + \sum_{i=1}^{k-1} \Delta_i$$

instrukcija izvršena na mašini X. Analizator instrukcije mašine X možemo realizovati na dva načina:

- 3.1. Napraviti nezavisne procedūre za svaku instrukciju iz instrukcijskog skupa modelirane mašine sa grananjem na jednom nivou:



3.2.

Da bi se izbjeglo dupliciranje sekvenci i stalno pretraživanje TABIN, instrukcije se mogu podijeliti u grupe pri čemu se izbor procedūra vrši na dva nivoa (nivo grupe i unutar grupe).

Izvršimo podjelu instrukcija na:

GRUPA 1 instrukcije koje zahtijevaju čitanje memorije (memory reference)

GRUPA 2 ostale.

GRUPA 1 se može podijeliti na podgrupe

- A: instrukcije grananja
- B: instrukcije čitanja
- C: instrukcije pisanja
- D: radne instrukcije

GRUPA 2 se može podijeliti na

- E: registarske
- F: ulazno-izlazne
- G: specijalne

Osnovna struktura simulatora, ako prihvatimo drugi način je:

```
while not END do
```

```
begin 'pročitaj sljedeću instrukciju'
```

```
klasa := grupa;
```

```
if klasa = GRUPA 1 then begin 'pročitaj operand'
```

```
tip :=
```

```
podgrupa';
```

```
case tip of
```

- A: 'procedure A'
- B: 'procedure B'
- C: 'procedure C'
- D: 'procedure D'

```
end
```

```
end
```

```
GRUPA 2 else begin 'tip := podgrupa';
```

```
case tip of
```

- E: 'procedure E'
- F: 'procedure F'
- G: 'procedure G'

```
end
```

```
end
```

```
end
```

ZAKLJUČAK

U radu je razmatran problem konstrukcije kros-asmblera i simulatora i data su rješenja koja su relativno jednostavna za realizaciju. Rješenja su izražena u formi apstraktnih algoritama koji uključuju relevantne detalje. Ovakvo izražavanje potpomaže prevodjenju (kodiranju) u date programske jezike i predstavlja pokušaj formalizovanja razvoja kros-asmblera i simulatora.

Na početku razmatranja je pretpostavljeno važenje $M_x \ll M_y$. Ako pretpostavimo da je $M_x \approx M_y$ onda su data rješenja primjenljiva na klasu miniračunara, što znači da se programi razvijani za jedan miniračunar X mogu izvršavati na organizaciono drugačijem miniračunaru Y.

Reference

1. Donovan: "Systems Programming", MC Grow Hill N.Y. 1972
2. Wirth Niklaus, "The Programming Language PASCAL", Acta Informatica, 1971, 1,1, str. 35-63
3. Digital Equipment Corporation, PDP-8/E Small Computer Handbook, 1971

specifičnosti mikroprocesorske izvedbe mjerila broja okretaja

b. kette

UDK 681.3-181.4:531.77

Elektrotehnički Fakultet u Zagrebu

SAŽETAK: Analizirana je točnost i brzina mjerenja broja okretaja u mjernom sistemu koji koristi mjernu metodu s dva uzastopna impulsa. Ova metoda je odabrana jer omogućava bitno brži rad od integracione mjerne metode. Utvrđena je nužnost izvođenja aritmetičkih operacija množenja i djeljenja. Opisana je konstrukcija sistema na bazi mikroprocesora 4040. Postignuti rezultati su: maksimalna greška manja od 0,2% ; brzina izračunavanja rezultata bolja od 20 rez./sek. Izvršena je usporedba sa jednom starijom konstrukcijom tahometra sličnih svojstava a na bazi MSI i SSI elemenata.

THE CONSTRUCTION OF THE TACHOMETER USING MICROPROCESSOR 4040

The two successive pulse method is chosen for the tachometer or velocity measurements. The accuracy and speed of the measurements is analysed. The multiply and divide operations should be performed. The construction of the measuring system on the microprocessor 4040 basis is described. The next performances are obtained: the accuracy better than 0,2% ; the speed of the computing better than 20 results/sec. . It is given the comparison with oneother tachometer construction based on MSI and SSI elements.

UVOD

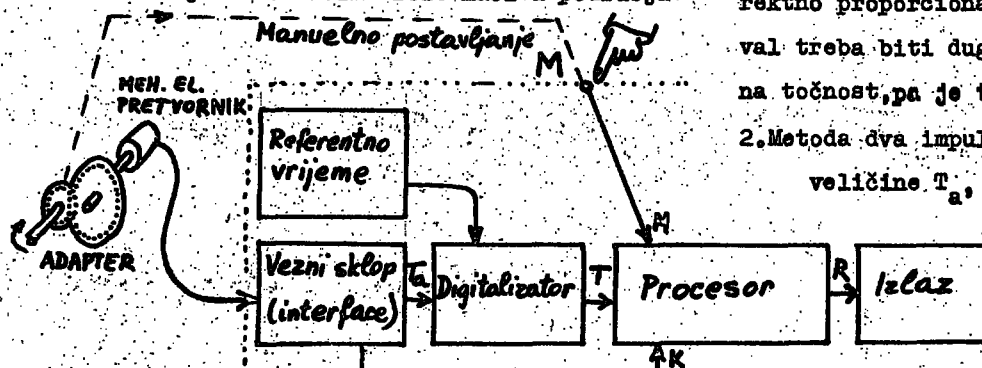
Blok shema mjernog sistema za određivanje broja okretaja, odnosno za indirektno određivanje brzine pokazana je na sl.1.. Rezultat mjerenja u digitalnom obliku može se koristiti u razne svrhe: optička indikacija, regulacija, alarm i sl.. Kao mehaničko električki pretvornik koriste se različiti tahogeneratori ili davači impulsa.

Ukoliko se mjerna veličina može naći u području

izvan nominalnog područja rada pretvornika, tada je potrebno upotrebiti adapter. Svakom tipu pretvornika pridružen je pripadni vezni sklop (interface) koji generira impulse ujednačenog oblika.

Mjerenje se može izvršiti na dva načina:

1. Integraciona metoda se zasniva na akumuliranju impulsa iz davača u vremenskom intervalu konstantnog trajanja. Akumulirani broj je direktno proporcionalan brzini. Vremenski interval treba biti dugačak da bi se postigla željena točnost, pa je to u osnovi sporo mjerenje.
2. Metoda dva impulsa se zasniva na mjerenju veličine T_a , koja je obrnuto proporcio-



Sl. 1.

nalna mjerenoj brzini. Ova metoda omogućava brzo mjerenje.

Na ulaze procesora privode se:

- T - Digitalizirana vrijednost kontinuirano promjenjivog perioda T_a .
- K - Konstanta davača. Podatak koji može biti trajno ožičen na veznom sklopu.
- M - Podatak adaptera. Tu veličinu namješta operater na nizu rotacionih preklopnika s indikacijom postavljene vrijednosti.

Nakon uključivanja mjernog sistema veličine K i M se ne mjenjaju, pa procesor treba stalno učitavati podatak T i izračunavati rezultat

$$R = \frac{K \cdot M}{T}$$

Rezultatom R izražava se izmjerena brzina s željenom težinom na pr. :okr/min, km/h i t.d..

PODRUČJE MJERENJA I TOČNOST

Maksimalna relativna greška izračunatog rezultata je:

$$\frac{\Delta R}{R} \cdot 100\% = \left(\frac{\Delta K}{K} + \frac{\Delta M}{M} + \frac{\Delta T}{T} \right) \cdot 100\%$$

Točnost rezultata ovisi o poznavanju ulaznih parametara i o dovoljno preciznom prikazivanju kako ulaznih parametara, tako i konačnog rezultata. Jedina mjerena veličina je parametar T. Ako je parametar T izražen u μ sek, tada će primjena davača 1 imp/okr ili 60 imp/okr., te rezultat izražen kao broj okr/min. zahtijevati pripadne konstante 60.000.000. odnosno 1.000000. U mjernom području od 100 okr/min do 10.000 okr/min ti davači će generirati impulse s periodom od 100 μ sek minimalno do 600 μ sek maksimalno. Upotrebom adaptera mjereno područje se proširuje, a veličina M predstavlja omjer broja zubaca u zupčanicima adaptera. Parametri K i M su apsolutno poznate veličine pa oni povećavaju grešku samo, ako se ne učitava i manje vrijedan dio numeričkih prikaza tih iznosa. Veličina M može biti iracionalan broj, pa u

takvom slučaju nužno postoji greška. Veličinu $\left(\frac{\Delta M}{M} + \frac{\Delta K}{K} \right)$ je moguće učiniti po volji malenom. Dakle, točnost rezultata R se može svesti na točnost koju se ostvaruje pri mjerenju intervala T. S rezolucijom mjerenja $\pm 0,5 \mu$ sek kod intervala T=100 μ sek maksimalno odstupanje $\frac{\Delta T}{T}$ će biti $\pm 0,5\%$, i ono brzo opada s porastom T. Grešku se čini i prilikom prikazivanja rezultata, ako se ne prikaže manje važan dio.

BRZINA MJERENJA

Ojelokupno vrijeme mjerenja sadrži komponente:

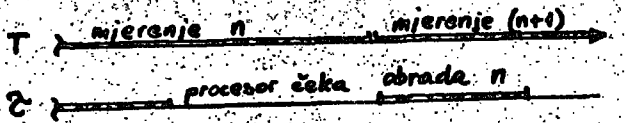
- T - vrijeme mjerenja
- τ - vrijeme obrade, t.j. vrijeme potrebno da procesor učita i obradi ulazne veličine te izda rezultat.

Vrijeme T ovisi kako o mjerenoj veličini, tako i o izboru davača s adapterom. U većini slučajeva područje T je od 100 μ sek do 1 sek.

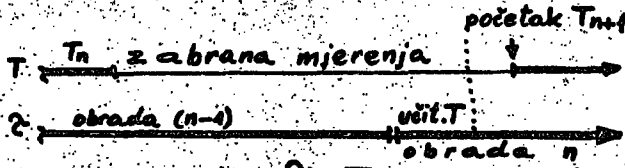
Vrijeme τ ovisi o izboru te o načinu rada procesora. Rješenje s minimumom sklopova zahtijeva τ u području 40 msek, što pruža mogućnost dobivanja oko 20 rezultata u sek, a to zadovoljava mnoge primjene.

Impuls iz davača označava završetak prethodnog i početak narednog mjernog intervala T, pa prema tome mjerenje može započeti samo u tom trenutku. Ako su skloповi izvedeni tako, da se istovremeno vrši mjerenje i obrada, tada brzina rada mogu ograničiti T ili τ , ovisno o njihovom međusobnom odnosu pokazanim na sl.2.

U slučaju $T > \tau$, ubrzavanje rada procesora nema smisla, jer on izbacuje rezultat brže nego što prima mjerene podatke, pa se nužno pojavljuju intervali u kojima procesor čeka. Važno je, da se u trenutku završetka prethodnog T mjerni sistem oslobodi za sljedeće mjerenje.

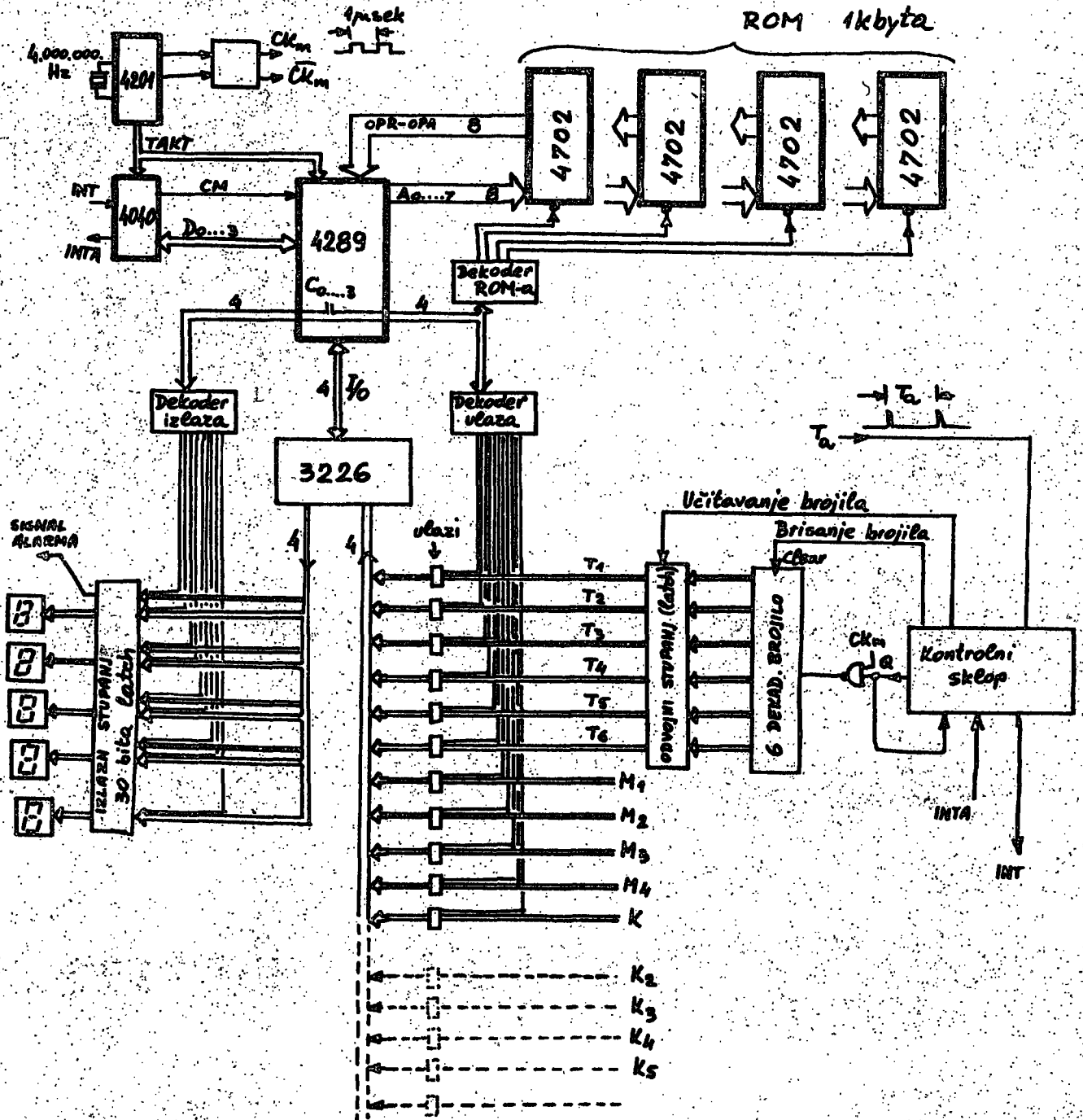


Sl. 2.a) $T > T_m$



Sl. 2.b) $T < T_m$

U slučaju $T < T_m$, brzina cjelokupnog procesa je ograničena brzinom rada procesora. Ako se u mjernom djelu isto brojilo koristi za mjerenje T i za čuvanje podatka T dok ga procesor ne učita, tada je nužno; po isteku prethodnog intervala T zabraniti novo mjerenje dok procesor ne obavi učitavanje. Nakon što je mjerenje dozvoljeno ono može započeti samo u trenutku kada se dobije impuls iz davača.



Sl. 3. Blok shema mjernog sistema s 4040 mikroprocesorom

MJERNI SISTEM S MIKROPROCESOROM 4040

Blok shema tahometra s minimumom sklopova na bazi 4040 mikroprocesora je pokazana na sl.3. Predviđeno je da procesor prihvaća ulazne podatke prikazane u NBCD kodu s primjenom dekad-ske aritmetike. Interval T smije poprimiti vrijednosti u području 100usek do 1 sek, a ako je izvan tog područja ostvaruje se pripadna signalizacija. Za mjerenje rotacija brzih od 10.000. okr/min potrebni su redukcioni zupčani i predviđen je prihvrat redukcionih omjera od 1.000 do 9.999. Nekoliko konstanti "K" za par tipova davača može biti upisane u ROM, a preko K- ulaza se dobiva podatak o tipu korištenog davača. Ako se želi omogućiti upotreba većeg broja različitih davača, tada je jednostavnije direktno učitati važniji dio pripadne konstante K.

Kontrolni sklop sadrži nekoliko bistabila, monostabila i logičkih sklopova. Taj sklop stalno prima impulse iz davača s kontinuiranom promjenjivim trajanjem T_a . Mjerenje intervala T označeno je signalom Q=1, pa se takt impulsi pribrajaju u brojilu. Po završenom mjerenju podatak T se prenosi u spremnik (latch), brojilo se čisti i počinje novo mjerenje/brojanje, samo ako je iz 4040 primljen signal INTACK. Taj signal označava završetak obrade prethodnog podatka. Poslije mjerenja intervala T kontrolni sklop posatlavlja signal INT, a mikroprocesor 4040 prihvaća taj zahtjev prekida ako je prethodna obrada završena.

Opisani sistem omogućava mjerenja u području 100- 15.000. okr/min s maksimalnom relativnom greškom manjom od $\pm 0,2\%$ ili $\pm 0,5$ digita. Greška je veća pri visokim brzinama koje generiraju $T \leq 250$ usek, ali to se može izbjeći pravilnim izborom davača i adaptera.

Za izvršenje pojedinih djelova programa potrebna su vremena:

- početno brisanje sklopa te učitavanje K i M	0,2 msek
- množenje K i M (5digita x x 4digita = 9digita)	7,0 msek
- učitavanje T	0,1 msek
- priprema za djeljenje s pripadnom kontrolom	0,5 msek
- djeljenje (9digita : 6digita = 4+0,5 digita)	28,0 msek
- kodiranje rezultata za izlaz na optičku indikaciju	0,3 msek
- izlaz	0,2 msek

Ukupno trajanje obrade 36,3 msek

Specifičnost opisanog sistema je da ne koristi RAM elemente. Za čuvanje međurezultata koriste se indeksi registri mikroprocesora. Prilikom izvođenja operacija podaci K i M su uništeni, pa je pri svakoj obradi potrebno ponovno učitavanje K i M te njihovo množenje.

Najpopularnije metode za izvođenje množenja su: metoda akumuliranja parcijalnih produkata i metoda množenja s tabelom množenja upisanom u ROM (Look up). Posljednja metoda je brža pa se primjenjuje u opisanom sistemu. Budući da je područje mjerenja definirano od 100 do 15.000. okr/min te veličine(K*M) i T nisu potpuno nezavisne. Prije početka djeljenja vrši se kontrola dali će rezultat biti unutar mjernog područja, s mogućnosti signaliziranja pojave neregularnih stanja. Djeljenje je izvedivo metodom sukcesivnih odbijanja s restauriranjem ostatka, ili metodom množenja s vjerojatnim digitom kvocijenta. Vjerojatni digit kvocijenta određuje se sukcesivnim odbijanjem samo najvišeg digita divizora od dva najviša digita dividenda. Posljednja metoda brže dovodi do rezultata, pogotovo kad se koristi tabela množenja u ROM-u.

Jedna starija konstrukcija tahometra sličnih svojstava koristila je 35 MSI i 90 SSI elemenata. Na izlazu su bili potrebni dodatni dekoderi za 7 segment ili koju drugu vrstu indikacije. U toj konstrukciji računanja su izvedena metodom brojanja. Takve metode se koriste u digitalnim diferencijalnim analizatorima. Isto brojilo se koristilo u procesu mjerenja i u procesu računanja. Takva konstrukcija ne dozvoljava isto-vremena mjerenje i obradu, pa je to pružalo sporiji rad.

MOGUĆNOSTI POBOLJŠANJA MJERNOG SISTEMA

Opisani sistem je jeftin. On koristi minimalnu količinu sklopova a upotrebljen mikroprocesor je skromnih mogućnosti. Postignuta točnost i brzina zadovoljavaju mnoge primjene.

Veću točnost se lako postiže korištenjem dužih riječi za prikaz ulaznih podataka i točnijim mjerenjem intervala T.

S današnjim mikroprocesorima operacije množenja i djeljenja se nužno izvode programima, čije izvođenje zahtijeva dugačka vremena. Brzinu rada opisanog sistema je moguće poboljšati dodavanjem RAM memorije kako bi se otklonila potreba stalnog učitavanja i izračunavanja konstanti. Daljnje ubrzanje procesa izračunavanja moguće je postići s mikroprocesorima koji obrađuju dužu riječ. U tom slučaju prikladan je binaran prikaz podataka, što zahtijeva binarnu aritmetiku koja je brža. Binarni prikaz podataka na ulazu se lako izvodi, jer K i M se postavljaju kodirano a za formiranje podatka T dovoljno je upotrebiti binarno umjesto dekadskog brojila. Na izlazu je potrebna konverzija programom ili sklopovima. Optimalni kompromisi između zahtjeva točnosti i zahtjeva brzine postižu se primjenom binarnih prikaza s pomičnim zarezom. Obzirom na standardno komercijalno dobavljive elemente pri-

kladan je format 12 bita za mantisu i 4 bita za eksponent. Kompleksniji mikroprocesori također zahtijevaju programsko izvođenje aritmetičkih operacija, pa je daljnje ubrzavanje moguće samo dodavanjem specijalnih sklopova. Procjena potrebnih vremena za izvođenje operacije djeljenja navedene su u tabeli.

Sistem 4040 - programom, dekadska aritmetika 9digita : 6digita	28 msek
Mikroprocesori 8-16 bita riječ - programom, 12 bita : 12 bita binarno	2 - 5 msek
Specijalna sklopovska rješenja - 12 bita : 12 bita binarno	1 - 20 μ sek

ZAKLJUČAK

Mjerni sistem mora obaviti operacije množenja i djeljenja. S minimumom sklopova ostvaruje se sistem čije su maksimalne netočnosti manje od $\pm 0,2\%$, a vrijeme obrade oko 40msek, što daje preko 20 rezultata u sekundi. Točnost mjerenja se lako poboljšava, a ograničena je s točnosti mjerenja vremenskih intervala. Brzina rada ovisi o brzini dobivanja podataka iz davača ili o brzini obrade podataka. Brzina obrade podataka ovisna je o trajanju procesa djeljenja. U današnjim mikroprocesorskim sistemima proces djeljenja se izvodi programom, a to zahtijeva relativno dugačka vremena. Za brzo izvođenje djeljenja nužno je korištenje specijalnih sklopova. Za postizanje većih točnosti i brzog rada treba koristiti binarnu aritmetiku i prikaz s pomičnim zarezom.

Tahometar je razvijen na Zavodu za Elektroniku. Posebno se ističe zalaganje studenta Željka Mikšić, koji je u okviru konstrukcionog i diplomskog rada doprinjeo realizaciji uređaja.

preizkušanje mikro- računalniških modulov

m. kovačević
d. novak
j. novak
a. p. železnikar

UDK 681.3 - 181.4.001.4

Institut "Jožef Stefan" Ljubljana

Članek opisuje praktične prijeme pri preizkušanju in odkrivanju napak v mikroročunalniških modulih modularnega mikroročunalniškega sistema. Izkušnje, ki so opisane, so nastale pri razvoju modularnega mikroročunalniškega sistema na osnovi procesorja 6800.

MICROCOMPUTER MODULE TESTING - The following article describes practical ways of testing and detecting of faults in microcomputer modules of the modular microcomputer system. The described experiences arose together with the development of the modular microcomputer system, based on the processor 6800.

1. Uvod

Modularni mikroročunalniški sistem gradimo nad univerzalnim vodilom, ki vsebuje podatkovne linije ter nabor kontrolnih linij, ki je bogatejši, če sistem obsega več možnosti in načinov delovanja. Koncept modularnega mikroročunalniškega sistema nam prinaša več različnih ugodnosti tako s stališča razvijalca sistema kot s stališča uporabnika. Za uporabnika je važno, da lahko konfigurira sistem v skladu s potrebami. Z drugimi besedami, uporabnik kupuje samo tiste module, ki so neobhodni za njegovo aplikacijo. Pri razvoju modularnega mikroročunalniškega sistema uporabljamo koncept "top-down" načrtovanja. Najprej specificiramo vse lastnosti bodočega sistema. Nato določimo module, njihove funkcije ter kontrolne signale vodila, ki krmilijo delovanje modula. Pri testiranju prototipnih modulov srečujemo več različnih tipov napak, ki imajo različne izvore. Trije osnovni tipi napak pri prototipnih modulih so logične ali konstrukcijske napake, realizacijske napake ter napake zaradi defektnih elementov. Seveda pri serijskih modulih ni konstrukcijskih napak; realizacijske napake ter napake zaradi defektnih elementov ostanejo aktualne tudi v tem primeru. Pri odkrivanju in identifikaciji teh napak uporabljamo standardno opremo (merilne instrumente, signal-generatorje, osciloskope itn.) ter razne druge hardwarske in softwarske pripomočke, ki jih moramo najpogosteje narediti sami.

2. Preizkušanje mikroročunalniških modulov v obstoječem modularnem sistemu

Najbolj enostaven način preizkušanja mikroročunalniških modulov ustvarimo, če imamo na razpolago modularni mikroročunalniški sistem z ustrežno programsko opremo, v katerega enostavno vstavljamo module, ki jih preizkušamo. Na ta način preizkusimo modul funkcionalno. V primeru, da po vstavitvi preizkusnega modula, sistem ne dela pravilno, pristopimo k identifikaciji napake za katero zagotovo vemo, da je na preizkušanjem modulu. Tak preizkusni modularni sistem vodi program

ki v času preizkusa angažira vse module sistema, hkrati pa daje poročila o rezultatih testiranja za preizkušeni modul. Blok shema takšnega sistema je podana na sliki 1.



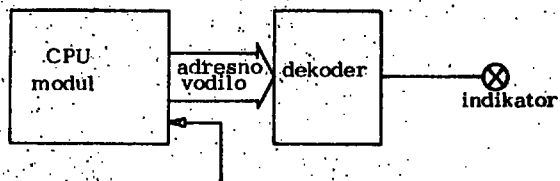
Slika 1. Bločna shema za funkcionalno preizkušanje modulov

Takšni sistemi so posebej primerni za funkcionalno preizkušanje industrijsko izdelanih, serijskih modulov. Pri preizkušanju prototipnih modulov se moramo posluževati drugih načinov in pripomočkov.

3. Preizkušanje prototipnih modulov

Pri preizkušanju realiziranih prototipnih modulov navadno začnemo pri CPU (Central Processing Unit) modulu, tako da najprej preizkusimo in v primeru nepravilnega delovanja usposobimo "srce" vsakega mikroročunalniškega sistema, uro. Pri sistemih s procesorjem 6800 je ura dvofazna in ima navadno možnost podaljševanja ene ali druge polperiode. Ta preizkus najlažje opravimo z osciloskopom, tako da s krmilnimi signali podaljšujemo eno ali drugo polperiodo in dobimo pri tem ustrezno sliko na zaslonu osciloskopa. Šele potem, ko smo prepričani, da deluje ura, lahko pristopimo k preizkušanju celega CPU modula. Jasno je, da v tej fazi testiranja ne moremo sprožiti uzvajanja nobenega programa glede na to, da v že "zgrajenem" sistemu nimamo še pomnilnikov. Lahko pa opazujemo kaj se zgodi ob resetiranju CPU čipa. Ob resetiranju CPU modula se mora na

adresnem vodilu pojaviti naslov prvega zloga restartne adrese ter READ signal na kontrolnem vodilu. Naslov prvega zloga restartne adrese identificiramo tako, da na adresno vodilo priključimo ustrezní dekodér z indikatorjem (slika 2).



Slika 2. Preizkušanje CPU modula na osnovi RESET signala

READ signal najlažje identificiramo z osciloskopom. Potem, ko smo opravili preizkus CPU modula pristopimo k razširitvi našega sistema. V tem trenutku bi bilo koristno usposobiti ter vključiti v sistem PROM pomnilnik. Če namreč želimo preizkušati sistem kot celoto, moramo to opraviti s programom, ki bo rezidenten v določenem pomnilniku (ROM ali RAM). PROM pomnilnik preizkusimo spet tako, da opazujemo reakcijo sistema na RESET signal. Če je v restartni lokaciji sistema, ki je znotraj pomnilniškega segmenta na PROM pomnilniškem modulu vpisana določena vrednost, jo moramo ob RESET signalu identificirati na podatkovnem vodilu. Tudi v tem primeru moramo uporabiti ustrezní dekodér z indikatorjem. Če želimo biti prepričani, da PROM pomnilniški modul pravilno deluje, moramo preizkusiti še pravilnost čitanja iz lokacije katere naslov je eniški komplement k naslovu restartne lokacije sistema. Pri tem je vsebina v tej lokaciji enaka eniški komplementu k vsebini restartne lokacije sistema. Ta preizkus lahko opravimo na naslednji način: v restartno lokacijo sistema vpišemo naslov lokacije, ki je eniški komplement k restartnemu naslovu sistema ter v omenjeno lokacijo vpišemo vsebino, ki je eniški komplement k vsebini restartne lokacije. Na podatkovno vodilo priključimo ustrezní dekodér z indikatorjem. Če v obeh primerih indikator reagira na RESET signal sklepamo, da PROM pomnilniški modul pravilno deluje. Do minimalne konfiguracije sistema manjka še samo določen I/O modul ter RAM pomnilniški modul.

3.1. Testiranje I/O modula za teleprinter

I/O modul za printer je serijski vmesnik, ki je narejen na osnovi kompleksnega integriranega vezja ACIA (Asynchronous Communication Interface Adapter - 6850). Modul bomo testirali s pomočjo programov, ki so v PROM pomnilniku. Najprej prekontroliramo, če na ACIA vezje prihajajo signali CS (Chip Select) ter R/W (Read/Write). To opravimo tako, da sprožimo delovanje programa, ki ima logični potek:

A (PODATKOVNI REGISTAR ACIA)
KONTROLNI REGISTER A
PONOVITEV PREDHODNIH KORAKOV

Ob izvajanju takšnega programa opazujemo na osciloskopu signale na pinih CS, RS0, RS1, ter R/W, ki morajo imeti periodični potek z ustrežno frekvenco (periodo določa čas izvajanja teh ukazov). S tem smo prekontrolirali

rali adresno dekodirno vezje I/O modula. Z osciloskopom prekontroliramo še delovanje ure, ki določa hitrost prenosa podatkov. Takšna dva testa nam zagotavljata, da I/O modul za priključitev teleprinterja deluje pravilno.

3.2. Testiranje pomnilnikov

Pri testiranju pomnilnikov moramo upoštevati stroge kriterije. Vsak bit bomo pretestirali na vpis "0", vpis "1" ter prehod $0 \rightarrow 1$ ali $1 \rightarrow 0$. Ta test opravimo s pomočjo programa (tabela 1), ki predstavlja tak mikroračunalniški sistem, ki ima monitor z naslednjimi podprogrami:

- CRLF - izpis znakov CR in LF na teleprinterju,
- OUTS - izpis presledka,
- OUT2H - izpis dvomestnega heksadecimalnega števila, ki je vsebovano v lokaciji, ki jo naslavlja indeksni register,
- PMES - izpis poročila vsebovanega v pomnilniku na naslovu, ki ga določa indeksni register, do lokacije, ki vsebuje vrednost 4.

Program je napisal tako, da v določenih lokacijah pomnilnika drži začetni in končni naslov pomnilniškega bloka, ki ga preizkušamo. Program preizkuša vsak bit na vpis '0', vpis '1' ter prehod '0 \rightarrow 1' ali '1 \rightarrow 0'. V primeru, da odkrijemo napako, program izpiše naslov in dejansko ter pričakovano vsebino lokacije, ki vsebuje napako. Iz seznama napak, ki se izpiše na teleprinterju lahko sklepamo o naravi napake. Če dobimo naprimer tak izpis:

```

4000 FF 55
4002 FF 55
4004 FF 55
4006 FF 55

```

Sklepamo, da adresna linija A0 ni pravilno pripravljena na pomnilniške elemente.

Pri testiranju dinamičnih pomnilnikov moramo skrbeti za testiranje pravilnosti delovanja ciklov za osveževanje. Program, ki testira dinamične pomnilnike mora priskrbiti dovolj časa med vpisom v lokacijo ter testiranjem vsebine, da se izvrši dovolj ciklov za osveževanje, ki nam garantirajo, da tudi sistem za osveževanje pravilno dela. Pri testiranju pomnilniških modulov se tudi (enako kot pri testiranju I/O modulov) priporoča testiranje naslovov ter pravilnosti R/W signala na pomnilniških elementih. Pri tem je logična zgradba programa, s katerimi si pomagamo, naslednja:

ČITANJE IZ POMNILNIŠKE LOKACIJE
VPISOVANJE V POMNILNIŠKO LOKACIJO
PONOVITEV PREDHODNIH KORAKOV

4. Sklep

Preizkušanje mikroračunalniških modulov pogostokrat zahteva predvsem vztrajnost pri delu. Če odkrijemo napako na modulu, je včasih potrebno pretestirati vsak element, vsako povezavo ter vsak signal. Napake, kot so slabi stiki med nogami integriranih vezij in prednožij, slabi stiki na wire-wrap trnih, razli loti itn. pogostokrat ne dajo osnove za logično sklepanje o izvoru napake. Pri preizkušanju modulov se moramo držati predvsem sistematičnosti.

graphics machine independ- ence with help of network graphics protocol

d. gojanović

UDK 681.327.13

Zavod za elektroniku
Elektrotehnički fakultet
Zagreb

MAŠINSKA NEZAVISNOST PROTOKOLA U GRAFICI POMIČU MREŽNOG GRAFIČKOG PROTOKOLA-Ovaj članak opisuje strukturu predloženog mrežnog grafičkog protokola 2) i neka iskustva dobijena prilikom implementacije i testiranja tog protokola.

This paper presents structure of the proposed Network Graphics Protocol 2) and some experience gained during its implementation and testing.

1. Introduction

Every time a graphical display terminal is attached to a computer in a unique configuration, a new graphics software system must be written to support it.^{1/} The design and implementation of this software is an activity that absorbs time and energy in large quantities.

Recognising this problem, some progress has been made in building so called "device independent" graphics systems. The systems must be sufficiently general purpose to support a wide variety of applications with a range of graphics peripherals. There is also the desirability that graphics system should also be high level. They should hide from the programmer the low features of hardware.

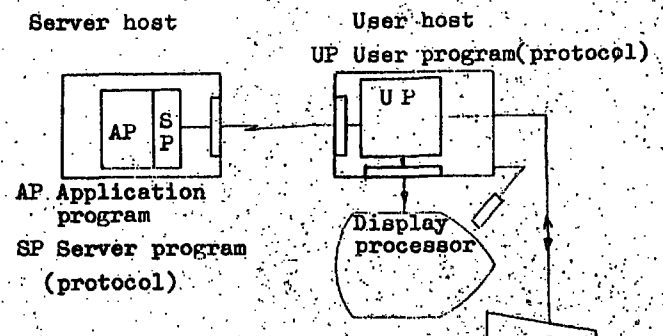
One step towards general graphics systems is graphics in computer networks, where it is possible that users with various different kinds of display hardware at different sites in a network use common application programs. In this way it would not be necessary to duplicate applications programs at every installation.

The benefits of such systems are wide. Without high initial investment in hardware and software it will be possible to use existed application programs located anywhere in network on easy reachable terminal at site of the user, because of adaptability of proposed network graphics protocol to variety of graphics terminal.

The Network graphics protocol is defined as interface between applications programs and graphics terminal (Fig.1)

Fig 1

A graphics application program running in one computer (the Server Host) drives a display console attached to another computer (the User Host)



Network graphics protocol is the forming of input/output information between application program and graphics terminal. Instead of generating device dependent code, logical code is generated which can be understood and converted to suit dependent code, with help of the Server and the User part of protocol.

2. Possible organisations of protocol

Graphics protocol can essentially be at one of three levels (A,B,C - designated on picture), dependent on the division of responsibilities between Server process and User process.^{2,4}

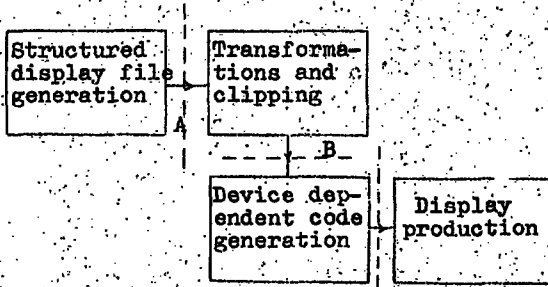


Fig. 2

If the interface between two sites is at point B then we are talking about transformed format protocol (all transformation processing is done at the Server). In these case, the role of the protocol is to build and modify a set of segments of a transformed display file stored in the User host. Segmentation of the display file helps to reduce the amount of information that must be transmitted through the network to affect a change.

Another possibility is to use structured format, which allows a calls to some other parts of the picture. Calls contain information for transformation to apply to the called picture part.

The structured format requires even less network bandwidth for updates than does the transformed format. In addition structuring can save space in satellite memory, when pictures include repeated symbols, such as logic and electric circuit elements. However the protocol proposal^{1/2} does not accommodate this requirement in the case of transformed protocol.

If we introduce the subpicturing in transformed format protocol we progress a bit to structured display file. Some applications do not require often transformations. Sometimes it is enough only to copy elements on different places of the screen.

To understand how to achieve subpicturing in transformed formatted protocol it is necessary to look at the description of the organisation of display file in the satellite (Fig.3)

3. Transformed protocol

In transformed protocol all transformations and generation of independent machine code describing picture segments are done at the Server host. Display file in case of intelligent terminal is maintained at satellite. It has two parts: Main Display file and common subpicture display file. From figure 4 we see that main file contains for every created segment: name, attributes, absolute coordinates for positioning and connection with

subpicture part of segment. Every subpicture part in common subpicture display file contains attributes, vectors, text and calls to other subpictures (segments).

3.1 Segment control

For segment control next commands are used:

```

<seg.open><name>
<seg.close>
<seg.kill><name>
<seg.append><name>
<seg.call><name>
  
```

Segment is created in the following fashion:

The open segment command is sent to the UP, together with a name for the segment. Any following graphical primitives sent to the UP are added to the currently open segment. The creation process is terminated by a "close segment" command.

All attributes (parameters) received immediately after <seg.open> and before <set.position> reset the default values (see table I) in Main file. These parameters could be called also "global" parameters, and can be changed at any time during the session with command <change.attribute>.

All other parameters are inserted in subpicture part of segment, and they are called "local" parameters. They overwrite during execution of display file the global parameters. Local parameters can not be changed during the life time of the segment.

attribute	Code	Value	default values
intensity	2	0-7	3 (visible)
blinking	8	0 or 1	0 OFF
light pen sensitivity	4	0 or 1	0 OFF
rotate picture	64	0 or 1	0 OFF
scale	1	0 - 15	0 (normal)
dash line	16	0 - 3	0 (solid)
off set	32	0 or 1	0 (normal)
name	128	0 - 127	0

TABLE I
attributes
(Concrete values are for VT-15 graphics display - internally)

Command <seg.close> signals the end of generation of the currently open segment.

With <seg.append><name> all subsequent graphical primitives are added to the end of the named segment, which must already exist.

Command <seg.kill><name> deletes the segment from

display file.

Proposed protocol² defines also commands for posting and unposting. These commands include and exclude logically the segment for displaying. Unposting can be implemented as JUMP over segment.

Any change in display image on screen is postponed up to the moment of receiving the command <end.batch of updates>

The flow diagram of segment control is given in figure 5.

After receiving information from Server host, first word of receive buffer is examined for "RS" character. If "RS" is received all subsequent data is supposed to be graphical data. In other cases the whole buffer is interpreted like ordinary terminal message inside predetermined area of the screen.

Generally, the contents of receive buffer can look like on figure 3.

```

<RS>
<seg.open>
<name>
<attributes>
.
<set.position>
<x>
<y>
.
<vectors>
.
<attributes>
<vectors>
<call>
<vectors>
.
<seg.close>

```

Fig. 3

The sequence of information sent from host to create segment

Recognising the command <seg.open> the building subpicture display file is initialised, the name of segment is inserted in Main file and the default values for attributes are set (see Table I). All subsequent attributes, up to receiving command <set position> are inserted in Main file resetting defaults values. After receiving the command for positioning <set.position><x><y> the appropriate code is generated and inserted in Main file. All succeeding attributes, vectors, text, and calls are inserted in subpicture display file, calling appropriate routines for generating display code. Command <seg.close> causes transfer of display code from subpicture building file to common subpicture display file.

3.3 Segment append

Once the segment is created, it is possible at any time to append graphical primitives to segment with command <seg.append><name>. Procedure for appending can be described with steps:

- finde in Main file referenced segment,
- finde the end of subpicture part in common subpicture display file,
- remember JMP return
- replace the JMP command for return with JUMP to free memory location,
- generate all display code in subpicture building file
- transfer contents of subpicture building file to common subpicture display file beginning from first free location,
- insert JMP return command at end
- allow display of appended informations.

3.4 Delete segment

With this command segment is deleted from display file. To delete the segment it is necessary to search for his name in Main file. Before deleting the segment, it must be checked if some other segment is referencing the same segment as subpicture. If no any segment references it, the segment is deleted.

Deletion could be logical and physical. Logical deletion deletes segments with JMP instruction in Main file over deleted segment. Physical deletion is deletion by new ordering in both, the Main file and subpicture file. This process of garbage collection could be done when space in memory starts to be critical.

3.5 Vectors

All received vector data are appended to open segment. Receiving code for vector data, satellite protocol stays in "vectore mode" until it receives information to exit from this mode. The beam is always moved from previous point to new one.

The starting point is set by command <set.position><x><y>. After receiving that command variables IX and IY contain absolute coordinates. Increments of every vector to be drawn are computed as difference :

$$IDX = IXN - IX$$

$$IDY = IYN - IY$$

where IXN, IYN are koordinates of points where ends new vector; IX and IY are coordinates of last beam position. IX and IY receive values of new point when vector is computed.

3.5.1 Format of vector data

byte 1 Hi x The bottom 5 bits of the byte are used to store top

		5 bits of the 11 bit value of x coordinate
byte 2	Lo x	The bottom 6 bits are used to store bottom 6 bits of x.
byte 3	Hi y	used as per Hi x
byte 4	Lo y	used as per Lo x

Bit 6 of byte 3 (Hi y) is used to transmit information about vector visibility. If bit 6 is set, a visible vector is drawn, otherwise an invisible move to x,y is performed.

The first byte Hi x of each group of four bytes is checked by the procedure before it is used to start a new coordinate pair. If bit 6 is found set, the byte is interpreted not as vector but as an instruction to exit vector mod. The next byte is then treated as a new command.

3.6 Call other segments

With this feature it is possible to use subpicturing. Receiving command <seg.call><name>the address of subpicture part of called segment is found and graphic return jump to called segment (subpicture) is generated and inserted in display file for open segment.

3.7 Close segment

This command causes a display of defined segment. The subpicture part of segment is connected to Main file part. Before that the subpicture part from building file must be transferred to common subpicture display file.

4. Implementation

The Protocol is implemented on PDP 15 with refresh graphics terminal and EMAS (Edinburgh Multi Access System) as a server host. EMAS is multiprogramming operating system supported on two ICL 4/75 main frames. The existing PDP 15 has 64 K of memory (unfortunately only 8 K usable by FORTRAN system), disk, DEC tapes, Tape unit and two VT -15 refresh displays. FORTRAN was chosen as programming language for the implementation because:

- possibility of using existing graphical package for VT-15,
- possibility of testing whole system only on PDP - 15,
- portability.

Implemented protocol occupies 4 KW of user site memory and 3 KW at server site. In these figures the sizes of the I/O handlers are not in-

cluded. At the server site (EMAS) the protocol is implemented as extension of graphical package with capabilities and syntax similar to PLOT package for Tektronix storage tube terminal. Majority of protocol functions are implemented on host and satellite so that is easy to configure system to suit application and optimise use of all resources..

It is reasonably to move high interactive parts of application program to satellite. Because all function of protocol are well isolated it is possible to modularise application program satisfactorily. Programming is consistent because protocol is implemented on both machine and moreover in same language.

Through experimentations we made these divisions of labor between Server host and satellite:

- whole system on satellite
- protocol split between two machines
- some applications modular transfered to satellite
- complete protocol at central site (in case of nonintelligent terminal)

The implemented protocol was tested with couple of simple programs. One of them was program for interactive drawing of circuit schematics. Moving highly interactive parts of program to intelligent terminal improvement in response time is obtained, simultaneously releasing server host of trivial services..

Summary

The experimentation with protocol has shown great flexibility in case when some parts of Server protocol are also implemented at user site. Using this approach it is easy to move some interactive parts of program to satellite when some computation and memory power exist locally.

References

1. W.Newman,R.Sproull: An approach to graphic systems design".
2. R.Sproull,E.Thomas: A network graphics protocol.
- 3.V.Dam,G.Stabler,J.Harrington: Intelligent Satellites for Interactive Graphics
- 4.P.Vaswani, D.Rayner: Another comparison of protocol A with proposed protocol of Sproull and Thomas.

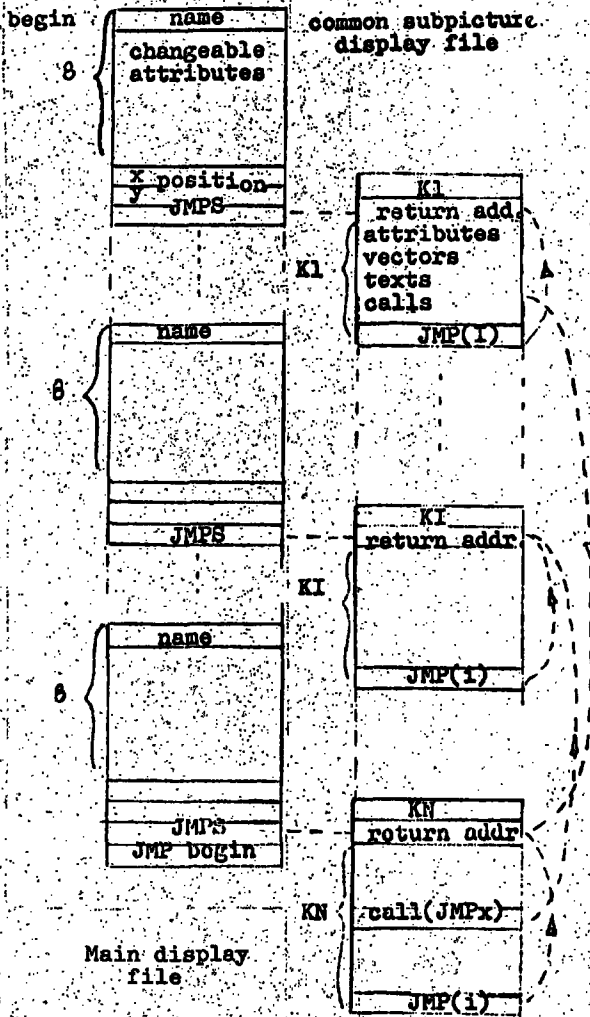


Fig 4
Organisation of display file

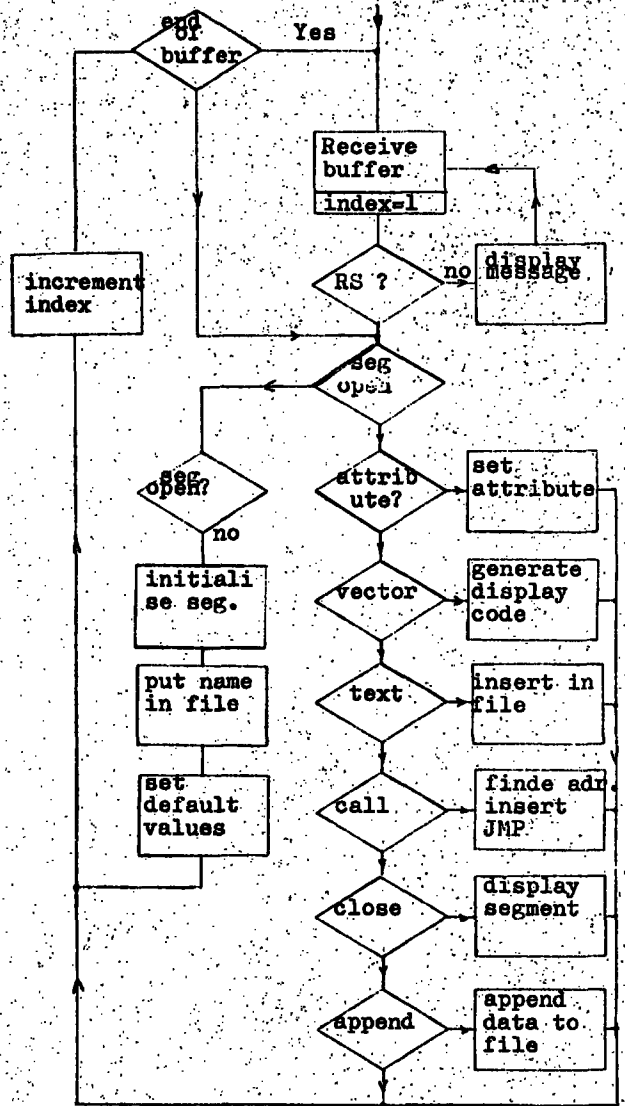


Fig. 5
Details of User part of protocol

primer koncepta računalniškega omrežja

j. kalan

UDK 621.395.74:681.3

INTERTRADE - TOZD Zastopstvo IBM
Ljubljana

Namen tega članka je, da pokaže eno od možnih realizacij računalniškega omrežja. Vsekakor obstoji želja, da se v tako omrežje lahko vključujejo tako obstoječi, kot bodoči terminali in sistemi. Prav ta cilj zasleduje tudi SNA. Opisani so cilji, ki jih zasleduje koncept računalniškega omrežja, nivoji funkcij, fizična in logična slika mreže, vrste vozlišč in povezave med vozlišči, podsistemi omrežja in možna uporaba različnih IBM-ovih računalnikov in terminalov v omrežju.

SYSTEMS NETWORK ARCHITECTURE (SNA) The objective of this article is to show one of the possible solutions of systems network architecture on example of IBM-s SNA. There is a need to include existing as well as future devices, and this is an objective of SNA too. Follows a description of objectives, functional levels, physical and logical view of the network, types of the nodes and links, network subsystems and the possibilities of different IBM machines and terminals.

SNA z ustreznimi produkti za realizacijo je bil v prvotni verziji objavljen leta 1974. Od tedaj je doživel koncept že nekaj dopolnitev. Prvotno je bil to koncept strogo hierarhične mreže z enim samim glavnim ali host računalnikom. Razširitev koncepta, ki je bila objavljena koncem leta 1976 in programi, ki naj bi bili predvidoma na razpolago konec leta 1977, pa omogočajo povezavo večih glavnih računalnikov in s tem oblikovanje univerzalne mreže za obdelavo podatkov.

Cilj vsakega računalniškega omrežja je, da nudi končnim uporabnikom vse potrebne funkcije na najbolj racionalen način. Želimo si predvsem univerzalnost. Vsak terminal naj bi bilo mogoče pooblastiti za dostop do katerekoli aplikacije v omrežju. Že nekaj časa je znano, da so primarnega pomena funkcije, ki jih omrežje nudi na terminalu. Uporaba teh funkcij mora biti za človeka kot končnega uporabnika čim bolj sprejemljiva in stalna, kajti večina delovnih mest je taka, da morajo zanje veljati trdno ustaljeni postopki. Tudi za programe velja, naj ne bodo podvrženi potrebam po čestih spremembah. Z aspekta računalniškega omrežja nam je znano, da so se zaradi hitrega razvoja tehnologije postopki za pridobivanje enakih funkcij na terminalih znotraj sistema bistveno spremenili. Predvsem smo v zadnjem času prešli iz prvotno uveljavljene popolnoma centralizirane obdelave zaradi novega razmerja med cenami komponent omrežja na distribuirano obdelavo. Če naj to ne vpliva na delo končnega uporabnika je potrebno točno definirati nivo končnega uporabnika. Nivo končnega uporabnika, tako imenovani aplikacijski nivo funkcij, moramo strogo ločiti od ostalih funkcij računalniškega omrežja. Na nastali meji moramo definirati standardni vmesnik ali interfejs. Tudi preostale funkcije računalniškega omrežja lahko dalje delimo na nivoje.

Danes razpolagamo z razmeroma velikimi možnostmi obdelave podatkov s pomočjo mikroprocesorjev, še vedno pa smo precej omejeni pri prenosu podatkov. Ker tudi na tem področju lahko pričakujemo v bodočnosti

določen razvoj, na primer optične povezave preko satelitov, je umestno, da se na to pripravimo, zato preostale funkcije omrežja razdelimo vsaj še na tako imenovani funkcijski nivo ter na prenosni nivo.

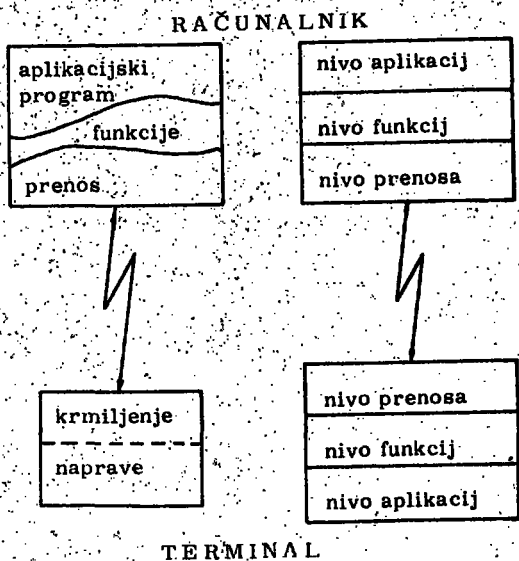
V RAČUNALNIŠKEM OMREŽJU ZASTOPANE FUNKCIJE

Osnovni cilj koncepta SNA je standardizacija arhitekture računalniškega omrežja. Standardizacija mora upoštevati tako današnje stanje kot bodoče zahteve, katere bomo lahko postavljali računalniškemu omrežju. SNA skuša omogočiti čim bolj fleksibilen razvoj računalniških omrežij s tem, da jasno ločuje odgovornosti treh osnovnih nivojev funkcije in standardizira povezave med tremi nivoji. Ti nivoji so:

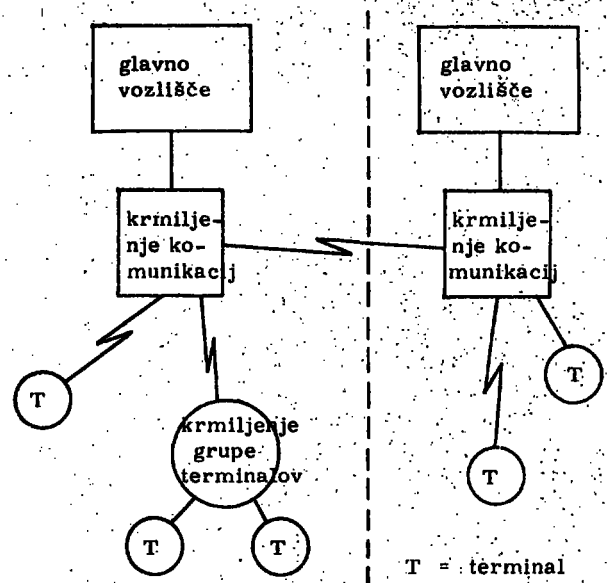
- nivo aplikacij
- nivo funkcij
- nivo prenosa

SNA te nivoje ločuje v vsakem vozlišču mreže. Realizirani so v potrebnem hardware-u in software-u. V omrežju so med dvema končnima uporabnikoma ti nivoji porazdeljeni simetrično. Slika 1 prikazuje primer povezave računalnika in terminala po stari ad hoc rešitvi in rešitev po konceptu SNA. Pred uvedbo SNA standardov so bile odgovornosti za krmiljenje prenosa in terminala često pomešane z obdelavo in pripravo podatkov. SNA zahteva strogo ločitev teh funkcij v vsakem vozlišču v diskretne nivoje. SNA določa način sodelovanja vsakega nivoja z odgovarjajočim nivojem v drugem vozlišču. To omogoča delitev odgovornosti med tema dvema nivojema. Po potrebi eden od njiju lahko prevzame večji del funkcij in s tem razbremeni drugega. Ker sta oba nivoja strogo ločeni enoti, ki na določen način komunicirata med seboj, se ta delitev ne odraža na ostalih nivojih. Tako se lahko posamezen nivo razmeroma enostavno modificira.

Nivo aplikacij je tisti del sistema, ki obdeluje le podatke, ki zanimajo oba končna uporabnika, ko je med njima že vzpostavljena povezava, ki jo imenujemo pogovor.



Slika 1 Delitev funkcij pred in v SNA



Slika 2 Vrste vozlišč v omrežju

Nivo funkcij je tisti nivo, ki končnemu uporabniku omogoča sprejemanje teh podatkov v čim preglednejši obliki, ali pa mu pomaga te podatke pripravljati s tem, da mu nudi orodje ali oporo pri delu, da ga opozarja na formalne napake, ter da vodi zapisnik opravljenega pogovora.

Nivo prenosa skrbi za to, da se podatki brez napak prenesejo na pravi naslov in tam predajo nivoju funkcij.

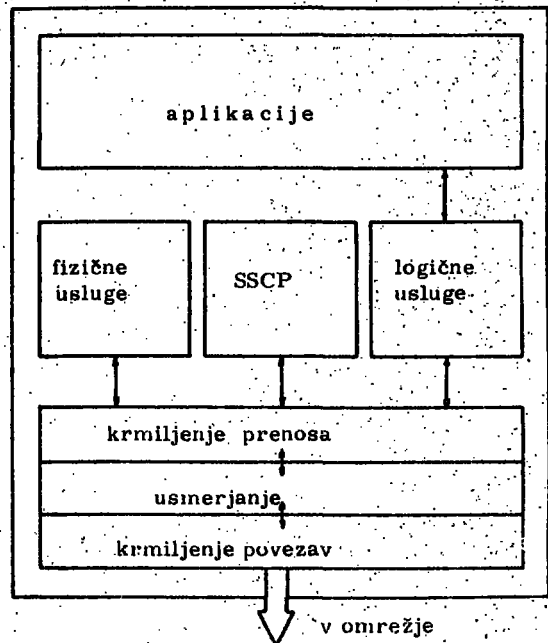
VOZLIŠČA IN POVEZAVE V SNA

Prednosti nivojske zgradbe omrežja se pokažejo posebno takrat, ko pristopimo k načrtovanju omrežja. Omrežje je fizično le skupek vozlišč, med seboj povezanih s povezavami. Vendar ni potrebno, da bi bila vsa vozlišča ali povezave popolnoma enake. Posebno ni potrebno, da bi vsako vozlišče obsegalo celotni spekter možnosti, ki jih obsega SNA. Prvotno je bila v sistemih za daljinsko obdelavo podatkov uporabljena le povezava med glavnim računalnikom in terminalom, torej le med dvema tipoma vozlišč. SNA definira štiri tipe vozlišč (slika 2):

- glavno vozlišče (host node).
- vozlišče za krmiljenje komunikacij (communications controller node)
- krmilno vozlišče za grupo terminalov (cluster controller node)
- vozlišče v terminalu (terminal node).

Glavno vozlišče

SNA zahteva v vsakem računalniškem omrežju vsaj eno glavno vozlišče. To navadno zadovoljuje glavne potrebe omrežja po podatkih in večjih obdelavah, predvsem pa krmili pripadajoče področje (domain) omrežja. Celotno omrežje delimo po SNA v toliko področij, kolikor imamo v omrežju glavnih vozlišč. Glavno vozlišče krmili svoje področje omrežja in skrbi za povezavo vozlišč svojega področja z vozlišči ostalih področij (slika 3).



Slika 3 Nivojska zgradba glavnega vozlišča

Na aplikacijskem nivoju nudi glavno vozlišče predvsem bogato izbiro podatkov in obdelav za potrebe svojega področja, kot tudi za potrebe ostalih področij, v kolikor je slednjim dovoljen pristop do posameznega podatka ali obdelave. To doseže s tem, da izkorišča vse možnosti svojega ozadja, ki ga predstavljajo: centralna enota s procesorji in hitrim pomnilnikom, perifernimi pomnilniki ter ostale naprave. Tu so locirani programi za proizvodnjo v bankah podatkov, aplikacijski programi za redno proizvodnjo, statistični programi, programi za interaktivno reševanje problemov, med katere lahko štejemo tudi jezike za interaktivno programiranje in računanje.

Tu se nahaja operacijski sistem, ki omogoča učinkovito izkoriščanje vseh ostalih možnosti. Operacijski sistem ter programi za DB/DC (Data Base/Data Communications) to je za banke podatkov in za komuniciranje omogočajo uporabo istih resursov omrežja s strani večih uporabnikov. Ti resursi so procseorji, realni in virtualni pomnilnik, eksterni pomnilnik, kanali za prenos podatkov, krmilne enote za komunikacije, krmilne enote terminalov in posamezni terminali. Program za komuniciranje krmili in kontrolira delovanje omrežja. Tu so na funkciji jskem nivoju zbrane vse niti, ki so potrebne, da dosežemo pravilno in nemotežno komuniciranje. Te se stekajo v krmilni in kontrolni točki SSCP (System Services Control Point). SSCP se nahaja le v glavnih vozliščih. Njene naloge so fizičnega in logičnega značaja. Fizično aktivira potrebna vozlišča in povezave, logično pa poveže dva končna uporabnika, da se lahko pogovarjata. SSCP vzpostavlja, vzdržuje in zaključuje logično povezavo med dvema končnima uporabnika, ki jo imenujemo pogovor. Ob vzpostavljanju pogovora SSCP preverja, ali sta uporabnika pravilno registrirana in ali se sploh smeta med seboj pogovarjati. Šele če je zadovoljeno vsem pogojem, dovoli pogovor, ki potem poteka strogo ločeno od vseh ostalih pogovorov. Med pogovorom SSCP meri število vprašanj in odgovorov ter število prenešenih znakov, ne vmeša pa se v vsebino pogovora.

Druga naslovna točka v glavnem vozlišču je fizična enota. Preko te se inicializirajo in aktivirajo posamezna vozlišča omrežja, obravnavajo se napake in vzdržuje fizična konfiguracija omrežja. Kot tretji tip naslovov v glavnem vozlišču nastopajo logične enote. Te predstavljajo vrata za vstop končnega uporabnika v mrežo. V glavnem vozlišču so končni uporabniki aplikacijski programi. Vsaka logična enota nudi uporabniku potrebne funkcije. Te so lahko logičnega značaja, kamor spada vzpostavitev pogovora, ali pa so to usluge predstavitve in kontrole podatkov.

Transmisijski nivo je v glavnem vozlišču zastopan z vsemi podnivoji. Ti so krmiljenje prenosa, usmerjanje in krmiljenje povezave.

Krmiljenje prenosa skrbi za pravičen prenos vprašanj in odgovorov med dvema končnima uporabnikoma. Druga naloga je nadziranje pogovora, kjer skrbi za vzpostavitev, prekinitve in obnovo pogovora v primeru morebitne slučajne prekinitve. Ne opravlja pa glavno vozlišče v SNA krmiljenja komunikacij, za katero, kot bomo videli skrbi specializirano vozlišče. V usmerjanje spada predvsem vzpostavitev adresibilnosti med glavnimi in ostalimi vozlišči. Usmerjanje lahko zahteva pogovor s krmilno enoto terminala, da se identificirata adresi obeh končnih uporabnikov, med katerima se vzpostavlja pogovor.

Specifične funkcije glavnega vozlišča so torej sledeče:

- krmiljenje celotnega omrežja ali področja
- krmiljenje glavnega vozlišča
- vzpostavljanje in krmiljenje pogovorov med dvema končnima uporabnikoma
- predelava podatkov za potrebe komuniciranja med dvema končnima uporabnikoma
- prenos podatkov preko prenosnega podsistema.

Vozlišče za krmiljenje komunikacij

Vozlišče za krmiljenje komunikacij le povezuje ostale vrste vozlišč. Nivo prenosa v tej vrsti vozlišč ima predvsem nalogo, da razbremeni glavno vozlišče za krmiljenje komunikacijskih povezav. S tem da deluje kot relejna postaja, skrbi to vozlišče predvsem za celovitost omrežja. Njegova glavna naloga je, da ščiti ostala vozlišča pred kompleksnostjo krmiljenja komunikacijskih povezav. To vozlišče ima sposobnost ponavljanja sporočil v primeru potrebe. Prav tako prilagaja sporočila sposobnosti sosednjih vozlišč, s tem, da uravnava dolžino in hitrost pošiljanja sporočil. Če je namreč sporočilo za sosednje vozlišče predolgo, ga razbije v več fizičnih stavkov. Krmilna enota za komuniciranje s temi funkcijami povezuje glavno vozlišče, ostala vozlišča tega tipa, krmilne enote terminalov ter posamezne terminale v enotno omrežje. Med njegove naloge spada še aktiviranje in deaktiviranje povezav, polnjenje oddaljenih krmilnih enot za komuniciranje s programi in prenos vsebine njih pomnilnika v centralni računalnik, poročanje o aktivnosti povezav in testiranje povezav.

Primer tega tipa vozlišč je IBM-ova krmilna enota za komuniciranje, opremljena s programom NCP/VS ali ACF/NCP/VS. Poudariti je treba, da se v primeru zamreženja, to je povezave večih glavnih računalnikov, posamezna področja vežejo med seboj preko tega tipa vozlišč. Vzpostavljanje pogovorov med področji je možno le pod kontrolo glavnega, ko je pogovor vzpostavljen pa nadaljne komuniciranje lahko poteka le preko krmiljen pa nadaljne komuniciranje lahko poteka le preko krmilne enote za komuniciranje. Ta vrsta vozlišč je lahko v omrežju priključena na glavno vozlišče, ali preko povezave na drugo vozlišče iste vrste. Tudi oddaljena vozlišča se polnijo s programom iz glavnega računalnika. To omogoča spreminjanje konfiguracije mreže; krmilno enoto za komuniciranje priključeno lokalno na glavni računalnik lahko po potrebi dodelimo kot oddaljeno drugemu glavnemu računalniku.

Druga v bistvu le začasna vloga vozlišča za krmiljenje komunikacij je, da deluje kot priključna točka. NCP je modularni program, ki omogoča prilagoditev priključnih funkcij različnih resursov omrežja, ki se direktno vežejo na krmilno enoto za komuniciranje. Poleg standardne možnosti priključevanja terminalov grajenih v skladu z SNA, je nanj mogoče priključiti skoraj vse vrste obstoječih terminalov ali računalnikov, da le odgovarjajo eni od IBM-ovih ali TTY specifikacij terminalskega protokola. Obratovanje terminalov iz časa pred uvedbo SNA je seveda potrjeno specifičnim zahtevam, ki jih še vedno pokriva glavni računalnik s priključeno krmilno enoto za komuniciranje, čeprav ne odgovarjajo SNA standardom. V tej vrsti vozlišč običajno aplikacijski nivo ne nastopa. Glavni povdarek je na funkcijah, ki spadajo v prenosni nivo. V dosledno izvedenem SNA konceptu bi morale biti priključne funkcije, ki spadajo v nivo funkcij, locirane v krmilnem vozlišču za grupo terminalov ali v samem terminalu.

Krmilno vozlišče za grupo terminalov

Krmilno vozlišče za grupo terminalov je orientirano predvsem na funkcijski nivo in v okviru tega na operaterja. Nekatere krmilne enote je mogoče programirati. V glavnem so del specializiranega sistema vhodno izhodnih enot. Vendar je na tem mestu lahko tudi računalnik

splošnega značaja. Naprav, ki se priključujejo na krmilno enoto navadno ni mogoče naslavljati iz poljubnega vozlišča omrežja, so le del podsistema. Ta vrsta vozlišč se lahko veže tudi direktno na glavno vozlišče, vendar aplikacijski program v tem primeru komunicira z njimi na enaka način, kot z oddaljenimi vozlišči. Logika te vrste vozlišč je lahko sestavljena iz hardware-a in software-a. Pri izvedbah, ki nimajo možnosti programiranja, se poslužuje operater za komuniciranje z aplikacijskim programom v glavnem računalniku le razpoložljivega hardware-a. Pri krmilnih enotah z možnostjo programiranja lahko program izdelava uporabnik sam, ali pa se poslužuje izdelanih programov. Krmilna enota grupa terminalov omogoča izvajanje večih različnih aplikacij istočasno. Programi za vsako aplikacijo naj bi bil izdelan s ciljem, da se čim bolj izkoristi možnosti lokalne obdelave in kapacitete prenosnih linij. Terminal lahko uporablja lokalne programe in podatke za reševanje večjega dela običajnih nalog in le tiste obdelave, ki lokalno niso izvedljive, se prenesejo v obdelavo v glavni računalnik. Krmilna enota za grupo terminalov je za posamezen terminal lahko transparentna, to pomeni, da v primeru, ko v krmilni enoti ni ustreznega programa avtomatsko preda nalogo glavnemu računalniku. Taka je naprimer krmilna enota IBM 3791 s terminali 3277.

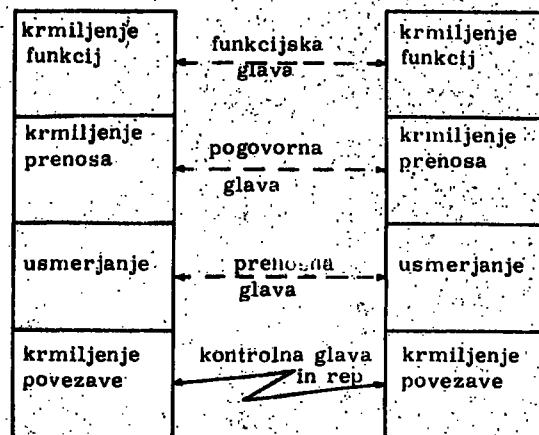
Vozlišče v terminalu

Vozlišče v terminalu predstavlja najnižji nivo v hierarhiji računalniškega omrežja. Največkrat opravlja le funkcijo zajema ali predajanja podatkov. Vseeno morajo biti tudi v takem vozlišču zastopani vsi nivoji funkcij. Rešitve so lahko samo hardware-ske. Aplikacijski nivo je lahko zastopan le z delovanjem operaterja. Lahko pa se na tem nivoju priključi v omrežje komplet računalnik s svojo na aplikacijski nivo usmerjeno periferijo in le z najnujnejšimi funkcijami za povezavo z omrežjem.

Povezave med vozlišči

SNA predvideva en sam standardni protokol za prenos podatkov med sosednjima vozliščema. To je SDLC (Synchronous Data Link Control) protokol, ki je bil razvit za namenom, da izboljša prenos podatkov na katerikoli tipu fizične povezave. Naloga SDLC protokola je torej, da nam tudi preko manj zanesljive fizične povezave omogoča zanesljiv prenos podatkov in s tem zanesljivo logično povezavo. SDLC skrbi za integriteto podatkov, ki se prenašajo iz enega v drugo vozlišče, tako v point to point, kot tudi v multipoint konfiguraciji ali v prenosni zanki.

Funkcijski nivoji v vozliščih komunicirajo z odgovarjajočimi funkcijskimi nivoji v drugih vozliščih preko podatkov, ki jih dodajajo osnovnemu sporočilu v obliki glave ali repa (slika 4). V glavah so definirana polja v kateri se vpisujejo krmilne informacije. Vsak nivo funkcij uporablja svojo glavo. Rep se uporablja le na nivoju krmiljenja povezav. Rep po eni strani razmejuje posamezne pakete podatkov, po drugi pa vsebuje polje s podatkom za verificiranje pravilnosti prenosa. SDLC zahteva obdelavo glave in repa v vsakem vozlišču, skozi katerega paket potuje. Vsako vozlišče loči od preostalih podatkov glavo in rep za krmiljenje povezave in na osnovi teh podatkov preveri pravilnost prenosa. Če ni skladnosti, zahteva ponovitev prenosa. Za pot do naslednjega vozlišča nato doda novo glavo in rep.



Slika 4. Komuniciranje med nivoji

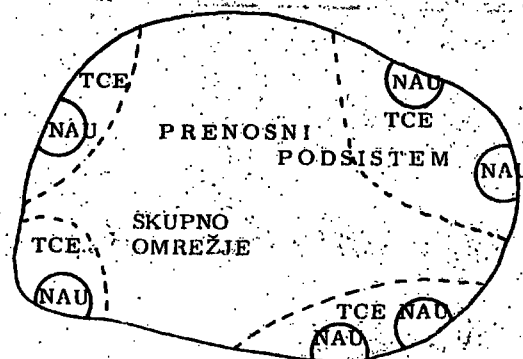
V glavi za krmiljenje prenosa se nahajajo podatki o tem, kdo pošilja, kdo je prejemnik, zaporedna številka vprašanja oziroma odgovora v okviru pogovora, prioriteta prenosa, ali je paket prvi, zadnji, srednji ali edini ter včasih še dolžina podatkov.

V glavi za krmiljenje pogovora mora biti podatek ali je sporočilo vprašanje ali odgovor, kateremu nivoju je podatek namenjen, ali je pričakovani odgovor in ostali podatki za krmiljenje pogovora.

V vsaki komponenti vozlišča se obdeluje le ta komponenti namenjen podatek iz odgovarjajoče glave. Prav to neodvisno obdelovanje glav in polj v teh glavah omogoča delitev funkcij v SNA v strogo ločene nivoje.

PODSISTEMI OMREŽJA

Na računalniško omrežje gledamo lahko na dva načina. Spoznali smo fizikalno gledanje na omrežje sestavljeno iz različnih vrst vozlišč, med seboj povezanih s povezavami. Drug način gledanja je logičen. Slika 5 predstavlja logično zgradbo omrežja. To tvorijo prenosni podsistem in okrog njega razporejene naslovljive enote (NAU - Network Addressable Units). Naslovljiva enota je logičen pojem, to so vrata, skozi katera imajo končni uporabniki dostop do omrežja, ki jim omogoča komuniciranje. Če želita dva končna uporabnika izmenjavati podatke se mora med njima vzpostaviti pogovor. Ta je običajno sestavljen iz večih vprašanj in odgovorov, ki si jih izmenjujeta udeleženca. Pogovor je začasna logična povezava med dvema naslovljivima enotama.



Slika 5. Logična slika omrežja

Nekatere naslovljive enote vodijo več pogovorov hkrati, vednar je vsak od teh pogovorov od ostalih neodvisen in izoliran.

Videli smo, da so vsa vozlišča zgrajena po istem načelu. Zato je odgovorjajoče nivoje funkcij v različnih vozliščih mogoče združiti v podsisteme. Te si lahko predstavljamo kot lupino, ki ovijajo posamezne nivoje funkcij. Definiramo lahko:

- komunikacijski sistem
- prenosni podsistem in
- skupno omrežje

Komunikacijski sistem tvorijo funkcije funkcijskega in prenosnega nivoja. Končni uporabnik se pri pogovorih poslužuje uslug komunikacijskega sistema, do katerega ima dostop preko naslovljivih enot.

Prenosni podsistem krmili pogovore in ureja promet v omrežje in iz njega. Predstavljajo ga elementi za krmiljenje prenosa (TCE - Transmission Control Element), ki so razporejeni okrog skupnega omrežja.

Skupno omrežje je podsistem, ki si ga lahko predstavljamo kot jedro celotnega sistema, sestavljata ga pravzaprav dve plasti, usmerjanje in krmiljenje povezav. Z logičnega stališča lahko gledamo na omrežje kot na črno škatlo, ki omogoča priključevanje naslovljivih enot. Fizikalno pa je to, kot smo videli, več med seboj povezanih vozlišč. Da se prenese informacija od enega naslovnika do drugega, je treba najti pravo pot preko vozlišč in povezav. Usmerjanje se odvija v vsakem vozlišču na poti. Za vsak paket podatkov, najsi pride od zunaj ali iz istega vozlišča, usmerjanje določi, v katero vozlišče ga mora poslati, da se približa cilju. Nato ga preda krmiljenju prenosa, ki se prav tako nahaja v vsakem vozlišču in skrbi za pravi prenos do naslednjega vozlišča.

MESTO RAČUNALNIKOV V RAČUNALNIŠKEM OMREŽJU

Obstoječi IBM-ovi računalniki in terminali, kot tudi računalniki in terminali ostalih proizvajalcev se lahko vključujejo v omrežje če so ustrezno opremljeni za komuniciranje in če v omrežju obstoje potrebne funkcije za priključitev. SNA je razmeroma nov koncept, zato je v uporabi še mnogo računalnikov in terminalov, ki temu konceptu niso prilagojeni, vseeno pa je normalno, da se tudi ti vključujejo v omrežje. Kot smo spoznali, razlikuje SNA štiri vrste vozlišč. Od načrtovalca omrežja je odvisno, na katero mesto bo postavil določen računalnik in kako bo realiziral priključne funkcije.

V glavnem vozlišču v SNA lahko deluje le računalnik tipa IBM/370 z enim od operacijskih sistemov za to serijo, ki uporablja virtualni pomnilnik. Da deluje uspešno, mora imeti ta računalnik dovolj velik pomnilnik. V tem mora biti prostor za operacijski sistem, komunikacijski program VTAM (Virtual Telecommunications Access Method) in uporabnikove aplikacije.

V lokalnem vozlišču za krmiljenje komunikacij je potrebna enota IBM 3705. V SNA omrežju deluje ta enota po programu NCP (Network Control Program). Ta program se pripravi na glavnem računalniku na osnovi istih podatkov kot VTAM. Glavni računalnik lahko na eksternem pomnilniku hrani za 3705 po več programov, kar omogoča delovanje v več režimih. IBM 3705 ima le hiter pomnilnik velikosti do 256 KByte. Nanj je mogoče priključiti do cca 350 linij. V SNA ima ta krmilna enota ostro omejen obseg delovanja. Vršni usmerjanje in krmili prenos podatkov po linijah do sosednjih vozlišč.

Nanjo je mogoče priključiti skoraj vse vrste linij in sisteme prenosa, asinhronski, sinhronski in SDLC. Na njo je možno priključiti skoraj vse vrste terminalov, ki so se pojavili v doslej razmeroma kratki dobi daljinske obdelave podatkov. Poudariti je potrebno, da ta enota predstavlja le del celovitega kompleksa za krmiljenje omrežja, katerega sestavni deli so še glavni računalnik, operacijski sistem ter programa VTAM in NCP. Ta enota s programom NCP le ščiti glavni računalnik pred kompleksnostjo krmiljenja povezav.

V oddaljenem vozlišču za krmiljenje komunikacij nastopa zopet enota 3705 ali 3704 s programom NCP, vendar je na tem mestu mogoče uporabiti tudi kak drug računalnik, za katerega pa je potrebno napisati program, ki naj zadovolji potrebnim funkcijam. Od IBM-ovih sistemov sta za to nalogo primerna Sistem/7 in Serija 1.

V vozlišču za krmiljenje grupe terminalov lahko nastopa katerikoli računalnik, na katerega je vezanih eden ali več terminalov. Vendar razlikujemo nekaj stopenj povezanosti z omrežjem. Lahko imamo čvrsto povezavo, ki omogoča s pomočjo standardnega hardware-a in software-a izkoriščanje vseh možnosti, predvidenih v SNA, ali pa imamo le ohlapno povezavo, ki se izraža v omejenih možnostih adresiranja in s tem manjši univerzalnosti terminalov.

Med čvrsto povezane sisteme lahko štejemo realizacije omrežja z uporabo IBM-ovih terminalskih sistemov 3270 in 3790. Ta dva se programira v glavnem le na glavnem računalniku in sta za uporabnika popolnoma zaprta sistema. Krmilna enota za grupo terminalov pri sistemu 3270 predstavlja popolnoma hardware-sko rešitev brez možnosti programiranja, pri 3790 pa obstoji možnost uporabe lokalnih obdelav in lokalnih datotek v krmilni enoti 3791. Za uporabo teh možnosti mora v enoti 3791 obstajati program, če pa tega programa ni, se sistem 3790 obnaša enako kot sistem 3270, pravimo, da je enota 3791 za priključene terminale transparentna. Glavni računalnik ima pregled nad vsebino datotek in stanjem programov v enoti 3790, saj se kopije datotek in programov nahajajo kot varnostna rezerva v glavnem računalniku. Taka rešitev predstavlja dosledno izvedbo mreže po načelih SNA. V njej se končnemu uporabniku na terminalu nudijo vse v omrežju razpoložljive funkcije.

Ostali računalniki omogočajo v glavnem le rahlo vezane sisteme. Od programa v računalniku, ki je postavljen v vlogo krmilne enote za grupo terminalov je odvisno, kakšne možnosti imajo terminali. Rešitve na splošno niso popolnoma standardizirane. Vzrokov za to je več. Ali je to razpoložljiv starejši hardware in software, ali pa manjše zahteve podsistema, ki jih lahko popolnoma zadovoljivo rešimo, vsaj v današnjem prehodnem obdobju, z manjšo univerzalnostjo. Vendar tudi v teh primerih skušamo zasledovati koncept SNA, ker si s tem olajšamo kasnejše vzdrževanje sistema. Od IBM-ovih računalnikov pridejo v poštev za take rešitve Serija 1, sistem/34, sistem/3, pa tudi sistemi/360 in manjši sistemi/370.

V vozliščih najnižjega reda lahko nastopajo raznovrstni terminali in tudi raznovrstnimi računalniki, v kolikor so ti brez priključenih terminalov. Terminali so po svojih funkcijskih sposobnostih zelo različni. Na funkcijskem nivoju SNA trenutno razlikuje pet profilov terminalov, prav tako razlikuje pet profilov pogovorov. Razlike med terminali so v izboru funkcij, načinu krmiljenja terminala, načinu prenosa podatkov, velikosti bufferjev itd. Obstoje terminali, ki jih je mogoče enostavno prekopirati iz enega načina dela v drugega. Terminali so definirani s terminalskim protokolom, ti pa so

največkrat označeni po tipičnih predstavnikih iz proizvodnega programa IBM-a ali kakšnega drugega proizvajalca. Kot primere naj navedemo IBM 3270 kot predstavnika s funkcijami bogatega multipoint terminala, IBM System/3 ali Serije 1 kot predstavnika sinhronnega point to point prenosa, IBM 2740 kot predstavnika asinhronskega terminala in Teletype 33 kot predstavnika ameriških teleprinterjev, katerega način prenosa je osvojilo mnogo proizvajalcev terminalov. K tem bi bilo potrebno dodati še evropski teleprinter. Tako smo dobili že zbirko različnih terminalov. V realnih mrežah moramo računati z možnostjo nastopanja vseh teh tipov terminalov. Po novih standardih lahko delujejo le terminali z novimi možnostmi obdelave podatkov, oziroma krmiljenja povezav s pomočjo mikroprocesorjev. Težiti moramo k temu, da v bodoče vzpostavljamo le sodobne povezave, ki odgovarjajo SNA, vendar bodo poleg teh najbrže še nekaj časa obstajali terminali in povezave z omejenimi možnostmi. O telefonskih linijah lahko rečemo, da so še vedno zelo drag način za množični prenos podatkov. V mnogih primerih bo še vedno zanimiv "off line" prenos podatkov z disketo ali magnetnim trakom. Smiselno bi bilo, da se podatki, ki jih človek kot uporabnik terminala takoj prebere in usvoji prenašajo preko linije, kjer pa se podatki pišejo za kasnejšo uporabo ali za razdeljevanje, bi bilo še vedno mnogokrat umestno razmisliti o "off line" prenosu. Kljub "off line" prenosu pa se lahko poslužujemo "on line" potrditve prejema in pravilnosti podatkov po analognih postopkih, kot veljajo za podatke prenešene po liniji.

ZAKLJUČEK

SNA sam po sebi je le koncept, ki definira potrebne funk-

cije računalniškega omrežja. Ta koncept predvideva porazdelitev funkcij na večje število komponent omrežja in skupno uporabo razpoložljivih zmogljivosti omrežja s strani večih uporabnikov. Aplikacijske programe je po tem konceptu mogoče razvijati tako, da so neodvisni od tipa uporabljenega terminala. Iste terminale je mogoče uporabljati za več različnih aplikacij. Omrežje se lahko nemoteno razvija od majhnega do velikega, od enega do večih glavnih računalnikov in se lahko tudi dinamično modificira med delom. Omogočeno je zasledovanje obremenitev, testiranje komponent in obračun.

Implementacija takega sistema zahteva izdelane standarde. Možno mora biti tudi vključevanje obstoječih obdelovalnih in prenosnih zmogljivosti. V članku je opisano sedanje stanje implementacije s strani IBM-a, vendar je to področje, ki še vedno dinamično razvija, predvsem v smeri čim večje univerzalnosti in uveljavitve splošno veljavnih standardov.

LITERATURA

1. Introduction to Advanced Communications Function, Multiple System Data Communication Networks IBM form nr. GC30-3033
2. System Network Architecture General Information IBM form nr. GA27-3102
3. J.H. McFadyen : Systems Network Architecture: An overview. IBM System Journal 15, No 1
4. P.G. Cullum: The Transmission Subsystem in SNA IBM System Journal 15, No 1
5. H.R. Albrecht and K.D. Ryder: The Virtual Telecommunications Access Method: A SNA perspective. IBM System Journal 15, No 1

pouzdanost prenosa podataka izmedju magnetnih medijuma i crc-kontrola

d. glušac

UDK 681.327.63:621.3.019.3

ELEKTRONSKI RAČUNSKI CENTAR
BEOGRADSKÉ BANKE BEOGRAD

Najbolji način povećanja pouzdanosti prenosa podataka izmedju magnetnih medijuma je da se koristi CRC-metoda. Osnovna ideja autora je da minimum matematičke pozadine pruži čitaocu, dovoljne da vodi direktno u praktični uvid.

RELIABILITY OF TRANSFER DATA BETWEEN MAGNETIC MEDIUMS AND CRC-CONTROL - The best way to improve reliability of transfer data between magnetic mediums is by using CRC-method. The basic ideal of author is to give the reader minimal mathematical background, to lead directly to practical insight.

POTREBA ZA CRC-KONTRLOM

Svi sistemi za obradu podataka zahtevaju veliki stepen pouzdanosti prenosa podataka uopšte, a posebno u saobraćaju sa magnetnim medijumima (ovde će biti reči o magnetnim diskovima i kasetama). Jedan od veoma moćnih instrumenata u ostvarjenju te pouzdanosti je CRC-kontrola.

Imajući u vidu učestalost grešaka kod pojedinih delova računskog sistema, na primer: centralni procesor 10^{-14} , periferni uređaji 10^{-9} , komunikacioni kanali 10^{-5} i operateri 10^{-2} , dolazimo do potrebe da se nivo pouzdanosti prenosa preko komunikacionog kanala poveća, što ostvarujemo CRC metodom. Podaci su dobijeni na seminaru "Intelligent terminals" održan 13. - 15. maja 1977. godine u Ljubljani, u organizaciji Elektro tehničkog fakulteta. U mnogim se novim i modernim računskim sistemima sve više ima mesta za CRC-kontrolu, a sudeći po literaturi koju sam ja koristio prvi ju je lansirao u svet računске tehnike IBM u svom sistemu IBM-2400, u realizaciji magnetnih traka (IV).

UVODNENJE POLINOMA

Delenje jednog polinoma $D(x)$ (polinom podataka) sa drugim $G(x)$ (generator polinom) daje rezultat $Q(x)$ (polinom rezultata) i obično $R(x)$ (polinom ostatka). Ako rezultat ima ostatak:

$$\frac{D(x)}{G(x)} = Q(x) + \frac{R(x)}{G(x)} \dots \dots \dots (2)$$

pomnožimo izraz (2) sa $G(x)$:

$$D(x) = Q(x) \cdot G(x) + R(x) \dots \dots \dots (3)$$

U primeni ove ideje na magnetne trake, "polinom podataka" $D(x)$ bi predstavljao niz bitova podataka na čijem kraju se doda "polinom ostatka" $R(x)$ ili CRC bajt. Tako, CRC bajt čini "polinom podataka" deljivim bez ostatka sa "polinom generatorom".

KAKO DA PODELIMO DVA POLINOMA

Ovom problemu ne prilazimo sa klasičnog aspekta deljenja polinoma polinomom, jer je on poznat u klasičnoj matematici, već postavljamo pitanje deljenja polinoma u moduo-2 binarnoj aritmetici. Osnovna razlika ova dva načina su u tome što se uvek sabira u moduo-2 aritmetici, te dolazimo do sledećeg postupka datog u narednom primeru:

$$\text{U izrazu } \frac{D(x)}{G(x)} = Q(x) + R(x)$$

dajmo za $D(x) = x^8 + x^5 + x^2 + 1$ a za $G(x) = x^5 + x^3 + x + 1$, a pronadjimo ceo izraz deljenja $Q(x)$ i ostatak $R(x)$?

$$\begin{array}{r} x^3 + x + 1 \\ x^5 + x^3 + x + 1 \overline{) x^8 + x^5 + x^2 + 1} \\ \underline{x^8 + x^6 + x^4 + x^3} \\ x^6 + x^5 + x^4 + x^3 + x^2 \\ \underline{x^6 + x^5 + x^4 + x^3} \\ x^2 + x \\ \underline{x^2 + x} \\ 0 \end{array}$$

dobili smo $Q(x) = x^3 + x + 1$ a $R(x) = 0$
U moduo-2 aritmetici se ne javlja pojam prenosa.

DOBIJANJE POLINOMA ZA PRENOS

U izrazu (3) oduzmimo $R(x)$ od obe strane jednačine i dobijamo sledeći izraz:

$$D(x) - R(x) = Q(x) \cdot G(x) \dots \dots \dots (4)$$

a polazeći od već nam poznate tvrdnje da su sabiranje i deljenje iste operacije izraz (4) možemo napisati u obliku:

$$C(x) = D(x) + R(x) = Q(x) \cdot G(x) \dots \dots (5)$$

U izrazu (5) proizved. je jasno deljiv sa $G(x)$, jer jedan od činioca, a znači i leva strana izraza. Ako ovaj polinom $C(x)$ primamo sa druge strane nekog prenosnog puta mi ga ponovo delimo sa $G(x)$. Ako je posle ovog deljenja ostatak jednak 0,

čvrsta je pretpostavka da smo primili "Polinom podataka" $D(x)$ sa nepromenjenim (bez greške) bitovima. Naravno, moguće je da se pojavi takva greška da je primljeni polinom ipak deljiv sa $G(x)$, te je ova greška neuhvatljiva, ali joj je mogućnost pojavljivanja veoma mala.

Bilo koji poremećaj u primljenom polinomu može biti predstavljen kao "polinom greške" $E(x)$. Teoretski gledano, kada bi $E(x)$ oduzeli, ili što je ovde isto, sabrali, od primljenog polinoma $Z(x)$ rezultat bi bio suma korektnih podataka i polinoma ostatka, tj. uzimamo da su $T(x)$ primljeni podaci i ostatak:

$$T(x) = Z(x) - E(x) = Z(x) + E(x)$$

Jasno je dakle, bilo koji pokušaj korigovanja greške zahteva da se nadje polinom greške.

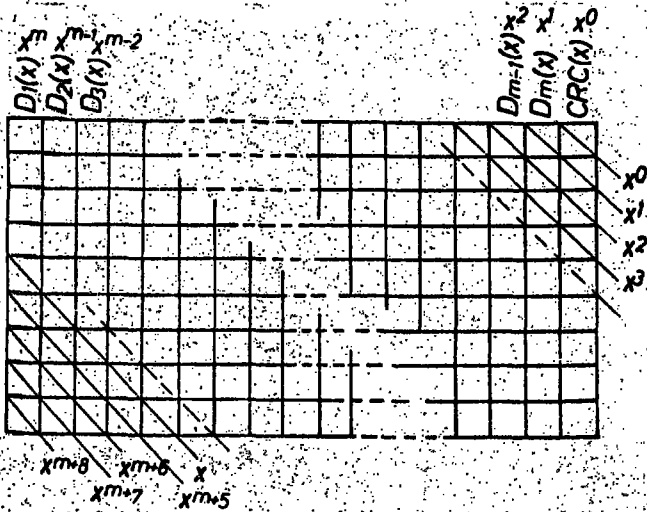
CRC - POLINOMSKA MREŽA

Za primenu CRC-kontrole u magnetnim trakama, gde je za 9-kanalnu traku veoma važan aspekt pouzdanosti, od strane AMERICAN NATIONAL STANDARDS INSTITUTE je izdat standard za "generator polinom".

$G(x) = x^9 + x^6 + x^5 + x^4 + x^3 + 1$. Ovak standard su preuzeli mnogi proizvođači opreme, jer obezbeđuje maksimum pouzdanosti.

Za slučaj magnetnih kaset standard za "generator polinom" je $G(x) = x^{10} + x^{15} + x^2 + 1$.

Podaci koji se nalaze zapisani na magnetnoj traci su 9-to bitni podaci - bajtovi, gde je 8 bitova podataka i bit za parnost. Ovi bajtovi čine formu polinomske mreže - slika 1.



(Primerba autora:

Za detaljnija obaveštenja o teoretskom prikazu rada sa CRC-kontrolom, čitalac može koristiti literaturu navedenu pod rednim brojem 1.)

Sl. 1 Zapis na drugoj traci. Ovde imamo slučaj od m-bajtova zapisanih na traci u obliku izdužene polinomske mreže - opšti slučaj. Ovo povećanje dužine se odražava, samo na povećanje stepena polinomske mreže.

Opšti izraz ove mreže je:

$$x^m \cdot D_1(x) + x^{m-1} \cdot D_2(x) + \dots + x \cdot D_m(x) = \sum_{i=1}^m x^{m+1-i} \cdot D_i(x)$$

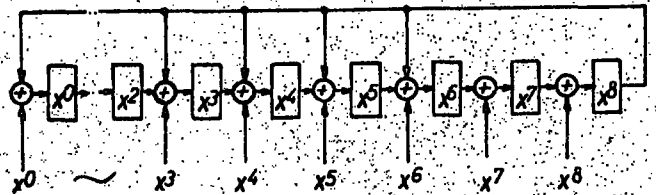
CRC-bajt je ostatak posle deljenja polinomske mreže sa "polinom generatorom" i kao takav postaje deo polinomske mreže (slika 1.). Ovak ostatak se iskazuje u obliku:

$$CRC(x) = \sum_{i=1}^m x^{m+1-i} \cdot D_i(x) \text{ mod. } G(x)$$

IZGLED POLINOMSKE MREŽE

Na slici dat je izgled polinomske mreže. Pri prikazivanju polinomske mreže koristimo dva logička simbola i to:

- a) simbol kružić sa krstićem u sredini čije je ime "ekskluzivno ili kolo" (EXCLUSIVE OR) i ostvaruje funkciju sabiranja binarnih brojeva po modulu 2.



Slika 2. Praktična polinomska mreža kod koje je deljenje "polinom generatorom" povratnom spregom na određena mesta mreže. Ova mesta su x^6, x^5, x^4, x^3 ($x^0 \equiv 1$) prema zadanom "polinomu generatoru", a signal stanja i se javlja kad član x^9 postigne tu vrednost i ne vodi se u kontinualni registar (slika 3) te se mreža ne širi povećanjem broja bajtova podataka koji se žele zaštititi.

Pomenuta mreža obezbeđuje deljenje sa "polinom generatorom" dakle sa fiksnim deliocem. U trenutku kad član x^9 postane 0, tada možemo "polinom generator" $G(x)$ izjednačiti sa 0 i rešiti ga po x :

$$G(x) = 0 = x^9 + x^6 + x^5 + x^4 + x^3 + 1$$

$$x^9 = x + x^5 + x^4 + x^3 + 1$$

a to su upravo razredi na koje se vraća povratna sprega sa kraja mreže ($x^9 = 1$).

LOKALIZOVANJE GREŠKE U PRENOSU

Od predajne strane prenosnog puta, gde smo podatke оформili sa CRC bajtom, prima se na prijemnoj strani ceo blok pomenutog sadržaja. Na prijemu se taj blok deli sa "polinom generatorom" i ako je ostatak 0, blok podataka je primljen korektno. Ako je ostatak nejednak 0, tražimo "polinom greške" $F(x)$.

Da bismo našli kanal i bajt sa greškom, uvodimo "vertikalnu redundantnu kontrolu" (VRC - VERTICAL REDUNDANCY CHECK), kontrolu neparnosti na svakom individualnom bajtu. Bajtovi bez greške proizvode $VRC = 0$, a bajtovi sa greškom $VRC = 1$ za odgovarajući bajt. Težina suma svih VRC u jednom bloku informacija čine $E(x)$. Paralelno sa prijemom bajtova podataka, u trenutku detektovanja $VRC = 1$, ovaj radnji ulazi u sličnu polinomsku mrežu (ili registar za deljenje) koja ima samo jedan ulaz.

Jasnije rečeno, kad se greška javlja samo u jednom kanalu, oba registra za delenje sadrže ostatak istog, "polinoma greške", međutim jedan je pomnožen sa stepenom od x koji odgovara pojedinom kanalu sa greškom. Da bi pronašli taj kanal, pomeramo "registar slike greške" (registar za delenje sa VRC ulazom) udesno, bez ulaza, poredeći ga svaki put sa sadržajem CRC registra (registar za delenje u cilju dobijanja CRC bajta). Posle t pomeraja, dva registra se podudare i broj pomeraja identifikuje kanal sa greškom.

PRAKTIČNA PRIMENA CRC - KONTROLE

Iz praktičnog primera bi trebalo da rezultuje jasnija slika ce-log postupka i da se možda bolje odgovor na neka postavljena pitanja iz prvog dela. Ceo primer bazira na zaštiti pet bajtova podataka, jer bi veći broj bio glomazan!

FORMIRANJE CRC - BAJTA PRI UPISU

Prilikom upisa nekog bloka podataka na magnetnu traku svaki se bajt "obrađi" u registru za delenje i dobijanje CRC bajta, a u slučaju da se tom prilikom pojavi greška VRC (Greška parnosti pojedinog bajta podataka) tada se formira i CRC - bajt, tzv. "slika greške". Na slici 3, imamo ceo blok od 5 proizvoljnih bajtova podataka sa CRC - bajtom i LRC - bajtom (o ovome će biti reči kasnije).

	D1	D2	D3	D4	D5	CRC	LRC
7	0	1	0	0	0	1	0
6	1	1	1	1	1	1	0
5	0	1	0	1	1	1	0
4	0	0	0	1	1	1	1
3	0	1	1	1	0	1	0
2	0	0	0	1	1	1	1
1	0	1	1	0	0	1	1
0	1	0	0	0	1	1	1
P	1	0	0	0	0	0	1

Slika 3. Na slici vidimo pet bajtova podataka D1 do D5 sa označenih 9 bitova sa 0, 1, ..., 7, P, što inače odgovara 9-kanalnoj magnetnoj traci. CRC bajt je rezultat postupka deljenja ovih 5 bajtova podataka, a ceo proces je dat na slici 6. Bajt nazvan LRC je bajt koji obezbeđuje za svaki kanal duž svih bajtova podataka "parnu parnost", tj: paran broj bitova vrednosti 1.

	P	0	1	2	3	4	5	6	7	
	0	0	0	0	0	0	0	0	0	CRC početna vred.
	0	0	0	0	0	0	0	0	0	Pomeraj
D1	1	1	0	0	0	0	0	1	0	Prvi bajt pod. ekskl. ili sa sadr. pomer.
	1	1	0	0	0	0	1	0		CRC bajta (nule) (1)
D2	0	1	1	0	0	0	0	0	1	Isto kao (1)
	0	0	1	0	1	0	1	1	1	Pomeraj
D3	0	0	1	0	1	0	0	1	0	Isto kao (1)
	0	0	0	0	1	1	0	0	1	Pomeraj
	1	0	0	0	0	1	1	0	0	Poz. 2-5 su invert. kad je P=1 (2)
D4	1	0	0	1	1	1	1	1	0	Četvrti bajt podat. je ekskl. ili sa gornjim (100110000)
	1	0	0	0	0	1	1	1	0	

Nastavak
tabele na sledećem stupcu

Nastavak tabele

	(P	0	1	2	3	4	5	6	7)	
	0	1	0	0	0	0	1	1	1	Pomeraj
D5	0	1	0	1	0	1	1	1	0	Isto kao (1)
	0	0	0	1	0	1	0	0	1	Posebni pomeraj
	1	0	0	0	1	0	1	0	0	Kon. izg. CRC bajta
	1	0	0	1	0	1	0	0	0	CRC bajt
	0	1	1	1	1	1	1	1	1	CRC bit pozicije su komplementir. izuzev pozic. 2 i 4.

Slika 4. Treba uočiti da se nakon postupka deljenja CRC-bajta na traku upisuje u komplementnom sadržaju osim bitova 2 i 4. Ova izuzetka je praktične prirode i nema suštinske veze sa CRC-kontrolom.

FORMIRANJE CRC - BAJTA PRI ČITANJU

Na prijemnoj strani prenosnog puta kad sa magnetne trake čitamo prethodno upisani blok, moguća su dva slučaja. Prvi, normalan slučaj, ako je pročitani blok bez gubitka bitova, a to zaključujemo tako što se ceo blok podeli sa $G(x)$ i dobijeni CRC-bajt se poredi sa CRC-bajtom primljenim sa trakom. Ako su dva bajta istog sadržaja nakon poredjenja ostatak je 0, a to je dokaz ispravnog prenosa podataka.

Drugi slučaj, kad je nastala greška na pojedinim bajtovima (umesto stanja 1 nastaje stanje 0, i obratno) zahteva pronalazanje mesta greške. Primljeni blok nema ima npr. sadržaj prema slici 5.

	D1	D2	D3	D4	D5	CRC	LRC
7	0	1	0	0	0	1	0
6	1	1	1	1	1	1	0
5	0	1	0	1	1	1	0
4	0	0	1	0	0(1)	0(1)	1
3	0	1	1	1	0	1	0
2	0	0	0	1	1	1	1
1	0	1	1	0	0	1	1
0	1	0	0	0	1	1	1
P	1	0	0	0	0	0	1

Slika 5. Greška se javlja u bajtovima podataka D(3), D(4) i samom CRC bajtu; a na kanalu 4, i to tako da je vrednost u zagradama - prava vrednost - izgubljena u prenosu. Kad se ovaj ceo blok - D(1) do D(5) - podeli sa $G(x)$ dobijeni CRC - bajt se poredi sa pročitanim, a kao rezultat se dobije sadržaj različit od nule. Ceo postupak je na slici 6.

	P	0	1	2	3	4	5	6	7	
	0	0	0	0	0	0	0	0	0	CRC početna vrednost
	1	1	0	0	0	0	1	0		Prvi bajt sa trake
	1	1	0	0	0	0	1	0		Ekskluz. ili CRC bajt sa prvim bajtom podataka
	0	1	1	0	0	0	0	1		Pomeraj
	0	0	1	0	1	0	1	1	1	Drugi bajt (bez greške)
	0	1	0	0	1	0	1	1	0	Ekskluzivno ili
	0	0	1	0	0	1	0	1	1	Pomeraj
	0	0	1	0	1	1	0	1	0	Treći bajt (sa greškom)
	0	0	0	0	1	0	0	0	1	Ekskluzivno ili
	1	0	0	0	1	0	0	0	0	Pomeraj
			1	1	1	1				Kad je P=1 pozicije 2-5 su invertovane
	1	0	0	1	1	0	1	0	0	

Nastavak
tabele na sledećem stupcu

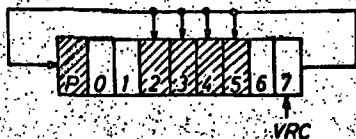
(P 0 1 2 3 4 5 6 7)	
0 0 0 1 1 1 1 0	Četvrti bajt
1 0 0 0 0 1 0 1	Ekskluzivno III
0 1 0 0 0 0 1 1	Pomeraaj
0 1 0 1 0 0 1 0	Peti bajt
0 0 0 1 0 0 0 1	Ekskluzivno III
1 0 0 0 1 0 0 1	Poseb. pomeraaj bez ul.
	1 1 1 1
1 0 0 1 0 1 1 0	
0 1 1 1 1 0 1 1	CRC bajt (sa greškom)
1 1 1 0 1 1 0 1	Ekskluzivno III

Po meraaj, inverzija i
ekskluzivno III se od-
vija u istom trenutku.

Slika 6. Dakle, u ovom trenutku raspolazemo sa dokazom da je neki kanal preneo pogresan bit (III bitove).

GENERISANJE "SLIKA GREŠKE"

Paralelno radu registra za delenje sa CRC bajt, radi i registar za delenje sa VRC-bitom na ulazu koji "snima" na kome bajtu podataka se javlja greška VRC=1, tj. greška parnosti bajta podataka VRC - registar za delenje dat je na slici 7.



Slika 7. Kolo je identično kolu sa slike 2, osim što ima samo jedan ulaz. Ceo postupak formiranja "slike greške" dat je na slici 8.

SG	P	0	1	2	3	4	5	6	7	
	0	0	0	0	0	0	0	0	0	Počet. vrednost
1										0 Nema vertik. greš.
	0	0	0	0	0	0	0	0	0	Ekskluzivno III
2										Pomeraaj
	0	0	0	0	0	0	0	0	0	0 Nema vertik. greš.
	0	0	0	0	0	0	0	0	0	Ekskluzivno III
3										Pomeraaj
	0	0	0	0	0	0	0	0	0	1 Detekt. vert. greška
	1	0	0	0	0	0	0	0	0	Ekskluzivno III
4										Pomeraaj
	1	0	0	1	1	1	1	0	0	0 Poz. 2-5 invertov.
	1	0	0	1	1	1	1	0	0	0 Rez inver. i pom.
	1	0	0	1	1	1	1	0	0	0 Nema greške
	1	0	0	1	1	1	1	0	0	Ekskluzivno III
5										Pomeraaj
	0	1	0	0	1	1	1	1	0	1 Vertikalna greška
	0	1	0	0	1	1	1	1	1	Ekskluzivno III
	1	0	1	0	0	1	1	1	1	Pomeraaj
										1 Poz. 2-5 invertovan.
CRC1	0	1	1	1	0	0	1	1	1	1 Vertikalna greška
	1	0	1	1	1	0	0	1	0	Ekskluzivno III

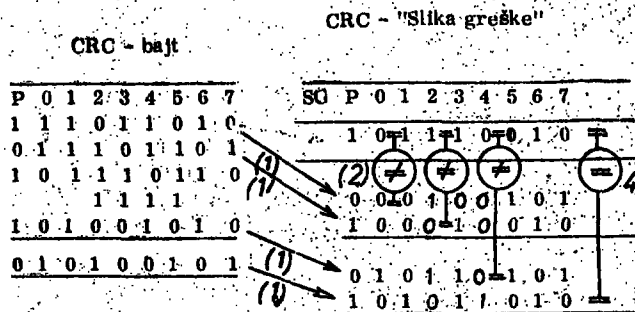
Slika 8. Dobljeni CRC-bajt nosi u sebi "sliku greške" (tj. poziciju greške u nizu bajtova).

TRAŽENJE KANALA SA GREŠKOM

Traženje kanala u kome se javlja greška se bazira na dva CRC bajta: CRC bajt dobljen postupkom deljenja bajtova podataka (slika 6) i CRC-bajt koji deli samo VRC-bit "Slika greške". Postupak sadrži sledeće korake:

- komplementiranje CRC-bajta (podaci) (1)
- poredjenje istog sa CRC-bajtom (slika greške) (2)

Ovaj postupak se zaustavlja kad je poredjenje uspešno, a broj poredjenja je broj kanala koji greši (u primeru sa slike 9, to je 4).



Slika 9. Bitovi CRC-bajt su označeni kao SG1 - 7, P.

Ako posle 9 ponovljenih postupaka poredjenja se ne pojavi koincidencija, znači više kanala izaziva grešku.

AUTOMATSKA KOREKCIJA GREŠKE

Znajući koji kanal greši i u kome se bajtu te greške javljaju moguće je automatski ispraviti grešku (promena binarnog stanja na mestu greške). Ovaj postupak je ostvaren delovanjem dela softvera računске mašine, koji kontrolise rad perifernih uređaja.

NAPOMENA AUTORA

Na ovom mestu se zahvaljujem mag. Pottč Veselinu, dipl. ing. Institut "Mihajlo Pupin", na stručnoj recenziji članka.

L I T E R A T U R A

1. ROBERT SWANSON, "UNDERSTANDING CYCLIC REDUNDANCY CODES", COMPUTING DESIGNS, TUSCON, ARIZONA
2. M. N. "CYCLIC REDUNDANCY CHECK - PRACTICAL SOLUTION, FACOM 230-45S, FUJITSU LIMITED
3. N. S. "CYCLIC REDUNDANCY CHECK FOR CASSETTE TAPE UNIT" FACOM 230-45S, FUJITSU LIMITED
4. TANG, CHIEN, "CODING FOR ERROR CONTROL" IBM SYST J, N° 1, 1969.

ceneni mikroračunalnik s procesorjem SC/MP

j. tasič
l. lenart
b. blatnik

UDK 681.3 - 181.4

Institut "Jožef Stefan" Ljubljana

Od leta 1973, ko se je pojavil prvi mikroprocesor druge generacije, pa do danes se je cena procesorjem manjšala celo hitreje kot so predvidevali največji optimisti. Tako je padec cen omogočil hiter prodor mikroračunalnikov v razne aplikacije. Kljub temu, pa še cena pada. Trenutno je najcenejši mikroprocesor druge generacije SC/MP, tovarne National (ZDA), ki je bil namenjen le za uporabo v enostavnih in ceneni aplikacijah.

Namen tega članka je seznaniti bralce in uporabnike z omenjenim mikroprocesorjem ter z enostavnim in ceneni računalkom zgrajenim na osnovi tega procesorja.

LOW COST MICROCOMPUTER WITH THE PROCESSOR SC/MP - The performances of low cost microprocessor SC/MP of National semiconductor's are presented as needed for simple self made microcomputer system. Instruction set and the simplest bench programming technique are discussed.

LASTNOSTI MIKROPROCESORJA

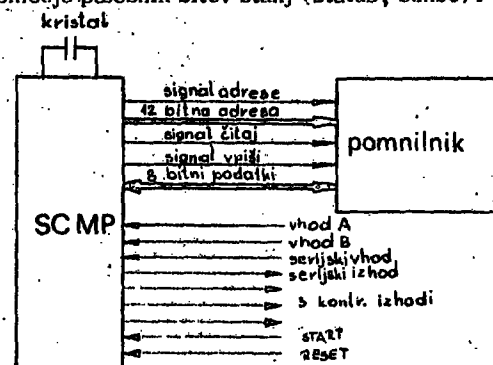
Kar v začetku postavimo bralca pred dejstvo, da sta na trgu na razpolago dva mikroprocesorja SC/MP. Prvi, ki je izveden v P-MOS tehnologiji, potrebuje dvoje napajalnih virov in sicer +5V in -7V, drugi SC/MP II ki je izvedenka prvega v N-MOS tehnologiji pa potrebuje le 5V in je tudi kompatibilen s prvim. V članku ne bomo posebej ločili enega od drugega, zato bomo uporabljali oznako SC/MP, le podatki, ki jih podajamo na koncu članka veljajo za SC/MP II.

SC/MP je 8 bitni mikroprocesor, ki ga je moč uporabiti brez posebnih dodatnih kontrolnih elementov v enostavnem mikroračunalniku. Tako lahko zgradimo ceneni mikroračunalnik le s 4-mi integriranimi vezji: SC/MP, ROM, 2x RAM. V sled enostavnosti proizvajalec ne dela dodatnih elementov, ki bi pripadali družini imenovani SC/MP, kar je razvidno pri drugih procesorjih kot so: Intel 8080, Motorola 6800, MOS TECH. 6500. etc. Proizvajalčev namen je bil proizvesti ceneni mikroprocesor, ki je uporaben v aplikacijah, kjer pomeni cena elektronskega dela pomemben kriterij. Procesor ima 40 priključkov in ima lastnosti, ki jih srečujemo le pri dražjih izvedbah. Tako ima notranji generator urinih impulzov, možnost komuniciranja s 65K pomnilnikom, serijsko paralelni in paralelno serijski prenos podatkov ter enoten nabor ukazov za komunikacijo s perifernimi enotami in s pomnilnikom.

Glavne prednosti mikroprocesorja SC/MP so v enostavnem povezovanju z ostalimi enotami, v enostavnem programiranju, v možnosti povezave večih procesorjev v multiprocesorski sistem, možnosti direktne komunikacije perifernih enot s pomnilnikom brez posredovanja mikroprocesorja (DMA), direktni kontroli perifernih enot z uporabo treh bitov stanj (status bits), v enostavni vhodno/izhodni materialni opremljeni, veliki pomnilnik enoti ter navsezadnje tudi v nizki ceni.

Komunikacija mikroprocesorja s perifernimi enotami

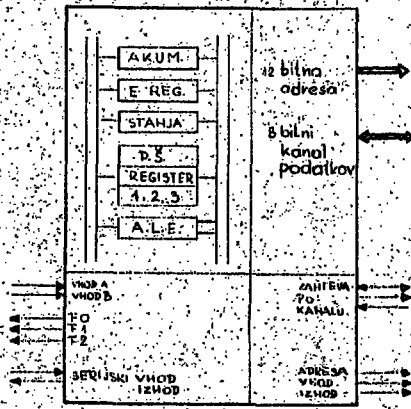
in s pomnilnikom se vrši preko skupnega dvosmernega kanala podatkov, ter preko 12 bitnega adresnega kanala. Ostale štiri bite adrese mikroprocesor generira na podatkovnem kanalu in jih moramo shraniti v začasnem pomnilniku. Ti štiri bite določajo strani pomnilnika, ostalih 12 bitov pa besedo na določeni strani. Mikroprocesor prav tako generira tri kontrolne signale (NADS, NWRD, NRDS), ki omogočajo sinhronizacijo perifernih in pomnilnih enot z mikroprocesorjem. Njegovo uporabnost povečuje tudi možnost serijske komunikacije s perifernimi enotami, ki je pod programsko kontrolo. Ostali kontrolni signali, ki jih je moč razbrati s slike 1, so namenjeni splošni kontroli pri komunikaciji in omogočajo start, ustavitve, zahteve po prekinitvi, vračanje v začetno stanje (restartiranje), dodeljevanje kanalov drugim procesorjem, programsko kontrolo perifernih enot s pomočjo posebnih bitov stanj (status, sense).



Slika 1. Osnovni komunikacijski kanal mikroprocesorja SC/MP

Kar v začetku lahko povemo, da ima SC/MP več načinov adresiranja, ki jih bomo spoznali kasneje. Omenjeni mikroprocesor vsebuje 10 notranjih registrov, od katerih je sedem programsko dostopnih registrov. Ti so

Razvidni iz slike 2, ki prikazuje notranjo organizacijo mikroprocesorja, ki jo mora bralec spoznati.



Slika 2. Organizacija mikroprocesorja SC/MP

REGISTRI MIKROPROCESORJA

Programski števec je 16 bitni register in vsebuje addresso ukaza, ki se bo ali se izvršuje. 12 bitov programskega števca je direktno dostopnih na adresnem kanalu, torej direktno na izhodnih priključkih SC/MP-ja.

Kazalni register se sestoji iz treh 16 bitnih registrov, P1, P2, P3, ki so pod programsko kontrolo in vsebuje referenčne addresse. Torej lahko služijo kot kazalo sklada adres, ali adres podprogramov itd. Prav tako se uporablja tretji kazalni register pri prekinjivnih zadevah. Le-ta vsebuje ustrezno addresso prekinjivnega podprograma.

Akumulator je 8 bitni delovni register, ki služi za obdelavo in shranjevanje rezultatov aritmetičnih in logičnih operacij. Prav tako služi za ostale naloge, ki jih opravljajo delovni registri pri drugih mikroprocesorjih.

Dotadni register E služi za serijsko vhodno/izhodno komunikacijo, ali pa kot druga beseda pri indeksnem ali avtoindeksnem adresiranju.

Register stanj omogoča shranjevanje aritmetičnih, kontrolnih in drugih programske bitov stanj. Pet bitov stanj je tudi dostopnih na priključkih mikroprocesorja.

Ukazni register je 8 bitni register, ki ni dostopen programerju. Register vsebuje informacijo, ki se prenese v času čitanja ukaza iz pomnilnika.

Prav tako je nedostopen vhodno/izhodni podatkovni register, ki služi le za začasno shranjevanje vhodno/izhodnih podatkov.

Adresni register je 16 bitni register, ki služi le začasnemu shranjevanju adrese v času vhodno/izhodnega cikla. Tudi ta ni dostopen programerju.

Navsezadnje moramo omeniti aritmetično-logično enoto, ki omogoča manipulacijo s podatki kot vsak drugi mikroprocesor.

KONTROLNA LOGIKA

Mikroprogram, ki generira posamezne sekvence, katere pripadajo posameznim ukazom, je del kontrolne logike mikroprocesorja. Tako se osnovni operaciji čitanja

in izvrševanja ukazov izvršujeta pod kontrolo mikroprograma. Rutina za čitanje ukaza najprej poveča programski števec in šele nato omogoči prenos adrese k perifernim enotam. Vsebinsko, ki je na dotični adresi shranjen ukazni register. Takoj nato se začne izvrševanje ukaza, pri čemer upošteva ali je ukaz eno ali dvo beseden. V slednjem primeru se druga beseda začasno shrani v vhodno/izhodnem registru. Pri izvrševanju programa ničnografe omenimo, da ukazni dekodirer dekodira kodo ukaza v ekvivalentno addresso mikroprograma, ki nato omogoča pravilno izvrševanje ukaza.

Prekinjivni ukazi so del mikroprocesorjeve logike. SC/MP ima samo eno prekinjivno linijo, ki ji programsko omogočimo razpoznavanje prekinjivnega ukaza s strani mikroprocesorja. To omogoča register stanj s svojim bitom IE (Interrupt Enable). V primeru, ko pride do zahteve po prekinitvi s strani perifernih enot je lahko bit stanja IE logično "1" ali "0". V prvem primeru je bila zahteva po prekinitvi sprejeta s strani mikroprocesorja, v drugem primeru pa jo je mikroprocesor prezrl. V primeru sprejete zahteve po prekinitvi, mikroprocesor najprej konča s tekočim mikroprogramskim paketom, nakar onemogoči nastop ostalih možnih prekinjivnih zahtev. Mikroprocesor tudi menja vsebino programskega števca z vsebino kazalnega registra P3, ki vsebuje začetno addresso prekinjivne rutine. V primeru, ko bi nastopilo v problemu, ki ga rešujemo z mikroprocesorjem, več prekinjivnih zahtev, tedaj SC/MP ni tako uporaben kot so drugi mikroprocesorji.

Oglejmo si primer serijskega prenosa podatkov v mikroprocesor. Začetna addressa prekinjivnega programa je v registru P3. Ustrezn program je sledeč:

SIN	SIO		vhod je naslednji serijski bit
	DLD	COUNT	zmanjšaj števec bitov in vstavi v A
	JNZ	END	skoci na END ce ni še konec
	LDI	8	
	ST	COUNT	
	LDI	0	vstavi besedo iz E v A
	XAE		počisti E register za nasled. besedo
	ST	al, IOBUF	vstavi besedo v reg. adresiran z PR 2
END	IEN		vrnitev iz programa
	XPPC		

Kot drugi mikroprocesorji, tudi SC/MP dopušta direkten dostop k pomnilniku poljubnim perifernim enotam. Pri tem je pomembno, da na ta način lahko pride do hitrega prenosa podatkov v pomnilnik brez posredovanja mikroprocesorja. Zunanja logika lahko komunicira z mikroprocesorjem na dva načina:

- ali, da uporabi NHALT signal in s tem zaustavi CPU,
- ali testira kontrolo linije v začetku cikla izvrševanja ukaza. V primeru, ko ukaz ne zahteva komunikacije s pomnilnikom, kar zazna zunanja logika v času 2 μ s, tedaj je pomnilnik dostopen zunanjim enotam.

Kot pomembno informacijo moramo smatrati tudi podatek o startu mikroprocesorja ob vklopu napajalnega vira. V tem primeru mikroprocesor v 250 nš resetira vse notranje registre, stabilizira uro tako nato že čita prvi ukaz iz začetne adrese 0001₁₆. Zato vse programe ki so namenjeni raznim uporabi pišemo od adrese 0001₁₆ dalje, kar omogoča pravičen start programa v vsakein primeru.

PRENOS PODATKOV

Prenos podatkov, torej komunikacija mikroprocesorja s pomnilnikom ali z zunanjimi enotami šele da pravo mesto mikroprocesorju. Tako mikroprocesor lahko komunicira z zunanjim realnim svetom preko paralelnih ali serijskih kanalov.

Tako se paralelni prenos podatkov dogodi pri vsaki komunikaciji procesorja s pomnilnikom ali s vhodno/izhodnimi enotami. Enotna struktura podatkovnega kanala poenoti ukaze za komunikacijo s pomnilnikom in s perifernimi enotami, kar je značilnost mikroprocesorja SC/MP. Tako lahko pri komunikaciji s perifernimi enotami uporabimo ukaze LD, ST, (load, store) za osnovne prenose, ILD in DLD (increment, decrement load) pa služijo za indeksiranje perifernih registrov. Pri perifernem prenosu informacij, mora imeti mikroprocesor dostop do adresnega in podatkovnega kanala. Pri tem mu za kontrolo služijo trije signali NBREQ, NEIN, NENOUT. Prvi le označuje, da je SC/MP zasedel oba systemska kanala, druga dva pa le označujeta prenos podatkov eno ali drugo smer, torej vpis ali izpis podatkov v periferno oz. pomnilnik. V primeru, ko mikroprocesor zaseda katerega od kanalov, sta NBREQ in NENIN vedno logično "0", dočim je NENOUT vedno logično "1" pri pozitivni logiki, ki jo ima SC/MP. V primeru, ko kanal podatkov ni zaseden, ko je NBREQ logično "1", tedaj sta NENOUT in NENIN vedno v enakem logičnem stanju. Pri komunikaciji s perifernimi enotami in s pomnilnikom, kjer moramo točno določiti posamezne funkcije, kot so adresiranje, čitanje, vpisovanje, generira mikroprocesor kontrolne signale kot so NADS, NRDS in NWDS. Pri tem je signal NBREQ vedno v stanju "0", ko pa se vhodno izhodna komunikacija konča preide slednji signal v stanje "1", kar pomeni, da je kanal prost tudi za druge mikroprocesorje ali za hitro komunikacijo s perifernimi enotami.

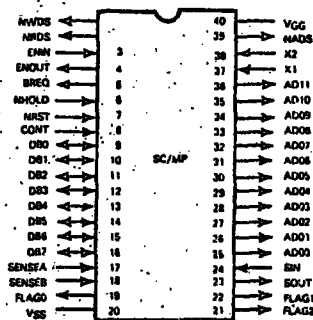
Pri komunikaciji s počasnimi perifernimi enotami lahko vsak vhodno/izhodni cikel podaljšamo z uporabo NHOLD signala, ki ga testira mikroprocesor sam. V primeru, ko je ta logično "1", tedaj procesor podaljša vhodno/izhodni cikel vse dotlej, dokler ne pride NHOLD signal v logično stanje "0".

Kot paralelni prenos pa je tudi pomemben serijski prenos podatkov, ki ga omogoča SC/MP. Ta je značilen predvsem za obravnavani mikroprocesor. Tak način prenosa je koristen pri komunikaciji s počasnimi perifernimi enotami, kot je teleprinter ali X-Y koordinatni pisalnik ali pa pri komunikaciji na večje daljave. Ta način prenosa je možno izvesti na tri načine:

- preko serijskega vhodno/izhodnega kanala, kar pomeni z uporabo registra E in SIO (serial I/O) ukaza. V primeru, ko se izvrši ukaz SIO, se vsebina registra E pomakne na desno za eno mesto. To pa pomeni, da se prečita v register E na mesto 7 logična vrednost, ki se je ta trenutek izvrševanja ukaza nahajala na priključku mikroprocesorja SIN, ter da se izpiše vsebina bita 0, preko SOUT priključka v periferno enoto.
- drugi način serijske komunikacije s perifernimi enotami je z uporabo registra stanj in bitov, ki so dosegljivi perifernim enotam. Tu uporabimo kot izhodni bit bit stanja, kot vhodni bit pa bit, ki omogoča programsko odtipavanje vhodnega stanja.
- tretji način je ta, da uporabimo standardne ukaze in vrata za paralelno komunikacijo, le da namesto osmih upoštevamo samo en bit.

PRIKLJUČKI IN SIGNALI MIKROPROCESORJA SC/MP

Slika 3 prikazuje osnovne priključke mikroprocesorja SC/MP in opis posameznih signalov



Slika 3. Osnovni priključki mikroprocesorja

Ime kontakta	Opis	Tip
X1, X2	priključek kristala	vhod
DB0-DB7	podatkovni kanal	izhod/vhod
AD00-AD11	adresni kanal	izhod
SENSE A, SENSE B	vhod zunanjega stanja	vhod
FLAG 0, 1, 2	izhodni biti stanja	izhod
NRST	reset	vhod
CONT	stop/nadaljaj	vhod
BREQ	signal za zahtevo kanala	dvosmeren
ENIN	podatkovni kanal dostopen	vhod
ENOUT	kanal zaseda CPU	izhod
NADS	adresa na kanalu	izhod
NRDS	čitanje podatkov	vhod
NWDS	izpis podatkov	izhod
NHOLD	zakasnitev	vhod
SIN	serijski vhod	vhod
SOUT	serijski izhod	izhod
Vgg, Vss	napajanje, masa	vhod

Oglejmo si detaljnije opise posameznih priključkov mikroprocesorja, ki jih s slike 3 ne moremo točno razbrati.

Poleg osnovnih kanalov so zanimivi priključki na register stanj. Ti so: SENSE A, SENSE B, FLAG 0, 1, 2, s katerimi programsko kontroliramo periferne enote. Prva dva sta vhoda v register stanj, ostali trije pa so izhodi iz registra. Poleg programsko dosegljivih vhodno/izhodnih signalov nastopajo še mikroprocesorjevi vhodno/izhodni signali, ki pa niso programsko dosegljivi. Te lahko razdelimo na tri skupine:

- kontrolni signali, ki krmilijo kanal podatkov so BREQ, ki določa zasedenost kanala, ENIN je kontrolni signal, s katerim zunanja logika sporoča mikroprocesorju potrebo po komunikaciji, a ENOUT je odgovor CPU-ja na ENIN signal. Ko je izhod mikroprocesorja v visokem impedenčnem stanju, tedaj ENOUT z logičnim nivojem "0" označuje, da je tudi ENIN v logičnem stanju "0", kar pa pomeni, da je kanal prost za komunikacijo. V primeru, ko pa signala ENIN in ENOUT prideta v logično stanje "1" pomeni, da se CPU pripravlja, da

zasede podatkovni kanal in ko sta ENIN in BREQ logično "1", tedaj je kanal zaseden.

- v primeru, ko CPU zaseda podatkovni kanal, tedaj trije signali določajo tip prenosa podatkov. Tako NADS določa prenos štirih bitov adrese po podatkovnem kanalu, NRDS določa, da se podatki prenašajo iz perifernih enot v mikroprocesor, NWDS pa, da se podatki prenašajo iz mikroprocesorja v periferne enote ali v pomnilnik.

- trije kontrolni signali pa določajo časovne funkcije mikroprocesorja. Tako je NRST sistemski reset signal, ki resetira vse registre mikroprocesorja, CONT je vhodni signal, ki omogoča prekinitev CPU enote med dvema ukazoma. Tretji kontrolni signal pa je NHOLD, ki je potreben pri komunikaciji s procesno periferijo.

NAHOR INSTRUKCIJ IN STANDARDNA PROGRAMSKA OPREMA

Mikroprocesor SC/MP in njegov nabor ukazov omogočata reševanje poljubnih problemov. Seveda moramo upoštevati hitrost izvrševanja posameznih ukazov, njihove lastnosti in velikost problema, ki ga rešujemo z mikroprocesorjem SC/MP.

Nabor vseh instrukcij porazdelimo na dvo in eno besedne instrukcije. Vseh 46 instrukcij pa porazdelimo še na:

- pomnilniške instrukcije:

		76543210	76543210
LD	Load	11000	ptr disp
ST	Store	11001	
AND	AND	11010	
OR	OR	11011	
XOR	Exclusive-OR	11100	
DAD	Decimal Add	11101	
ADD	Add	11110	
CAD	Complement and Add	11111	

- inkrement, dekrement instrukcije:

		76543210	76543210
ILD	Increment and Load	10101	ptr disp
DLD	Decrement and Load	10110	

- instrukcije za prenos:

		76543210	76543210
JMP	Jump	100100	ptr disp
JP	Jump if Positive	100101	
JZ	Jump if Zero	100110	
JNZ	Jump if Not Zero	100111	

- neposredne instrukcije:

		76543210	76543210
LDI	Load Immediate	11000100	data
ANI	AND Immediate	11010100	
ORI	OR Immediate	11011100	
XRI	Exclusive-OR Immediate	11100100	
DAI	Decimal Add Immediate	11101100	
ADI	Add Immediate	11110100	
CAI	Complement and Add Immediate	11111100	

- ostale dvobesedne instrukcije:

		76543210	76543210
DLY	Double-Byte Miscellaneous Instructions Delay	10001111	disp

- instrukcije za delo z E registrom:

		76543210
LDE	Load AC from Extension	01000000
XAE	Exchange AC and Extension	00000001
ANE	AND Extension	01010000
ORE	OR Extension	01011000
XRE	Exclusive-OR Extension	01100000
DAE	Decimal Add Extension	01101000
ADE	Add Extension	01110000
CAE	Complement and Add Extension	01111000

- instrukcije kazalčnemu registru:

		76543210
KPAL	Exchange Pointer Low	001100
KPAH	Exchange Pointer High	001101
KPPC	Exchange Pointer with PC	001111

- instrukcije za pomik:

		76543210
SIO	Serial Input/Output	00011001
SR	Shift Right	00011100
SRL	Shift Right with Link	00011101
RR	Rotate Right	00011110
RRL	Rotate Right with Link	00011111

- ostale enobesedne instrukcije:

		76543210
MALT	Half	00000000
CCL	Clear Carry/Link	00000010
SCL	Set Carry/Link	00000011
OINT	Disable Interrupt	00000100
IEN	Enable Interrupt	00000101
CSA	Copy Status to AC	00000110
CAS	Copy AC to Status	00000111
NOP	No Operation	00001000

Čas izvrševanja posameznih instrukcij je od 5us do 22-us. Zanimivost nabora instrukcij mikroprocesorja SC/MP je tudi ukaz za zakasnitev, ki lahko tvori zakasnitve od 13 do 593 us. Iz nabora vseh instrukcij je razvidno, da mikroprocesor nima posebnih instrukcij za komunikacijo s perifernimi vhodno/izhodnimi enotami, ampak pri tem uporablja isti nabor instrukcij, ki jih ima za komunikacijo s pomnilnikom.

Zanimivost tega mikroprocesorja je tudi ta, da se vse operacije vrše po straneh (4K) in da se ne izvrši prenos bita "carry" v enoto za izbiro strani. To mora izvršiti programer sam. Pri tem preskoku iz ene na drugo stran običajno uporabimo register P1 kot števec strani.

S to analizo so dane osnove za realizacijo enostavnega mikroročunalnika. Prednost omenjenega sistema je v tem, da nani ni potrebno dograjevati posebnega zunanega kontrolerja. Želimo ustvariti sistem, ki bo omogočal komunikacijo s teleprinterjem, izhodnimi 16 bitnimi vrati, ter 16 bitnimi vhodnimi vrati prav tako pa bo imel 1K RAM pomnilnika in 0.5K PROM pomnilnika. Sistem naj bo hkrati tudi razširljiv.

Osnovni podatek mikroprocesorja je ta, da lahko njegove kanale priključimo le po ena uporabniška vrata. Posledica te zahteve je ta, da moramo tokovno ojačiti vse adrese, podatkovne in kontrolne kanale. Prav tako moramo dvosmerni podatkovni kanal pretvoriti v dva enosmerna kanala s uporabo vmesnika za vhodne podatke in vmesnika za izhodne podatke. Ker imamo direktno na razpolago le 12 bitno addresso, ki omogoča komunikacijo s 4K pomnilnikom, moramo še dodati element, ki na osnovi bitov 42-15 določi stran spomina, s katero komunicira mikroprocesor. Z dodatkom vmesnih krnilnih elementov smo dobili sistem, ki omogoča komunikacijo z dvema vhodno/izhodnima podatkovnima besedama (slika 6).

V primeru, ko želimo realizirati najcenejši mikroročunalnik,

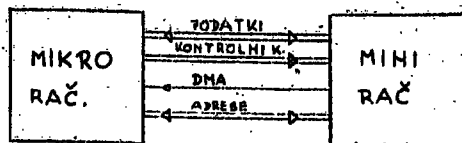
moramo uporabiti dva RAM 256 x 4 elementa, ROM pomnilnik ter TTY krmilni element. V prvem primeru lahko priključke mikroprocesorja X1, X2 priključimo 1MHz kristal ali pa ekvivalenten kondenzator. Primer mikro-računalnika s 4K pomnilnikom je prikazana na sliki. Konfiguracija omogoča dodatno dopolnitev: tipke REŠET, tipke HALT, tipke SINGLE STEP CONTINUE ter navsezadnje tudi enostavne konzole računalnika. Na kontakte konektorja moramo pripeljati vse krmilne signale in vse kanale podatkov in adres, saj le to omogoča razširitev cenene-ga mikro-računalnika v večji sistem (do 65K). Na kontakte konektorja povežemo tudi 16 bitna vhodno/iz-hodna vrata, ki jih kot ostale signale vodimo preko krmil-nih elementov.

Omenimo še, da je SC/MP kompatibilen s standardnimi elementi RAM, ROM, PROM, EPROM. Mikroprocesorjevi kontrolni signali omogočajo prenos adres pomnilnim elementom ter obojestranski prenos informacij, prav tako komunikacijo s perifernim elementom. S slike je razvidno, da isti dekoder naslavlja vhodno/izhodna vrata ter pomnilnik, kar omogoča komunikacijo s perifernimi elementi ter s pomnilnikom z istim naborom instrukcij.

Namesto PROM pomnilnika lahko uporabimo proizvajalčev ROM (KITBUG), ki vsebuje vse kontrolne programe za krmiljenje teleprinterja vnosa in popravljanje programov, izpisovanje ter startanje poljubnega programa na poljubni adresi. Ta kontrolni program v veliki meri poenostavi delo z mikro-računalnikom, saj omogoča razvoj manjših programov, v šestnajstiški kodli.

Pri uporabi mikro-računalnika SC/MP v večjih sistemih, ali v sistemih, kjer je potreben hiter prenos podatkov, je le primerno uporabiti njegovo pomembno lastnost, možnost direktnega pristopa perifernih sistemov k pomnilniku računalnika. Seveda mora zunanja kontrolna logika, ki omogoča omenjeno komunikacijo sinhronizirati prenos z mikroprocesorjem in generirati adrese v področju, kjer je to dopustno.

Tak primer enostavne povezave mikro-računalnika z večjim računalnikom, kjer le-ta predstavlja zunanji sistem prikazuje slika 4.

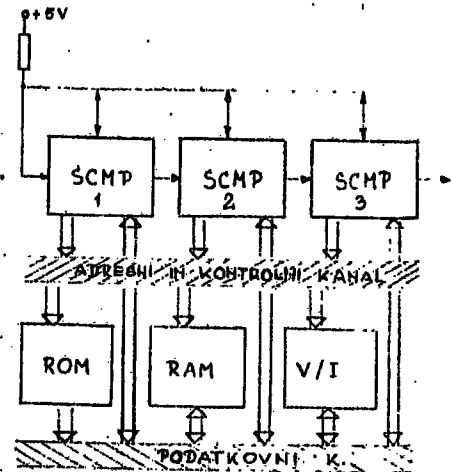


Slika 4. Enostavna povezava mikro-računalnika z miniračunalnikom

Pri tem lahko mini računalnik direktno spreminja program dela mikro-računalnika ali pa lahko sprotno neprimerno hitreje obdelava niz podatkov, ki jih je zbral mikro-raču-nalnik. Kontrolni kanal mikro-računalnika v tem primeru testira BREQ linijo in ko je ta logično "0", in ko je ENIN linija logično "0", kar je po zaključku vhodno/iz-hodne operacije, tedaj sme prekiniti delo mikroprocē-sorja in ga postaviti v visoko-impedično stanje, ter sme začeti komunikacijo s pomnilnikom.

Druga prijetna in že nekajkrat omenjena lastnost mikro-procesorja pa je tudi v možnosti povezave večjih mikro-procesorjev v multiprocesorski sistem. Tako blokovno

shemo povezave vidimo na sliki 5. Trije kontrolni sig-nali in prioritetno dodeljevanje funkcij posameznim mi-kroprocesorjem. Ta sistem



Slika 5. Shema multiprocesorskega sistema

večih povezanih mikroprocesorjev lahko razširimo še na povezavo multiprocesorskega sistema z miniračunalnikom. To pa že predstavlja manjšo računalniško mrežo. Iz tega lahko bralec sklepa, da je mikroprocesor SC/MP širo-ko uporaben in da ga lahko dokaj enostavno uporabljamo tudi v večjih a počasnejših sistemih.

V članku so podane lastnosti mikroprocesorja SC/MP, njegov nabor instrukcij. Del članka pa je tudi posvečen izdelavi cenene-ga mikro-računalnika s procesno enoto SC/MP.

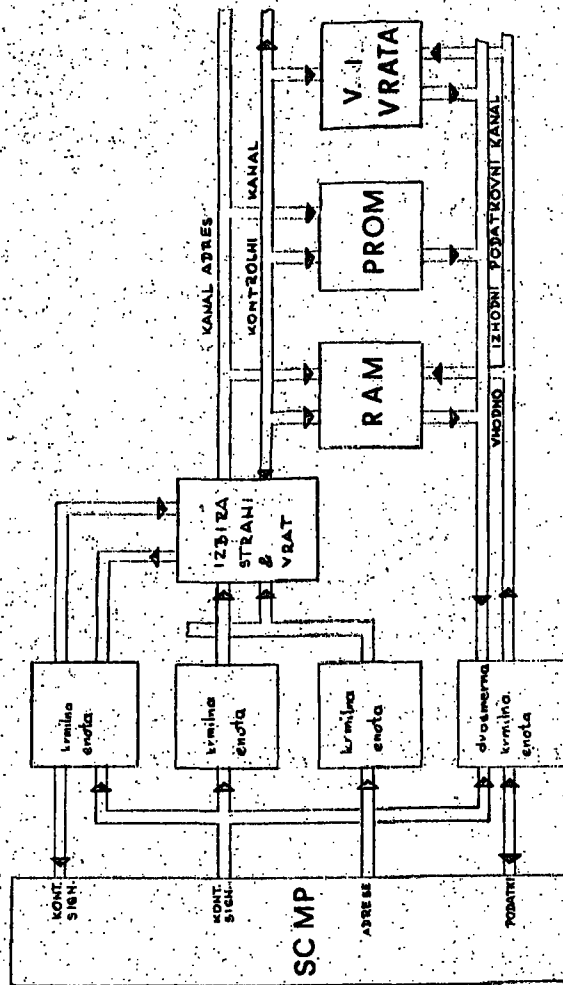
ZAKLJUČEK

Članek je v prvi vrsti namenjen začetnikom, ki lahko na podlagi opisa izdelajo podoben sistem. V naslednjem članku pa nameravamo ta članek kompletirati s podrobnejšimi opisi izvedbe enostavne komunikacije operater-računalnik s pomočjo konzole ter podati vse potrebne specifikacije elementov in distributerjev. Tako upamo, da bo članek koristil ne samo strokovnjakom, ki se pro-fesionalno ukvarjajo z mikro-računalniki, temveč tudi številnim entuziastom, ki želijo поблиže spoznati to mo-derno vejo elektronike.

LITERATURA

SC/MP Manual

SC/MP Technical Description



Slika 6. Bločna shema cenenege mikroračunalniškega sistema

moč udruženih mikroprocesora

U današnjem računarskom svijetu je veoma zastupljena tehnika multiprogramiranja. Multiprogramiranje znači, da se više od jednog programa istovremeno nalaze u radnoj memoriji računara te da centralni procesor može skakati iz jednog programa u drugi. Time je izključena mogućnost da centralni procesor čeka dok se kompletira određena ulazno/izlazna akcija. Ako program koji se izvršava, zahtijeva ulazno/izlaznu akciju, centralni procesor prelazi na izvršavanje onog programa koji ima kompletiranu ulazno/izlaznu akciju. Ako pri tome sudjeluje više procesora mnogi problemi će biti jednostavno riješeni - to je multiprocesiranje. Multiprocesorski sistem omogućava i sljedeći primjer: ako program koji se izvodi na jednom od procesora, zahtijeva ulazno/izlaznu akciju, izvođenje se prekida i može biti nastavljeno na drugom procesoru pošto je ulazno/izlazna akcija kompletirana. Uopšte, multiprocesorski sistemi mogu biti organizirani na dva načina: hijerarhijski i simetrični. U hijerarhijskim sistemima jedan procesor (nije potrebno da je isti kao preostali) kontrolira preostale. Primjer takve velike hijerarhijske multiprocesorske mašine je CDC 6600.

Mikroprocesori, sa vrlo pristupačnom cijenom, lako djeluju u tandemu tako da biraju poslove po nekom prioritetu. Ukoliko se pokaže da određena jedinica neće uspjeti izvršiti određeni posao, uključuju se rezervne jedinice. To nije slučaj u većini velikih multiprocesorskih sistema. Nekoliko jedinica u mikro sistemima su uvijek na raspolaganju. Sistem može upotrijebiti jedan procesor za upravljanje dodjeljivanjem memorije, drugi za komuniciranje te ostale za prevodjenje i izvršavanje programa, tabeliranje poslova (job scheduling) itd. Takav sistem pokazuje performanse ekvivalentne sistemu sa cijenom reda milion dolara, za svega nekoliko hiljada dolara.

Iako brzina pojedinog mikroprocesora može biti ograničena, svaka jedinica jednostavno djeluje nesinhronizirano sa drugima što obezbjeđuje jednostavno, brzo i moćno procesiranje. Odnos između brzine memorija i brzine mikroprocesora diktira novu arhitekturu multimikroprocesorskih sistema.

Mikro računari imaju nove funkcije, novu moć, nove brzine komponenti te mnogo novih sistemskih sposobnosti. Mikro računari se upotrebljavaju u najrazličitije svrhe - kontroliraju vrlo brze procese kao na primjer grafične displeje, plotere, radare i slično. Razmatranja o multimikroprocesorskim sistemima su vrlo slična razmatranjima o multiminiprocessorskim sistemima. Razlika se ogleda u tri sljedeće karakteristike: dijeljena memorija, niska cijena i mikroprogramske sposobnosti. Većina mikroprocesora ima ograničenu brzinu izvođenja programa. Dakle propusnost mikroručunarskog sistema ograničena je brzinom procesora i ne brzinom memorije. To znači da nekoliko mikroprocesora mogu dijeliti istu memoriju bez ozbiljnog smanjenja individualne brzine pojedinog procesora. Niska cijena mikroprocesora (u odnosu na cijenu memorije i periferije) doprinosi njihov veći broj u sistemu. To reducira važnost maksimalnih sposobnosti mikro procesora. Multiprogramiranje dozvoljava upotrebu specijalnih instrukcija koje zadovoljavaju odgovarajuće kontrolne zahtjeve. Sljedeće osobine multimikroprocesorskih sistema nam

dovoljno ubjedljivo govore o njihovoj korisnosti.

1. Propusnost sistema često raste direktno sa brojem procesora, dok cijena raste neznatno.
2. Dijeljenje sistemskih resursa eliminiira dupliranje pojedinih naprava što je često slučaj kod klasičnih sistema.
3. Dijeljeni resursi omogućavaju direktni pristup do podataka što inače zahtijeva prenos podataka od jednog sistema na drugog.
4. Više procesora omogućavaju nižu cijenu
5. Rezervni procesori mogu biti uključeni u sistem da zamijene one koji su prezaposleni ili pokvareni.
6. Trenutno nezaposlen procesor može biti upotrijebljen za izvršavanje sljedećeg zadatka, eliminirajući tako čekanje na specializirani procesor.
7. Pojedini procesori mogu imati fiksirane specijalne funkcije. Asimetrični (hijerarhijski) sistemi se obično upotrebljavaju u specializiranim aplikacijama gdje je tip, učestanost i relativna važnost zadataka unaprijed poznata. Za primjer jedan procesor može izvršavati sve ulazno/izlazne operacije, drugi može imati sposobnosti izvršavanja aritmetičnih operacija sa plivajućim zarezom, treći održava datoteke itd.
8. Specializacija pojedinih procesora može biti izvršena putem mikroprogramiranja ili izborom određene hardverske arhitekture (broj registara, procesiranje steka itd.)
9. Svi procesori, memorije i periferne naprave mogu biti multiplexirane nad jedinim podatkovnim vodilom. To u velikoj mjeri doprinosi vrlo niskoj cijeni takvih sistema.
10. Asimetrični procesor sa multiplexiranim vodilom doprinosi nižoj cijeni i jednostavnijoj organizaciji multimikroprocesorskih sistema od bilo kojih drugih.

Digital design, august 1977

Stacked Microprocessors

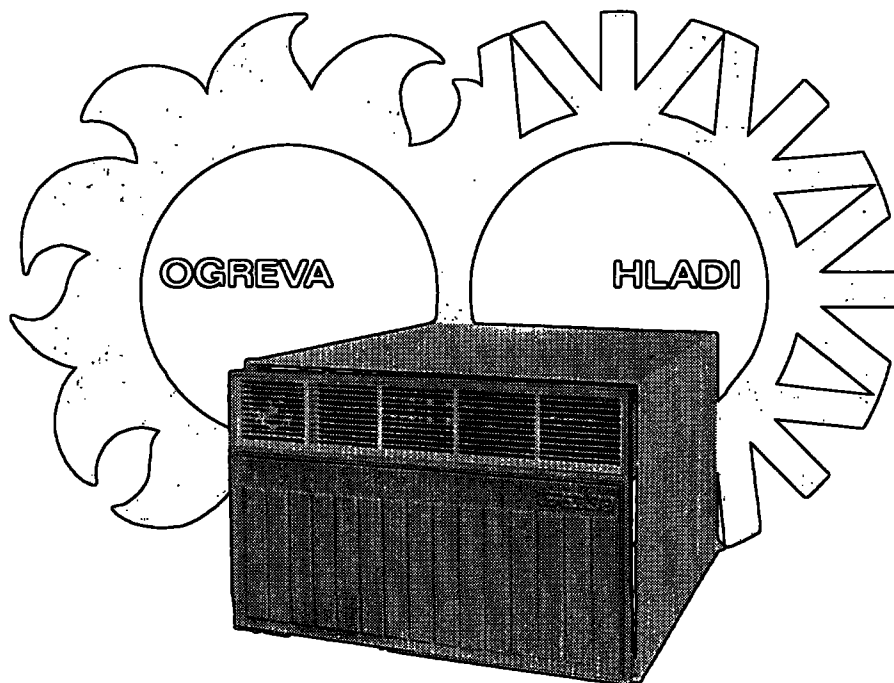
Stobodni prevod M. Kovačević

študentska vprašanja

Uspešna ekakurzija študentov računalništva in informatike na simpozij INFORMATICA 77

Simpozij, seminarji, razstava. Osrednje jugoslovansko srečanje teoretikov in praktikov iz področja računalništva in informatike, srečanje je letos organiziralo Slovensko društvo Informatika v sodelovanju z Institutom "Jovan Štefan" in Fakulteto za elektrotehniko. Skratka, nekaj fantastičnega. In že drugo leto smo se ga udeležili študentje računalništva in informatike. Nekaj še bolj fantastičnega.

Koliko lahko pomeni tako srečanje mladomu študentu računalništva in informatike pri širjenju njegovega obzorja, ve le on sam. Na simpoziju zvemo ne le samo za vsa mogoča (in nemogoča) področja računalniške znanosti in uporabe, ampak tudi za večino najnovejših dosežkov in



VEDNO PRIJETNO POČUTJE
Rešitev vseh problemov klimatiziranja:

KLIMATIZER TOBI 32
KLIMA OMARE KO

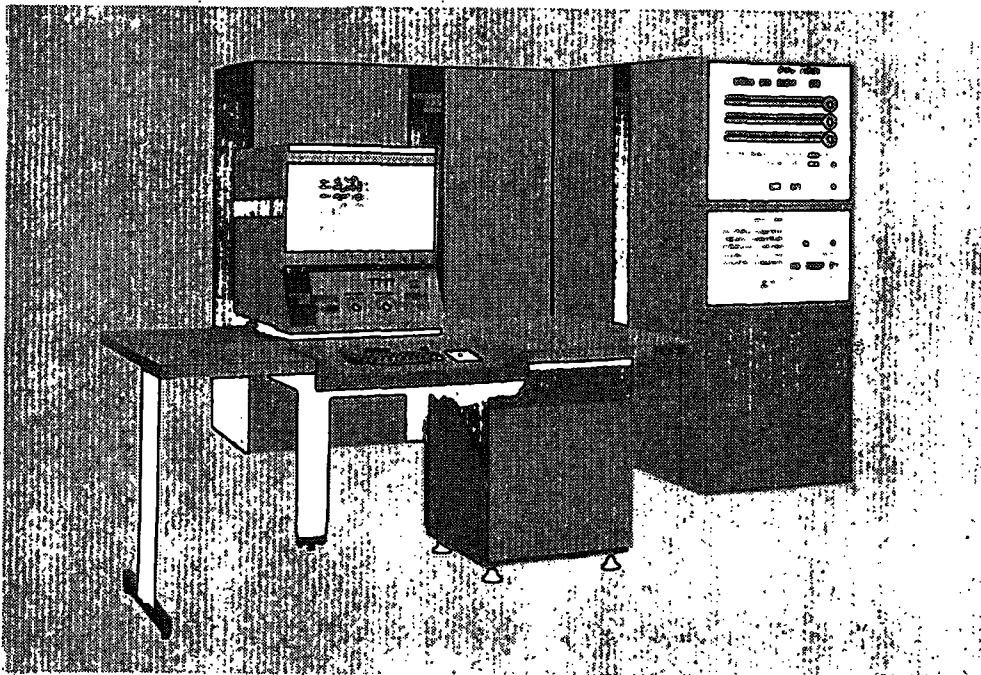
- Samodejno uravnavanje temperature v prostoru
- Enostavna montaža in vzdrževanje
- Servisna mreža in rezervni deli na celotnem področju SFRJ

INFORMACIJE O PRODAJI IN INŽENIRINGU:

EMO
63000 CELJE
Mariberska 86
tel.: (063) 23-921

ISKRA - TOZD INŽENIRINGI
61000 LJUBLJANA
Kotnikova 6
tel.: (061) 312-322

PORODICA FACOM FIRME FUJITSU



Radeći bez mnogo buke, ali marljivo posljednje četiri godine, FUJITSU je zajedno sa svojim zastupnikom ZPR-om (Zavod za primjenu elektroničkih računala) uspješno sklopila ugovore za više od 50 FACOM računala u Jugoslaviji.

Izenadjeni? Ne morate biti. FUJITSU, povrh toga što je vodeći proizvođač sistema za elektroničku obradu podataka na Japanskom tržištu, vrlo brzo preuzima jedno od vodećih mjesta i na svjetskom tržištu. Tajna uspjeha firme FUJITSU je u tome što ima vodeću tehnologiju u kombinaciji sa velikom pouzdanošću sistema i dobro organiziranom službom za održavanje i stručnu pomoć. Ne smijemo zaboraviti ni konkurentne cijene koje će vam dati najbolji mogući odnos cijena/performance.

Ako razmišljate o uvođenju elektronske obrade podataka u vašoj organizaciji ili želite da poboljšate svoj sadašnji sistem, obratite se predstavnicima firme FUJITSU da vas upoznaju sa svim novostima.

FUJITSU proizvodi sve - sastavne dijelove, memorije, off i on-line uređaje za prikupljanje podataka, inteligentne terminale, micro procesore i micro računala, malih, srednjih, velikih i super velikih kompjutera, uključujući najsnažnije kompjutore za opću svrhu koji se mogu kupiti na tržištu.

FUJITSU je poznata i u području telekomunikacija. To je razumljivo zbog toga što je FUJITSU jedan od vodećih proizvođača telefona i telekomunikacija u Japanu. FUJITSU je u Jugoslaviji izabrana da snabdije i pomogne kod razvoja najveće on-line real-time mreže koja je do sada ugovorena, uključujući oko 300 terminala. Mislimo da biste sebi i svojoj organizaciji učinili mnogo, ako saznate više o tome što Vam firma FUJITSU može ponuditi.

Servisni centri i uredi su u Ljubljani, Mariboru, Beogradu i Zagrebu

ZPR
ZAGREB, Savska 56
tel. 518-706
Telex 21689 YU ZPR FJ



LJUBLJANA, Topniška 45
tel. 311-059

FUJITSU LIMITED
Communications and Electronics

smeri bodočih raziskav. To pa se naravnost idealno dopolnjuje z v šoli pridruženim znanjem. Svojevrsten čar Informatiki daje tudi Bled s svojo enkratno jesensko lepoto in skrivnostnim jezerom (kjer se zadnje čase pojavlja pošast).

Bilo nas je kakšnih 20-30 študentov računalništva, nekaj jih je bilo tehničnih sekretarjev, vsi smo pomagali pri organizaciji, pošiljanju vabil itd., ter se s tem vsaj malo oddolžili za visoko kotizacijo. Študentje računalništva se ob tej priložnosti pristrčno zahvaljujemo prof. Železnikarju, mgr. Rajkoviču, prof. Hodžarju, doc. Pipanu in vsem ostalim, ki so nam pomagali. Mimogrede lahko povemo, da je tu in tja pri organizaciji malo zaškripalo, kar pa gre v precejšnji meri na račun naše nezkušenosti (mimogrede - je muza računalništva že znana, ali je še v pleničkah). Nekaj je pri vsej stvari res važnega. Za večino, vsaj za tiste bolj prizadevne, je teh nekaj dni na Bledu minilo v ognjemetu dobrih referatov in zanimivih pogovorov.

To je ta pravo!

Matjaž Gams

novice in zanimivosti

Poštovani čitatelji

U želji, da vam što više približimo ovu rubriku, pozivamo vas na saradnju pri objavljivanju novosti i zanimjivosti.

Želimo da nas obavještavate o zanimljivostima i novostima iz vaše radne organizacije (novi računarski sistemi, programski paketi, organizacijski pristupi, organizacijske strukture i slično). Pišite, također, o teškoćama sa kojima se susrećete (pomanjkanje opreme, materijala, specifičnost problematike koju rješavate i slično). U toj rubriki možete predlagati i rješenja svojih problema.

Radnim organizacijama, koje žele upozoriti računarsku javnost na svoje proizvode, predlažemo da to učine preko ove rubrike.

Ako ste otkrili zanimljiv članak iz neke druge revije, molimo da naš ukraćko obavijestite o njegovoj sadržini i izvoru (u rukopisu ili na formularu).

Uz vašu pomoć ćemo ovu rubriku oblikovati onako, kako to sami želite.

Uredništvo

Accutest Corporation iz ZDA nudi nov sistem 7700 ROM/PROM/EPROM programator in tester. Le-ta je popolnoma integriran, zelo hiter z računalniško krmljenim programiranjem, z zmožnostjo testiranja in z zmožnostjo procesiranja. Tako je programiranje pomnilnikov in testiranje združeno v enoten sistem, ki je pri-

meren tako za proizvajalce pomnilnikov kot za uporabnike. Sistem 7700 vsebuje tastaturo s televizijskim zaslonom in je preprost za programiranje in upravljanje. Tudi pri višjih hitrostih (nad 10 MHz) je sistem izredno točen in predstavlja izredno dober pripomoček za izčrpno preizkušanje sodobnih MOS in Bipolarnih bralnih pomnilnikov.

Mikro računar koji zadovoljava sve Military specifikacije i ima visoke performanse, je objelodanila Norden Division of United Technologies pod imenom LSI-11M. To je 16-bitni mikro računar baziran na arhitekturi poznatog DIGITAL LSI-11. LSI-11M je četiri puta brži od svog komercialnog imenjaka. LSI-11M sabirnice predstavljaju brz dvosmerni medjuspoj koji povezuje CPU memoriju i I/O. Naprave povezane na te sabirnice mogu komunicirati sa bilo kojom drugom napravom bez posredovanja CPU. Brzina I/O prenosa podataka je 833k riječi na sekundu. Cijena je \$2000. Za detaljnije informacije obratite se na: Norden Division of United Technologies, Norwalk, CT 06856. (203)838-4471.

Completni Mikro računar/Flopy disc sistem sa procesorom 6800 je objavila firma San Diego-Electronic Product Associates, Inc. Sistem sadrži 8k riječi RAM memorije, 1K MIK-BUG monitorski sistem i audio case, TTY, RS232C medjuspoj te 5V (20 A) in ± 12V (1 A) ispravljač Flopy Disc sistem je IBM kompatibilan i može biti u jednostrukoj ili dvostrukoj izvedbi. Sistem ima obezbijedeno napajanje, medjuspoj za mikroračunarski sistem i elektroniku neophodan za disk drajver. Vsak disk drži 1/4 miliona bajtova. Cijena za mikroračunarski sistem je \$1878. Jednostruki Flopy Disc sistem \$2595 i dvojni Flopy Disc sistem \$3295. Na raspolaganju je i programska oprema: Fortran IV, BASIC, Assembler, Editor i Flopy Disc operacijski sistem. Patti Neuman Director of Marketing Electronic Product Associates, Inc. 1157 Vega Street San Diego, CA 92110 (714)276-8911.

Computer Warehouse je objavila da ima na raspolaganju katalog za entuzijaste koji se privatno bave računarskom tehnikom (Personal Computing) te za korisnike malih sistema. Na 48 stranica kataloga je prikazanih pet različitih mikroračunarskih kit-sistema i obilje upotrijebljene periferije kao npr. CRT terminali, tastatura/printer terminali, video terminali, drajveri za kasete, drajveri za disk i printeri. "All About Hobby Microcomputer Systems" je specijalni izvještaj koji posreduje informacije od značaja za kompjuter-amatere. Usporedba raznih detalja za najpopularnije mikro procesore je također tema tog izvještaja (cijena, osobine, programska oprema, konfiguracija itd.) Katalog je na raspolaganju za \$1. Computer Warehouse Store, Dept. C, P.O. Box 68, Kenmore Station, Boston, Mass., 02215.

Novi Grafični terminal firme Hewlett-Packard pokazuje vrlo velike mogućnosti i dobre osobine uz nisku cijenu. HP 2648A je grafični CRT terminal sa kontrolerom na osnovu mikroprocesora koji upotrebljava tehnologiju skeniranja rastera (raster scan), ima neodvisnu grafičnu i alfanumeričnu memoriju, automatično iscrtaavanje tabeliranih podataka itd. Terminal, također, ima mogućnost zahvatanja podataka te komunikacijske sposobnosti HP 2645 A displej jedinice. Osnovna cijena za HP 2648A grafični terminal je \$5500; ako je opremljen sa drajverom za kasetu \$7100. Za detaljnije informacije pitajte: Hewlett-Packard Co., 1501 Page Mill Road, Palo Alto, CA 94304. (415)493-1501.

Za mikro sisteme sa procesorom 8080 se mogu dobiti kompilatori za FORTRAN IV po dosta ugodnoj cijeni (\$500), zvaní FORTRAN 80. Kompilator u potpunosti zadovoljava ANSI standarde izuzev dvojnè preciznosti i kompleksnih podatkovnih tipova. FORTRAN 80 uključuje tri tipa podataka: logične (jedan bajt), cijele (dva bajta) i realne (četiri bajta, plivajućí decimalni zarež). Kompilator je jedno-prolazni (one-pass) i zauzima nešto manje od 12K bajtova memorije. U toku prevodjenja zahtijeva još 6K bajtova. Nalagač sa poveziivačem (linking loader) je u istom paketu i ima sposobnost naganja premijestljivih programa. Dodatne osobine FORTRANA 80 uključuju optimizaciju mašinskog koda, izrazi sa miješanim argumentima i sve standardne Fortranske funkcije za realne i cijele brojeve. Paul Allen, 819 Two Park Central Tower, Albuquerque, N.M. 87108 (505) 256-3600.

Za leto 1978 napoveduje Motorola nov mikro procesor v družini 6800 z oznako MC 6809, ki bo imel več notranjih registrov kot procesor MC 6800 in razširjeno množico ukazov. Izdelan bo s posebno tehnologijo in z eno napajalno napetostjo. Proizvajalec je prepričan, da bo ta procesor močnejši od vseh dosedanjih 8-bitnih procesorjev (vključno z Z 80) in močnejši od večine 16-bitnih procesorjev.

Izgradnja nove FUJITSU network arhitekture proširuje primjenu kompjuterskog sistema

Nedavno je FUJITSU objavila dovršenje FUJITSU NETWORK ARCHITECTURE logički strukturirane i usmjerene porođice On-line kompjuterske mreže. Već dosada je FUJITSU razvila brojne On-line mrežne sisteme, vodeći pri tom računa da je standardizacija komponenti kompjuterskog sistema esencijalna za veću primjenu kompjutera. Prije, FUJITSU je bila prva japanska kompanija koja je uvela postupak kontrole povezivanja podataka visokog nivoa, pod nazivom HDLC.

U posljednje vrijeme upotreba kompjutera napredovala je od hatch-obrađe do jednostruke globalne mreže (single-host networks) i višestruke globalne mreže (multiple-host networks).

FNA proširuje primjenu i korištenje kompjuterskih sistema u svijetkim razmjerima i izlazi u susret zahtjevima za poboljšanje cijene i efikasnosti, te pouzdani servis tako važan na današnjem vrlo konkurentnom tržištu.

Svaka mreža u sklopu FNA sadrži brojne komponente: globalne (host) kompjutere, komunikacijske procesore, male centralne sisteme (sub-hosts), podsisteme (cluster controllers) i kompatibilne terminale koji su efektno spojeni po principu jednostrukih veza (single-line discipline). U osnovi, funkcije prenosa podataka unutar različitih komponenta odvojene su od funkcija obrade podataka, te na taj način sistem može preuzeti mnogo primjena na istoj liniji.

Zapisnici (protokoli) i slijed operacija za informacione transmisioné jedinice preko komunikacijskog sistema ipak su jasno definirani.

FNA smanjuje i zahtjeve, te obim posla za globalni kompjutor, a troškove dodatnih linija i dupliciranje programiranja svodi na minimum. Moćnost postojećih sistema se također proširuje bez većih dodatnih troškova. Kompatibilna kompjuterska mreža kao što je 230-5

Serijska, Serijska 230-8, te "M" Serijska mogu biti priključena na FNO preko postupka povezivanja kontrole podataka visokog nivoa (high-level data link control procedures), a mogući su i drugi postupci prenosa kontrole.

FNA također osigurava pristup iznajmljenim linijama, telefonskim centralama, telegraf-telefon komutacionom sistemu, "data highways" (linija velikih brzina instaliranih u tvornicama), te optičkim komunikacijama.

Oslanjajući se na provjerena pravila Internacionalne Organizacije za Standarde (ISO) i Comité Consultatif International Télégraphique et Téléphonique (CCITT), FNA je kompatibilan sa svim mrežama izvan i unutar Japana. Štoviše, ona sadrži Mrežnu Konstrukciju "M" Serijska (MSNA-"M" Series Network Architecture), a planira se njeno uravnavanje sa Mrežnom Konstrukcijom Prenosa Podataka (DCNA-Data Communication Network Architecture).

Zilog daje na tržište Z-80 diskovni mikroracunalski sistem, ki ima nabor 158 ukazov, obseg 64 K zlogov RAM, PROM ali EPROM pomnilnika, dvojne diskete z 600 000 zlogi, vmesnik za CRT ali TTY in dvoje paralelnih vrat za enostavne periferne vmesnike. Programska oprema vsebuje monitor na PROM-u, makrozbirnik, vzdrževanje datotek, urejevalnik, uporabniške rutine in rutine za preizkušanje programov. Na voljo je tudi jezik BASIC in kasneje MCS-COBOL ter PL/Z.

Intel Corp., Santa Clara, Calif., ZDA je izdal tvrdkama Signetics, ZDA in N.V. Philips, Nizozemska dovoljenje za proizvodnjo (dodatni dobavitelji) mikroracunalskih 8048 in 8035 ter pripadajoćih proizvodov. V zameno bodó tvrdki Intel dostopni Philipsovi patenti za MOS integrirana vezja.

iz časopisov

... Časopisi jugoslovenskog karaktera, sa rezimè oñi na stranim jezicima, omogućuju protok naućnih informacija i cirkulaciju naućne misli po ćitavoj zemlji, kao i razmjenu s inostranim časopisima. Nedovoljno je ipak razvijeno izdavanje i širenje sekundarnih publikacija, namijenjenih naućnoj javnosti u svijetu, u cilju signaliziranja domaćih primarnih (originalnih) izvora. Tako su naši znanstveni radovi vrlo slabo registrirani u svjetskim referalnim katalogima, kao što je npr. "Science Citation Index".

Znanstvene publikacije i skupovi služe ne samo protoku informacija nego i stroj valorizaciji naućnoistraživaćkih radova putem javne diskusije i kritike. To je danas dosta živa i razgranata aktivnost, kojom se u izvesnoj mjeri dobiva javna ocjena radova, iako sa općim stanjem znanstvene kritike ne možemo biti zadovoljni.

(Osnove za naućnu politiku u SFRJ, Zagreb, 1977)

Computer Club Europe, Postfach 11 08 68, 6100 Darmstadt
Izdaja časopis The Journal of Computer Encephalologia
and Neurosis. Prva izdaja je izšla v marcu 1977, predvi-
denih je 12 števk na leto. V prvi izdaji najdemo med
drugim članek z naslednjo vsebino: Supermožgani -
Igra v basicu, enostavni kasetni vmesnik, problemi ob-
delave tekstov, izboljšano programiranje, metode adre-
siranja, povezava tastature na teleprinterko z nanko z
60 mA, kaj je I2L, strukturirano programiranje, itd.
Prispevki so napisani v angleščini ali nemščini.

Nov časopis

S februarjem 1978 napovedujejo izhajanje novega dvo-
mesečnika "Computer Communications". Mednarodni
strokovni časopis bo namenjen izhajanju v praksi in
bo zanimiv za vse strokovnjake iz področja telekomuni-
kacij in računalništva. Sedež uredništva bo v Angliji:
(IPC Science Technology Press Ltd; IPC House, 32 High
Street, Guildford, Surrey, England GU1 3EW).

Automatika - Jugoslovanski časopis za avtomatizacijo
št. 5-6. letnik 1976 - vsebina:

- Anton P. Železnikar: O informatični definiciji strategije
- L. Gyergyek, S. Hanžel, E. Grobler: Optimizacija iz-
delava plana za foriniranje vlakov
- H. Pfüller: Numeričke integracijske metode i analiza
prilaznih pojavov u usmerivačkim sklopov-
ima
- B. Zajc, M. Vehovec: Topološka formulacija identifi-
kacije nelinearnih dvopolov
- N. Babunović, M.M. Millić: Jedno poboljšanje nemodi-
fikovanog brojenovog algoritma za rešavanje
nelinearnih rezistivnih mreža
- V. Potić, D. Hristović: Asociativna struktura za adre-
siranje spoljnih memorija
- S. Kostić, B. Čirilov, M. Šmelcerović: Primena Si
poluprovodničkog detektora za merenje komp-
leksnih smeša beta aktivnih izotopa
- V. Izgan, H. Barac, S. Tonković, Z. Stare: Jedna meto-
da analize EKG-a pomoću električnog računala
- P. Bešnjaković: Jedno rešenje problema preciznog me-
renja mrežne frekvencije
- M. J. Milčić: Funkcija osetljivosti aktivnih RC filtera
- S. Amoić: Analiza onosmernoga delovanja I²T celice
- H. Rusjan: Verige 3-členov RC s pritisnjeno napetostjo
in različnim bremenom
- Millić R., Stojić, Srbijanka Lazarević: Osobine stabilnos-
ti sistema s amplitudno ograničenim upravlja-
njem
- R. Murn, T. Pisanski: Diagnostika večkratnih napak v
drevesnih kombinacijskih vezjih

DIGITAL DESIGN - The magazine of digital systems
July 1977 - iz vsebine:

- E.A. Ross: Digital panel meters,
- S. Pellerin: Graphic display systems - adding dimensi-
on to computer I/O
- Micronotes: 8080 designers get second generation
system development tools.
- : Eliminating decoders: addressing compo-
nents in a 6800 Microcomputer.

literatura in srečanja

KONČAN SIMPOZIJ INFORMATIKA 77

V dneh od 3. do 7. oktobra 1977 je bil na Bledu 12. jugo-
slovanski mednarodni simpozij o obravnavanju podatkov
INFORMATIKA 77, ki ga je organiziralo Slovensko društvo
Informatika v sodelovanju z Institutom "Jožef Stefan" in
Fakulteto za elektrotehniko, Ljubljana. Od ponedeljka do
petka je preko 400 udeležencev iz 16 držav sledilo preko
250 referatov domačih in tujih strokovnjakov s področja ra-
čunalništva in informatike. Poleg tega so potekali tudi semi-
narji izbranih poglavij računalniške znanosti in tehnike. Naj-
bolj žgočim problemom na področju računalništva so bile na-
menjene posebne debate za okroglimi mizami. Prireditelj je
spremljala tudi razstava računalniške literature.

Namen tega srečanja ni bil le v izmenjavi mnenj in iskanju
pristopov na ozkem strokovnem področju. Problemi računalni-
štva in informatike že imajo širše družbene razsežnosti, kar
je v svojem otvoritvenem govoru poudaril tudi tov. Jože
Zakonjšek.

Z ustavo, družbeno-političnimi in drugimi dokumenti so pri
nas določeni splošni cilji in dani osnovni pogoji razvoja
družbenega sistema informiranja. Določila teh dokumentov so
jasna v opredelitvi, da se z družbenim sistemom informiranja
zagotavlja usklajeno evidentiranje, zbiranje, obdelavo in iz-
kazovanje podatkov in dejstev, ki so pomembna za spremlja-
nje in usmerjanje družbenega razvoja, ter dostopnost do teh
podatkov in dejstev. Pri tem je družbeni informacijski sistem
opredeljen kot oblika organizirane aktivnosti v kateri si bodo
delavci in delavni ljudje v vseh oblikah samoupravnega orga-
niziranja, zagotovili in izmenjevali resnične, sprotno in celo-
vite informacije, ki so pogoj za njihovo odločanje v združe-
nem delu in v družbi, na vseh ravneh njene samoupravne orga-
niziranosti.

V zvezi z družbenim sistemom informiranja je potekala tudi
ena izmed debat za okroglo mizo. Splošna ugotovitev je ta,
da z obstoječim stanjem na tem področju ne moremo in ne
smemo biti zadovoljni. Družbena pozornost je predvsem
usmerjena k novim nabavam in ekonomiziranju na novi infor-
macijski opremi (kar je tudi pomembno, vendar ne najbolj)
in nezadostno usmerjena k njeni uporabi v okviru celovitega
informacijskega sistema naše družbe. Vse prepogosto se ce-
lotni problem razvoja takega sistema obravnava kot tehnično
tehnoška akcija in ne kot družbeno odvisna naloga najširše-
ga pomena in interesa. Za ustrežno reševanje problemov v
tej zvezi je potrebna širša družbena akcija v smislu racional-
ne delitve in koordinacije nalog in dela.

Tudi ostale debate za okroglo mizo so bile deležne velikega zanimanja. V tem okviru so bili obravnavani tudi problemi vzgoje in izobraževanja, proizvodnje računalniške opreme pri nas in problem uvajanja mikroročunalniške tehnologije v proizvodne procese.

V zvezi z vzgojo in izobraževanjem je tekla beseda predvsem o nujnosti izgradnje celovitega vzgojno-izobraževalnega sistema na področju računalništva in informatike v smislu usmerjenega izobraževanja. Razprava o proizvodnji računalniške opreme v Jugoslaviji je pokazala da bo treba organizirano pristopiti k razreševanju dileme kaj uvažati in kaj delati doma in dileme med proizvodnjo na osnovi tujih licenc in proizvodnjo na osnovi domačega znanja tj. lastnega razvoja. Ne le razprava o uvajanju mikroročunalniške tehnologije v proizvodne procese, ampak tudi razprave o mestu in vlogi računalništva v gospodarstvu nasploh so terjale večji doprinos k povečanju učinkovitosti gospodarstva. V tej zvezi se je pokazal problem mesta in vloge računskih centrov v organizacijah združenega dela. Zaradi pomembnega in občutljivega mesta centra je to pereč problem, ki se ga bo treba celovito lotiti tudi iz stališča izobraževanja in kadrovanja.

Ob zaključku simpozija Informatika 77 lahko ponovno ugotovimo izreden pomen takih srečanj za široko izmenjavo mnenj, preverjanje rezultatov dela in za skupno iskanje pristopov za reševanje problemov s področja računalništva in informatike.

V. Rajković

DECEMBER

5-7 dec. Gaithersburg, Maryland, ZDA

WINTER SIMULATION CONFERENCE

Organizator: National Bureau of Standards-US Department of Commerce

Informacije: Dr Robert G Sargent (Gen Chm), Syracuse University, Syracuse, NY 13210, USA

5-9 dec. Rocquencourt, Francija

THIRD INTERNATIONAL SYMPOSIUM ON COMPUTING METHODS IN APPLIED SCIENCES AND ENGINEERING

Organizator: IRIA-LABORIA

Informacije: IRIA, Service des Relations Extérieures, Domaine de Voluceau, Rocquencourt, BP 5, 78150 Le Chesnay, France

5-9 dec. Versailles, Francija

THIRD INTERNATIONAL COLLOQUIUM ON METHODS OF SCIENTIFIC AND TECHNICAL CALCULATION

Organizator: IRIA

Informacije: IRIA, Service des Relations Extérieures, Domaine de Voluceau, Rocquencourt, BP 5, 78150 Le Chesnay, France

LETO 1978

23-25 jan. Tucson, Arizona, ZDA

FIFTH ANNUAL ACM SIGACT-SIGPLAN SYMPOSIUM ON PRINCIPLES OF PROGRAMMING LANGUAGES

Organizator: ACM

Informacije: Prof Larry Reeker and Prof David Ripley, Department of Computer Science, University of Arizona, Tucson, AZ 85721, USA

23-27 jan. Valparaiso, Čile

FIFTH PANEL DISCUSSION ON COMPUTATION TOPICS

Organizator: Computatio Centre-Valparaiso Catholic University
Informacije: Aldo Migliaro, Director, Centro de Computación, Universidad Católica de Valparaiso, Casilla 4059, Valparaiso, Chile

7-9 Februar, Brussels, Belgija

INTERNATIONAL CONFERENCE ON TRANSNATIONAL DATA REGULATION

Organizator: Online Conferences Limited

Informacije: Jean Seago, Online, Cleveland Road, Uxbridge UB8 2DD, Middlesex, UK

13-15 feb. Liège, Belgija

SYMPOSIUM ON COMPUTER NETWORK PROTOCOLS

Organizator: The University of Liège

Informacije: A Danthine, Symposium on Computer Network Protocols, Avenue des Tilleuls 49, B-4000 Liège, Belgium

21-23 februar, Detroit, Michigan, ZDA

ACM COMPUTER SCIENCE CONFERENCE

Organizator: ACM

Informacije: Orrin E. Taulbee, ACM Computer Science Employment Register, Department of Computer Science, University of Pittsburgh, Pittsburgh, PA 15260, USA.

21-23 Februar, Le Chesnay, Francija

AFCEP/IRIA CONGRESS: RECOGNITION OF FORMS AND TREATMENT OF IMAGES

Organizator: AFCEP, IRIA

Informacije: IRIA, BP 105, 78150 Le Chesnay, France.

21-23 Februar, London, Velika Britanija

FIRST EUROPEAN CONFERENCE ON PRAGMATIC PROGRAMMING AND SENSIBLE SOFTWARE

Organizator: Online Conference Limited

Informacije: Jean Seago, Online, Cleveland Road, Uxbridge UB8 2DD, Middlesex, UK.

23-24 feb. Detroit, Michigan, ZDA

SIGCSE/CSA SYMPOSIUM

Organizator: ACM Special Interest Group on Computer

Science Education, The Computer Science Association
Informacije: Kenneth Williams, SIGCSE/CSA Symposium Chairman, Computer Science Group, Western Michigan University, Kalamazoo, MI 49008, USA

7-10 marec Paris, Francija

INTERNATIONAL CONGRESS ON THE CONTRIBUTION OF COMPUTERS TO THE DEVELOPMENT OF CHEMICAL ENGINEERING AND INDUSTRIAL CHEMISTRY

Informacije: Monsieur Le Professeur A Brusset, Congrès International 1978, Société de Chemie Industrielle 28, rue Saint Dominique F-75007 Paris, France

marec, Grenoble, Francija

IFIP WC5.2 WORKING CONFERENCE ON ARTIFICIAL INTELLIGENCE AND PATTERN RECOGNITION IN COMPUTER-AIDED DESIGN

Organizator: IFIP WC5.2

Informacije: Mr Jean-Claude Latombe, ENSOP, IFIP WC5.2, BP 15, 38040 Grenoble, France.

15-17 marec Tampa, Florida, ZDA

ELEVENTH ANNUAL SIMULATION SYMPOSIUM

Organizator: ACM-SIGSIM, EE-CS

Informacije: ACM-HQ, 1133 Avenue of the Americas, New York, NY 10036, USA

27-30. marec, New York, ZDA

INTERNATIONAL CONVENTION AND EXHIBITION OF THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS

Organizator: IEEE

Informacije: J.H. Shumacher, IEEE, 345 East, 47th Street, New York, NY 10017, USA

14-17 marec, Toulouse, Francija

SYMPOSIUM ON MEDICAL INFORMATICS

Informacije: Prof. Francois Begon, CTI, Domaine de Voluceau, BP 105, 78150 Le Chesnay, France

29-31. marec, Lancaster, Velika Britanija

COMPUTERS IN HIGHER EDUCATION

Informacije: Dr E. Tagg, S Dept. of Mathematics, University of Lancaster, Lancaster LA1 4YI, UK

3-5 april, Palo Alto, California, ZDA

FIFTH ANNUAL SYMPOSIUM ON COMPUTER ARCHITECTURE

Organizator: ACM-SIGARCH, IEEE-CS

Informacije: ACM HQ, 1133 Avenue of the Americas, New York, NY 10036, USA.

11-13 april, Stocholm, Švedska

SEMINAR ON MINIS IN TERMINAL-BASED SYSTEMS

Organizator: On line Conference Limited

Informacije: Jean Seago, Online, Cleveland Road, Uxbridge UB8 2DD, Middlesex, UK.

11-22 april, Madrid, Španija

INTERGOVERNMENTAL CONFERENCE ON STRATEGY AND POLICIES FOR INFORMATICS

Organizator: IBI, UNESCO

Informacije: IBI Headquarters, POB 10253, 00144 Rome Italy.

9-12 maj, London, Velika Britanija

EUROCOMP 78

Organizator: Online Conferences Limited

Informacije: EUROCOMP 78, Online, Cleveland Road, Uxbridge UB8 2DD, England

22-25 maj, Taormina, Sicilija

SIXTH INTERNATIONAL CODATA CONFERENCE

Organizator: CODATA

Informacije: CODATA Sekretariat, 51 Boulevard de Montmorency, 75016 Paris, France

maj, Bled, Jugoslavija

CONFERENCE ON COMPUTERS IN BANKING AND FINANCE

Organizator: IAG

Informacije: IAG HQ, Paulus Potterstraat 40, Amsterdam-1007, The Netherlands

11-17 junij, Helsinki, Finska

IFAC CONGRESS

Organizator: Finish Society of Automatic Control

Informacije: Mr Olli Pezolanti, Höyläämötie 18, 00380 Helsinki 38, Finland

21-23 junij, Toulouse, Francija

1978 INTERNATIONAL SYMPOSIUM ON FAULT TOLERANT COMPUTING-FTCS-8

Organizator: FTC Technical Committee of the Institute for Electrical and Electronics Engineers Computer Society

Informacije: IEEE, 345 East 47th Street, New York, NY 10017, USA

11-15 junij, Prague, Češkoslovaška

IFAC/IFIP 2nd INTERNATIONAL SYMPOSIUM "SOFTWARE FOR COMPUTER CONTROL"

Organizator in informacije: IFAC/IFIP TC.5

22-23 junij, Gardone Riviera, Italija

ICPCI 78: INTERNATIONAL CONFERENCE ON THE PERFORMANCE OF COMPUTER INSTALLATIONS

Organizator: Politecnico di Milano, Università Cattolica del S. Cuore, Università Commerciale L. Bocconi, Università degli Studi di Milano, Università degli Studi di Pavia

Informacije: ICPCI 78 Conference Secretariat, CILEA, Via Raffaello Sanzio 4, 20090 Segrate (MI), Italia.

29-30 junij, Milano, Italija

INTERNATIONAL CONFERENCE ON MANAGEMENT DATA

Organizator: ACM Italian Chapter

Informacije: ACM HQ, 1133 Avenue of the Americas, New York, NY 10036, USA.

18-22 julij, Udine, Italija

FIFTH INTERNATIONAL CONFERENCE ON AUTOMATA LANGUAGES AND PROGRAMMING

Organizator: CISM, CNR

Informacije: CNR, Via Panciatichi 56/16, 50127 Florence, Italia.

september, Berlin, ZRN

THIRD INTERNATIONAL CONGRESS ON ELECTRONIC INFORMATION PROCESSING (IKD)

Organizator: AMK-Berlin, IKD Professional Commission

Informacije: AMK-Berlin, Ausstellungs-Mess-Kongress-GmbH, Messedamm 22, D-1000, Berlin 19, Germany

4-8 sept. MEDICAL INFORMATICS EUROPE: FIRST CONGRESS OF THE EUROPEAN FEDERATION FOR MEDICAL INFORMATICS

Organizator: European Federation for Medical Informatics
Informacije: Dr B Barber, Management Services Division, St Faith's Hospital, London Road, Brentwood, Essex, UK

4-8 sept. Manila, Filipini

SOUTH EAST ASIA REGIONAL COMPUTER CONFERENCE 1978 (SEARCC 78)

Organizator: Singapore Computer Society

Informacije: Robert Iau, President, Singapore Computer Society, c/o Central Provident Fund Board, Robinson Road, Singapore 1, Rep. of Singapore

POJMOVI I SKRAČENICE NA PODRUČJU RAČUNARSKE TEHNIKE

- Accumulator (akumulator) - 1, 8, 12 ili 16 bitni spremnik čija funkcija je da "drži" operand aritmetične, logične ili ulazno izlazne operacije.
- ADC (Analog-to-digital converter) - pretvarač analognih signala u digitalne.
- Address (adresa) - broj koji identificira registre, memorijske lokacije, periferne jedinice itd..
- ALU (Arithmetic logic unit) - Osnovna komponenta mikroprocesora koja je sastavljena iz spremnika i upravljačkog bloka ALU izvršava aritmetično logične operacije kao što su sabiranje, oduzimanje, konjunkcija, negacija itd.
- Algorithm (algoritam) - niz pravila koja predstavljaju korake pri rješavanju problema.
- Assembler - računarski program koji simbolične ulazne programe prevede u strojno izvršljive programe.
- ASCII - kratica za American Standard Code for Information Interchange. To je 7 bitna koda kojom predstavljamo alfanumerične znake kao binarne brojeve.
- Base register (bazni registar) - Registar koji je dovoljno velik da može sadržati adresu bilo koje lokacije u memoriji, upotrijebljen da "drži" apsolutnu adresu. Upotrebljava se za bazno relativno adresiranje memorije.
- Batch - procesiranje na sistemu gdje se programi izvršavaju jedan za drugim.
- Baud - brzina prenosa simbola u sekundi. Simbol je u ovom slučaju definiran kao skup bitova koji se prenose simultano.
- BCD - Binary Coded Decimal
- Breakpoint - Instrukcija umetnuta u program kako bi omogućila testiranje.
- Buffer - 1. Memorijski blok upotrijebljen za prilagođavanje brzine prenosa informacije.
2. Elektronska naprava koja povezuje dva elektronska kola tako da ih prilagođava jedno drugome.
- Bus - Zajednički put kojim informacije putuju od jednog izmedju nekoliko izvora do jednog ili više ciljeva.
- Compatibility - 1. Sposobnost dva računara da izmjenjaju električne signale.
2. Sposobnost dva računara da prihvate iste programsek naredbe.
- Console (konzola) - dijel računara koji sadrži većinu kontrola nad centralnom procesnom jedinicom. Većina indikatora internih operacija se također nalaze na konzoli.
- Core Image - forma koju ima program dok se nalazi u glavnoj memoriji.
- Cue - Tačka u programu na koju se prenese kontrola iz drugog programa.
- Cycle stealing - Način na koji se pri direktnom dostu-

pu do memorije suspenduje memorijski ciklus.

- Cycle time (Vrijeme ciklusa) - Vrijeme potrebno za čitanje riječi iz memorije te ponovni upis iste (zbog destruktivnosti čitanja).
- DDC (Direct Digital Control) - Direktna digitalna kontrola.
- Debuging - Otkrivanje i odklanjanje grešaka u hardveru i softveru.
- Display - Naprava koja omogućava vizualnu reprezentaciju podataka.
- DMA (Direct Memory access) - Direktni pristup do memorije. Prenos podataka direktno izmedju periferije i memorije, bez posredovanja centralne procesne jedinice.
- Editor (Uredjivač) - program za editiranje izvornih programa, podatkovnih paketa itd.
- File (Zbirka) - Organizirana kolekcija zapisa koja je obično na eksternoj memoriji.

avtorji in sodelavci

Drago NOVAK (1951), je diplomiral na Fakulteti za elektrotehniko v Ljubljani leta 1970. Zaposlen je na Institutu "Jožef Stefan". Ukvarja se s hardvarskimi in softvarskimi problemi mikro računalnikov in s problemi večprocesorskih sistemov. Sodeluje z našim uredništvom in vodi področje Mikro računalniki.

BRANISLAV POPOVIĆ (1948), diplomiral na Fakulteti za elektrotehniko v Ljubljani (1971). Po diplomi se je kot sodelavec katedre za kibernetiko na isti fakulteti ukvarjal s simulacijo in procesiranjem elektrofizioloških signalov. Leta 1972 je prešel v delovno organizacijo Iskra Elektromehanika, Kranj. V času 1972/73 je bil v Antwerpnu pri firmi BMT v iskrini skupini za osvojitve sistema Metaconta 10C. Magistriral (1974) je iz metodologije načrtovanja prevajalnikov za višje programske jezike. Delal je na programih za telefonske funkcije, diagnostiko in statistiko procesorsko vodenih telefonskih central. Kot samostojni raziskovalec se sedaj ukvarja z uvajanjem sodobnih metodologij načrtovanja operacijskih sistemov ter z uvajanjem mikro računalnikov v telekomunikacijske sisteme. V tem časopisu sodeluje kot član uredništva.

MEKINDA MILAN (1945), diplomiral je na Fakulteti za elektrotehniko v Ljubljani (1969). Zaposlil se je pri ŽTP Ljubljana in se ukvarjal z avtomatskimi signalno varnostnimi napravami. Kot zunanji sodelavec Prometnega instituta se je ukvarjal s simulacijo vožnje vlaka po avtomatizirani progi. Leta 1972 se je vključil v delovno organizacijo Iskra Elektromehanika v Kranju. V letih 1972-1973 je bil na izpopolnjevanju pri licenčnem partnerju BTM v Belgiji, kjer je sodeloval pri prenosu sistema

Metaconta 10 C. Magistriral (1974) je iz analize in strukturiranja sistema računalniško vodene telefonske centrale. Delal je pri analizi in oblikovanju nove aplikacije računalniško vodene telefonske centrale. Kot višji raziskovalec vodi izdelavo in preizkušanje programske opreme. Ukvarja se z raziskavami novih programskih tehnik ter novimi pristopi k organizaciji načrtovanja velikih sistemov realnega časa.

PETER KOLBEZEN (1932), diplomiral na Fakulteti za elektrotehniko v Ljubljani (1957). Istotam je magistriral (1969) iz problematike sinteze programiranih avtomatov in uspešno obranil (1974) doktorsko disertacijo pod na-

slovom "Algoritmčni jezik za opis problemov analize in sinteze avtomatov". V letu 1975 je bil habilitiran za docenta za področje "Računalništva in informatike". Na tretji stopnji študija predava predmet "Organizacija računalnikov in mikroprogramiranje". Od diplome II. stopnje dalje je redni sodelavec Instituta "Jožef Stefan". Kot zunanji sodelavec je sodeloval tudi na večih nalogah Združenega podjetja ISKRA. Na mestu višjega raziskovalnega sodelavca "Odseka za računalništvo in informatiko" na Institutu "Jožef Stefan" dela na aplikativnih in raziskovalnih nalogah na področju načrtovanja digitalnih sistemov.

CENIK OGLASOV

Ovitek - notranja stran (za letnik 1977)

2 stran ----- 16.000 din

3 stran ----- 12.000 din

Vmesne strani (za letnik 1977)

1/1 stran ----- 8.000 din

1/2 strani ----- 5.000 din

Vmesne strani (za posamezno številko)

1/1 stran ----- 3.000 din

1/2 strani ----- 2.000 din

Oglas o potrebah po kadrih (za posamezno številko)

----- 1.000 din

Razen oglasov v klasični obliki so zaželjene tudi krajše poslovne, strokovne in propagandne informacije in članki. Cena objave tovrstnega materiala se bo določala sporazumno.

ADVERTIZING RATES

Cover page (for all issues of 1977)

2nd page ----- 16.000 din

3rd page ----- 12.000 din

Inside pages (for all issues of 1977)

1/1 page ----- 8.000 din

1/2 page ----- 5.000 din

Inside pages (individual issues)

1/1 page ----- 3.000 din

1/2 page ----- 2.000 din

Rates for classified advertising:

each ad ----- 1.000 din

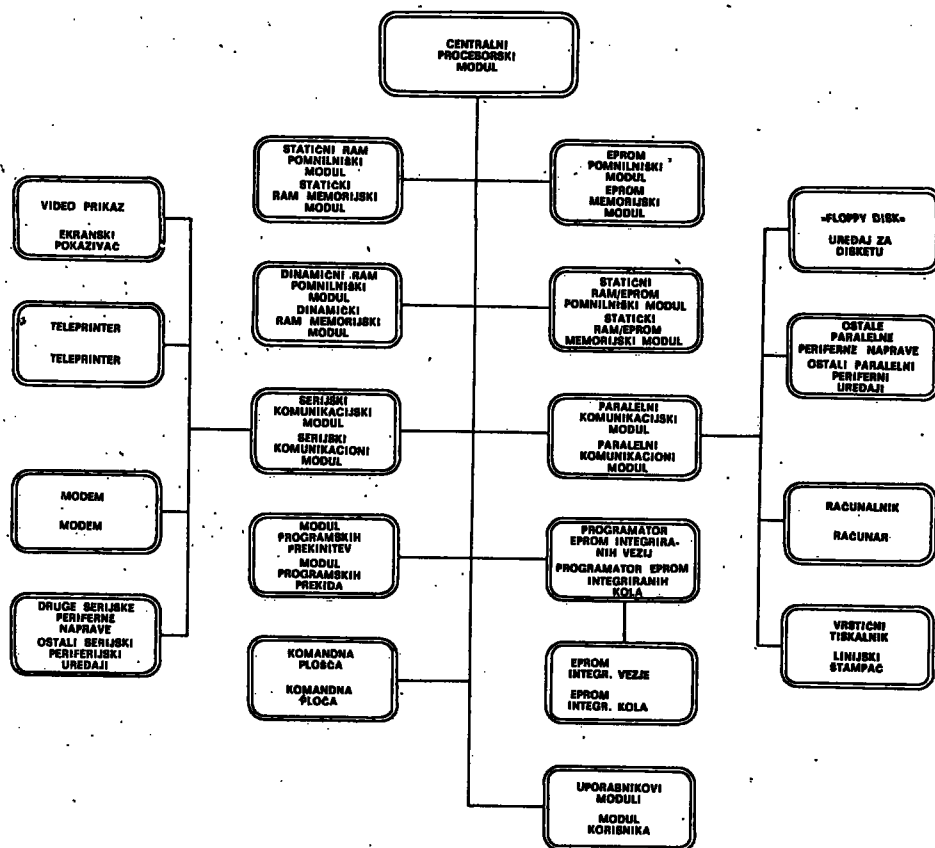
In addition to advertisements, we welcome short business or product news, notes and articles. The related charges are negotiable.



imamo, što ste tražili! iskradata 1680 prvi jugoslovenski mikroračunar

Konfiguracija mikroračunarnika Iskradata 1680

Konfiguracija mikroračunara Iskradata 1680



Mikroračunar Iskradata 1680 je zasnovan na mikroprocesorskoj seriji MOTOROLA 6800. Sistem je izradjen u modularnoj tehnici, tako da svaki modul (ploča sa štampanim kolima, format dvojna evropa) predstavlja električni zatvorenu celinu. Zajednički sistem vodjica za ploče, na koji dolaze signali zajednički svim modulima, omogućava korisniku proširivanje osnovnog sistema, odnosno slobodan izbor konfiguracije mikroračunara Iskradata 1680.

Ulazno/izlazni moduli povezuju uređaj sa proizvoljnim perifernim uređajima i time otvaraju široke mogućnosti komuniciranja.

Programski deo mikroračunara Iskradata 1680 obezbeđuje korisniku jednostavno pisanje programa i vezu sa programskim komandama.

Mikroračunar Iskradata 1680 sačinjavaju moduli sa zajedničkim vodjocima zbog toga mogu zauzimati proizvoljno mesto u uređaju. Moduli nemaju fizičke adrese i zato mogu dobiti proizvoljnu adresu u okviru celog adresnog područja. U svojoj strukturi svaki modul ima relativnu adresu, koja se uključuje na samom povezivanjem ili pomoću preklopnika.

Software mikroračunara Iskradata 1680 omogućava korisniku jednostavno programiranje sistema. Kontrolni program (monitor) sadrži sve potrebne rutine (naredbe) potrebne za upravljanje sa mikroračunarom (zapamti, pokaži itd). Kontrolni program je upisan na EPROM integrisanom kolu.

Rezidentni asemblerski prevodilac omogućava korisniku jednostavno pisanje programa u mnemoničkim rečima. (source program).

Makro assembler/disassemblerski prevodilac služi za pisanje programa u makro assembleru, odnosno za pretvaranje mnemoničkih naredbi u heksadecimalni kod.

Korisnik ima takodje mogućnost programiranja u višem programskom jeziku BASIC.

Korisnik može izabrati prevedeni program upisan u EPROM memoriji, odnosno na bušenoj papirnoj traci, koji potom sam svaki put unosi u RAM memoriju.

U budućnosti bićemo u mogućnosti da isporučimo pored mikroračunara Iskradata 1680 još i potrebne periferne jedinice (ekranski pokazivač, teleprinter, uređaj za disketu, linijski štampač itd.)

Pored modula prikazanih na slici, imamo i :
Brisač EPROM-a je u samostalno kućište ugradjena ultravioletna lampa za brisanje EPROM-a sa vremenskim uključenjem brisanja.

U pripremi su još sledeći moduli: časovnik realnog vremena, DMA moduli, D/A i A/D konverter, koji će omogućiti još veću univerzalnost i sveukupnost sistema mikroračunara Iskradata 1680,

ISKRA, Industrija za telekomunikacije, elektroniko in elektromehaniko, Kranj, TOZD Računalniki, 64000 Kranj, PE Ljubljana, 61000 Ljubljana, Titova 81 - telefon: (061) 326-367.

Tisk: Tiskarna Kresija, Ljubljana, Adamič-Lundrovo nabrežje 2
500 izvodov - december 1977