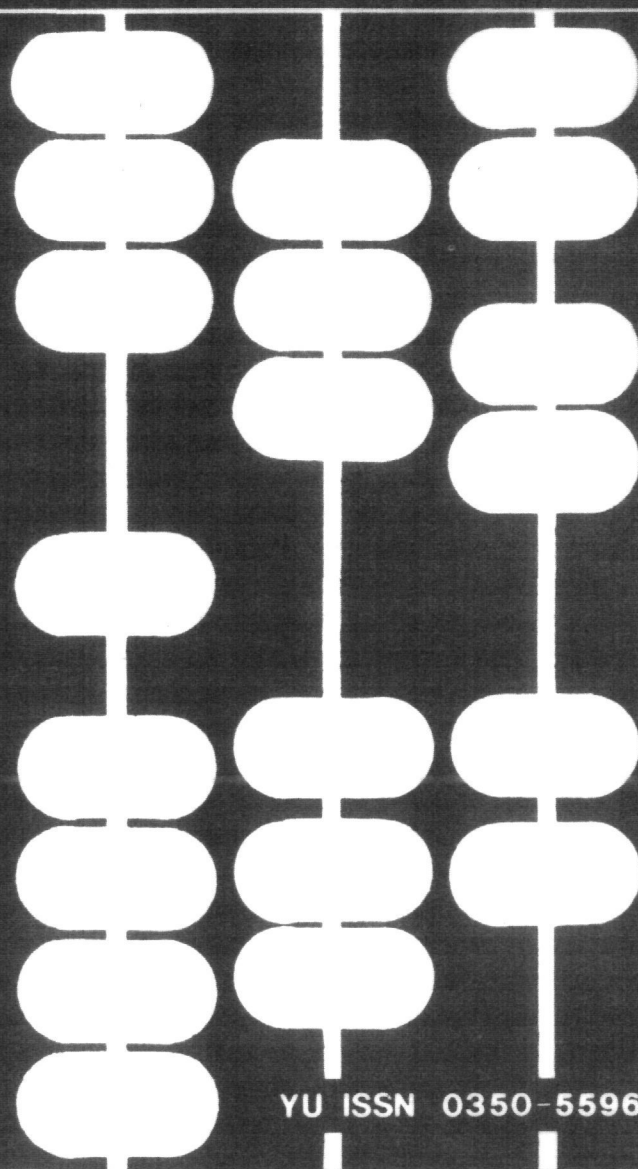
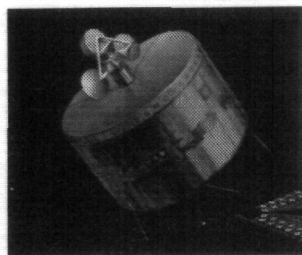


79 informatica 4

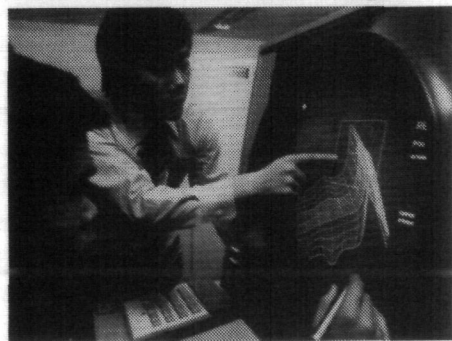
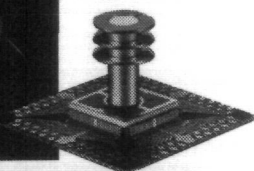




FACOM kompjutere proizvodi Fujitsu, tvrtka koja najveću pažnju posvećuje sistemima.



*LSI
s rebrima
za hlađenje*



Prije svega kompjuter je sistem, tj. sredstvo za obradu podataka koji u sebi sadrži hardware, software i aplikacionu tehnologiju. Naravno razne tvrtke bave se prodajom kompjutera. Ipak, malo je tvrtki koje mogu ponuditi potpuni izbor sredstava za automatsku obradu podataka – konstruirani tako, da osim optimalnih performanci, imaju mogućnost ugradnje u veće sisteme.

FUJITSU je jedna od tvrtki koja to može ponuditi. Kao vodeći proizvođač kompjuterskih sistema u Japanu, FUJITSU proizvodi široki asortiman proizvoda od minikompjutera s jednim LSI čipom do u svijetu najmoćnijih LSI sistema, kao i široki izbor periferne i terminalne opreme.

FACOM kompjuteri obavljaju važne aktivnosti u poslovnim i državno-administrativnim organizacijama u mnogim zemljama širom svijeta. U Japanu, drugom po redu najvećem tržištu kompjutera u svijetu, instalirano je najviše FACOM sistema u usporedbi s drugim modelima ostalih proizvođača. Ovi moćni, pouzdani FACOM kompjuteri sposobni su za obavljanje svih mogućih poslova. Oni upravljaju satelitima u svemiru, daju prikaz atmosferskih prilika real-time grafikonima u boji, obavljaju bankovno poslovanje pomoću on-line sistema za više od 7.000 filijala i ekspozitura i još mnogo, mnogo toga.

FACOM kompjuteri su potpuno integrirani sistemi gdje se kombinacijom visoko-kvalitetne tehnologije, moćnog softwera i već provjerenih aplikacionih programa postiže efikasnost i pouzdanost kojima nema premca.

Za dalje informacije obratite se na:

zpr

Zavod za primjenu elektroničkih računala
i ekonomski inženjering

41000 ZAGREB Savska c. 56 Telefon: 518-706, 510-760 Telex: 21689 YU ZPR FJ



FUJITSU



Fujitsu Limited Tokyo, Japan

informatics

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

JOURNAL OF COMPUTING AND INFORMATICS

EDITORIAL BOARD:

T. Aleksić, Beograd, D. Bitrakov, Skopje, P. Dragojlović, Rijeka, S. Hodžar, Ljubljana, B. Horvat, Maribor, A. Mandžić, Sarajevo, S. Mihalić, Varaždin, S. Turk, Zagreb.

EDITOR-IN-CHIEF:

Anton P. Železnikar

YU ISSN 0350 - 5596

TECHNICAL DEPARTMENTS EDITORS:

V. Batagelj, D. Vitas - Programming
I. Bratko - Artificial Intelligence
D. Čeček-Kecmanović - Information Systems
M. Exel - Operating Systems
A. Jerman-Blažič - Publishers News
B. Džonova-Jerman-Blažič - Literature and Meetings
L. Lenart - Process Informatics
D. Novak - Microcomputers
N. Papić - 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:

Rudi Murn

PUBLISHING COUNCIL

T. Banovec, Zavod SR Slovenije za družbeno planiranje, Ljubljana
A. Jerman-Blažič, Republiški komite za družbeno planiranje in informacijski sistem, Ljubljana
B. Klemenčič, ISKRA, Elektromehanika, Kranj
S. Saksida, 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)263 261, Cable; JOSTIN Ljubljana, Telex: 31 296 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: Rasto Kirn

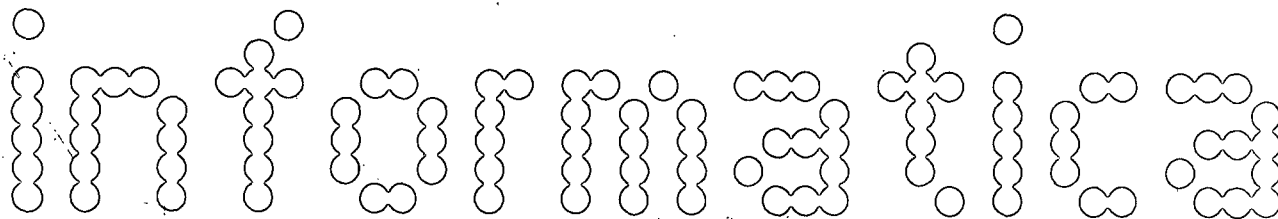
VOLUME 3, 1979 - No. 4

CONTENTS

S.Han	3	Biography
S.Han	4	Some Social Implications of Informatics
M.Gams N.Lavrač I.Bratko	11	Basic Principles and Structure of Expert Systems
N.Hadjina	17	Principles of System Software Design for Multiple-Microprocessor Systems
D.B.Popovski	23	A FORTRAN IV Subroutine for Finding a Bracketed Root
D.Milja P.Kolbezen	25	Some Aids for Microcomputer Systems Design
G.Kandus M.Špegel	30	Computer vision
F.Ružič	35	Information Systems for Public Data Communications
S.Prešeren M.Špegel	42	Sensing Systems in Robotics
D.M.Velašević	49	An Assembler Construction for the Interactive Program Development
D.Jemuović	53	Some Notices About Simple Hypothesis Formation on Computer (about chains of atoms)
A.P.Železnikar	58	Text Editor Improvement

News

Literature and Meeting



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

UREDNIŠKI ODBOR:

Člani: T. Aleksić, Beograd, D. Bitrakov, Skopje, P. Dragojlović, Rijeka, S. Hodžar, Ljubljana, B. Horvat, Maribor, A. Mandžić, Sarajevo, S. Mihalić, Varaždin, S. Turk, Zagreb.

Glavni in odgovorni urednik: Anton P. Železnikar

TEHNIČNI ODBOR:

Uredniki področij:

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

Tehnični urednik: Rudi Murn

ZALOŽNIŠKI SVET

- T. Banovec, Zavod SR Slovenije za družbeno planiranje, Ljubljana
- A. Jerman-Blažič, Republiški komite za družbeno planiranje in informacijski sistem, Ljubljana
- B. Klemenčič, Iskra, Elektromehanika, Kranj
- S. Saksida, Institut za sociologijo in filozofijo pri Univerzi v Ljubljani, Ljubljana
- J. Virant, Fakulteta za elektrotehniko, Univerza v Ljubljani, Ljubljana

Uredništvo in uprava: 61000 Ljubljana, Institut "Jožef Stefan", Jamova 39, telef. (061)263-261, telegram JOSTIN, telex: 31 296 YU JOSTIN.

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

Žiro račun št.: 50101-678-51841

Stališče uredništva se lahko razlikuje od mnenja avtorjev.

Pri financiranju revije sodeluje tudi Raziskovalna skupnost Slovenije.

Na podlagi mnenja Republiškega sekretariata za prosveto in kulturo št. 4210-44/79 z dne 1.2.1979, je časopis oproščen temeljnega davka od prometa proizvodov.

Tisk: Tiskarna KRESIJA, Ljubljana

Grafična oprema: Rasto Kirn

**ČASOPIS ZA TEHNOLOGIJO RAČUNALNIŠTVA
IN PROBLEME INFORMATIKE
ČASOPIS ZA RAČUNARSKU TEHNOLOGIJU I
PROBLEME INFORMATIKE
SPISANIE ZA TEHNOLOGIJA NA SMETANJETO
I PROBLEMI OD OBLASTA NA INFORMATIKATA**

YU ISSN 0350 - 5596

LETNIK 3, 1979 - št. 4

VSEBINA

S.Han	3	Kratek izvod iz biografije
S.Han	4	Neke društvene implikacije informatike
M.Gams N.Lavrač I.Bratko	11	Osnovni koncepti in struktura ekspertnih sistemov
N.Hadjina	17	Principi izgradnje sistemskih programskih komponenti u multi-mikro-procesorskim sistemima
D.B.Popovski	23	Jedan FORTRAN IV potprogram za nalaženje izolovanog korena
D.Miljan P.Kolbezen	25	Nekateri pripomočki načrtovanja mikroručunalniških sistemov
G.Kandus M.Špegel	30	Računalniški vid: obravnava vizualnih informacij in analiza slik
F.Ružić	35	Prilog definiciji informacijskih sistema za javno komuniciranje podacima i informacijama
S.Prešeren M.Špegel	42	Senzorski sistemi robotov
D.M.Velašević	49	Konstrukcija asemblera za iterativni razvoj programa
D.Jemuović	53	O postavljanju jednostavnih hipoteza pomoću računara (o lancima atoma)
A.P.Železnikar	58	Dopolnitev procesorja teksta

Novice in zanimivosti

Literatura in srečanja

PROF. STJEPAN HAN

KRATAK IZVOD IZ BIOGRAFIJE

Po završetku studija elektroinženjerstva na Tehničkom fakultetu u Zagrebu (1930.), drug Han se zapošljava u zagrebačkoj Gradskoj električnoj centrali, gde do rata pored ostalih poslova radi na planovima elektrifikacije šireg zagrebačkog regiona, kao i na stvaranju i uvodjenju novih tarifa za prodaju električne energije. To ga dovodi u vezu sa ekonomijom i naučnom organizacijom rada s jedne, a sa prvim koracima onda još mlade, interdisciplinarne energetike s druge strane.

Od decembra 1941. godine organizovano pomaže NOP, a od 1943. godine do kraja rata aktivno učestvuje u narodnooslobodilačkoj borbi, vršeći razne poslove i funkcije.

Posle demobilizacije u avgustu 1945. godine prelazi u bivšu Glavnu upravu elektroprivrede, gde kao načelnik, odnosno sekretar Elektroprivrednog saveta rukovodi izradom prvog plana elektrifikacije Jugoslavije. U oktobru 1946. godine prelazi u Ministarstvo industrije FNR Jugoslavije, gde rukovodi izradom prvog plana industrijalizacije zemlje. Kasnije postaje direktor Biroa za unapredjenje proizvodnje, zatim direktor Saveznog zavoda za produktivnost rada. Tu počinje bitku za sistematsko praćenje, merenje i povećavanje produktivnosti rada. Sredinom 1960. godine vraća se u saveznu administraciju gde je, kao sekretar Savezne komisije za radno vreme zadužen višegodišnjim zadatkom pripremanja i sprovođenja prelaska jugoslovenske privrede od 48-časovne na 42-časovnu radnu sedmicu.

Uporedo sa tim poslovima petnaestak godina vrši funkciju generalnog sekretara Jugoslovenskog nacionalnog komiteta Svetske konferencije za energiju, organizujući 1957. godine u Beogradu zasedanje Svetske konferencije za energiju, posvećeno tek danas akutnom problemu povezanosti energije i ekonomskog razvoja nedovoljno razvijenih zemalja.

Godine 1962. izabran je za redovnog univerzitetskog profesora, pa predaje nauku o radu i njegovom organizovanju na Elektrotehničkom fakultetu u Beogradu i Ekonomskom fakultetu u Subotici. Četiri godine kasnije oslobadja se dužnosti u SIV-u i zasniva radni odnos sa fakultetom u Subotici. Tamo je bio šef katedre, dekan i direktor Zavoda za organizaciju rada i poslovanja. Nabavio je za taj Zavod prvi ozbiljniji kompjuter u Vojvodini, obrazovao kadrove i uveo novi smer dodiplomskog i poslediplomskog studija informatike. Autor je stotinjak članaka, referata, studija i knjiga, a od Oslobođenja do danas je društveno veoma aktivan. Više od trideset godina je delegat Jugoslavije u mnogim međunarodnim vladinim i nevladinim organizacijama. Za svoju delatnost je dobio više priznanja, nagrada i odlikovanja.



NEKE DRUŠTVENE IMPLIKACIJE INFORMATIKE

Prof. S. HAN

UDK: 007:681.3

INSTITUT ZA INFORMATIKU, NOVI SAD

Referat ispituje neke od društvenih uticaja informatike na (1) svet profesionalnih informatičara; (2) ovo širi društveni sloj korisnika informatike, čije se aktivno učešće u projektovanju informacionih sistema preporučuje; i (3) društvo u celini.

SOME SOCIAL IMPLICATIONS OF INFORMATICS

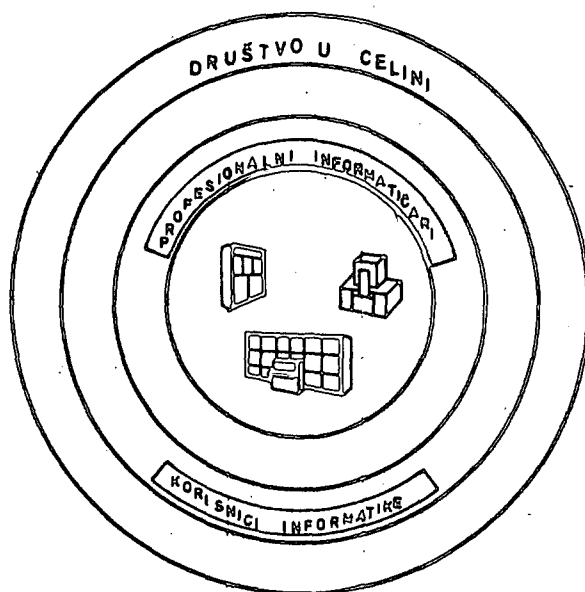
The paper examines some of the social impacts of informatics upon: (1) the world of computer specialists; (2) the ever larger social stratum of users of informatics, whose active participation in the process of designing information systems is recommended; and (3) society as a whole.

A. "TALASI PROMENE"

"Talasi promene" ("The Waves of Change") je naslov jedne uzbudljive knjige, koja podstiče na razmišljanje. Napisao ju je Čarls P. Lekt (Charles P. Lecht), i u njoj analizirao radikalne promene ordinatorske tehnologije i informatike tokom poslednje dve decenije. U toj knjizi, međjutim, ima i mnogo prognoza, pa bi bilo teško da ovde nabrojimo makar samo naslove glava i odseka tog dela, koje je postalo obavezno štivo

za svakoga, ko se za tehnološke i ekonomske aspekte revolucije u oblasti informatike ozbiljnije interesuje.

Ovaj članak, međjutim, treba da osvetli neke od društvenih aspekata tehnno-ekonomskog razvoja, tako da od Lokta posudjuje jedino naslov, dok je materija koju obradjuje veoma različita: uticaj nove tehnologije i brzog napretka u oblasti informatike na čove-



SLIKA 1

ka, i njegove društvene agregate. Čovečanstvo sada treba da očekuje te druge "talase promene" koji već jure kroz razvijeni deo sveta, i to u koncentričnim krugovima u čijem je centru ordinator, dolazeći prvo do profesionalnih informatičara, onda do korisnika informatike, a najzad do društva u celini.

Jasno nam je da je taj pristup pomalo proizvoljan, zbog neprekidne uzročno-posledične interakcije između tri uticajne sfere koje su ovde predložene. Ipak, zbog kratkoće i nekih drugih praktičnih razloga, mi ćemo pokušati da osvetlimo neke nasumce izabrane društvene implikacije informatike, služeći se modelom sa tri kruga.

Pre toga, međjutim, ukratko ćemo ilustrovati društveno relevantnu brzinu tehnoloških promena, obraćajući posebnu pažnju integrisanim elektronskim kolima. Za manje od dve decenije, parametri složenosti i performansi povećali su se za tri ili četiri reda veličine. Tabela koja sledi ilustruje jedan od značajnih napredaka, koji snažno utiče na razvoj hardware-a uopšte.

VRSTE ČIPOVA BROJ KOMPONENATA PO ČIPU

- 1) SSI
Nizak stupanj integrisanosti
(Small Scale Integration) 5 - 50
- 2) MSI
Srednji stupanj integrisanosti
(Medium Scale Integration) ... 50 - 500
- 3) LSI
Visok stupanj integrisanosti
(Large Scale Integration) 500 - 10000
- 4) VLSI
Veoma visok stupanj integrisanosti
(Very Large Scale Integration) 10000-100000
- 5) SLSI
Super visoki stupanj integrisanosti
(Super Large Scale Integration) ... > 100000

Čipovi tipa VLSI i SLSI još nisu na širokom tržištu, ali su već na putu da stignu. Verovatno će biti gotovi pre nego to pesimiste, pa i optimiste očekuju. Obazrivo pripremljene normativne prognoze govore, da će čipovi tipa SLSI pre kraja stoleća biti proizvodjeni masovno i automatizovano, nastavljajući na taj način eksponencijalni trend razvoja koji je bio dominantan do danas.

Pokušaćemo sada da ukratko ilustroujemo već spomenutu brzinu kojom se ova tehnologija

menja. Trebalo je ne više od 15 godina da od prvih, relativno jednostavnih integrisanih kola (SSI), tamo početkom šezdesetih godina, dodjemo do veoma složenih kola današnjice (LSI) koja imaju više hiljada komponentata po čipu. Tom zaista revolucionarnom razvoju nema premca u bilo kom drugom području tehnologije, jer su - u poredjenju sa ranim čipovima tipa SSI - današnji čipovi

- $10^3 - 10^4$ puta brži
- $10^3 - 10^4$ puta pouzdaniji
- $10^3 - 10^4$ puta ekonomičniji (mereno potrošnjom energije)
- $10^3 - 10^4$ puta jeftiniji.

Ovaj napredak koji "zaustavlja dah" (Höglund) može da se iskaže i dokaže, ali se zapravo ne može shvatiti, sve dok ne upotrebimo neku analogiju. Uzmimo stoga, kao osnovu naše analogije, automobil "ZASTAVA-750" koji je otprilike savremenik čipova tipa SSI. Ako bi u automobilskoj industriji napredak mogao da bude toliko brz kao u elektronici, performanse "Fiće" bi se do danas povećale u nezamislivoj meri.

NEKE PERFORMANSE AUTOMOBILA

"ZASTAVA-750"

(stvarne, odnosno hipotetične)

	RANIH 60-TIH GODINA	KASNIH 70-TIH GODINA
- maksimalna brzina	oko 100 km/h	više od 100000 km/h
- trajanje	oko 8 godina	oko 10000 god.
- potrošnja goriva	oko 8 l/100 km	oko 5 ml/100 km
- cena	oko 50.000 din.	manje od 50 dinara.

Nije, prema tome, neobično što svaki ozbiljan naučnik koji se bavi društvenim naukama, predviđa snažan uticaj informatike na društvo. Mnogo snažniji od neosporno veoma osetnog uticaja automobila.

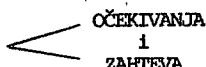
B. MALI SVET PROFESIONALNIH INFORMATIČARA

B.1 Profesionalni informatičari među sobom

U svakoj maloj socijalnoj grupi tražimo koheziju, zasnovanu na dobrim odnosima među ljudima na radu, kao preduslov visoke produktivnosti i niskih stopa fluktuacije. Kohezija, s druge strane, zavisi od zadovoljstva sa radom, za koje Enid Mumford misli da je plod dinamične ravnoteže između oče-

kivanja s jedne strane (šta on ili ona očekuje, želi, traži i zahteva od svog posla), i zahteva, s druge strane, (ono što radna organizacija očekuje, želi, traži i zahteva od nje, odnosno njega).

ZADOVOLJSTVO SA RADOM =

= SLAGANJE IZMEDJU  OČEKIVANJA
1
ZAHTEVA

(Mumford, 1973.). To važi za sve vrste poslova, bez obzira na predmete rada i tehnologiju koja se primenjuje.

Specifičnost uticaja ordinatora i informatike, međutim, izvire iz činjenica da ti poslovi zahtevaju prvenstveno intelektualne napore i novu hijerarhiju. Odnosi između analitičara sistema, programera, pa i operatera nisu isti kao odnosi između visoko kvalifikovanog poslovođe i priučenog radnika u masovnoj proizvodnji savremene industrije. Tu bi, prema tome, problemu kohezije trebalo drugačije pristupiti, nego što to čini Mumfordova. Jedan od tih drugačijih pristupa je Hercbergov (Herzberg), koji kombinuje faktore "motivacije" sa faktorima "higijene".

MOTIVACIONI FAKTORI

M.1 Značaj poslova

M.2 Raznolikost poslova

M.3 Prihvatanje inicijative

.

.

.

FAKTORI HIGIJENE

H.1 Plata

H.2 Uslovi za rad

H.3 Sredstva za rad

.

.

.

U svakom slučaju će primena dobro izabranih i korektno primenjenih psihometrijskih testova u toku profesionalne selekcije moći da poveća društvenu koheziju izabranih radnika, ali nas jedan od poznatijih autora upozorava da "... testove treba pre toga savesno ispitati. ... Treba osim toga odrediti stupanj pouzdanosti testova, to jest granice tačnosti rezultata. To, međutim, većina raspoloživih "testova sposobnosti za programiranje" ne daje. Taj test je u početku bila dobra ideja jednog bistrog programera, ali je kasnije bio "poboljšan" od drugih bistrih programera. Mnogi od takvih testova su sada veoma privlačni - kao bapsko priče, mitovi i popularne zablude". (Pennoy).

B.2 Informatičari i opšti rukovodioci

Informatičari katkada misle o opštim rukovodiocima kao o kasti kićenih "trutova", koji mogu samo formalno da predstavljaju svoju organizaciju, dok su oni pravi, visoko odgovorni poslovi rukovodjenja i upravljanja: predviđanje, planiranje, donošenje odluka, organizovanje, koordiniranje i kontrola sve više u njihovim rukama, pošto te upravljačke funkcije danas u velikoj meri zavise od pouzdanog funkcionisanja ordinatora.

U isto vreme je profesionalni informatičar "... visoko počašćen kadgod jedan od njih bude nadjen vrednim da ga prime u izabrano i privilegovano telo..." visokih rukovodilaca.

Priznajemo da smo pomalo zloupotrebili neke Engelsove ideje, iznete u "Utopijskom i naučnom socijalizmu", gde on analizira situaciju u kojoj stari, aristokratski gornji sloj, koji je izgubio ekonomsku moć, nastavlja da kontroliše državnu mašineriju, a kome je buržoazija sa realnom ekonomskom moći, ali nižeg društvenog statusa, iskreno zavidjala.

Priznajemo i to da je analogija pomalo šepava, ali je svakom poznavacu realne situacije poznato da postoje problemi, i to s jedne strane strahovanja, a s druge nade, neće li nova profesija informatičara, koja ima više organizacionih znanja, ali manji društveni ugled, postepeno preuzimati opšte poslovodne funkcije, ostvarujući na taj način Platonov - mi bismo danas rekli tehnokratski - ideal "vladara-naučnika".

Ovaj se članak slaže sa nekim autorima koji misle da je prekasno za istinsku "profesionalizaciju" informatičara. (Turton). "Profesionalizacija je proces osvajanja ekskluzivnih prava vršenja nekih poslova, kontrole kadrovske prinove i vrednovanja; moć daje ekskluzivnost znanja i umenja, a ne znanja i umenja kao takva". (Friedson). Turton smatra da je profesionalizacija, koja se razvijala tokom XVIII i XIX stoleća - lekari, advokati, arhitekto i knjigovođe - pod zaštitom i uz pomoć elitnih oligomonata društva, bila praćona relativno niskim stupnjem specijalizacije. Na osnovu toga zaključuju da su preduslovi za novo

profesionalizacije nestali, jer je druga polovina XX stoleća karakterisana veoma visokim stepenom specijalizacije. U malom svetu profesionalnih inforatičara takodje vlada sve razudjenija specijalizacija. To će reći da će birokratija - državna kao i privredna - uvek moći da uspešno sprečava pojavu nove "profesije" inforatičara.

C. KORISNICI INFORMATIKE

C.1 Ko su "korisnici"?

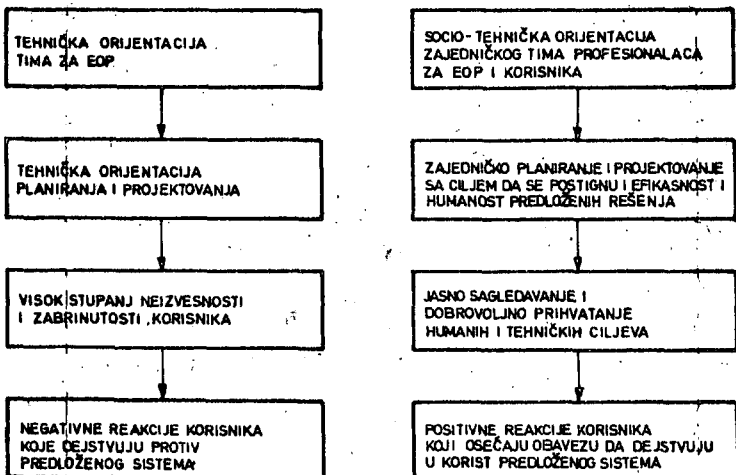
Pitanje prosto navodi na to da se na odgovori sa protupitanjem: "A ko nije"? Izvesno je da širina sloja korisnika zavisi od vremenskog horizonta koji ćemo odabrati. Za sada, dok broj terminala u stanovima i broj ličnih ordinatora bude zanemarljivo malen, mi ćemo smatrati "korisnikom" svakoga ko, iako nije profesionalni inforatičar, u izvesnoj meri shvata i sve više koristi usluge informatike. Toj grupi pripadaju neki lekari, mnogi inženjeri i praktično svi računovodje industrijalizovanih zemalja.

C.2 Uključivanje korisnika u projektovanje informacionih sistema

U poslednje vreme brzo raste broj veoma interesantnih istraživanja, posvećenih problematici uključivanja korisnika u proces projektovanja informacionih sistema. (Mumford, 1978.). Mumfordova razlikuje dva pristupa projektovanju sistema: "tradicionalni" i "participativni". Neke od karakteristika, prednosti i nedostataka tih pristupa prikazuje slika 2.

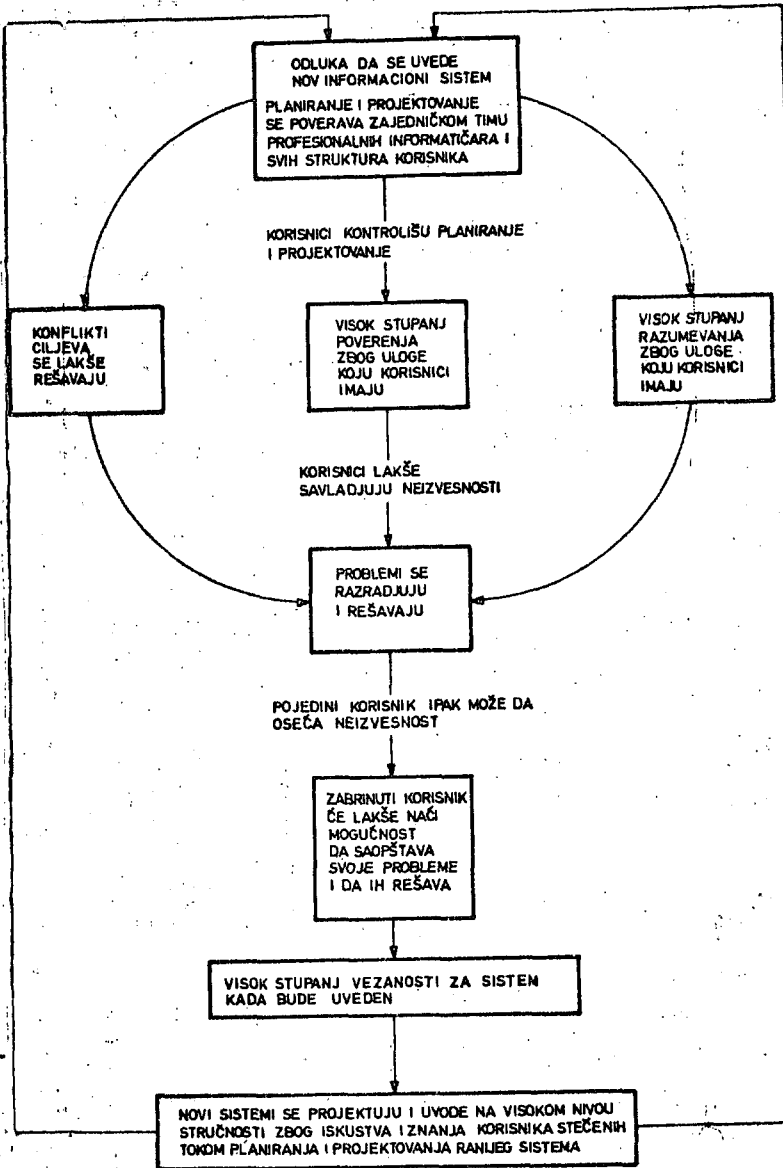
TRADICIONALNI PRISTUP

PARTICIPATIVNI PRISTUP



SLIKA 2.

Mikrosociološku dinamiku procesa projektovanja informacionog sistema sa participativnim pristupom ilustruje slika 3.



SLIKA 3.

D. DRUŠTVO U CELINI

D.1 Je li informatika privileg bogatih?

Zasada jeste. Jaz između visoko razvijenih zemalja i onih u razvoju je neverovatno dubok, a osim toga se stalno proširuje jer je u isto vreme i posledica i razlog tehnoloških, ekonomskih i kulturnih razlika među njima.

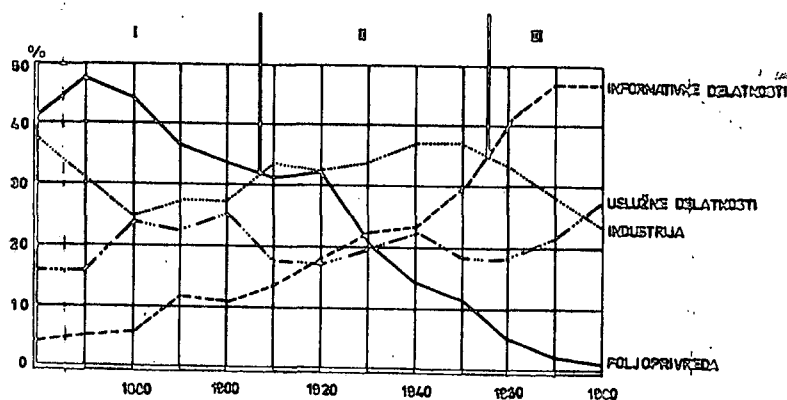
Istina je i to, da se ponegde čine napori za ublažavanje tih razlika, blagodareći postojanju i radu nekih međjuvladinih organizacija, kao što je na primer Međjuvladin biro za informatiku u Rimu (Intergovernmental Bureau for Informatics).

D.2 Da li je "informatičko društvo" na pomolu?

Visoko industrijalizovane zemlje kao SAD i Japan, pridaju sve više pažnje novim konfiguracijama socio-profesionalnih struktura, koje navodno najavljuju dolazak "informatičkog društva". Danas je većina privredno aktivnih ljudi u SAD oslobodjena prvenstveno fizičkih napora u poljoprivredi, industriji, zanatima i uslužnim delatnostima.

Ekonomija se inače može podeliti u dve polovine. U jednoj se transformišu materija i energija; dok se u drugoj informacija prevodi iz jednog pojavnog oblika u drugi.

Iako su te dve polovine ekonomije neraskidivo povezane, one su suštinski veoma različite, što ne može da nema ozbiljnog uticaja na društvo i našu sudbinu. Danijel Bel (Daniel Bell) je jednom citirao neku, onda već deset godina staru prognozu Rand korporacije, da će oko godine 2000. samo 2% američkog stanovništva proizvoditi sva industrijska dobra koja će onda biti potrebna (Nora-Minc, I aneks, str. 184.), a doktorska teza Marka Jurića Porata (Porat), daje statističke trendove koji, ekstrapolirani do 1980. godine, dele stodvadesetogodišnji proces profesionalne restruktuiranja SAD u tri stadija: (1) do oko 1907. godine, kada se broj zaposlenih u industriji izjednačio sa brojem privredno aktivnih ljudi u poljoprivredi; (2) od 1907. do oko 1958. godine, kada se udeo onih koji su bili zaposleni u sektoru informacija, izjednačio sa udelom ljudi zaposlenih u industriji; i (3) period posle 1958. godine. Vidi sliku 4. (Nora-Minc, I aneks, str. 187.).



SLIKA 4.

Treba time u vezi reći, da ljudi zaposleni u sektoru informacija nisu svi informatičari. Edwin Parker (Edwin Parker), koji se u svom izveštaju, napisanom za jedan od OECD-ovih seminara (Parker), snažno oslonio na Porata, čija teza onda još nije bila objavljena, podelio je radnu snagu SAD na 400 grupacija, od kojih 170 ide u red sektora informacija. U tom ćemo sektoru, pored profesionalnih informatičara naći i zanimanja kao na primer:

- službenici i rukovodioci,
- učitelji,
- sekretarice,
- računovodje i knjigovodje,
- inženjeri,
- pisci,
- poštari,
- grafički radnici,
- arhiviste,
- kancelarijski pomoćni radnici i mnoga druga.

I Parker je našao po neki prigovor ovoj klasifikaciji, ali opšte tendencije govore toliko jasnim jezikom, da se ne možemo o njemu oglašiti.

Ovaj je fenomen značajan, pogotovo ako se setimo da je dramatični pad agrarnog zaposlenja bio posledica mehanizacije, pa je evidentno da će smanjenje udela zaposlenih u sektoru informacija biti posledica sve veće upotrebe informatike i opreme za elektronsku obradu podataka, što će do nekog trenutka povećavati udeo, pa i apsolutni broj profesionalnih informatičara unutar sektora informacija.

Japanski koncept "informatičkog društva" je u neku ruku komplementaran viziji Porata i Parkera. Taj je koncept prikazao Jonaji Masuda, direktor Japanskog instituta za razvoj primene ordinatora (the Japan Computer Usage Development Institute, ili skraćeno JACUDI), prilikom OECD-ovog seminara održanog u Parizu od 13. do 15. novembra 1972. godine (Masuda). Informatička revolucija koja vodi do "informatičkog društva" podeljena je od Masude u četiri perioda.

Prvi je oslonjen na "veliku" nauku. Glavni su ciljevi bili u oblasti narodne odbrane i istraživanja međuplanetarnog prostora. Ta je faza trajala od oko 1940. do oko 1970. godine.

Drugi je period počeo oko 1955. godine i trajao je do 1980. godine. Masuda je zove periodom oslonjenom na privredne organizacije u kome privredna preduzeća i vlade koriste informatiku za racionalizaciju svoje delatnosti.

Treći je period orijentisan na društvene delatnosti. Treba da traje od oko 1970. do oko 1990. godine. Tokom ove faze će se informatika sve više uvoditi u obrazovanje, zdravstvo i neke druge segmente društva.

Četvrti period konačno, od oko 1980. do oko 2000. godine, biće orijentisan na pojedinca. Terminali u stanovima će postati sve mnogobrojniji.

Dajemo nešto skraćeni prevod Masudine tabele, koja je izazvala mnoge diskusije i ne malo polemika.

E. ZAKLJUČCI

Čitalac sada vidi da nije bila lažna skromnost, kada je naslov članka kazivao da će se baviti samo "nekim" društvenim implikacijama informatike.

Prvo smo pokušali da postavimo pitanje, da li ordinator i njegova primena utiču na socijalne odnose medju profesionalnim informatičarima; zatim da li pravilan odбір kandidata, primera radi pomoću psihometrijskih testova za određivanje sposobnosti za programiranje, može da poveća zadovoljstvo sa poslom i koheziju grupe; i konačno, da li će u svojim odnosima sa opštim rukovodstvom informatičarima uspeti da formiraju novu "profesiju" u tom smislu, u kome lekari, advokati, arhitekta i knjigovodje čine profesiju. Odgovor na poslednje pitanje bio je negativan.

Red. broj		1. PERIOD	2. PERIOD	3. PERIOD	4. PERIOD
		TEŽIŠTE JE NA REŠAVANJU PROBLEMA U OBLASTI			
		NAUČNO-TEHNOLOŠKE REVOLUCIJE I NARODNE ODBRANE	PRIVREBNIH ORGANIZACIJA	DRUŠTVENIH SLUŽBI	ŽIVOTA POJEDINACA
		1945.-1970.	1955.-1980.	1970.-1990.	1980.-2000
1.	CILJEVI	Usavršavanje protiv avionske odbrane i raketne tehnike, kosmonautike i korišćenja nuklearne energije.	Racionalizacija poslovanja privrednih organizacija.	Organizovanje zdravstva, obrazovanja i javne uprave.	Humanizacija života.
2.	MOTIVI	Prestić države	Privredni rast	Društveni stand.	Samoostvarivanje ličnosti.
3.	PODRUČJA	Čitava zemlja	Narodna privreda	Opština	Stan pojedinca
4.	GRANIČNE DISCIPLINE	Privredne nauke	Organizacione nauke	Društvene nauke	Biheviorističke nauke.

Hoćemo li ikada živeti u nečemu što će biti slično "informatičkom društvu"? A ako da, hoće li to biti dobro za čovečanstvo?

Obim kratkog pregleda kao što je ovaj, isključuje svaku mogućnost ma i površne analize živih diskusija čiji smo svedoci ili učesnici. Bilo bi dobro, ako bi ovaj članak izazvao razmenu misli, kritiku i polemike, što bi bolje osvetlilo probleme koji nastaju koliko iz menjanja uloge informatike, toliko iz promene socio-profesionalne strukture naših društava.

Naredna glava se bavi uključivanjem korisnika u projektovanje informacionih sistema. Raspoloživa je literatura korišćena, citirana i komentarisana, kako bi se preporučio novi, retko primenjivani, ali očigledno korisni "participativni" pristup.

U poslednjoj je glavi dato nešto dokumentacije o tendencijama socio-profesionalnih promena u najrazvijenijim zemljama sveta, ukazujući na brz porast sektora "informatičja". Ukratko je ukazano na pitanje da li ulazimo u "informatičko društvo".

CONCLUSIONS

We tried to pose, firstly, the question if the computer and its utilisation affected social relations between computer specialists; if proper selection of applicants with the help of programmer aptitude psychometric tests could improve job satisfaction and coherence within the group; and lastly if in their relations with general management, informaticians will succeed to become a "profession" in the sense physicians, lawyers, architects and accountants are. The answer to the last question was negative.

In a subsequent chapter the paper deals with the question of user involvement in the design of information systems. Available literature has been used, quoted and commented, in order to recommend the non traditional, new and rarely used, but seemingly highly advantagegeous "participative approach".

In the last chapter evidence has been assembled, concerning socio-professional changes in the most developed countries, showing rapid increase of the information sector. The possible advent of an Information Society is being briefly discussed.

LITERATURA

1. E. Friedson (Ed.): "Professions and their Prospects". Sage, 1973.
2. F. Herzberg: "The Motivation to Work". Wiley. New York, 1959.
3. I. Höglund: "The Development and Manufacturing of CMOS LSI Circuits at ASEA-HAFO". In "ASEA" Journal. Volume 52/1 Västerås, 1979.
4. Intergovernmental Bureau for Informatics "Considerations sur les effets sociaux de l'informatique". Rome, 1978.
5. C.P. Lecht: "The Waves of Change. A Techno-Economic Analysis of the Data Processing Industry". Advanced Computer Techniques Corp. New York, 1977.
6. Y. Masuda: "A New Development Stage of the Information Revolution". In: "Applications of Computer/Telecommunications Systems". OECD Informatics Studies 8. Paris, 1975. pp. 31-53.
7. E. Mumford: "Job Satisfaction: A Major Objective for the System Design Process". Management Information, 1973., 4, Pp. 191-202.
8. E. Mumford et al.: "A Participative Approach to the Design of Computer Systems". In: "Impact of Science on Society". Vol. 28/3. UNESCO. Paris, 1978.
9. S. Nora et A. Minc: "L'informatisation de la société". La Documentation Française. Paris, 1978.
10. E. Parker: "Background Report. Proceedings of the Conference on Computer/Telecommunications Policy". Organisation for Economic Cooperation and Development. Paris, 1975.
11. G. Penney: "Applying Knowledge to Reducing Recruitment Costs". In: "Computing and People". Andrew Parkin (Ed.) E.Arnold (Publishers) Limited. London, 1977.
12. M.U.Porat: "The Information Economy". Doctoral thesis submitted to the Institute for Communication Research. Stanford University, 1976.
13. R.Turton: "On the Relevance of Professionalism". In: "Computing and People". Andrew Parkin (Ed.) Edward Arnold (Publishers), Limited. London, 1977.

OSNOVNI KONCEPTI IN STRUKTURA EKSPERTNIH SISTEMOV

M. GAMS
N. LAVRAČ
I. BRATKO*

INSTITUT JOŽEF STEFAN, LJUBLJANA

* FAKULTETA ZA ELEKTROTEHNIKO, LJUBLJANA

UDK: 519.1:681.3

V referatu podajamo pregled področja ekspertnih sistemov, ki so eno najpomembnejših področij umetne inteligence in ki v zadnjih letih doživljajo velik kvalitativen in kvantitativen napredek. Referat opisuje osnovne metodološke koncepte, strukturo, delovanje in značilnosti ekspertnih sistemov. Podaja tudi pregled obstoječih sistemov z oceno njihovega delovanja in uporabnosti.

BASIC PRINCIPLES AND STRUCTURE OF EXPERT SYSTEMS: The paper presents a survey of expert systems which are one of the most important fields of artificial intelligence and have been rapidly advanced over the last years. The paper presents basic methodological concepts, structure, performance and characteristics of expert systems. It also presents a survey of existing systems together with the evaluation of their performance and use.

1. UVOD

Večina programov umetne inteligence temelji na ideji, da lahko kompleksne probleme rešujemo s heuristično vodenim iskanjem. Heuristično vodeno iskanje se skuša izogniti preiskovanju celotnega problemskega prostora, tako da s posebnimi triki pregleda samo najobetavnejše možnosti. Te metode so prvotno pripeljale do izdelave programov, pri katerih je bil poudarek na učinkovitih, od problemskega področja v glavnem neodvisnih algoritmih za preiskovanje (npr. program GPS - General Problem Solver, Winston, 77). Izkazalo pa se je, da so ti programi prešibki za učinkovito reševanje kompleksnih problemov in da leži moč visoko zmogljivih sistemov v znanju, ki ga ti sistemi vsebujejo. Na podlagi tega spoznanja so raziskave v zadnjih letih pripeljale do razvoja ekspertnih sistemov.

V najširšem smislu pod pojmom ekspertni sistemi razumemo inteligentne računalniške programe, realizirane z različnimi metodami umetne inteligence. Ime "ekspertni sistemi" izhaja iz zahteve, da ti programi delujejo podobno kot človek-strokovnjaki, ki zna na podlagi svojega specialnega znanja pametno sklepati, svetovati in razlagati svoje odločitve.

Kakršenkoli neinteligenten računalniški program ali program brez zmožnosti pojasnjevanja svojih odločitev pa ne spada v kategorijo ekspertnih sistemov.

Ekspertni sistemi so izrazito aplikativno usmerjeni in pogosto sredujemo njihove opise pod poglavjem "aplikacije umetne inteligence". Zaradi izrazite uporabniške usmerjenosti je vse podrejeno želji po čim boljših rezultatih in tako specifičnemu problemskemu prostoru.

V tem članku podajamo grobi pregled področja ekspertnih sistemov. Referat je izdelan na podlagi spoznanj, pridobljenih s prebiranjem svetovne literature o ekspertnih sistemih, in s preteklim in sedanjim delom skupine za umetno inteligenco Instituta Jožef Stefan v Ljubljani.

Dostoj smo se ukvarjali predvsem z raziskavami in implementacijo bazičnih metod umetne inteligence in ekspertnih sistemov. Ob že realiziranih ekspertnih sistemih za igranje šahovskih končnic pa načrtujemo tudi čisto aplikativne ekspertne sisteme.

2. OSNOVE DELOVANJA EKSPERTNIH SISTEMOV IN PRODUKCIJSKI SISTEMI

Inteligenca ekspertnih sistemov temelji na velikih bazah znanja, specifičnih za konkretna problemska področja. To znanje je razbito na čim bolj ločene module. Vsak modul vsebuje informacijsko zaključen kos znanja o specifični problemski domeni.

Večina ekspertnih sistemov deluje tako, da sistem pregleduje podatke, da bi ugotovil, kateri moduli ustrezajo dani situaciji v podatkih. Ko sistem najde modul, katerega iskalni vzorec se ujema z vzorcem v podatkih, se izvrši akcija oz. zaporedje akcij tega modula. Sistemi, ki delujejo na opisani način, se imenujejo vzorčno vodeni moduli.

Znanje v ekspertnih sistemih je najpogosteje predstavljeno v obliki pravil (ponavadi eno pravilo ustreza enemu modulu). Pravila imajo običajno obliko

situacija =====> akcija

Situacija določa pogoje (definira vzorec), pod katerimi je treba izvesti akcijo. Akcija je lahko trditev ali pa procedura, ki spremeni podatkovno bazo ali npr. usmeri kontrolne mehanizme sistema na aktiviranje določenega dela znanja.

Ekspertni sistemi pogosto uporabljajo produksijske sisteme; to so sistemi, ki imajo znanje predstavljeno v obliki pravil.

Oglejmo si enostaven primer produkcijskega sistema na modelu termostata za vzdrževanje temperature med 18 in 20 stopinjami Celzija.

temperatura>18 in temperatura<20 ==> miruj.
 temperatura<10 ==> pokliči_popravilo
 vključi_električno_gretje.
 temperatura<18 in stanje_pebi=ugasnjen
 ==> prižgi_peb.
 temperatura>20 in stanje_pebi=prižgan
 ==> ugasni_peb.

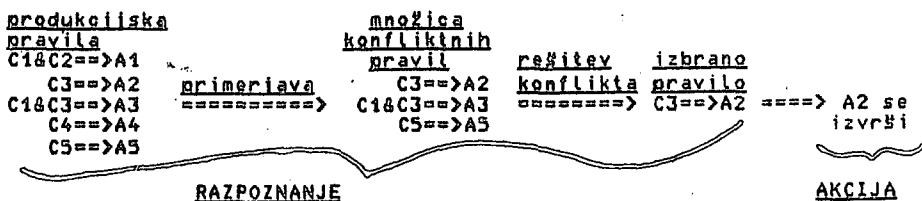
To produkcijski sistem je sestavljen iz štirih pravil. Vsaka leva stran (pogoj) pravila določa vzorec vnaprej izbranih spremenljivk. Sistem poišče pravilo, katerega pogoji so izpolnjeni glede na podatkovno bazo. To pravilo se sproži in izvrši se zaporedje akcij, ki ga definira desna stran pravila.

V tem preprostem primeru termostata smo si ogledali samo vloženo znanje, ne pa tudi njegovo dejansko uporabo. Ta lastnost omogoča "top-down" pristop in učinkovito strukturiranje sistema.

Pravila v produkcijskih sistemih se izvajajo v ciklih "razpoznavaj-ukrepaj" (glej sliko 1.).

Na sliki 1 vidimo, kako se izvrši en cikel "razpoznavaj-ukrepaj" produkcijskih sistemov. Interpretor najprej pregleduje, katera pravila izmed množice produkcijskih pravil ustrezajo danim podatkom iz podatkovne baze. V našem primeru pridejo v poštev tri pravila in ta pravila predstavljajo ti. konfliktno situacijo. Kot rešitev konfliktno situacije se lahko proži več, običajno pa samo eno pravilo iz konfliktno množice.

Podatkovna baza: C5 C1 C3



RAZPOZNANJE

AKCIJA

Slika 1: Cikel "razpoznavaj-ukrepaj" produkcijskih sistemov.

Produkcijski sistemi so praviloma seveda mnogo kompleksnejši. Naštajmo le nekaj najbolj pogostih karakteristik:

- (a) Podatki lahko izpolnjujejo pogoje več pravil - ta pravila potem tekmujejo za izvajanje.
- (b) Pravilom lahko pridamo faktorje zanesljivosti, tj. števila na intervalu [-1,1], ki izražajo, v kolikšni meri zaupamo danemu pravilu. (Vzemimo primer neformalno napisanega pravila iz medicinskega ekspertnega sistema MYCIN: "Gramnegativna paličasta bakterija v krvi je zelo verjetno (0.8) E.COLI.")
- (c) Pravila se lahko pri izvajanju vežejo v bolj ali manj komplicirane strukture (verige).
- (d) Veritazenje pravil lahko spremlja metoda za določanje faktorjev zanesljivosti verige iz faktorjev zanesljivosti posameznih pravil.

Produkcijski sistemi so našli dvojce uporabnih področij uporabe (Davis,77): za modeliranje človeškega mišljenja (PSG,PAS2,VIS..) in v ekspertnih sistemih (MYCIN,DENDRAL,MOLGEN...).

Prve uporabe PS za modeliranje človeškega mišljenja so bile raziskave na področju šaha in kriptografskih problemov (Simon in Newell).

Po Newellu in Simonu so PS primerni predvsem zaradi:

- (1) PS so ravno tako splošni kot Turingov stroj in omogočajo gradnje modelov z raznovrstnimi možnostmi procesiranja informacij;

- (2) PS omogočajo enostavno spreminjanje in dodajanje pravil, pri tem lahko pravila dodajamo postopoma;
- (3) grobo gledano, produkcijska pravila funkcionalno ustrezajo modelu človeškega dolgoročnega spomina; dinamični del podatkov v podatkovni bazi pa človekovemu kratkoročnemu spominu.

Zanimiva je ugotovitev, da sistem z dodajanjem novih pravil običajno pridobi na kvalitoti, če pa pravila postopoma izločamo iz sistema, po dobimo model okrajšenega človeškega razmišljanja.

Pri gradnji ekspertnih sistemov so misli usmerjene v čim bolj kvalitetno delovanje sistema na specifičnem problemskem področju, zato so izbrane metode bolj ali manj strogo vezane na izbrano problemsko področje. Kljub temu so PS najobetavnejši in se pojavljajo v večini uspešnih aplikacij ekspertnih sistemov.

Prednosti pri uporabi PS so predvsem:

- (1) enostavno dodajanje in spreminjanje pravil;
- (2) znanje v obliki pravil je našeloma vedno dostopno vsakemu delu programa in s tem raste moč sistema;
- (3) specialni primeri kot npr. hierarhična kontrolna struktura omogočajo prilagoditev PS ustreznemu problemskemu področju;
- (4) PS so najmočnejše orodje za predstavitev znanja na področjih, kjer je znanje zajeto v velikem številu med soboj večinoma relativno neodvisnih pravil;
- (5) PS omogočajo "inteligentno" komunikacijo z uporabnikom, saj omogočajo razlago svojih sklepov, pojasnjevanje posameznih pravil itd. Pri tem imajo nekatera pravila priložen komentar, drugod pa znajo programi sami predstaviti pravila v naravni jezik.

Kljub razlikam vsi omenjeni sistemi spadajo pod pojem "produkcijski sistemi".

Nekateri strokovnjaki kot Newell (72) gledajo na PS ne samo kot na primarno orodje za raziskave človeškega modeliranja, ampak bolj kot na metodologijo, katere moč v veliki meri izvira iz velike podobnosti s človeškim razmišljanjem. Zdi se, da uspešna uporaba PS v ekspertnih sistemih ni samo slučajna, ampak da je strukturiranje znanja v PS učinkovita metoda za uporabo zelo velikih količin znanja. Podobne metode naj bi tekom evolucije razvili in izpopolnili naravni inteligentni sistemi (živa bitja), z odkritjem takih metod pa smo dobili v roke uspešno orodje za izgradnjo kvalitetnih ekspertnih sistemov.

Uspeh nekaterih sistemov umetne inteligence, ki bazirajo na PS, v določeni meri potrjuje zgornja razmišljanja. Prav tako je dejstvo, da je metodologija PS zelo učinkovita. Vendar ni povsem jasno, ali učinkovitost PS res izvira iz podobnosti s človeškim razmišljanjem in ali bi morali biti zaradi tega tudi inteligentni racionalistiški programi podobno grajeni.

Izkušnje kažejo, da je moč ljudi predvsem v specializaciji, v izkušnjah in znanju z določenega področja. Tako je malo verjetno, da bi bil šahovski velomajster hkrati vrhunski matematik ali kemik. "Ekspert" je strokovnjak s

specialnim znanjem o svojem področju; s specialnimi metodami in hevristikami. Odtod izvira ime "ekspertni sistemi".

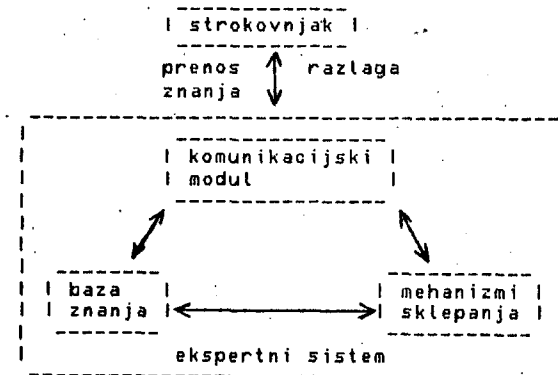
3. STRUKTURA IN DELOVANJE EKSPERTNIH SISTEMOV

V splošnem so ekspertni sistemi sestavljeni iz baze znanja o problemskem področju (knowledge base) in ustreznih mehanizmov sklepanja (inference engine). Strukturo ekspertnih sistemov prikazuje slika 2. Baza znanja vsebuje vse informacije o objektih in relacijah med objekti problemskega področja ter navodila, kako oz. kdaj uporabiti posamezne dele znanja. Mehanizmi sklepanja pa so algoritmi za uporabo tega znanja pri reševanju problemov in temeljijo na bolj splošnih in od problemskega področja manj odvisnih mehanizmih.



Slika 2: Osnovna struktura ekspertnih sistemov

Znanje v bazi znanja je predstavljeno tako, da omogočimo uporabniku enostavno vnašanje, spreminjanje in dopolnjevanje znanja; razumevanje ter razlago znanja in izpeljanih sklepov (rezultatov). Pogosto vsebuje vse to znanje poseben komunikacijski modul, ki omogoča uporabniku inteligentno interakcijo s sistemom v skoraj naravnem jeziku. Delovanje takega sistema prikazuje slika 3.



Slika 3: Prikaz delovanja ekspertnih sistemov

V nadaljnji obravnavi se bomo omejili na osnovno strukturo ekspertnih sistemov (glej sliko 2) ter razložili njihove sestavne dele in delovanje.

Ekspertne sisteme sestavljata dva osnovna modula: (a) baza znanja in (b) mehanizmi sklepanja.

(a) Baza znanja o problemski domeni je sestavljena iz (a1) znanja in (a2) podatkov.

(a1) Znanje vsebuje specifično znanje o problemski domeni. Tvorijo ga informacije o objektih domene in o relacijah med temi objekti. Včasih vsebujejo ekspertni sistemi tudi ti. meta-znanje, ki ga tvorijo informacije o uporabi tega specifičnega znanja. Del specifičnega znanja, ki mora biti v danem operacijskem ciklu sistema na voljo v procesu sklepanja, tvori ti. "aktivno znanje", medtem ko je preostali del specifičnega znanja v ti. "spečem" stanju. Specifično znanje je praviloma razbito v množico vzorčno vodenih modulov.

Učinkovitost delovanja ekspertnih sistemov je v veliki meri odvisna od predstavitve tega znanja. Znanje je lahko predstavljeno v obliki modelov, semantičnih mrež, pravil, itd. Izredno zanimiva in učinkovita je predstavitve znanja v obliki pravil, ki ne zahtevajo stroge formalizacije.

Čeprav je z izbiro predstavitve znanja že določena večina iskalnih mehanizmov in ustreznih hevristik, je koristno vključiti v sistem še dodatno znanje, ki še poveča učinkovitost delovanja sistema. To meta-znanje dodatno določa, kako naj se moduli aktivirajo, kako naj se izmed vseh modulov, ki ustrezajo danemu vzorcu podatkov, izbere in izvrši najustrenejši, itd.

(a2) Podatki so ali vhodni podatki ali vmesni rezultati, saj se podatki praviloma spreminjajo v vsakem operacijskem ciklu sistema. Podatki opisujejo tekoče stanje problema, tekoče kontekste, tekoče piane, zastavljene cilje in tekoče hipoteze, ki jih želimo dokazati.

Tudi za predstavitve podatkov imamo cel spekter možnosti: sezname, drevesa, mreže, pravila, itd.

(b) Mehanizmi sklepanja se sestojijo iz algoritmov za uporabo znanja pri reševanju problemov. To so različni iskalni mehanizmi (kot npr. alfa/beta iskanje v and/or drevesih, hevristične funkcije za evaluacijo vozlov v iskalnih drevesih) in različni drugi algoritmi, ki so v veliki meri vezani na obliko, v kateri je predstavljeno znanje. To so npr. algoritmi za primerjanje vzorcev v podatkih in modulih, algoritmi za izbiro in aktiviranje modulov, algoritmi za izvrševanje akcij, ipd. Pogosta in uspešna tehnika za reševanje problemov je "generiranje in testiranje", tj. generiranje hipotez, ki jih potem sistem preverja.

4. KRATEK PREGLED OBSTOJEČIH EKSPERTNIH SISTEMOV

DENDRAL

Z deli na ekspertnem sistemu DENDRAL so začeli leta 1965 v Stanfordu, ZDA (Buchanan, 77). Sistem stalno dopolnjujejo. To je verjetno največji projekt umetne inteligence, ki je tudi prvi dokazal, da je metode umetne inteligence mogoče učinkovito prenesti v prakso.

DENDRALova naloga je, da v interaktivni komunikaciji z uporabnikom (kemikom) ugotovi kemično strukturo neznane snovi. Za neznane snovi je namreč zelo težko ugotoviti njihovo strukturo. DENDRAL uporablja različne vire znanja: znanje o valencah (povezavah atomov), znanje o stabilnih in nestabilnih konfiguracijah atomov, pravila o razpadu snovi v masnem spektrometru, pravila o magnetni resonanci, pravila za planiranje in evaluacijo hipotez o neznani snovi in dodatno uporabnikovo znanje o neznani snovi.

Kemične strukture so predstavljene kot grafi: atomi so vozlišča, vezi med atomi pa povezave med vozlišči. Omejitve so podane kot podgrafi, ki se jim mora program izogibati. Znanje o razpadu snovi v masnem spektrometru je podano v obliki pravil:

<u>situacija:</u>	<u>veri, dogodka</u>	<u>akcija:</u>
določena konfiguracija atomov	=====>	razbijanje določene konfiguracije (razpad povezav podgrafa)
(podgraf)		

DENDRAL deluje v ciklih, sestavljenih iz faz "planiraj in generiraj"-"testiraj". Vsako od teh faz izvaja poseben modul. Modul "generiraj" (CONGEN) je kombinatorični algoritem za generiranje vseh topološko dovoljenih struktur. Modul "planiraj" (PLANNER) omejuje generiranje strukture glede na podane omejitve. Modul "testiraj" (MSPRUNE in MSRANK) nekatere izmed zgeneriranih struktur zavrže, ostale možne strukture

pa razvrsti (oceni) glede na znanje v sistemu. DENDRALova kvaliteta je na ravni najboljših strokovnjakov za snovi, za katere ima vloženo še specialno znanje. Za iskanje strukture brez merjenja z instrumenti pa je DENDRAL (oz. CONGEN) bistveno boljši od človeka. DENDRAL je v ZDA dostopen preko računalniške mreže TYMNET.

MYCIN

Tudi MYCIN je plod dolgoletnega dela strokovnjakov, ki se je začelo leta 1972 v Stanfordu (Shortliffe, 76).

MYCINova naloga je, da v interaktivni komunikaciji z uporabnikom (zdravnikom specialistom) postavi diagnozo povzročitelja infekcije krvi in meningitisa in svetuje najboljšo terapijo (doziranje zdravil). Komunikacija poteka v omejeni angleščini in s standardnimi ukazi.

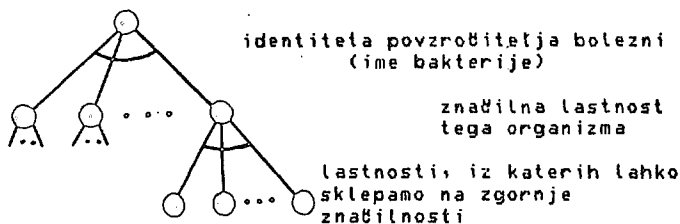
MYCIN vsebuje strokovno zdravniško znanje v obliki produkcijskih pravil kot je npr.

ČE: 1) BARVANJE ORGANIZMA JE GRAMNEGATIVNO, IN
2) OBLIKA ORGANIZMA JE PALIČASTA, IN
3) AEROBNOST ORGANIZMA JE AEROBNA
POTEM: OBSTAJA VELIKA VERJETNOST (0.8) DA SPADA ORGANIZEM V RAZRED ENTEROBACTERIACEAE.

Podatki v podatkovni bazi so četvorke oblike: (lastnost objekt vrednost faktor_zanesljivosti)
Primeri:

(IDENTITETA ORGANIZEM-1 E.COLI D.8)

MYCIN deluje po principu "generiraj in testiraj". Sistem išče povzročitelja bolezni tako, da primerja iskani organizem z vsemi znanimi organizmi iz podatkovne baze, pri čemer MYCIN vsakič razvije and/or drevo. Produkcijska pravila se torej v tem postopku vežejo v vzvratni smeri.



MYCIN preiskuje and/or drevo najprej v globino ("depth-first search"), zato za marsikateri organizem že kmalu ugotovi, da ne ustreza, in drevesa ni treba razvijati do konca.

MYCIN na svojem področju dosega in presega najboljše zdravnike-specialiste. Ugotovili so, da zdravniki največkrat predpisujejo antibiotike preširokega spektra in premalo pazijo na negativne medsebojne vplive zdravil, česar MYCIN ne dela. Kljub izredni kvaliteti pa MYCIN ni praktično v rabi, zlasti zato, ker zahteva velik računalnik in ker je težko sprejemljiv tako bolnikom kot zdravnikom.

DENDRAL in MYCIN sta klasični deli s področja ekspertnih sistemov, zato smo ju podrobneje opisali. Navedimo le še nekatere značilnejše primere ekspertnih sistemov.

META-DENDRAL je nastal kot del DENDRALa. Sistem skuša odkriti nova pravila o razpadu molekul v masnem spektrometru. Pravila so kreirajo na osnovi primerov spektrov že znanih molekul. Pri kreiranju pravil uporabnik interaktivno sodeluje.

TEIRESIAS je nastal pri izpopolnjevanju MYCINa. Sistem skrbi za enostavno dodajanje in spreminjanje pravil, za preverjanje protislovnosti pravil, za iskanje in odpravljanje napak in pomankljivosti v pravilih. TEIRESIAS je nadgradnja nad običajnimi ekspertnimi sistemi.

EMYCIN je nastal kot MYCINov modul. EMYCIN je ekspertni sistem za gradnjo ekspertnih sistemov. Njegovo delovanje temelji na spoznanju, da je mogoče z uporabo istega modula "mehanizmi sklepanja" z zamenjavo baze znanja dobiti nov ekspertni sistem za podobna problemska področja. Z uporabo EMYCINa so zgradili več sistemov: **PUFF** - program za ugotavljanje dihalnih obolenj z merjenjem izdihane sapa; **HEADHED** - program za psihofarmacijsko področje; **SACON** - program za ugotavljanje najboljših metod za testiranje materialov; itd.

PROSPECTOR je ekspertni sistem za odkrivanje nahajališč rud in nafte. Znanje v sistemu je podano v obliki pravil, ki so povezana v semantično mrežo. V tipični aplikaciji da PROSPECTOR kot rezultat geografsko karto, na kateri so označene verjetnosti nahajališč določene rude.

Omenimo še sisteme **MOLGEN** (svetuje genetikom, kakšni poskusi pri raziskavah DNA so najbolj obetavni); **AM** (poskuša odkriti nove zanimive matematične koncepte); **MACSYMA** (rešuje nekatere matematične probleme kot so reševanje diferencialnih enačb, simbolično integriranje, itd.); **EL** (sistem za diagnostiko električnih vezij); **AGE** (sistem z bazo znanja za gradnjo sistemov z bazo znanja); **SU/X** (sistem za razpoznavanje identitete objektov v prostoru); itd.

5. OCENA DELOVANJA EKSPERTNIH SISTEMOV

Omenili smo že, da je pri gradnji ekspertnih sistemov vse podrejeno čimboljšim rezultatom in s tem problemski domeni. To pa ne pomeni, da metoda ni važna, ampak da izbiramo take metode, od katerih pričakujemo najboljše rezultate. Prav tako to ne pomeni, da so ekspertni sistemi čista aplikacija teoretičnih dosežkov, tj. prenos teorije v prakso. Ne - ekspertni sistemi so velik dosežek umetne inteligence in so eno od najbolj cvetočih področij znanosti.

(a) Ocena ekspertnih sistemov z znanstvenega stališča

Najprej moramo odgovoriti na malce filozofsko vprašanje: "Ali so ekspertni sistemi inteligentni?" Odgovor je zagotovo "Da". Najboljši ekspertni sistemi dosegajo raven najboljših človeških strokovnjakov na področjih, ki zahtevajo inteligentnega pristop od kateregakoli nam znanoga bitja ali naprave. Pri tem nas ne sme motiti, da je ekspertni sistem pravzaprav računalniški program in kot tak vsaj teoretično predvidljiv, tj. da lahko do potankosti razložimo vsako navidez inteligentno operacijo (tega pri človeku ne moremo narediti). Poleg tega so znanstveniki prišli do prepričanja, da lahko pojem inteligence skoraj v celoti zamenjamo s predstavitvijo in uporabo znanja.

Najmočnejše orodje ekspertnih sistemov, produkcijske sisteme, so zelo uspešno uporabili za modeliranje človeškega razmišljanja. Tako so na ožjih področjih dostaj najuspešnejše posnamali človeško obašanje z vsemi najpomembnejšimi lastnostmi kot so učenje, pozabljanje, sklepanje, ...

Ekspertni sistemi so veliko prispevali k razvoju umetne inteligence (AI). Njihov razvoj je dokončno opravil z zmotnim prepričanjem, da morajo raziskovalci umetne inteligence odkriti globalne principe človeškega razmišljanja, s čemer bi že lahko pisali inteligentnejše programe, ki bi na izredno hitrih računalniških posnamali človeški (boljši) način razmišljanja. Ekspertni sistemi so pokazali, da ne gre za ono samo metodo, temveč da moramo poleg bolj splošnih metod izbrati tiste specialne metode, ki so ozko povezane s problemskim prostorom. Tako mora npr. robotski ekspertni sistem vsobovati znanje o dinamiki, senzorijskih in metoda za učinkovito

rokovanje z njimi. Tudi sistemi, ki imajo veliko skupnih lastnosti (npr. MYCIN in DENDRAL oba uporabljata produkcijska pravila) se običajno v konkretni izvedbi močno razlikujejo.

Velik korak naprej je bila ugotovitev, da lahko uporabimo iste mehanizme sklepanja na podobnih problemih s tem, da zamenjamo bazo znanja o problemski domeni.

Pri izgradnji zahtevnejših sistemov je praviloma sodelovalo več najboljših strokovnjakov v najbolj znanih svetovnih obeh umetne inteligence. Izdelava takih sistemov je vzela nekaj ali nekaj deset blovek-let dela. Vse to nam pove, da je gradnja kvalitetnih ekspertnih sistemov izredno zahtevno delo. Še nekaj: pogosto se v znanosti obeta dober skok naprej na določenem področju in takrat se veliko znanstvenikov loti dela na tem področju. To nam dokazujejo številna istočasna odkritja (npr. telefon; itd.). Zdi se, da so tudi ekspertni sistemi tako področje.

Delo na področju ekspertnih sistemov je prispevalo k umetni inteligenci kot znanosti vsaj naslednje:

1. Izdelavo metod za uspešno predstavitev in uporabo znanja in s tem tudi za hitrejšo izdelavo novih ekspertnih sistemov.

Za izdelavo ekspertnega sistema DENDRAL je bilo potrebnih okoli 50-60 blovek-let, za MYCIN 20-30. Ti sistemi so kot pionirski dosežki odprli pot nadaljnjemu razvoju. Danes nastajajo manjši sistemi v nekaj blovek-mesecih. To pohitritev vsaj za faktor 10 so dosegli predvsem z uporabo splošnejših, od problemskega prostora skoraj neodvisnih mehanizmov sklepanja, ki omogočajo hitro vlaganje specifičnega problemskega znanja. Lep primer je sistem EMYCIN, ekspertni sistem za gradnjo ekspertnih sistemov. Poleg tega so pri gradnji ekspertnih sistemov razvili množico novih metod za predstavitev in uporabo znanja. Te metode omogočajo gradnjo modularnih programov, ki jih je enostavno spreminjati, jim dodajati novo znanje, graditi učeče se sisteme, itd.. Največkrat pri tem ni šlo za iznajdbo novih metod, ampak za izvirne izboljšave prej večinoma teoretičnih metod. Ni dovolj, da poznamo metodo, dostikrat je pomembnejša konkretna izvedba metode, njena praktična uporaba tako je še zlasti na področju umetne inteligence.

2. Izdelavo metod za inteligentno komuniciranje z uporabnikom v skoraj naravnem jeziku.

Pri izgradnji ekspertnih sistemov so razvili tudi metode za inteligentno komuniciranje z uporabnikom, saj bi bil sicer celoten sistem za uporabnika nerazumljiv. Zato imajo običajno taki sistemi naslednje lastnosti:

- sistemi znajo popravljati brkovne napake (to lahko delajo predvsem zato, ker predvidevajo možne odgovore);
- njihov slovar besed je vezan na ozko specializirano področje in je kot tak uporabniku razumljiv in zadosten kljub manjšemu številu besedi;
- sistemi so sposobni pojasnjevati svoje odločitve;
- sistemi znajo odgovarjati na vprašanja o svojem delovanju, o vloženi znanju, itd. v skoraj naravnem jeziku.

(b) Ocena ekspertnih sistemov z uporabnega stališča:

Še se vprašamo, ali so ekspertni sistemi prinesli več dobrih, kot je bilo vloženega dela, je odgovor, merjeno samo z denarjem, seveda "ne". Sicer pa to velja za vse raziskovalne dosežke, dokler ne prodrejo v rutinsko rabo. Zdi se, da si ekspertni sistemi v zadnjih letih z velikimi koraki utirajo pot v vsakdanjo rabo. Realno je pričakovati, da bodo dobri ekspertni sistemi s stal-

nimi dodatki in izboljšavami dajali vedno boljše rezultate in da bodo v uporabi vsaj še nekaj deset let.

Naslednja značilna lastnost je, da so to zelo obsežni programi, praviloma kodirani v zelo zmogljivih programskih jezikih na velikih računalnikih. Najpogosteje uporabljeni jezik je INTERLISP, ki je v interaktivni rabi kar nekaj desetkrat počasnejši od FORTRANa (ta pa je nekajkrat počasnejši od zbirnega jezika). Težko bi rekli, da je zmogljivost kodiranja prenosorazmerna počasnosti, še zlasti zato, ker bolj zmogljivi jeziki omogočajo prijeme, ki v nižje-nivojskih jezikih praktično niso mogoči. Ti veliki programi z velikimi prevajalniki na velikih računalnikih onemogočajo prodor ekspertnih sistemov na široko področje. Druga negativna posledica je velika cena zahtevane programske opreme in obratovanja. Prekodiranje sistema v nižjenivojski jezik pa ni preveč obetavno, saj onemogoča enostavno spreminjanje, itd..

Ekspertni sistemi se ukvarjajo s problemi, značilnimi za velike računalniške centre (univerze, raziskovalni inštituti, itd.). Zdi se, da bo pomemben korak naprej omogočil razvoj računalniških mrež, ki širokemu krogu uporabnikov omogočajo uporabo najnovejših verzij programov. Primer naj bo računalniški program DENDRAL, ki je preko mreže TYMNET dostopen po celi ZDA.

Naslednja navidez banalna lastnost prav tako otežuje uspešen prodor ekspertnih sistemov. Medtem ko so ljudje računalniške programe hitro sprejeli v pomoč pri računanju (ker slabo računajo), pa so ljudje po naravi inteligentna bitja in tu ne čutijo potrebe po pomoči. Tako so bolj uporabni ekspertni sistemi (npr. MACSYMA, DENDRAL, itd.), ki za uspešno obvladovanje prostora potrebujejo inteligentnost in računanje hkrati.

Tudi posrednost komuniciranja negativno vpliva na sprejemljivost ekspertnih sistemov. Medtem ko ljudje s pomočjo svojih čutil učinkovito in neposredno komunicirajo med seboj, je komuniciranje z računalnikom preko vtipkavanja podatkov in odčitavanja rezultatov precej omejeno. Kot primer vzemimo MYCIN (sistem za ugotavljanje diagnoze in terapije infekcijskih bolezni). Šeprav je MYCIN-ova kvaliteta nesporna in jo presegajo le najboljši svetovni zdravniki specialisti, ravno to posredniško komuniciranje bistveno ovira sprejemljivost sistema. Predstavljam si samo, da zdravnik pomotoma odtipka napačno težo pacienta in že bo natančno izračunano doziranje zdravil dalo povsem napačne rezultate.

Vendar pa so ekspertni sistemi običajno bolj zanesljivi kot ljudje, saj niso podvrženi motnjam, boleznim, utrujenosti itd. Poleg tega so praviloma hitrejši. Najboljši ekspertni sistemi dosega in na nekaterih ožjih področjih celo presegajo najboljše bloveške strokovnjake. Omenimo vsaj tri zelo uspešne ekspertne sisteme. To so DENDRAL (program za ugotavljanje kemične strukture neznane snovi), MACSYMA (program za reševanje mnogih matematičnih problemov kot so simbolično integriranje, itd.) in MOLGEN (program za načrtovanje eksperimentov v molekularni genetiki). Kolikor nam je znano, so ti trije sistemi v vsakodnevni rabi. Z njimi si pomagajo uporabniki pri reševanju problemov, ki jih ne znajo rešiti sami - zanje je ta računalniška pomoč hitrejša, cenejša in lažje dostopna kot pomoč človeških strokovnjakov.

Visoko kvaliteto dosega še nekaj drugih sistemov, vendar ti še niso poželi večjih uporabniških uspehov (npr. MYCIN). Ti sistemi pa so z znanstvega stališča izredno zanimivi in je precej verjetno, da bodo z dodatnimi izboljšavami le našli tudi konkretno uporabo. Nekateri

drugi sistemi (npr. AM za odkrivanje novih konceptov v matematiki) pa so čisto znanstveno usmerjeni in so mišljeni predvsem kot raziskave.

7. ZAKLJUČEK

Ekspertni sistemi so velik raziskovalni dosežek umetne inteligence. To so sistemi, ki so sposobni inteligentnega reševanja problemov in na nekaterih ožjih področjih dosegajo kvaliteto najboljših strokovnjakov. Da bi zgradili dober ekspertni sistem, je potrebno vložiti veliko dela, zato je izdelava metod za hitro gradnjo kvalitetnih ekspertnih sistemov pomenila velik napredek. S tem se je poraba blovčkega bosa zmanjšala vsaj za faktor 10, hkrati pa se je povečala učinkovitost teh sistemov. Razen redkih izjem ekspertni sistemi še niso dali najboljših rezultatov v konkretnih aplikacijah, vendar lahko pričakujemo v bližnji bodočnosti velik razvoj. Obetavnost tega področja nam dokazuje poleg izrazitega znanstvenega napredka v zadnjih letih tudi naraščajoče število znanstvenikov, ki se s tem področjem ukvarjajo v tehnično najrazvitejših državah sveta.

Poročilo bo naši skupini služilo kot osnova za podrobnejše proučevanje ekspertnih sistemov in za načrtovanje konkretnih aplikacij. Trenutno načrtujemo dvoje ekspertnih sistemov, enega za demografsko napovedovanje in drugega za olajšanje samoupravnih sporazumov.

8. LITERATURA

1. Bratko I.: Predstavitev znanja v robotskih sistemih; delovno poročilo IJS DP-1697, Ljubljana, 1979
2. Buchanan B.G., Feigenbaum E.A.: DENDRAL and METADENDRALI v reviji Artificial Intelligence, Vol.11, No. 1,2, 1978
3. Davis R., King J.: An Overview of Production Systems v Machine Intelligence 8 (str. 300), Edinburgh, 1977
4. Feigenbaum E.A.: The Art of Artificial Intelligence: Themes and Case Studies of Knowledge EngineeringI v Proceedings of the 5th IJCAI (str. 1014), Cambridge, USA, 1977
5. Gams M., Lavrač N.: Ekspertni sistemi; delovno poročilo IJS DP-1867, Ljubljana, 1979
6. McDermott D.: The Last Survey of Representation of Knowledge v Proceedings of AISB/GI Conference on Artificial Intelligence, Hamburg, 1978
7. Shortliffe E.H.: Computer-Based Medical Consultation: MYCINI Elsevier Scientific Publishing Company, 1976
8. Waterman D.A., Hayes-Roth F.: Pattern-Directed Inference Systems; Academic Press, 1978
9. Winston P.H.: Artificial Intelligence; Addison Wesley Publishing Company, 1977

PRINCIPI IZGRADNJE SISTEMSKIH PROGRAMSKIH KOMPONENTI U MULTI- MIKROPROCESORSKIM SISTEMIMA

N. HADJINA

UDK: 681.3 : 519.682.6

SVEUČILIŠNI RAČUNSKI CENTAR, ZAGREB

U radu je dana izgradnja sistemskih programskih komponenti razmatrana sa stanovišta pridjeljivanja sredstava procesima u multi-mikroprocesorskim sistemima. Pod sredstvima su razmatrane kako sklopovske komponente tako i programske komponente multi-mikroprocesorskog sistema. Pod procesima su razmatrani sahtjevi za izvodenjem na sklopovskim komponentama kako na sistemskoj, tako i na korisničkoj razini. Također je razradjena kontrola pristupa tim sredstvima, komunikacija između tih sredstava, te različiti postupci pridjeljivanja sredstava kod izgradnje operacionih sistema i kompilatora za multi-mikroprocesorske sisteme.

PRINCIPLES OF SYSTEM SOFTWARE DESIGN FOR MULTIPLE-MICROPROCESSOR SYSTEMS: System software design, concerned from the standpoint of allocation of resources to processes in multiple-microprocessor systems, is given in this paper. Hardware components and software components are concerned as resources of multiple-microprocessors systems. All requests for execution, either from system or user level are concerned as processes. Control of resource accesses, communications between resources and diferent resource allocation procedures during operating systems and compilers design for multiple-microprocessor systems is also presented.

1. UVOD

Kako cijena mini i mikro-računala postaje sve manja, jasno se nameće potreba za projektiranjem takovih konfiguracija koje po nekim svojstvima (kapacitet računala, odnos cijene i performansi, fleksibilnost i dr.) mogu imati podjednake karakteristike kao veliki jednoprocesorski računarski sistemi. Glavni je problem koji se ovdje pojavljuje kako povezati te sklopovske komponente, te kako izgraditi programske komponente (operacijske sisteme, kompilatore i dr.), da bi se dobile željene karakteristike, posebno sa stanovišta prodjeljivanja sklopovskih i programskih komponenti takovih sistema.

Naziv više-procesorski sistemi može imati šire značenje, pa ga treba pojasniti. Više računarskih sistema instaliranih unutar iste sobe mogu se zvati više-procesorski sistem iako nisu fizički povezani. Naravno, pod tim nazivom treba podrazumjevati samo računarske sisteme ili procesore koji su fizički povezani. Prema tome se više-procesorski sistemi mogu dijeliti prema načinu na koji su procesori međusobno povezani.

Mrežni sistemi sastoje se od nezavisnih računarskih sistema koji su povezani preko zajedničkog komunikacionog kanala. Taj kanal može biti npr. telefonska linija, ili vrlo brz procesorski kanal. Glavna karakteristika mreže je da se svaki sistem pojavljuje prema drugome kao udaljeni uređaj, a razmjena podataka se obavlja preko za-

Jedničke veze.

Spojeni sistemi također se sastoje od nezavisnih sistema, a veza medju njima je ostvarena preko zajedničkih uređaja vanjskih memorija (diskovi, bubnjevi i dr.). Ti sistemi omogućavaju paralelno korištenje baza podataka, zbog razmjene velike količine podataka medju sistemima i dr. U ovim sistemima ti vanjski uređaji predstavljaju višestruko korištenje sredstava, te se na njih primjenjuju postupci pridjeljivanja i zaključavanja sredstava, koji su prethodno uvedeni.

Više-procesorski sistem, kako mu i samo ime kaže, predstavlja računarski sistem sa više procesora. Procesori koriste zajedničku memoriju i eventualne periferne uređaje. Ukoliko su klasični procesori zamijenjeni mikroprocesorima, tada se sistem naziva višestrukim mikroprocesorskim sistemom (MPPS). I u ovom sistemu se pristup višestruko korištenoj memoriji kontrolira postupcima zaključavanja na kritične odsječke u memoriji (sistemske tabele, mape pridjeljivanja i dr.)

Daljnja kategorizacija se može napraviti na osnovi međusobnog odnosa sistemskih programskih komponenti implementiranih na sklopovskoj konfiguraciji. Npr. mogu postojati specijalizirani procesori kojima se pridružuju specifični zadaci (program za upravljanje radom teleprintera,

procesor za obradu polja podataka) ili procesori kojima zadatke pridružuje jedan glavni procesor (dispečer) u sistemu.

2. PROJEKTIRANJE VIŠESTRUKIH MIKROPROCESORSKIH SISTEMA (μ MPS)

Da bi se dobio uvid u problematiku, te da se istraže mogućnosti primjene postupka pridjeljivanja sredstava unutar sistemskih programskih komponenti, potrebno je analizirati rad takovih sistema kroz:

- * pridjeljivanje sredstava (procesori, U/I kanali, memorije) procesima (zadacima);
- * kontrolu pristupa sistemskim sredstvima;
- * komunikaciju medju procesima.

PRIDJELJIVANJE SREDSTAVA PROCESIMA

Pridjeljivanje sredstava μ MPS-a, te sinhronizacija između mikroprocesora, predstavlja ozbiljan problem za projektiranje operacionog sistema μ MPS-a. To uključuje identifikaciju paralelnih procesa, dijeljenje procesa na zadatke, postavljanje sheme prioriteta izvodjenja, pridjeljivanje i raspoređivanje zadataka između različitih mikroprocesora, sinhronizaciju mikroprocesora kako bi proces završilo uspješno, dinamičko preraspoređivanje u slučaju greške procesnog modula (PM) i dr. PM predstavlja mikroprocesor (centralnu jedinicu ili U/I kanal). Takav operacioni sistem mora biti tako projektiran, da koristi postupke za realizaciju paralelne obrade, raspolaze s mogućnošću stvaranja paralelnih procesa, te da ima mogućnost detekcije paralelizma za vrijeme kompilacije i dr.

Za manji broj mikroprocesora (2-4) problemi još nisu preveliki za uspješno rješavanje, ali za veći broj procesora problem postaje vrlo kompleksan, te mu treba posvetiti veliku pažnju.

KONTROLA PRISTUPA SISTEMSKIM SREDSTVIMA

Procesi, zahtjevi za izvodjenjem na μ MPS-u, višestruko koriste zajednička sredstva radi povećanja performansi sistema. Pod sredstvima se podrazumijevaju sklopovske (procesori, memorija, U/I kanali, registri, sabirnice i dr.), te programske (programi, datoteke, spremnici, tabele i dr.) komponente sistema. Što je više takovih višestruko korištenih sredstava, zahtjeva se i veća (kompleksnija) kontrola za pridjeljivanje sredstava, te za detekciju i razrješenje nastalih zahtjeva medju procesima [1]. Velik broj tih sredstava i procesa povećava vjerojatnost pojave zastoja, te dovodi do smanjenja propusnosti cijelog sistema. Kod projektiranja sistema mogu se koristiti postupci pridjeljivanja i otklanjanja zastoja navedeni u [1].

Kontrola pristupa sredstvima može se provoditi:

- * arbitražom,
- * postavljanjem i uklanjanjem ključova (status bitova) i
- * prekidima.

Arbitražna - arbitar prima zahtjev od PM-a, analizira zahtjev i izvještava PM o njegovoj odluci. Centralni arbitar sastoji se od jedne samostalne sklopovske komponente (kontroler sabirnice, mikroprocesor za pridjeljivanje, IBUS arbitar i dr.). Decentralizirani arbitar je onaj u kojem je logika kontrole distribuirana po procesnim modulima koji su povezani sa višestruko korištenim sredstvima. To zahtjeva složeniji procesni modul, ali povećava pouzdanost u radu u slučaju greške. Oba pristupa koriste iste metode za arbitražu, postavljanje prioriteta, postavljanje ključeva i izazivanje prekida, postavljanje zahtjeva u lanac zahtjeva i dr. Izbor metode ovisi o zahtjevu na jednostavnost izvedbe, zahtjevu za posluživanje, vjerojatnosti pojave greške, kabliranju, blizini kontrolera i dr. Brzina rada arbitra mora biti takova da ona iznosi samo mali dio u ukupnom vremenu korištenja sredstava. Arbitar dijeljenog ciklusa brze memorije mora biti izveden na nivou sklopa, dok ulogu arbitra za dijeljivi blok memorije može imati mikroprocesor.

Ključevi - konflikti koji se pojavljuju na višestruko korištenoj memoriji, te na U/I kanalima mogu se razriješiti postavljanjem i uklanjanjem ključeva (status bitova). Procesor koji zahtjeva sredstvo ispituje postojanje ključa koji je, zapravo, indikator da je sredstvo zauzeto. Ako je sredstvo zauzeto, mikroprocesor mora čekati na njegovu oslobađanje. Ako je sredstvo slobodno, mikroprocesor ga zauzima i postavlja ključ. Po završetku rada uklanja ključ sa sredstva. Postupci zaključavanja mogu dovesti do pojave neželjenih situacija (zastoja), te ih treba prvenstveno spriječiti, detektirati i otkloniti ugradnjom postupaka pridjeljivanja u sklopovske i u programske komponente sistema [1]. Iz razloga pouzdanosti i sigurnosti u radu operacije postavljanja i uklanjanja ključeva (status bitova) moraju biti vremenski nedjeljive.

Ako se ključ primjenjuje na memoriju, tada je potrebno svakoj lokaciji omogućiti ciklus čitanja-izmjene-pisanja prije nego se omoguće drugi pristupi. To zahtjeva zaključavanje memorijskih adresa, što je puno teži zahtjev od zaključavanja cijelog memorijskog bloka, kada ostatak bloka ostaje neiskorišten za druge procese. Iz tih razloga ključevi se katkada i implementiraju preko skupa vanjskih registara koji sami provode ciklus čitanja-izmjene-upisa.

Prekidi se mogu koristiti za obradu internih grešaka procesora (paritet, krivi operacijski kod, adresa izvan dozvoljenog područja i dr.), vremenskih signala vanjskih uređaja (znak spremnosti za prijenos, prijenos izvršen i dr.)

ili za sinhronizaciju komunikacije medju procesima (korištenje memorije, preraspoređivanje zadataka ili uređaja i dr.)

KOMUNIKACIJA IZMEDJU SREDSTAVA

U μ MPS-u komunikacija izmedju sredstava u principu ostvaruje se preko memorijskog medija, poštanske kutije. Procesori i U/I kanali komuniciraju preko poštanske kutije, višestruko korištenog sredstva koje se sastoji od poruka, datoteka, zahtjeva za blokove i repove. Procesor koji odašilje, strukturiranu informaciju sprema je u poštansku kutiju. Procesor koji prima, pretražuje sadržaj kutije ili sredstvo koje odašilje informaciju indicira da postoji nešto za prijemni procesor. Upozoravanje procesora na prijem informacija može se provesti periodičkim upitima, status bitova, uvjetnim skokovima, instrukcijama prekida ili nekim drugim specijalnim instrukcijama. Status bit koji je u memoriji ili nekom vanjskom sklopu pokazuje da li je stanje u kutiji značajno ili ne. Pristup memorijskoj kutiji najčešće se određuje postupkom arbitriranja.

Glavni problemi μ MPS-a su nezavisni od mikroprocesora. U ovom momentu je tehnologija bitno ispred arhitekture i systemske programske podrške. Operacijski sistemi za μ MPS-a trebaju, dakle, efikasno riješiti slijedeće probleme:

- * Identifikacija paralelizma i podjela procesa u zadatke;
- * Pridjeljivanje sredstava zadacima;
- * Raspoređivanje procesora za paralelnu obradu procesa;
- * Medjuprocessorska komunikacija;
- * Detekcija i sprečavanje zastoja kada se pristupa višestruko korištenim sredstvima;
- * Identifikacija i izolacija greške, te raspoređivanje zadataka.

Iako je danas već napravljen napredak na ovim područjima, trend razvoja μ MPS-a biti će značajno u zakašnjenju usprkos niskoj cijeni mikroprocesora, U/I kanala, memorija (PROM, RAM) i dr., zbog sporog razvoja operacijskog sistema koji treba efikasno izvršiti navedene zadatke.

3. PROJEKTIRANJE SISTEMSKIH PROGRAMSKIH KOMPONENTI

Ovdje će se razmatrati struktura dviju glavnih systemskih programskih komponenti, operacionog sistema i kompilatora, sa stanovišta obrade paralelnih procesa u μ MPS-u.

3.1. PROJEKTIRANJE OPERACIONOG SISTEMA

Tip konfiguracije može biti vrlo različit, od jako povezanih sistema, gdje više procesora dijele svu memoriju, do vrlo slabo povezanih sistema, gdje su procesori dio nezavisnih sistema koji komuniciraju preko direktnih kanala za prijenos podataka. U slučaju višestrukog korištenja sklopovskih komponenti (npr. memorije, diska), mora postojati procedura zaključavanja koja dozvoljava da procesor pristupi sredstvu tako dugo dok ne završi kritični zadatak na toj komponenti. U slučaju mrežnih sistema ne postoje djeljive komponente koje su dostupne svim sistemima. Svako referenciranje sistema od strane drugog procesora predstavlja direktnu razmjenu podataka medju sistemima preko zajedničke komunikacione linije.

U tipičnom multiprogramskom sistemu sa jednim procesorom, više procesa ili zadataka koriste za izvodjenje jedan procesorski sistem. Proces se definira kao apstraktna cjelina koja se pomiče kroz instrukcije programa kako se one izvode na procesoru. Radna okolina procesa je zamišljena kao da se sastoji od brojača lokacija koji pokazuje na instrukciju u izvodjenju i od adresnog prostora u koji se smještaju instrukcije i podaci.

Operacioni sistem multipleksira jedan realni procesor izmedju takovih procesa koji se izvode na sistemu, tj. za svaki proces to izgleda kao da se izvodi na svom vlastitom procesoru. Dakle, ovi procesori mogu biti zamišljeni kao da komuniciraju na isti način kao sklopovski procesori, budući da postoji preslikavanje 1:1 izmedju procesa i virtuelnih procesora (multipleksni dio realnog procesora).

Prema tome, mogu se razmatrati procesi koji su povezani preko zajedničke memorije ili periferije kao u slučaju sklopovske komunikacije, ili preko direktnih kanala za razmjenu podataka kao u slučaju mrežnih sistema. Ako su oni povezani kao u prvom slučaju, tada pristup zajedničkim sredstvima mora ići preko postupaka zaključavanja, kao i u sklopovskoj realizaciji. U drugom slučaju taj problem iščezava.

POSTUPCI KOMUNICIRANJA U POSTOJEĆIM OPERACIJSKIM SISTEMIMA

Istraživanjem tipičnih sistema mogu se pronaći mnogi oblici razmjene podataka ili komunikacije izmedju različitih procesa ili komponenti sistema. Korisnički procesi (aplikacioni programi) u principu komuniciraju sa systemskim procesima (systemske datoteke i dr.) preko programiranih prekida za posluživanje, kao npr. otvaranje datoteke, punjenje programa i dr. Korisnički procesi gradi listu informacija u svom adresnom prostoru i prekida procesor predajući mu adresu i liste. Sistem prenosi informaciju na odgovarajući systemski proces, koji tada izvodi posao za korisnika i obično smješta odgovor u njegov adresni prostor.

Sistemske procese komuniciraju s drugim sistemskim procesima preko višestruko korištenih tabela, te postupaka zaključavanja na kritičnom kodu tabele. Sistemske procese također komuniciraju sa sklopovskim uređajima (sistemski program za disk na disk kontroler) punjenjem U/I registra ili postavljanjem liste U/I komandi, te postavljanjem U/I instrukcija za prekid kontrolera i njegovo korištenje. Konačno, u mnogim sistemima postoji i komunikacija između samih korisničkih procesa. Neki sistemi, kao što je rečeno, koriste poštansku kutiju koju jedan proces puni informacijom i upozorava druge procese da je čitaju. Drugi sistemi koriste višestruke ulaze na sklopovskim komponentama za komunikacije između procesa.

Projektiranje ovakvih operacijskih sistema, u uvjetima μ MPS-a, bi dovelo do preopterećenja operacionog sistema i korisničkog programa dodatnim kodom. Ta neujednačenost postupaka unutar sistema dovodi do problema sinhronizacije i raspoređivanja, te dovodi do strukture koja postaje nerazumljiva i teška za održavanje. Također se pojavljuje veliki problem uvođenja novih procesa u sistem.

DIREKTNJA RAZMJENA PORUKA

Iz gore navedenih razloga poželjno je da se projektiraju takvi operacioni sistemi gdje je razmjena podataka jednostavna i da se efikasno izvodi na μ MPS-u. Ovi ciljevi se mogu postići ako svi procesi u sistemu komuniciraju direktnom razmjenom podataka. Ukoliko se procesi ograniče da komuniciraju kao da su direktno povezani upotrebom dobro definiranih protokola, mnogi od navedenih problema mogu se riješiti. Sve metode neujednačenog komuniciranja zamjenjuju se direktnom razmjenom poruka. Problemi višestrukih pristupa memoriji, ključeva i modifikacija od nedozvoljenog ponašanja procesa nestaju. Reduciranjem raznih metoda komuniciranja na nivo direktne razmjene, kao u slučaju mrežnog sistema, sistem postaje lako prilagodljiv svim konfiguracijama μ MPS-a.

Bazični operacioni sistem izgleda, dakle kao brzi preklopnik poruka. Funkcija jezgre sistema je da prima poruke odašiljaoca i upućuje ih na određite. Procesni nisu ograničeni u izvodjenju postupcima zaključavanja, te su na taj način nezavisni, koliko je to moguće, od svih drugih procesa u sistemu. Time je omogućen maksimalan paralelizam u obradi procesa u μ MPS-u. Procesni se sinhroniziraju čekanjem na potvrdu prethodno odaslanih poruka. Svaki proces se izvodi u vlastitom adresnom prostoru i potpuno kontrolira pristup preko sheme za protokol razmjene poruka. Problemi zaštite sada leže u procesu koji prima poruku, koji ima mogućnost pretraživanja dospjelih podataka, te se na taj način sistem može obraniti od pogrešnih procesa. Budući da se sva razmjena podataka u sistemu izvršava na isti način, to je jednako pogodno slati poruke kako lokalnim, tako i udaljenim procesima.

TRANSPARENTNOST LOKACIJA PROCESA I HOMOGENOST KOMUNIKACIJE

Jedna od glavnih dobiti projektiranja sistema na osnovi prespajanja poruka je transparentnost lokacije odredišta prema odašiljaocu procesa. Odašiljaoc procesa uključuje kod odredišta (logičko ime ili identifikaciju) u poruku koja se prenosi primaocu. Jezgra operacijskog sistema je odgovorna da se poruka prenese primaocu. Prema tome, odašiljaoc procesa ne vodi brigu o memoriji i sistemskim lokacijama primaoca poruke. Ta transparentnost lokacija dozvoljava pomicanje procesa između memorijskih jedinica ili sistema, bez ponovnog programiranja, jednostavnim izmjenom lokacije procesa u jezgri sistema. Npr., korisnik koji želi komunicirati sa lokalnom datotekom sistema, treba samo promijeniti kod odredišta u poruci koju šalje sistemskim procesima za korištenje datoteke. Slično, proces koji je pristupio datoteci šalje informaciju natrag procesu koji je tražio uslugu, bez obzira gdje je taj proces lociran u μ MPS-u.

Druga dobit ove metode komuniciranja leži u testiranju programa. U postojećim sistemima, ukoliko se želi testirati novi sistemski proces, potrebno je zamijeniti stari ili dodati novi proces u programski vektor prekida, interno u sistemu. U ovom sistemu dodaje se novi proces kao i svaki drugi proces, a samo se promijeni kod odredišta u porukama koje su prethodno odaslane starom sistemskom procesu. Upotrebom ove metode, novi procesi za korištenje diska, datoteke i dr. mogu se dodati i testirati on-line, tj. korisničkim podacima za testiranje, dok svi ostali korisnički procesi i dalje koriste stari sistemski proces. Ovaj princip *homogene komunikacije procesa*, gdje svi procesi komuniciraju preko iste sheme za razmjenu poruka, dozvoljava pomicanje sistemskih procesa unutar cijele konfiguracije, pa čak i unutar sklopovskih komponenti, bez narušavanja rada cijelog sistema.

Sada se može definirati proces, tj. proces se sastoji od adresnog prostora (lokalne memorije) za smještaj instrukcija i podataka, procesora za izvodjenje tih instrukcija i komunikacionog kanala koji ga povezuje sa svim drugim procesima u sistemu. Za korisničke procese komunikacijski kanal je u biti simuliran jezgrom operacijskog sistema. Za sklopovski tip procesa (disk kontroler) to može biti stvarni U/I kanal preko kojeg se prenose poruke. Primjenom nove tehnologije (mikroprocesora) razumno je U/I kanale smatrati inteligentnim procesima koji komuniciraju sa stvarnim diskovima na jednoj strani i sa procesima koji zahtijevaju obradu na drugoj strani, preko sheme protokola za razmjenu poruka.

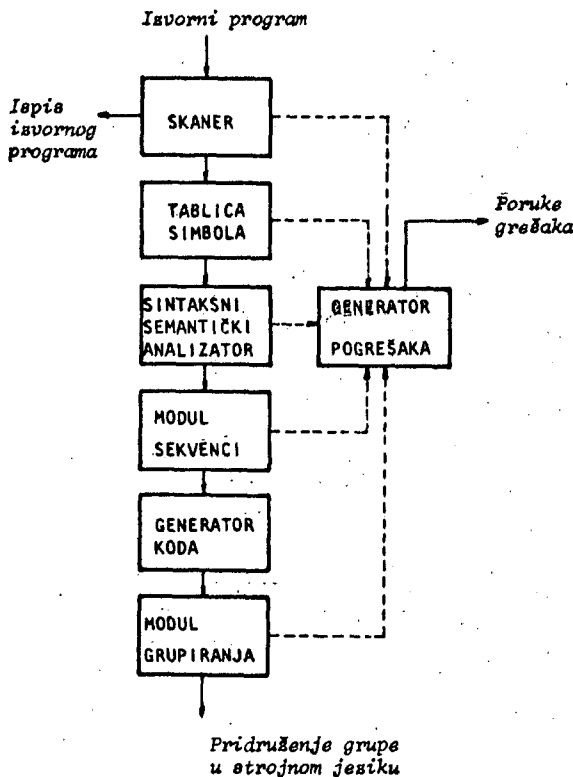
Princip direktne razmjene podataka kao metode komuniciranja unutar operacijskih sistema ima dva glavna cilja.

Prvi je metoda za projektiranje operacijskih sistema sa formalnom organizacijom upravljanja procesa i komunikacijom izmedju procesa. Drugi je da takovi operacioni sistemi primijenjeni na μ PS pružaju mogućnost efikasne distribuirane obrade.

3.2. PROJEKTIRANJE KOMPILATORA

Da bi se što bolje iskoristile potencijalne mogućnosti arhitekture μ PS-a, naročito u pogledu propusnosti sistema, potrebno je ovaj problem sagledati kao dekompoziciju na paralelne procese. Moguća su dva rješenja za dekompoziciju, funkcijska podjela i podjela prema zavisnosti podataka.

Prvo rješenje dijeli program u kontrolirane dijelove prema njihovoj funkciji sa što manjom komunikacijom i punom upotrebom raspoloživog paralelizma u arhitekturi μ PS-a. Tako npr., se kompilator može podijeliti na diskretne faze, a svaka faza izvodi jednu od funkcija. Faze se mogu organizirati za paralelnu obradu, kao na slici 1. Svaka faza je locirana na zasebnom procesoru. Naredbe izvornog programa prolaze kroz faze po redu, bez povratne veze.



Slika 1. Struktura organizacije kompilatora

Drugo rješenje dijeli program u nezavisne (paralelne) grupe prema zavisnosti izmedju varijabli u programu, ali se funkcija programa ne uzima u obzir. Grupe koje se paralelno izvode smanjuju ukupno vrijeme izvodjenja, ali dvije zavisne naredbe u različitim grupama izazivaju potrebu za izmjenom poruka i potencijalna čekaња. Ovo rješenje primjenjuje se u modelu grupiranja za sliku 1.

Organizacija kompilatora kao na slici 1. zahtijeva neka ograničenja na projektiranje. Prvo, da bi procesni moduli bili zaposleni, svaka faza izvodi po jednu naredbu u vremenu. Nakon što je naredba obradjena, transformirana naredba se prenosi u slijedeću fazu. Drugo, pojedinačnim fazama nije dozvoljen pristup globalnim tabelama, jer to unosi znatan utrošak sistemskog vremena. Treće, svaka faza može pristupiti samo lokalnoj memoriji procesora, iako je ukupna memorija sistema mnogo veća.

Da bi se smanjila komunikacija poruka, originalni izvorni program se ne šalje od faze do faze.

Moduli tablice simbola, sekvenci i grupiranja grade i analiziraju strukturu podataka, kako bi se odredila zavisnost izmedju naredbi. Modul tablice simbola treba informaciju o zavisnosti kako bi mogao detektirati grešku (nedefiniranost varijable i dr.). Ako je jezik programiranja nesekvencijalan, modul sekvenci koristi podatak o zavisnosti izmedju naredbi kako bi odredio pravi redoslijed naredbi. Modul grupiranja razmatra program kao graf. Čvorovi grafa su varijable, a strelica izmedju čvorova grafa označavaju zavisnost izmedju dvije varijable. Modul grupiranja dijeli graf u grupe presijecanjem strelica. Svaka pojedinačna strelica nužno zahtjeva poruku izmedju dvije varijable pridružene krajnjim čvorovima strelice. Pored minimalnog vremena za izvodjenje ovih nezavisnih grupa potrebno je osigurati i vrijeme za komunikaciju poruka izmedju varijabli povezanih strelicama.

Iako jedna centralizirana, složena struktura podataka za određivanje zavisnosti može biti implementirana za sva tri modula, ipak svaka faza gradi svoju lokalnu strukturu podataka. Zasebna struktura podataka je efikasnija i jednostavnija, ukoliko je podešena za specifične potrebe svakog modula. Što više, sve tri strukture podataka se konstituiraju i analiziraju istovremeno.

Budući da naredbe prolaze kroz kompilator u paralelnoj formi, naredbe se ne mogu dohvatiti od generatora sve dok modul sekvenci ne odredi poredak izvodjenja. To se može odrediti tek kad se analizira cijeli niz naredbi. Generator koda si ne može dozvoliti čekaње na taj poredak, pa umjesto toga svaka naredba koja dodje u generator koda, u poljskoj notaciji, se odmah translata u oblik zatvorenih programa. Kada modul sekvenci osigura poredak, tada ga generator koda prevodi u sekvencu poziva potprograma. Modul grupiranja, grupira potprograme i njihove pozive u grupe,

generira poruke izmedju grupa i pridružuje ih procesima u μ MPS-u.

U svakoj fazi se koristi dijagnostički mod za onemogućavanje izvodjenja narednih modula, ukoliko je otkrivena fatalna greška.

4. ZAKLJUČAK

Analiza postupaka pridjeljivanja sredstava μ MPS-a predstavlja značajan problem kod projektiranja kako na sklopovskom, tako i na programskom nivou. U biti se problem pridjeljivanja na jednom i drugom nivou može svesti na slijedeće. U μ MPS-u može postojati kontrola pridjeljivanja sredstava koja je ili centralizirana ili decentralizirana po pojedinačnim procesnim modulima.

Centralizirana kontrola znači, sa sklopovskog stanovišta, da postoji jedan procesor kojem je pridjeljena logika za kontrolu pristupa zajedno sa višestruko djeljivom memorijom (poštanska kutija), a sa programskog stanovišta znači da postoji jedna jezgra operacionog sistema u μ MPS-u kao nadzornog elementa (monitora) za upravljanje pristupa programskim i sklopovskim komponentama sistema.

Decentralizirana kontrola podrazumjeva, kako sa sklopovskog, tako i sa programskog stanovišta, da su funkcije i odgovornost svih procesora u μ MPS-u ravnopravne, te da svaki od mikroprocesora raspolaže sa logikom za kontrolu pristupa sklopovskim komponentama, te sa djeljivom memorijom i dijelom operacionog sistema mikroprocesora koji je odgovoran za komunikaciju te pridjeljivanje sredstava sistema procesima.

Izbor metode za kontrolu postupaka pridjeljivanja (centralizirana ili decentralizirana) prvenstveno ovisi od broja procesa (sklopovskih i programskih) koji moraju međusobno komunicirati. Za slučaj velikog broja procesa

poželjnija je decentralizirana struktura zbog smanjenja vjerojatnosti pojave zastoja, zbog distribuiranosti procesa na većem broju sredstava (logika, memorija). Kod malog broja procesa, jednostavnija je i jeftinija za izvedbu centralizirana struktura u kojoj svi procesi upisuju i čitaju informaciju sa jednog mjesta. U prvom slučaju su kritični odsječci (logika, memorija) distribuiranih po mikroprocesorskim sistemima, dok je u drugom slučaju kritični odsječak centraliziran. Ovakva analiza naročito je efikasna sa stanovišta kontrole pristupa sklopovskim komponentama.

Za ugradnju postupaka pridjeljivanja na nivou sklopovskih komponenti, potrebno je ugraditi jednostavne metode, koje jednostavnim operacijama dovode do rezultata za postojanje, sprečavanje i detekciju potpunog zastoja [1].

Kod projektiranja programskih komponenti, iz definicije procesa (program, proces za izvodjenje i komunikacioni kanal) razabire se potreba dinamičke kontrole većeg broja procesa (korisnički i sistemski), pa je pogodno koristiti metode sprečavanja detekcije zastoja razvijene na modelu grafa sistema [1]. Postupci pridjeljivanja razvijeni na takvom modelu, mogu se vrlo jednostavno implementirati na μ MPS u slučaju programskih komponenti preko jezgre operacionog sistema (monitora) μ MPS-a. Tako da se za ne preveliki broj procesa mogu primijeniti postupci pridjeljivanja koji su razvijeni na reduciranju grafa sistema ili na detekciji gnijezda u grafu sistema, dok se za vrlo složene slučajeve velikog broja procesa i ovdje mogu koristiti postupci razvijeni na detekciji ciklusa u grafu sistema [1].

REFERENCE:

- [1] N. Hadžina: POSTUPCI DETEKCIJE STANJA POTPUNOG ZASTOJA RAČUNARSKOG SISTEMA NA MODELU GRAFA SISTEMA, Informatika 79, 2, st. 14-17.

A FORTRAN IV SUBROUTINE FOR FINDING A BRACKETED ROOT

D. B. POPOVSKI

UDK: 681.3 : 51

DEPARTMENT OF ENGINEERING, UNIVERSITY OF BITOLA

In this paper a FORTRAN IV subroutine for finding a bracketed root is presented in which King's method F is used.

JEDAN FORTRAN IV POTPROGRAM ZA NALAŽENJE IZOLOVANOG KORENA. U ovom radu prezentiran je jedan FORTRAN IV potprogram za nalaženje izolovanog korena u kome se koristi King-ova metoda F.

King¹ gave a procedure, call it method F, for finding a bracketed root which may be described in the following manner:

Suppose that a real root of

$$y(x)=0 \quad (1)$$

has been bracketed by initial approximations x_1 and x_2 . Thus, $\text{sign}y_1 \neq \text{sign}y_2$, $y_1 = y(x_1)$.

Find x_3 by the secant step

$$x_3 = x_2 - \frac{y_2}{\frac{y_2 - y_1}{x_2 - x_1}} \quad (2)$$

and calculate y_3 . If $\text{sign}y_2 \neq \text{sign}y_3$, set $x_B = x_2$ and $y_B = y_2$. Otherwise set $x_B = x_1$ and $y_B = y_1$.

(a) Do the P-type step

$$x_4 = x_3 - \frac{y_3}{\frac{(y_3 - y_2) + (y_2 - y_1)}{(x_3 - x_2) + (x_2 - x_1)} - \frac{y_2 - y_1}{x_2 - x_1} + \frac{y_3 - y_2}{x_3 - x_2}} \quad (3)$$

provided this point is between x_3 and x_B . Otherwise get x_4 by the Snyder step

$$x_4 = x_3 + y_3(x_3 - x_B) / (0.5y_B - y_3), \quad (4)$$

remembering to replace y_B by $0.5y_B$. Calculate y_4 . If $\text{sign}y_4 \neq \text{sign}y_3$, set $x_B = x_3$ and $y_B = y_3$. In any case, replace (x_1, y_1) by

(x_{i+1}, y_{i+1}) , $i=1,2,3$. Return to (a).

This means that the step of type P is taken whenever possible, and only the three latest iterates are used even though they may all be on the same side of the root. Also one other iterate is retained—the opposite bracketed point x_B (x_B may, in addition, be one of the current iterates). The current point x_3 is used with x_B in the Snyder step whenever the P step fails to fall within the interval bounded by x_3 and x_B .

King shows¹ that method F is faster than Snyder's method², Muller's method³, the Pegasus method⁴, the Anderson and Björck's method⁵ and the Improved Pegasus method⁶.

The following is a FORTRAN IV subroutine in which King's method F for finding a bracketed root is used.

Description of parameters:

- Y - Name of the external function subprogram used.
- A - Initial approximation x_1 .
- B - Initial approximation x_2 .
- T - Tolerance for the relative error.
- R - Root of the equation $Y(X)=0$.
- M - Maximum number of iterations permitted.
- N - Error parameter coded as follows:
 - N=0 - No error.
 - N=1 - No convergence after M iterations.
 - N=2 - Basic assumption $\text{sign}y_1 \neq \text{sign}y_2$ is not satisfied.

Remarks:

The procedure assumes that function values at initial approximations A and B do not have the same sign.

The external function subprogram Y(X) must

be furnished by the user.

```

SUBROUTINE F(Y,A,B,T,R,M,N)
N=0
C=A
D=Y(C)
E=B
G=Y(E)
IF(D)1,19,2
1 IF(G)17,18,3
2 IF(G)3,18,17
3 H=E-C
O=G-D
P=O/H
Q=-G/P
S=Q+E
DO 15 I=3,M
U=Y(S)
IF(U)4,16,5
4 IF(G)7,18,6
5 IF(G)6,18,7
6 C=E
D=G
7 E=U-G
G=E/Q
H=Q+H
IF(H)8,12,8
8 P=(E+O)/H-P+G
IF(P)9,12,9
9 P=-U/P
R=S+P
IF(R-C)10,12,11
10 IF(P)12,16,13
11 IF(P)13,16,12
12 D=D*0.5
P=D-U
P=(S-C)*U/P
R=S+P
13 IF(ABS(R)*T-ABS(P))14,20,20
14 O=E
E=S
S=R
H=Q
Q=P
P=G
15 G=U
N=1
16 R=S
RETURN
17 N=2
18 R=E
RETURN
19 R=C
20 RETURN
END

```

Subroutine F was tested for a considerable number of examples on an IBM 1130 computer using floating-point arithmetic with 24-bit mantissa (standard precision) and 32-bit mantissa (extended precision). A selection of eight examples is given in table I.

REFERENCES

1. R.F.King: Methods without secant steps for finding a bracketed root, *Computing* 17, 49-57 (1976).
2. J.N.Snyder: Inverse interpolation, a real root of $f(x)=0$, University of Illinois Digital Computer Laboratory, ILLIAC I Library Routine H1-71, 4 pages, 1953.
3. D.E.Muller: A method for solving algebraic equations using an automatic computer, *MTAC* 10, 208-215 (1956).
4. M.Dowell and P.Jarratt: The "Pegasus" method for computing the root of an equation, *BIT* 12, 503-508 (1972).
5. N.Anderson and A.Björck: A new high order method of Regula Falsi type for computing a root of an equation, *BIT* 13, 253-264 (1973).
6. R.F.King: An improved Pegasus method for root finding, *BIT* 13, 423-427 (1973).

Table I. Numerical examples: $T=1.E-8$, extended precision is used.

	$y(x)$	A	B	R
1	$(x-1)x^2-1$	5.	0.	1.46557123
2	$(x-5)(x+1)^2-1$	0.	8.	5.02752466
3	$x(x-2)(x+1)^2-1$	5.	0.	2.05230034
4	$(x^2-1)x^3-10$	5.	0.	1.72099577
5	$(x-1)x^6-1$	3.	0.	1.25542287
6	$(x-3)(x+1)^3x^4-1$	0.	5.	3.00019282
7	$(x-2)(x+1)^4x^4-1$	0.	3.	2.00076963
8	$(x-5)x^9-1$	0.	6.	5.00000051

NEKATERI PRIPOMOČKI NAČRTOVANJA MIKORARAČUNALNIŠKIH SISTEMOV

D. MILJAN
P. KOLBEZEN

UDK: 681.3.06

INSTITUT JOŽEF STEFAN, LJUBLJANA

Na kratko so opisani nekateri pripomočki v materialni in programski opremi, ki so koristni pri načrtovanju mikroraračunalniških sistemov. Pripomočki so direktno uporabljivi pri načrtovanju sistemov, ki so osnovani na procesorju INTEL 8080, medtem ko so iznešene ideje tudi splošno uporabljive.

SOME AIDS FOR MICROCOMPUTER SYSTEMS DESIGN. In this paper some hardware and software aids for eases microcomputer design are described. Aids which are described here are directly applicable for Intel 8080 microprocessor based systems, but ideas carry out here for other microprocessor systems are applied too.

1. UVOD

V primeru, ko sami gradimo mikroraračunalnik ali želimo že obstoječi mikroraračunalnik nekoliko obogatiti, so nam koristne kakršnekoli ideje, ki nam olajšujejo projektantsko in programersko delo. Pripomočki, ki so opisani v tem prispevku, so prilagojeni za mikroraračunalnik INTEL 8080. Z določenimi prilagoditvami pa so uporabni tudi pri nekaterih drugih mikroraračunalnikih.

Opisani sta dve programski ter dve materialni rešitvi s programsko podporo. Prvi primer kaže, kako na enostaven način dosežemo relativne skoke pri mikroraračunalniku Intel 8080. Drugi primer pa opisuje pripomoček, ki omogoča hitro prepisovanje podatkov znotraj pomnilnika in predstavlja povsem programsko rešitev tega problema. Nadaljna dva primera opisujeta testiranje programske in materialne opreme mikroraračunalnika s pomočjo dodatne materialne opreme, ki jo dodamo standardnemu mikroraračunalniškemu vezju.

2. Relativni skoki

Pri programiranju na mikroraračunalniku 8080 hitro začutimo potrebo po relativnih skokih znotraj programa. S tem se izognemo togi vezanosti na določene adrese in dobimo možnost prestavljanja programa znotraj pomnilnika mikroraračunalnika.

Pogojni in brezpogojni skoki v naboru množice instruk-

cij procesorja 8080 omogočajo razmestitev programov povsod po pomnilniku. Vsi skoki so absolutni in določeni z šestnajstbitnim absolutnim naslovom.

Kratka rutina, sestavljena od nekaj instrukcij, pa omogoča spremembo vrednosti programskega števnika (PC) in s tem tudi relativen skok. Skok je možen naprej ali nazaj od trenutne vrednosti programskega števnika. Ker je velikost relativnega skoka shranjena v paru - registrov (DE ali HL), je možen skok na vsako lokacijo znotraj pomnilnika kapacitete 64 tisoč zlogov.

```
RJDE: SP HL      /vrednost PC v HL
      DADX SP     /prištej DE k HL
      SP<>HL      /nova vrednost v PC
      RET

RJHL: DADX SP     /prištej HL k PC
      SP<>HL      /nova vrednost v PC
      RET
```

Rutina, ki omogoča relativen skok, preprosto prišteje velikost relativnega skoka (ki je v paru registrov DE - pri prvi rutini, ali v paru registrov HL - pri drugi rutini) k trenutni vrednosti programskega števnika (PC). Željeni skok bo zapisan v obliki dvojiškega komplementa.

Pozitivni premik za X bo pomenil nadaljevanje izvajanja programa na lokaciji PC + X. Če je premik negati-

ven, pa se bo program nadaljeval na lokaciji PC - X.

2.1. Uporaba rutin relativnih skokov

Kadar želimo relativen skok v programu, bomo uporabili eno od zgoraj opisanih rutin. Pri prvi rutini RJDE mora biti vrednost relativnega skoka zapisana v paru registrov DE. Če uporabimo rutino RJHL, pa mora biti vrednost relativnega skoka zapisana v paru registrov HL.

Primer a: :
:
LXI DE RSKOK /vrednost rel.skoka v DE
CAL RJDE / klic rutine
:
:
Primer b: :
:
LXI HL RSKOK /vrednost rel.skoka v HL
CAL RJHL /klic rutine
:
:
:

Ko želimo brezpogojen relativen skok, uporabimo za klic rutine instrukcijo CAL. Če potrebujemo pogojni relativni skok (JTZ, JFC, ...), uporabimo za klic rutine ustrezni pogojni klic (CTZ, CFC, ...)

3. Hitro prenašanje blokov podatkov s pomočjo sklada

Pri uporabi kazalca na sklad, s katerim naslavljammo blok podatkov, ki ga prenašamo iz enega dela pomnilnika v drugi del, bomo potrebovali 40% časa manj, kot če bi isto nalogo opravljali s klasičnimi rutinami. Ker delo s skladom omogoča prenos dveh računalniških besed istočasno, in ker je inkrementiranje vrednosti kazalca na sklad avtomatično, je prenos podatkov s pomočjo sklada hitrejši, kot če bi podatke prenašali posamično, to je besedo za besedo.

Klasične rutine, ki prepisujejo bloke podatkov znotraj pomnilnika, navadno rabijo par register HL za shranjevanje naslova izvira podatka, par register (BC ali DE) za kazalec, ki kaže na ponor podatkov, ter register ali par register (B, C, D in/ali E) kot števniki preostalih zlogov. Pri takšnih programih, se podatek, ki je naslovljen z HL registrom, prenese v akumulator, potem pa na lokacijo, ki jo označuje register DE, ali BC. Pri vsakem koraku se dekrementira vrednost števnik zlogov. Ta se potem testira. Če je v števniku ničla pomeni, da je prenos končan, v ostalih primerih pa se povečajo vrednosti kazalcev na naslov izvira in ponora in opisani korak ponovi.

Pri mikroprocesorju 8080 je kazalec na sklad (sklad je lahko kjerkoli v pomnilniku) v bistvu kazalec na naslov v pomnilniku. Poleg tega, instrukcije za delo s skladom avtomatično inkrementirajo ali dekrementirajo

jo vrednost kazalca na sklad. Ko shranimo podatek v pomnilnik (z instrukcijo PUSH), se kazalec na sklad (SP) dekrementira. Ko pa preberemo podatek iz pomnilnika (instrukcija POP), pa se kazalec sklada avtomatično inkrementira. Torej, če bi pri klasični rutini nalogo shranjevanja naslova izvira podatkov prepustili kazalcu sklada, bi pri vsakem ciklu pridobili čas izvajanja instrukcije za inkrementiranje.

Rutina za prepis bloka podatkov:

```

PREP: LXI HL DZAC  /naslov cilja
      LXI SP SZAC  /naslov izvira
      XR A
      LAI DOLZ     /dolžina bloka
      RAR
ZANK: POPX DE      /podatki v DE
      LM E
      INX HL
      LM D
      INX HL
      DCA          /dekrement števca dolžine
                  bloka
      JFZ ZANK
      HLT
      END
  
```

Število besed, ki se prepisujejo, mora biti sodo. Rutina namreč prepisuje besede v parih. Iz programa je jasno, da se največji del časa porabi v zanki za prenos podatkov. Za prenos dveh besed se rutina izvaja 49,5 μsekund, medtem ko se klasična rutina izvaja 37 μsekund za prenos samo ene podatkovne besede. Torej prihrani opisani program 13 μsekund pri prenosu ene podatkovne besede.

Dodajmo še naslednjo možnost: Če želimo s klasično rutino za prenos prenesti več kot 256 besed, moramo uporabiti par register kot števniki preostalih zlogov. Pri tem se odkrivanje konca rutine za prepis (to je vsebine nič števnik zlogov) podaljša. Odkrivanje namreč podvoji čas izvajanja enega cikla zanke.

V našem primeru lahko z minimalno spremembo prvega dela rutine PREP prepisujemo do 512 računalniških besed.

```

PREP: LXI HL DZAC
      LXI SP SZAC  ZA VEČ KOT
      XR A         256 besed
      LAI DOLZ     }  SETC
      RAR          }  LAI XHH
                  }  RAR
ZANK: :
      :
      :
  
```

X^{HH} se izračuna po izrazu: $X^{HH} = DOLZ - 256$
pri tem je $DOLZ > 256$

4. Koračno izvajanje programa

Veliko pomoč pri testiranju, tako programske kot materialne opreme, daje možnost koračnega izvajanja programa. Za to je potrebna posebna materialna oprema, ki je izvedena s tremi integriranimi vezji, tipko, stikalom in svetlečo diodo. Glej sliko 1!

Tipka, stikalo in svetleča dioda je lahko montirana na prednji plošči mikroročunalnika. Koristen dodatek tej opremi je prikazovalnik podatkovnega in adresnega vodila, ki je lahko izveden z ustreznim številom alfanumeričnih prikazovalnikov. Ti naj bi kazali vrednosti na podatkovnem in adresnem vodilu.

Pri testiranju nove mikroročunalniške opreme (programske ali materialne), lahko s pomočjo vezja za koračno izvajanje programa le-tega v vsakem trenutku ustavimo in pregledamo stanje adresnih, podatkovnih in kontrolnih linij. Zatem nadaljujemo s koračnim ali navadnim izvajanjem programa.

Normalni izvajanje programa ustavimo s stikalom STOP. Ko je stikalo STOP v položaju "1", se program nemoteno izvaja, v položaju "2" pa se program ustavi. Ko je program ustavljen, se s tipko KORAK nadaljuje izvajanje programa po korakih. Kontrolna svetleča dioda je med koračnim izvajanjem programa prižgana. Tedaj je mikroročunalnik v stanju WAIT.

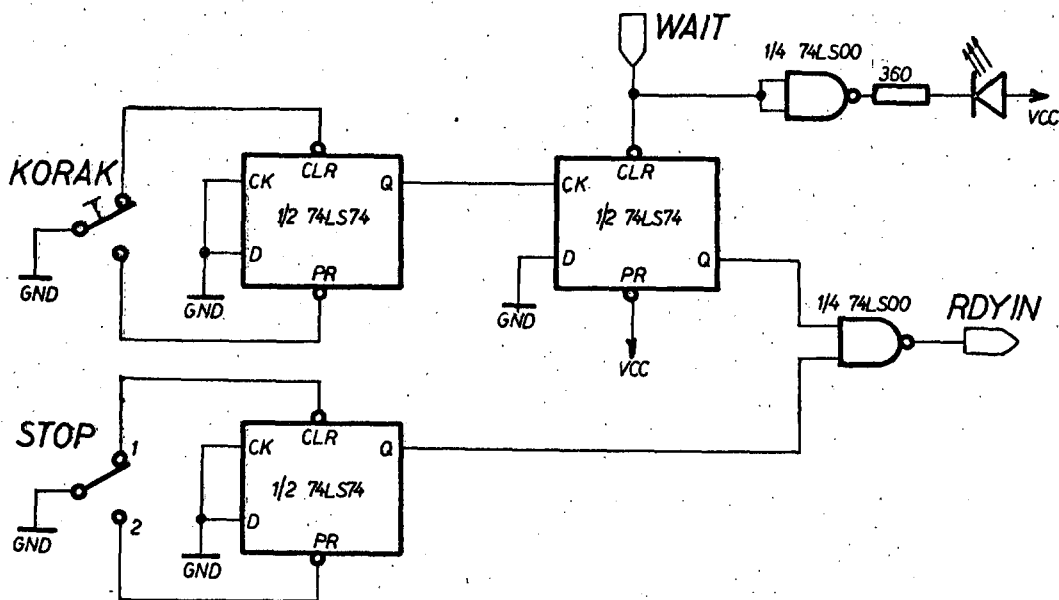
5. Materialna oprema za testno prekinitev programa

Prikazan je pripomoček, ki omogoča testno prekinitev programa pri mikroprocesorju 8080. Na sliki 2 je podana tehnična shema logike, ki omogoča testno prekinitev.

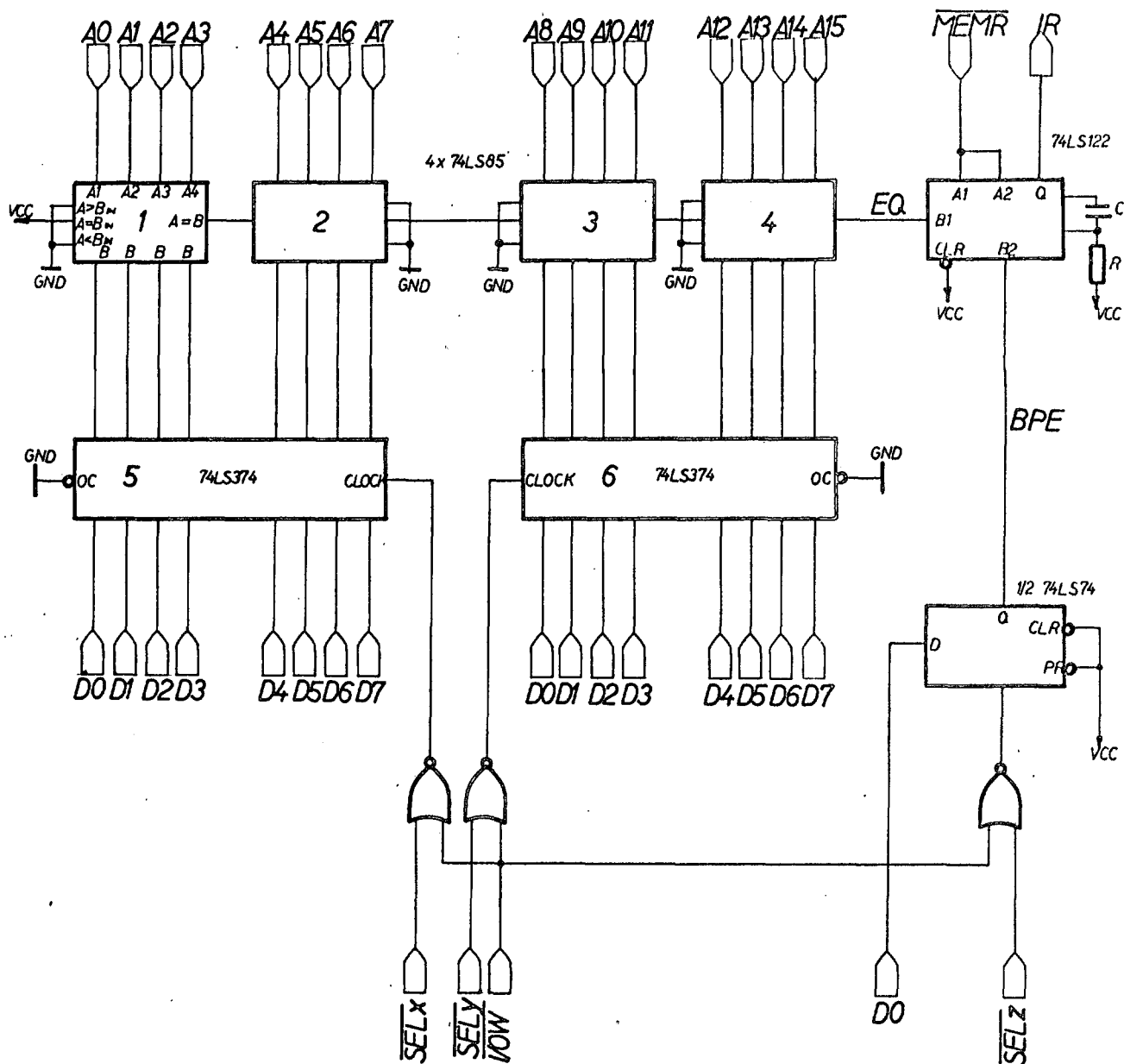
Adresa, pri kateri želimo prekinitev programa, se vpiše preko podatkovnega vodila. Vsebina podatkovnega vodila se vpiše v pomnilniške elemente 5 in 6, ko je aktivna ena od selekcijskih linij x ali y (\overline{SELx} ali \overline{SELy}).

Adresa, pri kateri želimo prekinitev, je določena s 16 bitno besedo. Ker je podatkovno vodilo osembitno, se bo адреса vpisovala v dveh korakih. Kadar je aktivna selekcijska linija \overline{SELx} , se vrednosti podatkovnega vodila vpiše v pomnilniške elemente 5, ki hranijo bite adrese z nižjo utežjo. Ko je aktivna selekcijska linija \overline{SELy} se vrednost podatkovnega vodila vpiše v pomnilniške elemente 6. Le-ti vsebujejo nižje adresne bite.

V času izvajanja programa, ki ga želimo testirati, se primerja vrednost programskega števnik (PC), oziroma vrednost adresnega vodila A0-A15, z adresno točko testne prekinitve. Enakost adres ugotavljajo komparatorji (1-4). Pri enakih adresah postane aktiv-



Slika 1.



Slika 2.

na kontrolna linije EQ. Ko se pojavi v programskem številku adresa točke testne prekinitve, prekinitvena logika testira še pogoja $\overline{\text{MEMR}}$ in BPE.

Linija BPE (break point enable) postane aktivna, če pred začetkom izvajanja testiranega programa vpišemo s selekcijske linije $\overline{\text{SELZ}}$ vrednost $\text{DO} = 1$.

Ko so izpolnjeni vsi omenjeni pogoji, bo kontrolna prekinitvena logika poslala mikroračunalniku prošnjo za prekinitvev IRx (interrupt request).

Nadaljnji potek testiranja je odvisen od organizacije prekinitvene logike konkretnega računalnika. V splošnem bo prekinitvev povzročila nadaljevanje

programa v prekinitveni testni rutini.

Če zbirko v EPROMu shranjenih pomočnih rutin obogatimo še z naslednjo kratko rutino za vpis vseh pogojev v kontrolno logiko, postane določanje točke testne prekinitve z opisano dodatno materialno opremo za testno prekinitvev zelo enostavna.

```
BRPT: PSHX PSW /shrani ACC na F
      LA L
      OUT x /vpis L-adrese
      LA H
      OUT y /vpis H-adrese
      LAJ 1
```

```

OUT z /enable B.P.
POPX PSW
RET

```

Potek testiranja je naslednji:

- i) Na naslov določene prekinitvene (interrupt) rutine vpišemo testni program s določenimi nalogami (izpis vseh ali le določenih registrov, izpis stanj v V/I pomnilniških elementih in podobno),
- ii) V par registrov HL vpišemo addresso željene točke testne prekinitve ter pokličemo rutino BRPT, kot kaže primer:

```

:
:
LXI HL nasl
CAL BRPT
:
:

```

- iii) Začnemo z izvajanjem programa, ki ga testiramo.

Prepoved aktivnosti opisane logike dosežemo tako, da vpišemo $DQ = 0$ s pomočjo selekcijske linije \overline{SELZ} (LAI 0; OUT z). Po tem koraku se bo program izvajal nemoteno dalje.

6. ZAKLJUČEK

Opisane ideje so podane dovolj natančno in kot take takoj uporabljive pri razvoju mikroračunalniških sistemov, ki so osnovani na procesorju INTEL 8080. Opisane rutine so lahko shranjene v mikroračunalniškem pomnilniku EPROM, saj zasedejo malo prostora, medtem ko se pogosto uporabljajo.

Materialno opremo za končno izvajanje programa ter opremo za testno prekinitvev programa lahko realiziramo na eni sami kartici tiskanega vezja, ki vsebuje enajst integriranih vezij.

7. LITERATURA:

Contol Logic, Inc. Priročniki za mikroračunalnik
INTEL 8080, November 1974.

RAČUNALNIŠKI VID: OBRAVNAVA VIZUALNIH INFORMACIJ IN ANALIZA SLIK

G. KANDUS
M. ŠPEGEL

UDK: 681.3 : 007.52

INSTITUT JOŽEF STEFAN, LJUBLJANA

V članku je omenjenih nekaj algoritmov in metod ter podan pregled teorije računalniškega vida. Poudarek je na procesiranju vizualne informacije in na analizi scene. Še posebej je računalniški vid samostojno področje, so v zadnjih letih pospešeno razvijajo predvsem kot najsplošnejši senzor inteligentnih robotskih sistemov.

COMPUTER VISION: In the article some algorithms and methods are mentioned and an overview of computer vision theory is given. The emphasis is on visual information processing and on scene analysis. Though the computer vision is an independent field, its fast progress in last few years is made as most general sensor for intelligent robot systems.

1. Uvod

Pri človeku je gledanje neposredna iskušnja, ki ji ni mogoče priti do konca s še tako skrbno introspekcijo. Med informacije, ki jo sprejemajo naša oči in podoba, ki si jo na podlagi nje izoblikujejo možgani, je še precej korakov zavutih v možgane. Nevrofiziologija ve marsikaj o individualni živčni celici, a le malo o tem, kako celice kontaktirajo med seboj pri prenašanju, spreminjanju in zapisovanju informacije, kar je bistveno za razumevanje problemov gledanja. Biološki sistem gledanja je prekompleksen, da bi ga bilo mogoče simulirati.

Računalniški vid je trd oreh. V literaturi je mogoče zaslediti vrsto člankov, ki so problemsko orientirani in opisujejo različne algoritme, metode in tehnike, ki so uporabne ob določenih predpostavkah, ni pa mogoče najti dela, ki bi nudil celovit pregled in analizo že znanih postopkov ali pa skušal utemeljiti teorijo računalniškega vida.

Naloga sistema za računalniško gledanje je, da opiše tisto, kar vidi. Vhod je ona ali več slik, ki jih določajo dvodimenzionalna polja vrednosti (običajno intenziteta svetlobe, včasih barva), ki nam jih posredujejo senzorji (običajno TV kamera) iz območja zornega kota (scene). Izhod je običajno v simbolni obliki. Održati mora neki aspekt tri dimenzionalne realnosti in biti pripravljen za izvajanje določenih nalog.

Področja, ki se deloma prekrivajo s področjem računalniškega gledanja so:

- procesiranje slik
- prepoznavanje vzorcev
- analiza scene.

Kratek opis teh področij:

Procesiranje slik je spreminjanje in bogatenje slikovne vsebine. Je odino od zgoraj omenjenih področij, ki ima sliko za vhod. Žal pa ima sliko tudi za izhod. Metode, ki jih procesiranje slik uporablja kot svoja glavna orodja (linearno, prostorno invariantno metodo in transformacijsko

tehnike), igrajo pri računalniškem gledanju le omejeno vlogo.

Bistvena metoda, ki jo uporablja prepoznavanje vzorcev, je klasifikacija vzorcev, ki prestikujejo vektorje v celo števila. Komponente vektorjev predstavlja množico vrednosti, celo števila pa označujejo razrede, katero množico produkti pripadajo. Za računalniško gledanje so zanimive tehnike, ki se uporabljajo za izračun numeričnih vrednosti značilnosti, izločenih iz slikovnih podatkov.

Analiza scene spominja in dopolnjuje opis slik v bolj zapletene in uporabne opise. Ena od prvih nalog tega področja je bilo iz prvotne slike dobiti risbo, sestavljeno iz črt mojih ploskev predmetov. Da bi poenostavili komplicirano analizo prvotne slike so raziskovalci pričeli uporabljati metode, ki izkoriščajo pridobeno znanje o sceni. Na ta način so je mogoče dokopati do ustrezni opisev tudi na podlagi raznorodnih nepopolnih ali motenih podatkov. Paziti pa je treba, da ne opisujemo tistega, kar želimo videti ali kar pričakujemo, namisto dejanske scene.

2. Procesiranje vizualne informacije

Pri vseh treh prej omenjenih področjih pogosto uporabljamo metode, ki bi podatke o intenziteti prvotne slike prevedle v simbolni opis. Problem je tudi, v kateri fazi procesa naj se zgodi ta transformacija. Nevarno raziskavo sugerirajo prehod na simbolni opis v začetnem delu procesiranja vizualne informacije. Opis sestavlja relativno velike število osnovnih podatkov, iz katerih izhaja vsa poznejša analiza.

2.1 Izračun osnovne slike

Številka skoraj tako dobro zaznava predmete, ki jih vidi na fotografiji, kot so jih vidi narisan, šeprov je informacije, ki v obliki intenzitete svetlobe pade na očesno mrežnico

v obeh primerih bistveno različna. To nas napeljuje na misel, da risarjevi simboli odgovarjajo simbolom, ki si jih človek izračuna pri tem, ko si skuša razložiti sliko. Torej ne izgubimo mnogo informacije, če vrednosti, ki ustrezajo intenziteti slike transformiramo v opis, kako se te vrednosti spreminjajo na robu približno homogenih območij. S tem zelo zmanjšamo količino podatkov potrebnih pri nadaljni analizi slike ob ne bistvenem zmanjšanju množine informacije.

Nekaj osnovnih pojmov, ki sestavljajo simbolni opis:

- rob
- linija
- okroglina

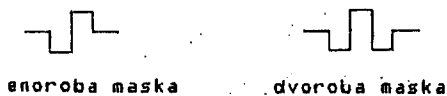
Njihovi parametri pa so:

- orientacija
- velikost
- kontrast
- pozicija
- zabrisanost

če odkrijemo na sliki rob predmeta, ga zapišemo v simbolni obliki na primer takole:

(ROB(POZICIJA(34 48)(73 48))
(KONTRAST 34)
(NEJASNOST 17)
(ORIENTACIJA 0))

Herskovitz in Binford (1970) sta empirično ugotovila, da so stopnice, izbokline in strehasti profili najpogostejše oblike spremembe intenzitete svetlobe, ki prihaja s scene, ki je sestavljena iz poliedrov. Znano je, da je človeški vid občutljiv ne le na spremembo, ampak tudi na hitrost spremembe intenzitete. Približke prvemu in drugemu odvodu intenzitete slike lahko dobimo s konvolucijo med lokalnimi vrednosmi intenzitete ter enorobo masko in dvorobo masko v obliki palice. Enoroba maska meri gradient intenzitetnega polja v določeni smeri; dvoroba maska pa lokalne spremembe gradienta intenzitete (Slika 1).



Slika 1

Algoritem za izračun simbolnega opisa lahko strnemo v naslednje štiri korake:

- poišči in primerjaj med seboj vrhove dobljene s konvolucijo in maskami različnih širin
- določi značilne vrhove
- grupiraj bližnje vrhove
- iz konfiguracije vrhov, ki so posledica sprememb intenzitete, določi tipe osnovnih pojmov (rob, linija, okroglina) predstavljajo ter njihove parametre (kontrast, pozicija, zabrisanost, itd).

Tako dobljeno skico imenujemo izvirno osnovno skico. Popolno osnovno skico dobimo z grupiranjem pojmov. Grupiramo ne le enake pojme, ampak tudi pojme, ki se po legi in orientaciji ujemajo s sosednjimi elementi. Kratke segmente tretiramo kot šum, če so šibkejši in jih upoštevamo, če so močnejši od daljših segmentov, ki jih sekajo. Zelo kratke samostojne segmente ignoriramo. Včasih nam pri presoji ali naj vključimo segment v skico ali ne, pomagajo abstraktno zanižljene linije. Pričakovati je namreč, da tudi kontne točke linij, čeprav nepovezane, leže na gladkih krivuljah.

2.2 Ločevanje predmetov od ozadja

Naslednji računalniški problem pri procesiranju vizualne informacije je segmentacija. Na tej stopnji se pojavi problem, kako ločiti predmet od ozadja ali predmete med seboj. Ločevanje se običajno izvede z diferenciacijo prvega reda ali z grupirnimi operacijami elementov osnovne skice.

Linjski zbir pomeni grupiranje površinskih elementov, tako da se orientacija grupe ujema z orientacijo elementov. To je v bistvu eno dimenzionalno grupiranje, ki je zelo važno, kajti tudi večja območja se najlažje poiščejo z določenjem njihovih meja.

Theta zbir se razlikuje od linijskega po tem, da se notranja orientacija elementov, ki jih grupiramo, razlikuje od orientacije grupe kot celote.

Elemente osnovne skice lahko grupiramo tudi na podlagi statističnih zakonitosti, če zberemo naslednje podatke:

- celotna dolžina črt, število okroglin, število različnih kontrastov
- celotno število elementov in celotna dolžina črt za posamezne orientacije
- porazdelitev parametrov velikosti
- porazdelitev kontrastnih parametrov.

če se izkaže, da neka lastnost prevladuje, je to dodatna informacija o obliki in orientaciji grupe, če predno ugotovimo točne obrise območja.

Nekateri avtorji (Bacjy (1972)) so uporabljali za ločevanje Fourierjevo analizo. Nekajkrat je bilo mogoče s pomočjo močnostnega spektra ločiti območja z različnimi strukturami. Drugi avtorji so uporabljali razne diferencialne operatorje. Eden najstarejših je Robertsov gradient (Roberts (1963)).

2.3 Uporaba rezultatov začetne faze procesiranja

a) Stereo tehnika:

Na razpolago imamo dve sliki scene, posneti iz dveh bližnjih točk na isti vodoravni višini. Sliki se rahlo razlikujeta, toda vsaki točki na eni sliki odgovarja točka na drugi sliki. Iz relativnih razlik, ki jih izračunamo, je mogoče uoločiti globino, to je razdaljo opazovane točke od točke opazovanja.

Stereo problem se tako prevede na primerjavo dveh prvotnih opisov. Pogoja, katerima mora biti zadoščeno, sta:

- enkratnost: vsaki točki na eni sliki ustreza natanko ena točka na drugi sliki
- zveznost: globina predmetov se zvezno spreminja.

Marr, Poggijo, Palm (1977) razvijajo metodo za izračunavanje lokalnih razlik para stereogramov, znano pod imenom kooperativni algoritem.

b) Ugotavljanje oblike predmetov s pomočjo gibanja

če se predmet giblje glede na opazovalca, lahko iz razlik na zaporednih posnetkih dobimo informacijo o njegovi obliki. Ullman (1979) razdeli problem na:

- primerjavo elementov na časovno zaporednih posnetkih

- ugotavljanje oblik predmetov s pomočjo neritov o spremembi njihovega položaja

Ullman postavi teoreme, ki pravi, da lahko določimo lego togega telesa v prostoru, če poznamo lego štirih njegovih točk, ki ne leže na isti ravnini, če jih opazujemo iz treh različnih smeri. Iz teorema izpelje metodo za izračun oblike gibajočih se predmetov, ki daje rezultate za katere trdi, da prekašajo človeško zmogljivost določanja oblike s pomočjo podatkov o gibanju teles.

2.4 Nivojni opisov

Za območno orientirane predstavitve ločimo naslednje nivoje opisov:

- pixel: slikovni element
- krpa: skupina bližnjih slikovnih elementov, ki imajo podobne lastnosti
- območje: smiselna skupina krp, ki odgovarja istemu predmetu
- podslika: del slike, ki odgovarja predmetu ali množici predmetov
- slika: predmeti in njihovo ozadje

Linjsko orientirane opise dobimo, če zamenjamo

- krpo z delom linije
- območje z linijo
- podsliko z množico linij, ki odgovarjajo predmetu

Območno orientirani opisi so primernejši za opis zaobljenih predmetov, medtem ko so linjsko orientirani opisi primernejši za opis planarnih predmetov.

Grobo lahko trdimo, da se nižji nivoji ukvarjajo s konkretnimi podatki, medtem ko se višji nivoji ukvarjajo s simboličnimi abstrakcijami. Podatkov na spodnjih nivojih je ogromna množina, če pri sceni, ki jo analiziramo, uporabimo določeno znanje, ki je bodisi splošno in opisuje splošno znana dejstva bodisi posebno in opisuje, kar vemo o sceni, lahko občutno zmanjšamo količino podatkov za opis predmetov pri prehajanju iz nižjih na višje nivoje. S tem zelo zmanjšamo potrebne računalniške kapacitete in pridobimo na času računanja, kar je pomembno, ker morajo biti rezultati pri računalniškem gledanju čisto dobljeni v realnem času.

3. Analiza scene

Raziskovanje analize scene je potekalo v dveh smereh. Zaradi nerazvitih matematičnih orodij so se najprej pojavile analize onostavnih scen, sestavljenih iz pravih geometrijskih teles, v večini primerov iz poliedrov. Pionirji na tem področju so bili Guzman (1968) in Clowes (1971), Huffman (1971) in kasneje Waltz (1976).

Simbolni opis scene je bil podan kot črna risba (projekcija) teles v splošni legi. Rezultat analize scene pa je bila lista, iz katere je bila vidna pripadnost površin črne slike posameznim predmetom scene.

Iz želje po analizi realnih scen, za katere poliedrska aproksimacija ne zadošča, se je razvil opis s tako imenovanimi posplošenimi valji, na katere je prvi opozoril Binford (1971).

Marr (1979) v svojem preglednem članku ne omenja analize scene, ampak govori o končni fazi procesiranja vizualne informacije.

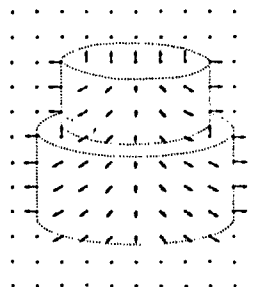
Črtni sliki odgovarja dvo in pol dimenzionalna skica; opisu teles pa tri dimenzionalna predstavitev.

Končna faza procesiranja vizualne informacije ima nalogo, da s pomočjo informacije o legi in položaju površin, katerih opis je odvisen od mesta opazovanja, pridemo na opis, ki ni odvisen od opazovališča in je z njegovo pomočjo mogoče prepoznati oblike predmetov iz različnih oddaljenosti in pod različnimi koti. Središče koordinatnega sistema se pri tem premakne od opazovalca k predmetu.

Lahko je opaziti, da vsebujejo slike velik del informacije o tridimenzionalni naravi njenih površin. Horn (1975) pokaže, kako je mogoče razviti diferencialne enačbe o osvetljenosti slike tako, da lahko iz gradienta osenčenja ugotovimo obliko predmetovih površin, če poznamo njihovo odbojno funkcijo.

3.1 Dvo in pol dimenzionalna skica

Velikokrat je koristno upoštevati podatke dobljene iz več virov hkrati: iz stereo slik, iz analize gibanja, barve in osenčenja ter jih kombinirati. Predstavitve, ki eksplicitno kaže obliko in orientacijo površin imenujemo dvo in pol dimenzionalna skica (Slika 2). Orientacijo površin predstavljajo puščice, ki pokrivajo sliko. Dolžina puščice je sorazmerna nagljenosti ravnine v tisti točki glede na os opazovanja.



Dvo in pol dimenzionalna skica

Slika 2

Cilj te faze procesiranja vizualne informacije je odkriti, kakšne so orientacije površin na sliki, kateri obrisi v osnovni skici odgovarjajo nezveznostim površin v dvo in pol dimenzionalni skici in kateri obrisi manjkajo v osnovni skici in jih je treba vstaviti v dvo in pol dimenzionalno skico tako, da je zadoščeno zakonom tri dimenzionalnega prostora.

3.2 Tri dimenzionalna predstavitev

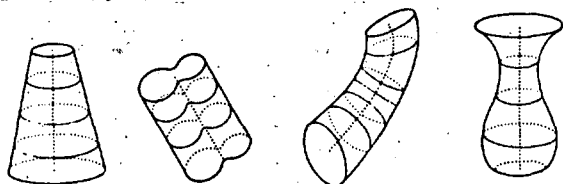
Pri tri dimenzionalni predstavitvi predmetov nas zanima predvsem:

- masno težišče
- površina, volumen in razsežnost
- rotacijske in zrcalne simetrijske osi.

Zato je ugodno, če je opis predmetov prostorski ter v koordinatnem sistemu, ki jo za opisa naraven. Tem zahtevam se približamo z opisom s posplošenimi valji (Slika 3).

Posplošeni valj (Binford 1971) je ploskev, ki jo opiše zaključena ravninska krivulja pri tem, ko ravnina te krivulje potuje vzdolž neke osi. Krivulja ni nujno, da je krog in os ni nujno, da je pravica. Krivulja

se lahko vzdolž osi spreminja; vendar tako da ohranja podobnost.



Nekaj primerov posplošenih valjev.

Slika 3

Zahtevnejši predmeti so lahko sestavljeni iz več različnih posplošenih valjev.

3.3 Klasifikacija spojišč

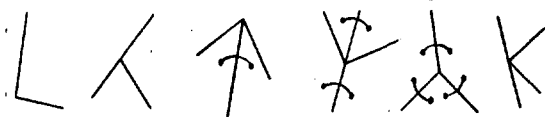
Guzman (1968) je napisal program, ki ugotavlja, kateri predmeti v obliki poliedrov sestavljajo sceno. Predmeti se lahko dotikajo drug drugega ter deloma prekrivajo.

Simbolni opis scene predstavljajo spojišča, linije in ploskve. Simbolni opis odgovarja dvo dimenzionalni črtni sliki.

Razmerja med dvo dimenzionalno črtno sliko in tri dimenzionalno sceno so naslednja:

- ločimo dve vrsti spojišč: tista, ki pripadajo ogliščem predmetov na sceni in tista, ki so posledica delnega zakrivanja robov
- črte pripadajo vidnim ali deloma vidnim robovom teles
- površine pripadajo vidnim ali deloma vidnim ploskvam teles

Največ informacije o temi, katere ploskve pripadajo istim in katere ploskve pripadajo sosednjim telesom, dobimo iz spojišč.

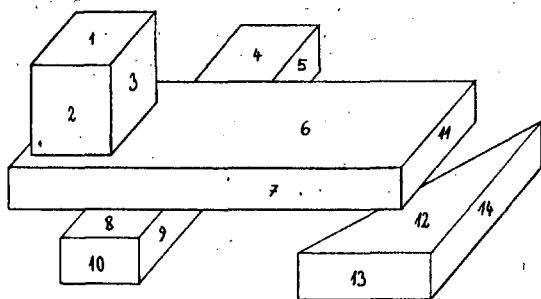


L T PUSZICA PSI VILICE K

Znak označuje pripadnost dveh ravnin istemu telesu.

Nekaj osnovnih tipov spojišč

Slika 4



Primer triedrsko scene

Slika 5

Rezultati, ki ga da program kot opis scene na sliki 5 jet

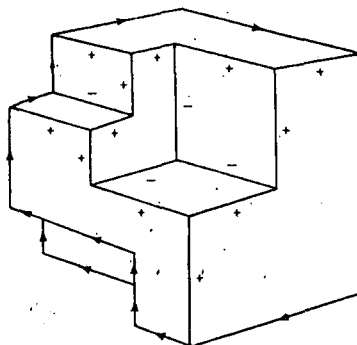
- (TELO 1 JE: 1:2:3)
- (TELO 2 JE: 4:5:8:9:10)
- (TELO 3 JE: 6:7:11)
- (TELO 4 JE: 12:13:14)

Guzmanov program lahko analizira precej zapletene scene; ni pa sposoben analizirati konkavnih predmetov in prepoznavati nesmiselnih predmetov.

3.4 Označevanje robov

Razen spojišč nam lahko tudi črte dvo dimenzionalne slike, ki predstavljajo robove teles, nudijo vir informacij pri analizi scene.

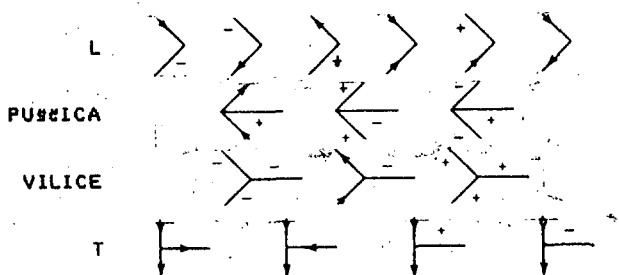
Ločimo dve vrsti robov: konkavne in konveksne robove. Konkavni robovi so na sliki vidni le, če sta vidni obe stični ploskvi. Označimo jih z znakom (-) minus. Konveksne robove označimo z znakom (+) plus; če sta vidni obe stični ploskvi, če pa je vidna le ena ploskva, pa govorimo o prikritem robu in ga označimo z znakom (→) puščica. Puščica je usmerjena tako, da je na desni vidna, na levi pa prikrita stična ploskva; če se premikamo v smeri puščice. Primer označevanja v skladu z omenjenimi pravili je na sliki 6.



Označevanje robov

Slika 6

Število vseh kombinacij je precej večje od števila dejansko možnih kombinacij različno označenih črt. Primeri možnih kombinacij za nekatera spojišča so na sliki 7. Vidimo da so na primer pri vilicah možne le 3 od skupno 81 kombinacij.



Možne kombinacije označitev za nekatera spojišča

Slika 7

S pomočjo informacije o tipih robov, ki se stekajo v spojišča je mogoče rešiti primere, v katerih je bil Guzmanov program nemočen. Analizirati je namreč mogoče tudi telesa z odprtini in vdolbinami.

Možno so še nadaljnje razširitve in izboljšave označevalnih metod in izločevanje namogočih kombinacij označenih črt v spojitvah.

Waltz (1976) razvije algoritem, ki upošteva osenčenje scene, s čimer pa se precej poveča število možnih kombinacij različnih označitev.

4. Uporaba računalniškega vida

Pri računalniškem vidu imamo opravka s prenosom in obdelavo velike množine informacije. Informacijo sprejmemo s pomočjo TV kamere ter pošljemo preko A/D vmesnika v računalnik, kjer jo obdelamo tako, da iz nje izluščimo opis, ki ga lahko potem koristno uporabimo.

Najpogosteje uporabljamo računalniški vid v industriji bodisi samostojno, bodisi kot dopolnilo inteligentnemu robotskemu sistemu.

Nekaj primerov uporabe robotov opremljenih s TV kamero:

- pobiranje različnih predmetov s tekočega traku in polaganje na določeno mesto (sortiranje, pakiranje, itd.)
- zajemanje različnih predmetov s kupa in polaganje na določeno mesto (sortiranje, pakiranje, podajanje stroju, itd.)
- barvanje na tekočem traku obešenih predmetov
- oblačno varjenje vpetih predmetov
- vrtnanje ploščic tiskanih vezij

Sistem za gledanje sestavlja niz programov za izračun ploskovno ali linijsko orientiranih opisov predmetov na podlagi slike, sprejete preko TV kamere. S pomočjo značilnosti kot so: površina, obseg, simetrijske osi in ukrivljenost predmeta, sistem identificira predmete, ki jih vidi. Naslednji korak je določitev lege predmetov in njihove orientacije. Lego določa običajno masno središče, orientacijo pa zasuk predmeta glede na osnovno orientacijo, ki jo določimo v učni fazi, ko predmet predstavljamo sistemu. Podatki o identitoti, legi in orientaciji predmeta sprožijo pri robotu ustrezno akcijo.

Uporaba računalniškega vida v industriji ni nujno vezana na robotski sistem. Nekatore samostojne aplikacije so:

- analiza in kontrola velikosti predmetov;
- analiza in kontrola površinskih lastnosti predmetov: barve, gladkosti, homogenosti, itd.

Računalniški vid uporablja tudi medicina, vojska in uprava javne varnosti.

Uporaba v medicini:

- analiza očiščnega tkiva
- analiza krvne slike
- analiza roentgonske slike

Uporaba v vojaške namene:

- analiza zračnih posnetkov
- odkrivanje maskiranih objektov
- orientiranje na nepoznanem terenu

Uporaba pri UJV:

- analiza prstnih odtisov
- analiza in kontrola prometa

Zanimivo in koristne pa so tudi druge aplikacije računalniškega vida:

- v meteorologiji pri napovedovanju vremena
- v astronomiji pri vodenju vesoljskih vozil

- v ekologiji pri določanju onesnaženosti voda
- v optični analizi tekstov, itd.

Računalniški vid je v zadnjih letih skokovito napredoval predvsem zaradi razvoja tehnologije mikroprocesorjev in TV kamer, ter pocenitve digitalnega spomina na tržišču. Vse kaže, da se bo dinamični razvoj nadaljeval tudi v prihodnosti.

5. Literatura

1. Bacjy R. (1972) Computer Identification of the Textured Visual Scenes. Stanford AI Lab Memo 180.
2. Binford T. O. (1971) Visual Perception by Computer. Presented to the IEEE Conference of Systems and Control, Miami.
3. Clowes M. B. (1971) On Seeing Things, Artificial Intelligence Journal 2, pp 79-116
4. Guzman A. (1968) Decomposition of Visual Scene into Three Dimensional Bodies, AFIPS proceedings of the Fall Joint Computer Conference 33, pp 291-304
5. Gerskowicz A., Binford T. (1970) On Boundary Detection, MIT AI Lab Memo 183
6. Horn B. K. P. (1975) Obtaining Shape from Shading Information, Psychology of Computer Vision, Ed. P. H. Winston; McGraw-Hill, New York pp 115-155
7. Horn B. K. P. (1978) Vision, Proceedings of AISB/GI Conference on Artificial Intelligence, Hamburg, pp 147-163
8. Huffman D. (1971) Impossible Objects as Nonsense Sentences, Machine Intelligence 6, Edinburgh University Press, Edinburgh
9. Marr D., Poggio T., Palm G. (1977) Analysis of the a Cooperative Stereo Algorithm, Biological Cybernetics 28, pp 223-239
10. Marr D. (1975) Early Processing of Visual Information, MIT AI Memo 340
11. Marr D. (1979) Visual Information Processing, IJCAI 79 pp 1108-1126
12. Roberts L. (1963) Machine Perception of Three Dimensional Solids, Tech. Report 315, MIT Lincoln Lab
13. Ullman S. (1979) The Interpretation of Structure from Motion, Proceedings of Royal Society London
14. Waltz D. (1976) Automata Theoretical Approach to the Visual Information Processing, Ed. Yeh R. T. in Applied Computation Theory, Prentice-Hall Inc., Englewood Cliffs, pp 468-529
15. Ward M. R., Rossol L., Holland S. W. (1979) CONSIGHT: A Practical Vision-Based Robot Guidance System, 9-th International Conference on Industrial Robot Technology, Washington D.C., pp 195-211
16. Winston P. H. (1977) Artificial Intelligence, Addison-Wesley Publishing Company, Reading

PRILOG DEFINICIJI INFORMACIJSKIH SISTEMA ZA JAVNO KOMUNICIRANJE PODACIMA I INFORMACIJAMA

F. RUŽIĆ

UDK: 007 : 681.3

ZAVOD ZA PRIMJENU ELEKTRONIČKIH RAČUNALA, ZAGREB

Informacijsko-komunikacijski sistemi (Inko sistemi) predstavljaju razvojni oblik sistema distribuirane obrade podataka. Istovremeno su Inko sistemi daleko više iskoristivi u odnosu na distribuirane, budući omogućuju najširem krugu korisnika pristup velikim memorijskima. Inko sistemi omogućuju pružanje najpotpunijeg oblika kompjutorskih usluga-u domaćinstvima, u školama, u bolnicama odnosno kod svih aktivnosti gdje je moguć pristup javnim bazama podataka pomoću javnih komunikacionih medija. Inko sistemi pružaju najširem krugu korisnika informacije koje oni traže i u vrijeme kada su im potrebne bez potrebe udaljavanja sa svojih radnih mjesta i domova. Krajnji korisnik u Inko sistemima interaktivno komunicira sa lokalnim i centralnim datotekama kao i sa javnim bazama podataka.

Information Systems for Public Data Communications

The Inco systems are development of distributed data processing systems, but are much more powerful system since they can access a very much larger store of information and provides true interactive facilities. Inco system opens the way to an universal computer service-in the house, in the business, into schools, in the hospitals etc., in any location that can access the public communication facilities and public data bases, too. Inco system makes it possible for the ultimate users to get the information they want, when they want, without setting foot outside their home or office. In the Inco system, the end users interacts directly with both local and central files of data.

1. UVOD

U posljednje vrijeme elektronska obrada podataka i informacija dobija novu dimenziju svojom transformacijom u oblike masovnog-potpuno javnog pristupa bazama podataka od strane najšireg kruga korisnika. Čitavo razdoblje dosadašnjeg razvoja elektronske obrade podataka i informatike bilo je isključivo domena profesionalnih informatičara dok se danas informatičarska sve više približava svim zainteresiranim korisnicima čime se u potpunosti, raskida sa kompjutorskim totemizmom.

Činjenica je da, danas još uvijek, kompjutor predstavlja za široke mase predmet tajanstvenog i mitskog značaja sa općim uvjerenjem u svemogućnost ovih uređaja. Proces miniaturizacije, odnosno tehnološki napredak na području elektronske obrade podataka nastao usvajanjem mikroprocesorske tehnologije, doveo je do proizvodnje izuzetno pristupačnih kompjutorskih uređaja. Ovi uređaji obično predstavljaju kompjutore za osobnu namjenu individualnih koris-

nika i u stanovitom stupnju su ograničenih mogućnosti iako se često pokazuju kao pravi zamjenik velikih kompjutorskih uređaja. Istovremeno sa usavršavanjem mikroprocesorske tehnologije došlo je do izuzetnog razvoja telekomunikacionih uređaja i procesa koji se danas sve više orijentiraju na operacije sa podacima i informacijama u digitalnom obliku.

Svi ovi razvojni procesi doveli su do konačnog rušenja kompjutorskog mita s razvojem sistema za javno komuniciranje podacima i informacijama te za pretraživanje banaka podataka od strane najšireg kruga korisnika.

Inko sistemi predstavljaju logičan nastavak izuzetnog razvoja komunikacija i sve većeg stupnja zahtjeva koje korisnici postavljaju i očekuju od novih komunikacionih usluga. Inko sistemi donose revoluciju na polju telekomunikacija istovremeno proširujući interakciju telekomunikacija i kompjutora što korisnicima donosi veći kapacitet usluga, brže i kvalitetnije.

2. DEFINIRANJE InKo SISTEMA

InKo sistemi predstavljaju informacijski medij za sve radne ljude i građane kojima se pruža mogućnost da pomoću standardnog kućnog televizora ili telefonskog aparata pristupe jednoj od javnih banaka podataka ili pak specijaliziranom kompjutorskom centru.

U stvari, sistemi za javno komuniciranje podacima i informacijama mogu se zasnivati u osnovi na prenošenju digitalnih podataka i informacija (interaktivnim putem) na slijedeća tri modaliteta

-pomoću televizijske mreže (prijem na standardnom TV prijemniku)

-pomoću telefonske mreže (prijem putem terminala priključenih akustičkim adapterima na telefonsku liniju)

-pomoću TV i telefonske mreže (pomoću TV prijemnika koji se pretvara u terminalski uređaj priključen na telefonsku liniju).

1. Sistemi za javno komuniciranje podacima i informacijama koji kao informacijski medij koriste televizijsku mrežu omogućuju svim građanima koji posjeduju standardni televizijski prijemnik da pristupe jednoj od javnih banaka podataka jednostavnim biranjem kanala. Međutim ovi sistemi omogućuju prijenos veoma ograničene količine informacija budući se ovakav oblik davanja informacija odnosi na veoma promjenljive banke podataka koje se koriste veoma često ažurirati (dnevne novosti, financijske vijesti, sportske novosti i slično). Usavršavanje ovih sistema vezano je za veći broj kvalitetnijih informacija što ovisi od uvođenja kvalitetnih komunikacionih linija koje omogućuju javno pristupanje bankama podataka.

Ovdje je potrebno napomenuti da u okviru kapaciteta televizijskih kanala postoji još uvijek mnogo slobodnih kanala koje ovi sistemi mogu koristiti za prenošenje podataka i informacija za najširi krug korisnika.

2. Sistemi za javno komuniciranje podacima i informacijama koji kao medij koriste telefonske linije ograničavaju masovnost dostupnosti javnim bankama podataka na one korisnike koji posjeduju telefonsku liniju (priključak). Za razliku od sistema opisanih pod 1., ovi sistemi daleko više podržavaju interaktivno komuniciranje i u izvjesnom broju slučajeva korisnici mogu putem svojih terminala priključenih na telefonsku liniju pristupiti u željeni kompjutorski centar za stanovite obrade. U stvari, telefonskim mrežama korisnik šalje i prima pisane poruke koje se u osnovi sastoje od grupiranih podataka.

3. Sistemi za javno komuniciranje podacima i informacijama koji kao informacijski medij koriste televizijsku i telefonsku mrežu, u stvari su pravi kompjutorizirani sistemi u kojima se jednostavnim adaptivnim priključcima standardan televizijski prijemnik pretvara u kompjutorski terminal. Takav se uređaj, priključen na telefonsku mrežu, veže sa javnim bankama podataka, sa specijaliziranim kompjutorskim centrima ili sa drugim korisnicima umutar mreže. Ovi će se sistemi u budućnosti još više usavršiti masovnim uvođenjem kablovizije, odnosno TV mreže unutar koje se korisnici vežu sa TV centrom pomoću kablovskih linija. Time će se mnogostrukije usavršiti kvaliteta prijema digitaliziranih podataka i informacija budući se u današnjim komunikacionim sistemima poruke, koje korisnici primaju putem televizijskih prijemnika, prenose pomoću valnih dužina koje u izvjesnom smislu otežavaju prijem digitalnih podataka. Kod uspostavljanja ovih sistema neophodno je da se riješe dva osnovna problema - da se omogući adaptacija standardnog TV prijemnika za prijenos i primanje digitaliziranih podataka ugradnjom određenih elektronskih komponenata, te da se pripremi stvaranje javnih banaka podataka i njihovo ažuriranje kao i pripremanje informacija upotrebljivih za najširi krug korisnika (potrošača informacija).

3. NEKI PREDUVJETI UVOĐENJA InKo SISTEMA

U cilju razvoja sistema za javno komuniciranje podacima i informacijama, neophodno je u prvom redu izraditi takove terminalske jedinice koje će biti prikladne za primanje kompjutoriziranih podataka od strane posve nestručnih korisnika. Tu je bitno istaknuti da se ovim sistemima pojam korisnika u infosistemu iz osnova mijenja i da to nije više stručni ili priučeni korisnik informacijskih usluga, već korisnik/potrošač informacijskih usluga, koji ne poznaje i ne želi da nauči osnovne informatike u istom opsegu kao što ne želi znati osnovne telefonije iako se gotovo svakodnevno koriste telefonskim uslugama. Iako su danas već znatno usavršeni terminalski uređaji koji bi bili prikladni za kućnu upotrebu, ostaje pitanje njihove dostupnosti i masovnosti. S druge strane, ovi sistemi zahtijevaju terminalske uređaje koji su sposobni vezivati se na infomrežu priključkom na telefonsku liniju što ujedno omogućuje da se ti uređaji razlikuju od telefonskih aparata koji su već u upotrebi.

Međutim sistemi sa govornim izlazom informacija pomoću telefonskih aparata prilično su ograničenog opsega te se ovakvi sistemi mogu koristiti isključivo za vrlo jednostavne poruke. I upravo iz tog razloga, treba razvijati sisteme koji će uz telefonski aparat uključivati i televizijski prijemnik za primanje opširnih i cjelovitih informacija. Standardni TV prijemnik može u svakoj kući postati kompjutorski terminal jednostavnim adaptivnim priključcima za običan kazetofonski uređaj, jeftin štampač te tastaturom sa oznakama za obavljanje određenih operacija. Klasičan televizijski prijemnik u ovakovim sistemima, prihvaća informacija i podatke koji se prenose telefonskim linijama na koje se priključuju preko ugrađenih modema. Za što masovnije korištenje ovih sistema potrebno je kreirati takav kompjutorski sistem čiji će pripadajući software omogućiti dostupnost i komuniciranje podacima i informacijama korisnicima/potrošačima bez ikakvog iskustva u radu sa kompjutorskim uređajima. Ovdje se ne smije zaboraviti na ekonomski efekt, odnosno da se mora omogućiti troškova pristupačnosti najširem krugu korisnika (nabavna cijena, cijena dodatnih komponenti, pretplata za korištenje informacionih usluga). Prijenosu najrazličitijih oblika informacija u prvom redu prethodi pisanje i otipkavanje podataka što većina ljudi nije u mogućnosti izvoditi. Stoga je potrebno usavršiti tehnike unošenja podataka koje neće tražiti izuzetnu vještinu otipkavanja i pisanja podataka, odnosno treba razviti metode pretvaranja podataka koje unose korisnici u svom najprirodnijem obliku (ljudski govor, pismo) u podatke pogodne za obradu u elektronskim uređajima. Ti se zahtjevi naročito ističu u sistemima za javno komuniciranje podacima koji koriste kao prenosni medij standardni TV prijemnik. U tim se sistemima zahtjeva da se kodovi za pretvaranje podataka u oblik pogodan za njihovu obradu i prijenos (digitalni oblik) pohranjuju na memorijske jedinice koje su sa svoje strane povezane sa tastaturom za unos podataka i uređajima za prijenos. Ovdje je u svakom slučaju potrebno razlikovati kodove za pretvaranje pisanih i govornih podataka (poruka) pri čemu treba naglasiti da će sistemi za pretvaranje govornih podataka u podatke pogodne za prijenos i obradu u elektronskim uređajima biti razvijeni u potpunosti tek krajem ovog ili početkom idućeg desetljeća. Iako su u posljednje vrijeme terminalski uređaji sa ekranom doživjeli izuzetan razvoj, oni su u principu i dalje orijentirani na prikazivanje tekstualnih info-

rmacija na ekranu. Međutim, činjenica je da elektronski uređaji u velikom opsegu smanjuju količinu papirnog materijala čime se izuzetno povećava brzina i efikasnost obrade podataka. Zbog toga treba razviti tehniku rada terminalskih uređaja sa ekranom budući predstavljaju osnovu svih sistema za javno komuniciranje podacima i informacijama. Što se tiče komuniciranja udaljenih korisnika, potrebno je izvršiti transformaciju informacija prilagođenih čovjeku u oblik pogodan za prijenos komunikacionim linijama u okviru kojih postoje određeni frekvencijski opsezi. Kako bi se izbjegao kaos u komunikacionim linijama uz poštivanje točno određenih frekvencijskih opsega neizbježno je uvoditi nacionalne i međunarodne standarde i norme za korištenje komunikacionih linija. Sve veći porast količine informacija zahtjeva sve jače korištenje komunikacionih linija i stalno proširenje frekvencijskog opsega za njihov prijenos. Sistem telefonije kojim započinje era komunikacija ne može zadovoljiti sve veće zahtjeve društva za komuniciranje podacima kako na nacionalnom tako i na međunarodnom polju te je potrebno razviti nove komunikacione tehnike. Klasična se telefonija razvija uvođenjem digitalnog prijenosa podataka. Digitalni oblik prijenosa podataka i informacija razvija se sve više uvođenjem komunikacionih linija koje koriste radio-valove, mikrovalnih sistema, satelitskih sistema i kablovizijskim sistemima. Kapacitivne mogućnosti pojedinih komunikacionih linija mogu se sagledati sa prosječnim vrijednostima na slijedeći način

telefonski kabel	5M bit/sec	500 linija
koaksijalni kabel	300 M	30000
mikrovalni sistem	10^9	10^5
satelitski sistem	10^9	10^5
staklena vlakna	10^{12}	10^9

Iz tabele je vidljivo da je najveći učinak prijenosa kod komunikacionih linija pomoću staklenih vlakana, međutim uvođenje ovih sistema tek prethodi, dok će se u slijedećem razdoblju razvijati tehnike prijenosa zasnovane na telefonskim i TV linijama.

Budući da InKo sistemi uključuju komunikacionu tehnologiju, kompjutore i televiziju kao integralne elemente, biti će potrebno usavršiti standardizaciju komunikacionih tehnika te standardizaciju televizijskih prijemnika i kompjutorskih sistema. Istovremeno je potrebno usavršiti telefonsku mrežu i prihvaćanje međunarodnih standarda koji su već razvijeni za digitalni prijenos podataka te usavršiti kodove za prijenos digitaliziranih podataka i informacija.

4. UVOĐENJE InKo SISTEMA

Kompjutorski sistem koji će omogućiti izgradnju sistema za javno komuniciranje podacima i informacijama posredstvom TV i telefonske mreže sastoji se od mreže kompjutorskih stanica smještenih u neposrednoj blizini potrošača distribuiranih tako da korisnici dobivaju informacije sa najbližeg (lokalnog) kompjutorskog centra kako bi se istovremeno smanjilo vrijeme mnogostrukih priključivanja iz jednog telefonskog mrežnog kruga u drugi. Istovremeno će se korisnik priključivati na lokalnu banku podataka posredstvom lokalnog kompjutorskog centra odnosno daljim priključivanjem na ostale kompjutore u mreži, poštujući hijerarhijski redoslijed.

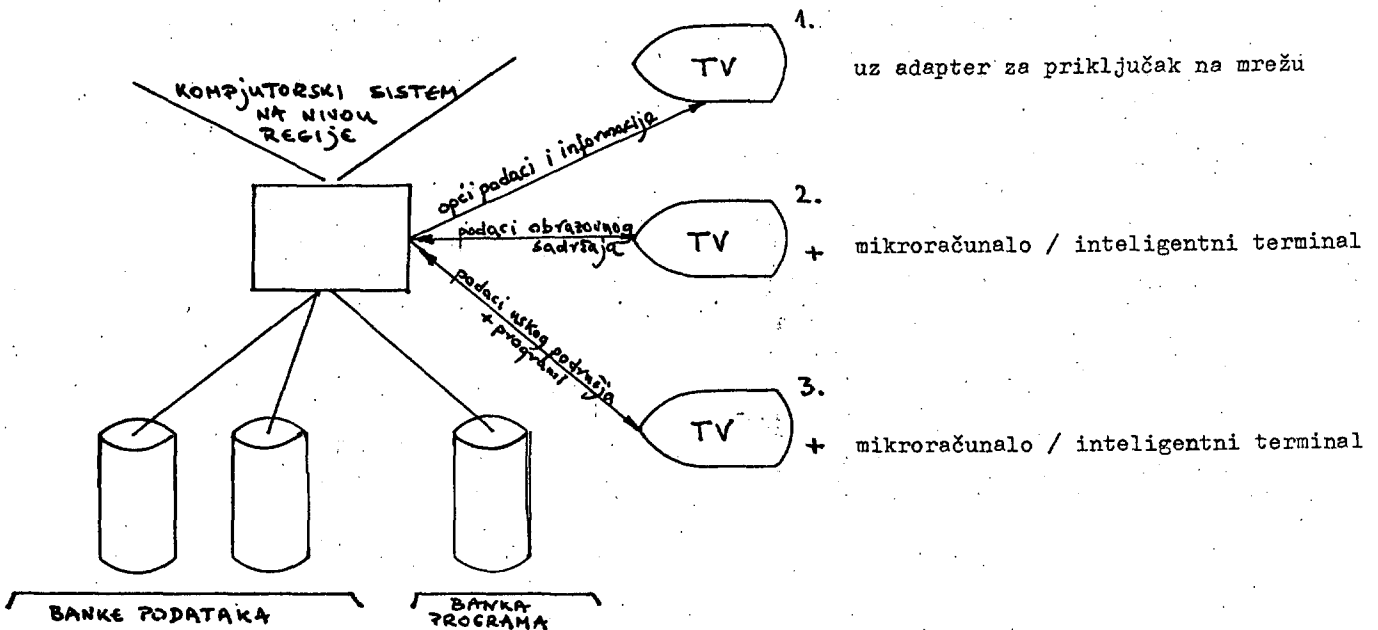
Korisnik će se vezivati na jedan od kompjutorskih centara i njihovih pripadajućih banaka podataka, biranjem određenog broja na telefonskom aparatu poput najobičnijeg telefonskog razgovora/poziva. Uspostavljanje veze se signalizira zvučnim signalima koji korisnika upozoravaju da pritiskom određene tipke na telefonskom aparatu, pretvori telefonski aparat u terminal za primanje poruka. Nakon uspostavljanja veze, korisnik mora u mrežu unijeti svoj pretplatnički broj čime se identificira ova veza. Razvijeniji sistemi će omogućiti korisnicima uspostavljanje veze sa infosistemima i bez biranja određenog telefonskog broja i to jednostavnim pozivom na prekidaču kojim se korisnik može direktno uključiti u komunikacionu liniju.

Najveći problem kod ovih sistema je prilagoditi njihovu upotrebu za masovno korištenje od strane nestručnih korisnika, odnosno isto kao što se zbivalo kod potrošača uporedo sa razvojem automobilske industrije. Nakon što su usavršeni prvi modeli automobila kojima su mogli upravljati jedino profesionalni inženjeri, razvojem tehnologije danas se proizvode automobili kojima upravljaju korisnici koji poznaju jedino tehniku vožnje dok o automobilskom motoru i drugim mehaničkim sklopovima nemaju nikakovo znanje. Zato se nameće osnovno pitanje-kako usavršiti infosistem u kojem se nalaze korisnici svih profila, profesija, dobnih uzrasta i obrazovnog potencijala?, odnosno kako usavršiti infosistem koji će biti oslobođen od kompjutorskih grešaka? Ključno pitanje za kreiranje dobre komunikacije između korisnika i kompjutora je dijalog (čovjek-kompjutor). Da bi se taj dijalog ostvario, potreban je zajednički jezik te se može napraviti kompromis između strojnog i ljudskog jezika-

korištenjem viših programskih jezika. Međutim, poznavanje tehnike programiranja u višim programskim jezicima nalazimo kod malog broja profesionalnih programera te je taj pristup veoma težak i kompliciran za uvođenje u sisteme za javno komuniciranje podacima i informacijama.

Drugi pristup omogućuje kreiranje unaprijed programiranih kompjutorskih sistema u kojima korisnik jednostavnim izborom odlučuje koja će se od mnogobrojnih, postojećih programskih operacija i izvršiti. Ovakav pristup ograničava korisnikovu slobodu ali se njime uklanjaju problemi koji su vezani za korištenje kompjutorskih jezika. U tu svrhu sastavljaju se indeksi iz kojih korisnik odabire informacije po svojim potrebama. Informacije se mogu sastaviti u obliku hijerarhijskog trokuta na čijem se vrhu nalaze najvažnija područja koja se zatim dijele na oblasti iz kojih se selekcijom uzimaju željene informacije, što omogućuje sa svoje strane mnogo brže i točnije komuniciranje. Implementiranje sistema za javno komuniciranje podacima i informacijama uzrokovati će smanjenje izvora informacija tako da će se na taj način izvršavati stanovita selekcija informacionih sadržaja. Uspostavljanjem sistema za javno komuniciranje podacima formira se nova kategorija potrošača koja će pokazivati iste zakonitosti kao i ostale kategorije potrošača. Ovu novu kategoriju možemo označiti kao potrošače informacija koji će imati velike zahtjeve na efikasnost sistema za pretraživanje informacija koje će moći prikazivati na ekranu svojih televizora uz istovremenu mogućnost izvršavanja stanovitih obrada. Ovdje je veliku ulogu odigrao razvoj inteligentnog terminala koji omogućuje lokalnu obradu i kreiranje upita koje će korisnik postavljati sistemu za javno komuniciranje podacima. Individualni korisnici sa većim zahtjevima i složenijim aplikacijama moći će korištenjem inteligentnih terminala opsluživati i vlastite uređaje za štampanje kao i vlastite baze podataka.

U skladu sa osnovnim kategorijama korisnika sistema za javno komuniciranje (opći tip korisnika, korisnici obrazovnih sadržaja, korisnici malih specifičnih aplikacija) kreirati će se kako banke podataka tako i programski paketi koji će se prenositi unutar nacionalne komunikacione mreže. Shematski prikaz toka prenošenja podataka i programa u sistemu za javno komuniciranje podacima i informacijama dat je na slici 1.



1. korisnici općeg tipa (gledaoci TV programa)
2. korisnici obrazovnih sadržaja (obrazovne i znanstvene institucije)
3. korisnici malih, specifičnih aplikacija (male organizacije održanog rada, zanatske radionice znanstvene institucije)

5. PRIMJENA InKo SISTEMA

Primjena sistema za javno komuniciranje podacima i informacijama posredstvom TV i telefonske mreže gotovo je neograničeno. Stvarna primjena će u svakom slučaju ovisiti o intezitetu interesa i potražnje pojedinih grupacija korisnika/potrošača, što znači da će ovisiti o informacionom tržištu.

Najznačajnija područja na kojima se zasniva primjena ovih sistema su slijedeća:

-pružanje informacionih usluga- Ovo područje je najprihvatljivije za veliki dio korisnika i upravo će to područje najviše da se razvija uključujući sve djelove koje će obuhvaćati. Jedan dio informacionih usluga obuhvaćati će najaktualnije informacije privlačne za najveći dio korisnika a u vezi sa političkim, ekonomskim i kulturnim novostima. Za razliku od sistema u kojima se prijenos podataka i informacija odvija samo putem telefonskih linija u sistemima koji za prijenos podataka i informacija koriste TV i telefonsku mrežu, podaci se prikazuju mnogo detaljnije i raznovrsnije. Njima će se moći također, pružati ne samo najaktualnije novosti nego i razni komentari, osvrti i savjeti. Ovi će sistemi isto tako omogućavati kako pružanje primarnih tako i sekundarnih informacija odnosno podatke o izvornim informacijama.

Jedan dio informacionih usluga biti će orjentiran na pružanje pomoći korisnicima kod pojedinih kupovina, rezervacije prevoznih karata, osiguranja hotelskog smještaja i slično. Ovako široki spektar informacionih usluga omogućen je interaktivnim pristupom u odnosu korisnik/sistem, čime se sistem u potpunosti orjentira ka neposrednom korisniku.

-pružanje komunikacionih usluga- Sistemi za javno komuniciranje posredstvom TV i telefonske mreže omogućuju uz pružanje informacionih usluga također i mogućnost prenošenja poruka sa određenih kompjutorskih instalacija ka zainteresiranom korisniku. Korisnik će biti u mogućnosti da izabere jednu od pohranjenih poruka u kompjutorskom centru te da ju kroz sistemsku mrežu uputi primaocu. Poruka se upućuje u centar koji je najbliži primaocu poruke te se zatim automatski bira telefonski broj primaoca, nakon čega je primaoc u mogućnosti da prihvati poslanu mu poruku. Primaoc je također u mogućnosti primiti poruku i kada nije fizički prisutan u vrijeme poziva čime se u potpunosti nastoji pružanje informacionih usluga prilagoditi korisnikovim zahtjevima. Ova aplikacija omogućuje primanje poruka i na televizijskom ekranu (time se omogućuje gluhoj i oslabljenoj osobama da uspješno komuniciraju i na udalje-

nima lokacijama što im do sada, klasičnim telefonskim linijama nije bilo moguće).

-pružanje obrazovnih usluga- Sistemi za javno komuniciranje podacima i informacijama posredstvom TV i telefonske mreže omogućuju korisnicima dobivanje informacionih usluga o mogućim obrazovnim sadržajima, institucijama i vremenskom rasporedu te o načinu sudjelovanja u određenim obrazovnim sadržajima. Ovi sistemi također, omogućuju provođenje obrazovnih postupaka neposredno kod korisnika koji se uključuju u sistem, čime se provodi proces obrazovanja uz pomoć kompjutera te je zbog toga potrebno usavršiti elemente programirane nastave uz pomoć kompjutera koji se u našoj zemlji polako ali ipak uvode. Time će se u znatnoj mjeri osloboditi radno vrijeme nastavnika koji će tako biti u mogućnosti da se mnogo više i svrsishodnije posvete svakom pojedinom učeniku te da se usavrše elementi individualizirane nastave. Sposobnost interaktivnog rada sa ovim sistemima omogućuje korisnicima/učenicima da provode kontinuirano samokontrolu postignutih rezultata te da proces nastave prilagode svojim sposobnostima i interesima.

Osnovu sistema za obrazovanje širokog kruga korisnika čine disk jedinica za memoriranje vidljivih (slikovnih) podataka pa se te jedinice mogu u praksi naći pod nazivom video-disk. Stanoviti obrazovni sadržaj pohranjuje se na video-disk sa kojega se tekst, slika, grafikon i sl. prikazuju na TV ekranu u obliku programiranih nastavnih lekcija ili pak sa zvučnom pratnjom pripremljenih nastavnih sadržaja. Normalno je očekivati da se za uvođenje ovakvih sistema obrazovanja treba u prvom redu ostvariti određeni nivo informatičke kulture što sa svoje strane zahtjeva prethodno informatičko prosvjeđivanje.

-pružanje računskih usluga- Prvenstvena uloga sistema za javno komuniciranje je pružanje informacionih i komunikacionih usluga. Međutim, ovi sistemi omogućuju i provođenje izvjesnih računskih usluga namjenjenih učenicima, poslovnim ljudima i malim poslovnim organizacijama odnosno korisnicima koje zadovoljava ograničeni opseg obrada podataka. Istovremeno, treba nastojati da jednostavne operacije računanja ne zahtjevaju od korisnika temeljno poznavanje programiranja elektroničkih računala odnosno treba omogućiti korisniku da jednostavnim instrukcijama na svom terminalu/TV prijemu + telefonski aparat, zatraži operaciju računanja koju želi.

-informacijski sistemi za pružanje usluga klasičnih časopisa i novina- Ovi se sistemi

već uspostavljaju u svom najjednostavnijem obliku i imaju izvanrednu sposobnost odgovaranja na raznovrsne zahtjeve korisnika. U ovim sistemima su u stvari najzastupljeniji uređaji za obradu teksta i oni čine osnovicu on-line povezanih sistemskih komponenti. Ovi uređaji omogućuju sastavljanje pisanih materijala (novina, časopisa, dopisa) bez mukotrpnog i multipliciranog prepisivanja i umnožavanja. Istovremeno, mnogobrojni korisnici mogu da preko svojih TV prijemnika "prelistavaju" tražene publikacije čime se proširuje krug pretplatnika a samim time i korištenje raznovrsnih publikacija i drugih pisanih informacija.

-sistemi za slanje pošte elektronskim putem- Ovi sistemi koriste telefonske linije za slanje poruka u digitalnom obliku koje se mogu kod korisnika uključenih u sistem za javno komuniciranje, prezentirati na korisnikovom ekranu. Ovi se sistemi u praksi veoma često susreću pod nazivom elektronska pošta-pružanje poštanskih usluga elektronskim putem, koji omogućava brže, efikasnije, gotovo trenutačno prenošenje poruke. Time otpadaju ogromna ulaganja u poslovnu korespondenciju koja uključuje veliki broj tipkačica, inikorespodenata, stručnjaka za poslovno dopisivanje, lektore i sl. S druge strane, ne opterećuju se poštanske ustanove slanjem velikog broja jednovrsnih poruka te se skraćuje vrijeme između slanja i primanja poruka gotovo na minimum.

-sistemi za održavanje konferencija pomoću TV i telefonskih linija- Ovi se sistemi u praksi često nazivaju sistemi za telekonferencije. Uvođenjem elektronskih sistema na osnovi TV prijemnika i telefonskih linija biti će moguće ostvariti sudjelovanje i onih sudionika konferencije koji nisu fizički prisutni. Teoretski se mogu održavati konferencije gdje se istovremeno svaki sudionik nalazi u svojoj instituciji iz koje direktno komunicira sa ostalim sudionicima. Međutim, obzirom na samu organizaciju i tok konferencije biti će nužno uočiti probleme na polju komunikacionih linija, jezika, gubitka ljudske prisutnosti i mogućeg pojavljivanja otuđenosti.

Sistemi za javno komuniciranje podacima i informacijama posredstvom TV i telefonske mreže omogućuju također, uvođenje distribuirane obrade podataka za mnogobrojne i potpuno neovisne korisnike odnosno mogu se promatrati kao nacionalni sistemi za distribuiranje obrade podataka i informacija. Povezujući TV prijemnike u domaćinstvima i malim poslovnim organizacijama preko telefonske mreže sa centralnim bankama podataka postaju pravi javni informa-

cijski sistemi. Ovi će sistemi, omogućujući priključenje mikrokompjutora, malih poslovnih sistema te inteligentnih terminala, proširiti aplikacije unutar sistema te pružiti daleko veće usluge svim zainteresiranim korisnicima. Sistemski software koji se odnosi na rad terminala omogućiti će korisnicima da vrše prijenos podataka i programa iz jednog kompjutorskog centra u drugi odnosno iz ili u svoj terminalski priključak. Jedan dio sistemskog kao i aplikativnog softwarea ukomponirani su u korisnikov terminal koji je priključen na sistemsku mrežu. Time se korisniku omogućuje da vrši izbor sadržaja iz sistemskih banaka podataka. Isto tako, korisnici će moći da putem sistemske mreže elektronskim putem šalju podatke, informacije i programe u udaljene, dislocirane organizacione jedinice. U tu svrhu razvijaju se programski jezici koji se mogu prenositi iz jednog kompjutorskog sistema u drugi neovisno o vrsti i modelu, budući se mora osigurati takovo prenošenje programa da njihov prijem ne ovisi o medelu kompjutorske konfiguracije.

6. ZAKLJUČAK

Informacijski sistemi za javno komuniciranje podacima i informacijama predstavljaju logičan nastavak izuzetnog napretka komunikacija i sve većeg broja zahtjeva koje korisnici postavljaju i očekuju od telekomunikacionih usluga. Proces uvođenja ovih sistema u najvećoj mjeri ovisi o razvoju tehnika digitalnog prijenosa informacija. U narednom razdoblju prirodno je očekivati veću integraciju sistema za obradu informacija usmjerenog ka uspostavljanju potrošačkog aspekta informacionih usluga koji će se razviti izgradnjom raznih oblika elektronskih sistema za pružanje i održavanje informacionih usluga. Nekoliko najznačajnijih i praktički upotrebljivih sistema nužno je što prije uspostaviti u vezi ekonomskih, društvenih i kulturnih spoznaja o razvoju opće informatičke djelatnosti u suvremenom društvu.

Ključni kompjutori će priključkom na sistemsku mrežu biti u stanju da šalju i primaju poruke što drugim rječima znači da će korisnik moći pristupiti podacima i informacijama pohranjenim u velikim bankama podataka na republičkom i regionalnom nivou. Normalno, biti će potrebno da televizijska mreža i PTT organizacije pristupe usaglašavanju pristupa pri izgradnji banke podataka i komunikacionih linija pri čemu treba voditi računa da se prijenos digitalnih podataka i informacija vrši praktički

vremenski neograničeno (što znači i u vrijeme odvijanja televizijskog programa).

Usvajanjem nove tehnologije od strane najšireg kruga korisnika porasti će informatička kultura a sa njome će porasti potrošnja/potražnja za sve većim brojem informacija kao i za višom kvalitetom traženih informacija.

Uvođenje sistema za javno komuniciranje podacima i informacijama implicira određene društvene i pravne odnose. Normalno je očekivati stanovite promjene nastale povećanjem korištenja elektronike, kompjutora i mikrokompjutora na području privrede, obrazovanja, socijalnih službi i domaćinstva. Jedno od najznačajnijih područja primjene ovih sistema sa stanovišta socijalnih odnosa je pitanje tajnosti osobnih podataka naročito onih koji su o nekoj osobi pohranjeni u kompjutorizirani info sistem sa javnim pristupom pohranjenim podacima. Zato se postavljaju izuzetno veliki zahtjevi za potpunu pravnu i socijalnu zaštitu svih osoba čiji se podaci nalaze unutar sistema. Mreže međusobno povezanih kompjutorskih sistema omogućuju neograničeni pristup i prenošenje raznovrsnih podataka vezanih za pojedince pa je također potrebno točno utvrditi ovlaštena lica koja će moći koristiti osobne podatke.

LITERATURA

1. D.R. Doll, Data Communications
The Telecom Library, New York, 1978.
2. S. Nora/A. Minc, L'informatisation et societe
La Documentation Francaise, 1978.
3. Computer Technology and Employment, AUEW
The National Computer Centre, 1979.
4. Informatique et Societe-Premier Plan
Ministere de l'industrie, Paris, 1979.
5. F. Ružić, Sistemi distribuirane obrade podataka
Praksa, 12/1977
6. F. Ružić, Standardisation As The Key Element
in DDP Systems
MFK, Interbiro 79, Zagreb, 1979.
7. F. Ružić, Information Systems for Public Data
Communications
ACM Computer Science Conference, Kansas City,
1980.

SENZORSKI SISTEMI ROBOTOV

S. PREŠERN
M. ŠPEGEL

UDK: 007.52

ODSEK ZA RAČUNALNIŠTVO IN INFORMATIKO
INSITUT JOŽEF STEFAN, LJUBLJANA

Članek opisuje senzorske sisteme, ki so ob mikroprocesorski kontroli ključ do inteligentnega obnašanja robotov. Tako v raziskovalnih laboratorijih za robotiko kot tudi v industrijski robotiki je glavna teža študija posvečena razvoju sposobnih senzorskih sistemov, ki omogočajo prilagajanje robotovega delovanja spreminjajočim se pogojem v okolju. Sodobni roboti zmorejo s svojo umetno kožo in rokami tipati, meriti sile prijema in momente v zapestju svoje roke, čutijo oddaljenost do predmetov v okolju in s pomočjo vida razpoznavajo objekte v svoji okolici.

SENSING SYSTEMS IN ROBOTICS. The paper describes the sensing systems of robots and the corresponding microprocessor control which is the key to intelligent behaviour of robots. In research laboratories as well as in industrial robotics, the sensors and sensing systems are in the focus of research and development interests. Modern robots are capable of tactile sensitivity by means of artificial skin, they can measure the grasping force in the artificial hands and the momentum in the artificial wrist, they can feel the distance to the surrounding objects and see their environment. Intelligent robots became, with the powerful sensing systems and with the application of microprocessor control systems, the reality.

1. UVOD

Takšno revolucijo, kot so v sodobnem svetu povzročili avtomati in uporaba mikroročunalnikov, bodo kmalu v robotiki povzročili senzorski sistemi krmiljeni s multiprocesorskim sistemom mikroročunalnikov. Tako bo poleg varjenja in barvanja postalo sestavljanje mehanskih, elektromehanskih in elektronskih naprav eno od glavnih področij uporabe inteligentnih robotskih sistemov v industrijski proizvodnji. Ta napoved temelji na objavljenih raziskovalnih rezultatih vodilnih laboratorijev za robotiko in umetno inteligenco na univerzah (Stanford University, MIT, University of Edinburgh, Politecnico di Milano itd.), kot tudi na raziskavah vodilnih proizvajalcev elektromehanskih, elektronskih, in računalniških naprav. Nekateri med njimi (na primer IBM, Olivetti, Fujitsu itd.) so sestavljalne robote opremljene s ustreznimi senzorskimi sistemi ne samo vključili v različne faze proizvodnje teh naprav, temveč sestavljalne robote tudi serijsko proizvajajo za tržnišče (ŠPEG79).

Tako pri industrijski robotiki kot tudi pri raziskavah in aplikacijah na področju umetne inteligence se pojavlja vse večja potreba po natančnosti in prilagodljivosti robotskih sistemov. To je mogoče doseči s uporabo senzorskih sistemov, preko katerih robot komunicira s svojim fizičnim okoljem in s povratno sanko ustrezno korigira svoje akcije. Izбира ustreznega senzorskega sistema in krmilne mikroročunalniške logike je odvisna od namena uporabe robota.

Prilagodljiv "inteligenten" robot mora torej imeti sensor za komuniciranje s zunanjim svetom. Pomislek, da v idealnem avtomatisiranem procesu ni potrebe po robotih, katerih akcije podpirajo senzorski sistemi, je poenostavljanje realnih situacij. Zamislimo si na primer avtomatski tekoči trak. Če bi bila orientacija in položaj vseh komponent na traku točno določena vse od začetka proizvodnje pa do konca procesa obdelave, bi bila slepa avtomatika lahko precej zadovoljiva. V praksi pa obstoja veliko vzrokov sakač ta sačelani oifj ni mogoče doseči. Zaenkrat rešujejo take probleme praviloma s pomočjo človeka, ki posreduje v avtomatskem procesu, večkrat pa je sačeljeno, da bi sam robot imel ustresno čutilo s katerim bi zasnal ostopanja od idealnih pogojev in posredno odločil o njegovem upravljanju (HALE75).

2. INTELIGENTNI ROBOTI

Inteligentno obnašanje robotov sajema študij idej s pomočjo katerih delujejo računalniki za upravljanje robotov na tak način, da postane delo robotov inteligentno. S tem posegamo v področje umetne inteligence katere oifj je razširitev uporabe računalnikov in razumevanje principov, ki omogočajo inteligentne akcije. Te akcije so pa velikokrat možne le, če ima robot informacije o stanju okolice, ki jo dobi s pomočjo sensorja (WINS77).

Vemo, da danes najrazličnejši računalniki ali pa roboti izvajajo številne sahtevne inteligentne operacije, kot so na

primer sledeče:

- roboti so se sposobni učiti
- opravljati geometrični inteligenčni test
- razumevati preproste risbe in risanke
- razumevati preprost govorni jezik
- reševati zahtevne manipulacijske naloge
- izvajati precizne montažne operacije
- modelirati psihološke procese
- igrati damo, šah, go, bridge
- komponirati glasbo in drugo

3. ENOSTAVNA IN SESTAVLJENA ČUTILA ROBOTOV

3.1. Enostavna čutila

S pomočjo senzorskih sistemov robot, na primer, opazuje svojo fizično okolico in dobi od njih informacije o dolžini predmetov ali njihovi legi, o sili prijema robotove roke, o momentu in sevda o obliki ter orientaciji predmetov. Spodnja tabela kaže načine merjenja oz. danes najpogosteje uporabljene sensorje (PRES79) za posamezne količine.

količina	vrsta sensorja
dolžina	-strain gauge* -potenciometer -pomikanje jedra v tuljavo -spremenljivi kondenzator -piezoelektrični element -polprevodniki -prekinjanje žarka pred fotoelementom -laser -infrardeč izvor
sila	-strain gauge -potenciometer -piezoelektrični element -polprevodniki
moment	-strain gauge
oblika	-TV kamera -polvodniške kamera

*strain gauge je spremenljivi upor, kjer je žica pritrjena na raztegljiv material, ki se pod vplivom sile raztegne. V odsotnosti ustreznega domačega izrasa uporabljamo angleški izras.

Zaradi ključne vloge sensorjev pri robotih se postavlja vprašanje, kakšen sensor določen robot potrebuje. Splošnega odgovora na to vprašanje ni. Robot potrebuje tak sensor, ki najbolje ustreza operacijam in funkcijam konkretnega robota. Ogleдали si bomo nekaj posebno zanimivih senzorskih sistemov ter opisali njihovo uporabo v inteligentnih robotih. V grobih obrisih bomo pregledali kako se informacija, ki jo s pomočjo sensorja zaznamo obdela, kako s senzorsko povratno zanko povečamo natančnost robotov, shranjevanje senzorske informacije v spomin računalnika ali pa na kaseto magnetnega traku in selekcijsko programa akcije.

3.2. Sestavljena čutila

Skupno obravnavanje čutil za tip, silo in moment je osnovano na dejstvu, da te tri količine merimo s enakimi sensorji. Če nas zanima le binarni odgovor sensorja, služi tak sensor kot preprosto čutilo za tip. Če nas zanima večji merilni obseg sensorja, je lahko tak sensor merilnik sile. Če pa take sensorje sil ustrezno prostorsko razporedimo, postane

senzorski sistem merilnik navora.

Za ilustracijo uporabe strain gauge pogledajmo razvoj sensorjev sil in navorov, ki so ga naredili v laboratoriju za umetno inteligenco na MIT (AI-TR). Namesto prostorske rasporeditve sensorjev sil v obliki maltskega križa predlagajo drugačno prostorsko porazdelitev strain gauges, ki omogoča delovanje učinkovite senzorske povratne zanke.

Določanje prijemališča in velikosti sile

Predstavljajmo si nosilec, ki je oprt na dva sensorja sil A in B. Če pritisnemo v točki "A" s silo F, dobimo na sensorjih rezultat

$$A = F \quad B = 0$$

Če pritisnemo s silo F v točki "B", dobimo

$$A = 0 \quad B = F$$

Če pa pritisnemo v točki "C", ki je za "X" oddaljena od "A" dobimo

$$A+B = F \quad (\text{celotna sila pritiska})$$

$$AX - (1-X)B = 0 \quad (\text{vsota vseh navorov je 0})$$

Če je dolžina nosilca $l=1$, sledi

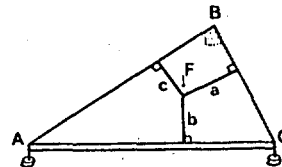
$$X = B / (A+B)$$

Iz dveh sensorjev lahko torej neposredno dobimo informacijo o lokaciji prijemališča sile in o njeni velikosti. S podobnim razmišljanjem spoznamo, da v trikotni plošči, ki je podprta s tremi sensorji lahko kjerkoli v ravnini določimo velikost sile in točko prijemališča sile.

Za določitev vektorja sile na togo telo v prostoru potrebujemo 6 podatkov, torej že tri dodatne sensorje. Če poznamo premico na kateri deluje vektor sile, lahko določimo točko prijemališča sile, če poznamo obliko površine predmeta na katerega deluje sila. Tako lahko šestosni sensor sile služi kot tipalni sensor na vsej površini predmeta.

Rasporeditev strain gauge

Naj bo enakostranična trikotna plošča podprta s tremi sensorji A, B in C. Daljice a, b in c naj bodo pravokotne razdalje od poljubne točke prijemališča sile F do odgovarjajočih stranic (Slika 1, AI-TR).



Slika 1.: Trikotna plošča s tremi sensorji A, B, C.

Če je F velikost sile, potem velja $F=A+B+C$, saj je sistem v vodoravnem ravnovesju. Tri sile A, B, C so v preprosti svesi s tremi razdaljami a, b, c. O tem se prepričamo, če pogledamo ploščo v ravnini, ortogonalni na svesnico BC (Slika 2, AI-TR). Velja namreč

$$AH = Fa$$

$$BH = Fb$$

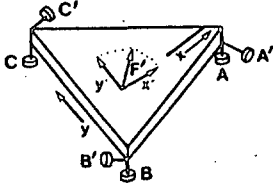
$$CH = Fc$$



Slika 2.: Določanje prijemališča sile F

Velikosti sil v treh sensorjih določajo točko prijemališča sile F v ravnini sensorjev.

Sila na ravnino ima lahko tangencialno in normalno komponento. Tangencialno komponento lahko določimo s pomočjo dodatnih treh strain gauge. Vsak dodatni sensor meri diferencialno gibanje vsoli ost, ki je vnapredna nasprotni stranici trikotnika (Slika 3, AI-TR).



Slika 3.: Razporeditev sensorjev pri tangencialni sili F

Tangencialna sila F_t , ki leži v ravnini ima komponente x' in y' . V ravnovesju velja na komponenti x' in y' zveza

$$\begin{aligned} x' &= \sqrt{3} (C' - B') / 2 \\ y' &= -A' + (B' + C') / 2 \end{aligned}$$

S tremi meritvami sil A' , B' , C' dobimo torej smer $\arctg(y'/x')$ in velikost tangencialne sile

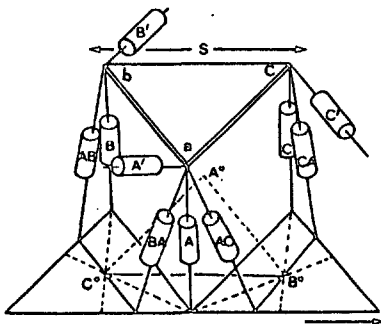
$$\sqrt{A'^2 + B'^2 + C'^2 - A'B' - A'C' - B'C'}$$

Kadar ima sila tangencialno in normalno komponento, sensorji A, B, C še vedno določajo točko prijemališča sile, ker trije dodatni sensorji A', B', C' ne vplivajo na meritve v normalni smeri.

Če poznamo tangencialno komponento F_t in normalno komponento sile F_n , dobimo s tem popoln tridimenzionalen opis sile v prostoru in jo določata kota ϕ in ϵ

$$\begin{aligned} \text{dvišni kot } \phi &= \arctg (A+B+C) / |F'| \\ \text{azimut } \epsilon &= \arctg (2A'+B'+C') / 2X' \end{aligned}$$

Povrnimo se k oktaederskem sistemu strain gauge (Slika 4, AI-TR). Nadomestimo



Slika 4.: Prostorska razporeditev sensorjev

normalno in tangencialno ležeče sensorje A' in A s poševno ležečimi sensorji BA in AC . Sensorji A, A', BA in AC ležijo v isti ravnini, zato velja

$$\begin{aligned} A &= pBA + qAC \\ A' &= rBA + sAC \end{aligned}$$

Pritrdimo sensorja BA in AB v točko a . Zopet lahko upoštevamo ravninsko konfiguracijo točk a, b, c in sensorjev AB ter BA . Hkrati lahko nadomestimo par

sensorjev AB in BA s parom BC' in $C'A$ (na sliki jih zaradi preglednosti nismo vrisali) in dobimo

$$\begin{aligned} BA &= p'BC' + q'C'A \\ AB &= r'BC' + s'C'A \end{aligned}$$

Celotno transformacijo lahko združimo v matriko velikosti 6×6

$$A = p''BC' + q''C'A + r''AB' + s''B'A C$$

itd.

Naše razmišljanje je vezano na male premike celotne strukture in predstavlja tako rekoč primer kvazi izometričnih strain gauge sensorjev. Pri večjih premikih kjer ima tipalo tudi svoje prostostne stopnje gibanja, ne bi bil sistem več linearen, koeficienti niso konstantni in je možno priti do praktičnih rešitev le s različnimi aproksimacijami.

4. ČUTILNI SISTEMI INDUSTRIJSKIH ROBOTOV

4.1. Sensorji tipa, sil in momentov

Sodobni industrijski roboti, ki opravljajo najrazličnejše funkcije, kot so na primer avtomatsko prijemanje in sestavljanje, varjenje, ravnanje isboklin po varjenju itd., so opremljeni s sensorji tipa, sil in navorov. Sledeča tabela prikazuje nekatere tipične vrste sensorjev, ki jih uporabljajo obstoječi roboti, in za kakšno funkcijo so ti sensorski sistemi namenjeni.

FUNKCIJA ROBOTA	VRSTA SENZORJA	OPAZOVANA KOLIČINA
mikromanipulator	piezoelektrični element	dotik, sila prijema
tipalo lege svara	strain gauge upogib nosilca induktivno	sprememba razdalje
podvodni ladijski robot	strain gauge	dotik, prepoznavanje oblike
ravnanje isboklin po varjenju	strain gauge	sprememba razdalje
regulacija sile prijema	strain gauge	sprememba razdalje
korekcija gibanja	strain gauge	sila prijema
avtomatsko sestavljanje	strain gauge	sila
prijemalo	magnetni upor	sila, momenti
prijemalo in identifikacija s umetno kožo	sprememba impedance	dotik

Na kratko bomo opisali posamezne sensorje tipa, sil in momentov, ki jih uporabljajo naj sodobnejši roboti.

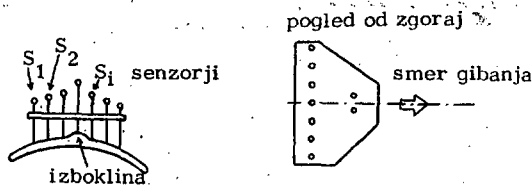
Mikromanipulator:

Pri manipulatorjih, ki imajo opravka s mikroskopskimi objekti, je potrebna velika občutljivost tipala. Dosežemo jih s čutili na osnovi sobatih kolesa, toplotnega razstosanja in na principu piezoelektričnega elementa

(UMET78). Z uporabo piezoelektričnega efekta je mogoče izdelati tipalo, ki je sposobno prijematii objekte s premerom približno nekaj mikronov pri čemer ima tipalo občutljivost prijema približno 100 mikrogramov. Piezoelektrični element je posebno primeren zaradi njegove lastnosti pretvarjanja mehanske energije v električno in obratno. To nam omogoča konstrukcijo preprostega manipulatorja, ki ima hkrati sposobnost majhnih premikov in sposobnost tipa. Piezoelektrični kristal je vgrajen v prst mikromanipulatorja, kontrolira pa ga elektronsko vesje.

Ravnalo izboklin

Tipalne senzorce uporabljajo tudi industrijski roboti za ravnanje izboklin, ki nastanejo kot kapljice po varjenju avtomobilskih karoserij (STUT79). Velikost varilnih izboklin ni večja od nekaj milimetrov in sato je merilni obseg senzorca do 8 mm. To tipalo, ki uporablja strain gauge, omogoča kljub preprosti konstrukciji veliko zanesljivost merjenja in je namenjeno obratovanju v grobem industrijskem okolju. Digitalna resolucija signala je 12 bitov. Več takih senzorjev lahko linearno združimo, da s njimi kot s "glavnikom" pretipamo površino svara (Slika 5).



Slika 5.: Tipalo - glavnik

Celoten proces tipanja ima sledeče stopnje:

- robot otipa svar tam, kjer naj bi se svar nahajal (podatek o lokaciji idealnega svara ima v spominu)
- dobi merilne podatke o izboljšani lokaciji svara
- izračuna izboljšano tirnico svara
- izračuna gladilno funkcijo iz nove tirnice svara
- če je potrebno, proces ponovi in še enkrat otipa svar.

Avtomatsko sestavljanje:

Oblika tipalnega senzorca je tudi sensor, ki meri sile in momente. Pri industrijskih robotih za avtomatsko sestavljanje potrebujemo senzorski sistem v realnem času s povratno zanko (BRUS78). Tak sensor lahko sasnna komponente sile F_x , F_y in F_z (ki lahko variirajo od $-20N$ do $+20N$) in momente M_x , M_y in M_z (v obsegu $-1Nm$ do $+1Nm$). Sile in momente merimo s strain gauge in s pomočjo deformacije upognjenega nosilca. Te podatke potem vodimo v analogni multiplekser in v analogni digitalni pretvornik ter od tam v računalnik.

Umetna koža:

Za potrebe, ko želimo spoznati položaj in orientacijo predmeta pa tudi predmet identificirati in nam okoliščine (podvodno okolje, dim, maščoba, ...) onemogočajo uporabo TV kemere, je možno uporabiti kot sensor umetno kožo (BRIO79). Ta sensor sestoji iz tiskanega vesja, na katerem je izolirana vrsta senzorskih točk. Te senzorske točke so kvadratne oblike, enakomerno razporejene po površini ter tvorijo matrično obliko. Okrog senzorskih točk je varnostni prstan, na katerega je priključena napetost. Vse to je

prekrito s plastjo prevodne snovi, katere impedanca se spreminja v odvisnosti od pritiska. To povaroči v vsaki senzorski točki spremembo toka, ki ga zaradi lažjega merjenja pretvorimo v spremembo napetosti. Računalnik preko analogne digitalne konverzije zbira podatke o napetosti v vsaki posamezni točki matrike. Ker je struktura vrhnje prevodne plasti še vedno neidealna, je treba pred obdelavo podatke še binarizirati.

Tipalo reže

V razvojnem centru tovarne FIAT (NIC080), so izdelali mehansko tipalo, ki ga uporablja varilni robot pri odčitavanju natančne lege reže svara. Izdelali so dve izvedbi mehanskega tipala. Prva sestoji iz dveh pravokotno ležečih stopenjskih motorjev, ki omogočata vertikalno in horizontalno gibanje tipalnega trna. Število impulzov v stopenjskem motorju daje informacijo o položaju trna ob dotiku. Gibanje stopenjskih motorjev vodi računalnik z ustreznim programom.

Druga izvedba deluje na principu analognih elektromagnetnih pretvornikov za določanje premika. Sensor sestoji iz štirih pretvornikov, ki ležijo v ravnini, pravokotni na smer gibanja varilne šobe in se nahajajo nekaj centimetrov pred varilno šobo. Zunanja dva od teh senzorjev merita vodoravne odmike in širino reže.

Pri obeh izvedbah tipala vodimo informacijo iz senzorca preko povratne zanke v robotov kontrolni program, ki zagotovi gibanje varilne šobe in priključenega senzorca v ustrezen položaj, ter s tem odpravi napako, ki lahko nastane zaradi odstopanja tirnice reže od idealnega modela.

4.2. Senzorji oddaljenosti

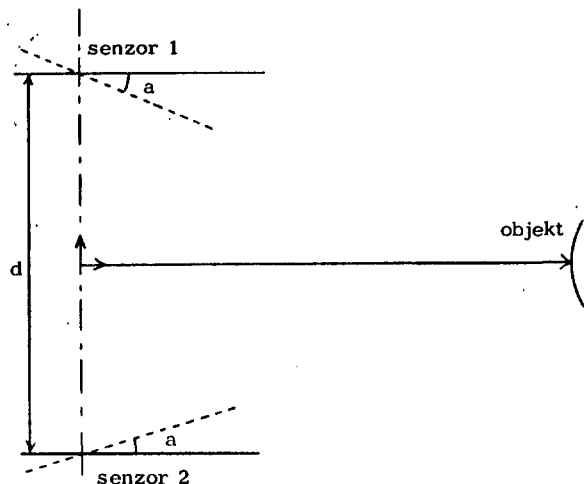
Druga večja skupina senzorjev so senzorcji za merjenje oddaljenosti. Posebno zanimiva sta dva tipa te vrste senzorjev, ki sta v uporabi pri robotih: to je infra rdeči sensor, ki pogosto podpira funkcijo avtomatičnega prijemanja in laserski sensor, ki je namenjen robotu za barvanje ali varjenje.

Infrardeči sensor oddaljenosti

Infrardeči sensor omogoča približno ugotavljanje oddaljenosti do predmeta v bližini nekaj decimetrov od senzorca. Zaradi sipanja žarka je posebno primeren za določanje razdalje do predmetov v gibanju. Nadaljnja prednost tega senzorca pa je niska cena. Koristno ga uporabljajo za podporo funkcije prijemanja pri kvadrilegičnih pacientih (CATR78) ter pri robotskem sestavljanju.

Infrardeče senzorce je možno povezati v sistem s optiko vlaken. Vsak sensor ima dva ločena sistema vlaken: en sistem je priključen na moduliran LED, drugi pa na foto-detektor. Tak senzorski sistem uporablja dva koplanarna žarka, ki sta razmaknjena za razdaljo d izbrano v odvisnosti od dimenzij predmeta, ki ga želimo prijeti. Žarka sta za kot α obrnjena drug proti drugemu. Na ta način parametri d in α določajo karakteristike senzorca. Na primer velika vrednost d in majhen α pomeni sensor z majhno natančnostjo in velikim obsegom. Ko je izbran sensor obrnjen v določeno ravnino, zavisi odgovor samo od razdalje predmeta od senzorca (Slika 6).

Seveda pa je infrardeč sensor občutljiv na obliko in celo barvo tarče. Pri neravnih



Slika 6.: Infrardeči senzor oddaljenosti

tarčah senzorsko polje ni točkasto, integracijski učinek pa lahko povzroči deformacijo krivulje intenzivnosti. Tudi orientacija predmeta je za ta senzor zelo pomembna. V idealnem primeru je normala pravokotna na ravnino opazovanega predmeta šarkov, večji odklon od 90 stopinj pa povzroči možno oslabitev signala in premik maksimuma.

Laserski senzor oddaljenosti

Drugi zanimiv senzor oddaljenosti je laserski senzor, ki ga je na področju industrijske robotike možno uporabiti tudi pri robotskem barvanju in robotskem varjenju krivih površin (UEDA79). Ta senzor uporablja žarek He-Ne laserja, vrteče ogledalce in foto senzor. Moč laserja je 10mW, vrteče ogledalce naredi poln obrat v 33 msec, tako, da je mogoče dobiti podatek o razdalji ob vsaki rotaciji ogledalca. Prednost tega laserskega sensorja je, da ta ni odvisen od barve površine obdelovanca. Uporablja se za merjenje razdalj od 10 do 200 cm s natančnostjo do nekaj mm.

4.3. Senzorji vida

Raziskave računalniško zasnovanega vida so že nekaj let eden od osrednjih problemov robotike. Ker je problem zahteven in je za enkrat na voljo le malo v industrijski praksi preverjenih metod in tehnologij, je razumljiva isredna pestrost načinov pridobivanja vizualnih informacij in metod za njihovo obravnavanje.

Tehnološka zasnova za robotski vid so televizijske in polvodniške kamere, diskretne fotodiode in linearne kamere.

Tudi metode zajemanja slikovnih podatkov so zelo pestre: kamera je lahko fiksno pritrjena nad delovni prostor robota, lahko se njena smer, lega in snemalni parametri računalniško vodljivi, lahko pa je tudi pritrjena na robotsko roko. Za opazovanje zahtevnejših prostorskih operacij je potrebno več kamer (2, 3 ali celo 4) hkrati.

Veliko sistemov za gledanje je namenjenih prepoznavanju oblike predmetov. Sposobnosti prepoznavanja so različne, od sposobnosti prepoznavanja omejenega števila oblik ob predpostavki, da je objekt pravilno nameščen, do prepoznavanja poljubne oblike celo v realnem času. Metode prepoznavanja so zasnovane na programih, ki so se sposobni

učiti in na primerjanju vizualne informacije s notranjim modelom. V ta namen je pogosto treba spreminjati položaj kamere, ki je vgrajena v robotovo roko, dokler predmeta ne snemamo s referenčne smeri.

Za razpoznavanje objektov s vidom je izredno pomembna osvetlitev predmetov. Praviloma je potrebna dodatna osvetlitev, ki poudarja kontrast med temnim ozadjem in svetlim predmetom.

Robotski sistemi vida so praviloma dokaj počasni in razpoznavajo v realnem času le preproste objekte in scene. Najverjetneje pa je vzrok temu saenkrate še dokaj kratka zgodovina robotike. Zavedati pa se moramo, da je procesiranje podatkov iz vidnega sensorja zelo zahteven proces saj dobimo informacijo o obliki in orientaciji predmeta šele s zapletenimi algoritmi in metodami prepoznavanja, ki jih dosedanja kratka razvojna pot robotike še ni uspela zadovoljivo razviti.

Primeri sensorjev vida

Pri vidnih sensorjih so oblike sensorjev, metoda njihove uporabe in njihova omejitve zelo različne. Poglejmo nekaj značilnejših uporab vidnih sensorjev pri obstoječih robotih, funkcijo robotov, ki te sensorje uporabljajo in velikost občutljivostnega polja sensorja, kar kaže naslednja tabela.

FUNKCIJA ROBOTA	VRSTA SENZORJA	ORČUTLJIVO-STNO POLJE SENZORJA
manipulator za nameščanje in sortiranje	1 TV kamera	128x128 slikovnih elementov
zasnova lokacije	1 TV kamera	50x50 do 244x256 slikovnih el.
vrtnanje ploščic tiskanega vezja	1 TV kamera	300x400 slik. elementov
identifikacija gibajočih se predmetov	1 TV kamera	573x512 slik. elementov
pritrjevanje priključkov na LSI	2 TV kamere	320x240 slik. elementov
lokacija in prepoznavanje ploskih predmetov	1 TV kamera	256x256 slik. elementov
prepoznavanje položaja v prostoru	4 TV kamere	
manipulator	linearno polje diod	256x1 slik.el.
identifikacija predmetov	linearno polje diod	128x1 slik.el.
razpoznavanje predmetov	polje polprevodniških elementov	50x50 fotosl. elementov
opis scene	multikanalni senzorski sistem s TV kamero	

Uporaba vida v robotiki

S sensorji vida so vse pogostejše opremljeni številni manipulatorji za sortiranje in nameščanje predmetov. Tipični senzorski sistem industrijskega manipulatorja, ki ga upravljamo s pomočjo računalniškega vida (GLEA79) sestoji iz n. pr. GE model TN-2200 TV kamere s 128 x 128 elementi ter iz mikroračunalniškega vmesnika, ter DEC LSI-11 mikroračunalnika.

Omanjena TV kamera ima ravno pravšnjo prostorsko resolucijo in dinamičen obseg ter razmeroma nisko ceno. Njena pomanjkljivost, značilna za številne silicijeve polvodniške kamere pa je njena omejena uporaba, da pri močnih svetlobah snemanje ni možno.

Elektronika:

Tudi robot za vrtnanje ploščic tiskanega vezja (HALE75) uporablja za senzor TV kamero, ki snema polprevodniško ploščico, računalnik Honeywell 516 pa obdela TV sliko. Računalnik upravlja vrtnalni stroj samo s pomočjo informacij, ki jih dobi iz TV slike. Signal TV kamere, ki snema ploščico tiskanega vezja preko polposrebnega arcala, vsorčimo s maksimalno resolucijo 300 x 400 slikovnih elementov. Vsak slikovni element lahko digitaliziramo v 5 bitov. Slikovne podatke je treba s primerno matematično metodo obdelati in s pravo strategijo določiti vrtnala mesta. Vizualni senzor služi po izvršenem vrtnanju hkrati tudi za kontrolo opravljenega dela.

Identifikacija gibanja:

Robot za identifikacijo gibajočih se predmetov (JAIN77) v zaporedju slik okolice uporablja za vidni senzor TV kamero, ki snema cestno križišče. Digitalizirana TV slika sestoji iz 573 vrstic s 512 slikovnimi elementi. S pomočjo obdelave slikovnih podatkov robot identificira gibajoče se predmete.

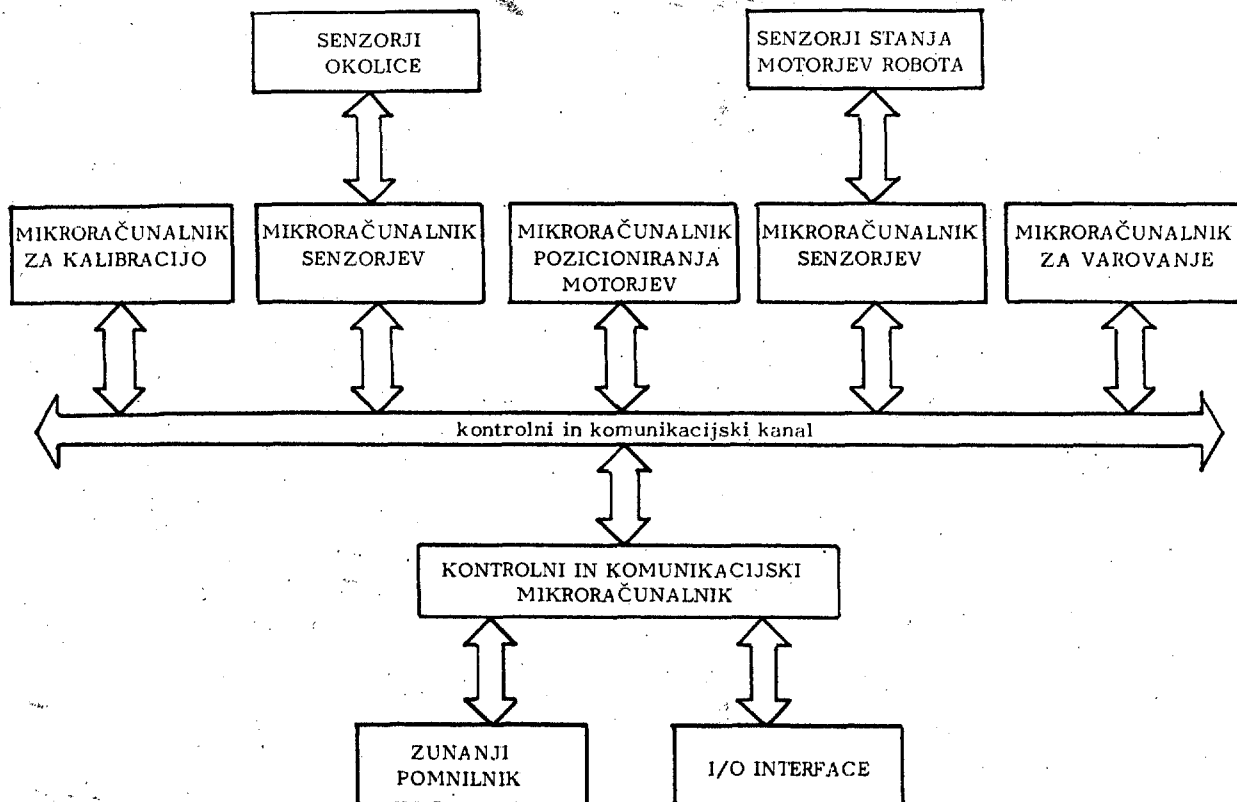
Mikroelektronika:

Popolnoma avtomatičen robotski sistem za pritrjevanje žic na LSI vezja (MESE77) uporablja dve TV kameri kot senzor za določanje pozicij. Kameri posnameta slike dveh različnih zunanjih delov pocamersne LSI ploščice. Na osnovi informacije iz sensorja se s pomočjo posebnega hardvarnega vezja za obdelavo slike in miniračunalnika približno določi položaj za pritržitev žic v vsakem vidnem polju. Računalnik izračuna položaj za vse priključke in te podatke posname na magnetni trak za obdelavo sledečih LSI elementov. Ustrezno resolucijo so zagotovili s mikroskopskimi ležami. Analogen video signal iz vsake kamere vzorčimo in efektivno posneto polje je razdeljeno na 320 x 240 slikovnih elementov. Tako ustrezna en slikovni element kvadratu s stranico 2,5 mikrona.

5. MIKROPROCESORSKI KONTROLNI SENZORSKI SISTEM

Za kontrolo večine senzorskih sistemov se uporabljajo mikroračunalniki, ki omogočajo obdelavo senzorskih informacij in upravljanje v realnem času. V primeru zahtevnejših obdelav informacij je mikroračunalnik na nivoju sensorja povezan z večjim centralnim miniračunalnikom, ki iz višjega nivoja koordinira robotove akcije v odvisnosti od senzorskih informacij okolja.

Ker pa pri krmiljenju robotov zahtevamo procesiranje v realnem času in je cena zmogljivega miniračunalnika relativno visoka, se vedno bolj uveljavlja izvedba krmilne logike z multiprocesorskim sistemom mikroračunalnikov. Ti so razporejeni ob osrednjem komunikacijskem kanalu in vsak



Slika 7.: Mikroprocesorski kontrolni senzorski sistem

mikroračunalnik lahko komunicira z vsemi drugimi. S programsko opremo in kontrolnim mikroračunalnikom je določeno mesto mikroprocesorja v hierarhiji upravljanja. Na ta način multiprocesorski sistem omogoča vzporedno izvajanje algoritmov na različnih nivojih in omogoča sensorju neposredno komunikacijo z aktuatorjevimi mikroprocesorji.

Ob uporabi povratne zanke za natančno in hitro upravljanje robota potrebujemo dva senzorska sistema: to je sensorje fizične okolice robota in sensorje stanja motorjev vseh prostostnih stopenj robota (Slika 7). Vsak senzorski sistem vodi ustrezen mikroračunalnik, ki hkrati pretvarja električni signal iz sensorja v digitalno obliko in na senzorskem nivoju obdela informacije iz sensorja. Na skupni komunikacijski kanal, ki ga kontrolira kontrolni in komunikacijski mikroračunalnik so priključeni še mikroračunalnik za pozicioniranje motorjev robota, mikroračunalnik za kalibracijo in mikroračunalnik za varovanje. Tu vidimo pomembno vlogo sensorjev pri kontroli in upravljanju robota in mesto, ki ga sensorjiasedajo v hierarhiji multiprocesorskega sistema.

6. SKLEP

Pričujoči članek o senzorskih sistemih pri robotih analizira najzanimivejše dosežke robotike, kjer s pomočjo sensorja postane delo robota bolj samostojno, kjer sensor omogoča večjo fleksibilnost robotskega sistema, večjo točnost pozicioniranja in omogoča operacije, ki brez sensorja sploh niso mogoče. Potreba po ustreznih sensorjih v povratni zanki robotskih upravljaljskih sistemov je vsak dan večja, saj ima le na ta način upravljaljski sistem neprestano vpogled v stanje okolice in stanje robotskega manipulacijskega sistema. Današnji trend razvoja kaže velik razmah robotov opremljenih s senzorskimi sistemi, ki so krmiljeni s multiprocesorskim sistemom mikroračunalnikov. Po drugi strani pa tak silovit razvoj robotov vabuja pri ljudeh strah pred industrijskimi roboti in sprašujejo se kakšne bodo posledice za zaposlene delavce, ki se bodo počutili odvečne in zavržene. Mogoče prehod iz homo faberja v homo ludens ne bi smel biti prenaagal.

7. LITERATURA

AI-TR-310 ČNEW PROGRESS IN ARTIFICIAL INTELLIGENCE, Arm Theory, MIT, Artificial Intelligence Laboratory, 1974

BRI079 Briot M.: THE UTILISATION OF AN "ARTIFICIAL SKIN" SENSOR FOR THE IDENTIFICATION OF SOLID OBJECTS Proc. 9th Int. Symp. on Industrial Robots, Washington D.C. 1979, pp. 529-547

BRUS78 Brussel H.V. et al.: AUTOMATIC ASSEMBLY BY ACTIVE FORCE FEEDBACK ACCOMODATION Proc. 8th Int. Symp. on Industrial Robots, Stuttgart, 1978, pp. 181-193

CATR78 Catros, J. et al.: AUTOMATIC GRASPING USING INFRARED SENSORS Proc. 8th Int. Symp. on Industrial Robots, Stuttgart, 1978, pp. 132-142

GLEA79 Gleason G. J. et al.: A MODULAR VISION SYSTEM FOR SENSORS-CONTROLLED MANIPULATION AND INSPECTION Proc. 9th Int. Symp. on Industrial Robots, Washington D.C. 1979, pp. 57-70

HALE75 Hale, J. A. G. et al.: CONTROL OF A PCB DRILLING MACHINE BY VISUAL FEEDBACK Proc. 4th IJCAI, Tbilisi, Georija, SZ, 1975 pp. 775-781

JAIN77 Jain, R. et al.: SEPARATING NON-STATIONERY FROM STATIONARY SCENE COMPONENTS IN A SEQUENCE OF REAL WORD TV-IMAGES, Proc. 5th IJCAI, Massachusetts Institute of Technology, Boston, 1977, pp. 612-618

MESE77 Mese, M. et al.: AN AUTOMATIC POSITION RECOGNITION TECHNIQUE FOR LSI ASSEMBLY, Proc. 5th IJCAI, Massachusetts Institute of Technology, Boston, 1979, pp. 650-655

PREŠ79 Prešern, S. et al.: MODEL TIPALNEGA SENZORSKEGA SISTEMA PRI ROBOTU ZA AVTOMATSKO OBLOČNO VARJENJE, IJS, E/4 1979, v delu

PREŠ79 Prešern, S. et al.: SENZORSKI SISTEMI ROBOTOV IN SENZOR ZA VARJENJE Delovno poročilo raziskovalne naloge "Aplikativna robotika" IJS E/4, 1979

STUT79 Stuts, G. et al.: THE CONTROL DESIGN OF AN INDUSTRIAL ROBOT WITH ADVANCED TACTILE SENSITIVITY Proc. 9th Int. Symp. on Industrial Robots, Washington D.C. 1979, pp. 519-527

ŠPEG79 Špegel, M. et al.: PREGLED RAZISKOVALNIH PROGRAMOV ROBOTIKE IJS DP-1979, Inštitut J. Stefan, Ljubljana, januar 1979

UEDA79 Ueda, M. et al.: A SIMPLE DISTANCE SENSOR AND A NEW MINI-COMPUTER SYSTEM Proc. 9th Int. Symp. on Industrial Robots, Washington D.C. 1979, pp. 477-488

UMET78 Umetani, Y.: PRINCIPLE OF A PIEZO-ELECTRIC MICRO MANIPULATOR Proc. 8th Int. Symp. on Industrial Robots, Stuttgart, 1978, pp. 406-413

AN ASSEMBLER CONSTRUCTION FOR THE INTERACTIVE PROGRAM DEVELOPMENT

D. M. VELAŠEVIĆ

UDK: 681.3 : 519.682

ELEKTROTEHNIČKI FAKULTET, BEOGRAD

A new technique of assembler construction for the interactive program development is described. The separate editing phase for the creation of the source program file is avoided, contributing in this way to an efficient program development. The first pass of the assembler creates the program file and concentrates in itself the maximum possible error diagnostics and tasks, thus giving the opportunity to the user to correct the whole instruction line immediately, in the assembly phase. Basic modules of the assembler - lexical, syntax and semantic analyzers - are presented generally in the context of new conception.

KONSTRUKCIJA ASEMBLERA ZA INTERAKTIVNI RAZVOJ PROGRAMA. U radu je opisana nova koncepcija konstrukcije asemblera namenjena efikasnijem interaktivnom razvoju programa. Njome se izbegava odvojena faza za formiranje izvornog programa. U prvom prolazu, asembler stvara programsku datoteku koncentrišući u sebi maksimum zadataka i dijagnostičkih mogućnosti, pružajući istovremeno priliku korisniku da ispravi celu liniju instrukcije neposredno, u fazi prevodjenja. Osnovni moduli asemblera - leksički, sintaksni i semantički analizatori - prikazani su uopšteno u okviru nove koncepcije

Introduction

In realization of programs written in assembly language in an interactive environment there are the following phases:

- editing
- assembling
- loading
- execution

The editing phase is a characteristic property of the interactive program development and the first necessary step to start with in the program realization. Let the realization of a program requires an average time T (session time), i.e.

$$T = T_E + T_A + T_L + T_{EX} + T_E^{call} + T_A^{call} \quad (1)$$

where

- T_E - average time of editing phase
- T_A - average time of assembly phase
- T_L - average time of loading phase (including time to call the loader and start the execution)
- T_{EX} - average time of execution phase
- T_E^{call} - time to call and load the editor
- T_A^{call} - time to call and load the assembler

The average values are derived from a collection of programs realized in a given time interval. The value of T relates to the case when there are no formal and informal errors in the created program file. For small T_{EX} , T_E dominates.

If a program contains formal errors, T increases and becomes now

$$T = T_E + \sum_{i=1}^{n-1} T_{C_i} + nT_A + T_L + T_{EX} + n(T_E^{call} + T_A^{call}) \quad (2)$$

where

- T_{C_i} - average time of i^{th} correction of the source program file
- n - number of editor and assembler calls

Let the arrival rate of programmers to the terminal station follows Poisson's distribution. The average queue of programmers, $E(q)$, waiting for the terminal is given by Khintchin-Pollaczek equation,

$$E(q) = \frac{\rho^2}{2(1-\rho)} \left(1 + \left(\frac{\sigma_T}{T} \right)^2 \right) \quad (3)$$

where

$\rho = E(n)T$, $E(n)$ - average number of programmers arriving at the terminal station in the given time interval. In fact, ρ is the utilization factor of a terminal station.

σ_T - standard deviation of T

q - number of programmers waiting for the terminal in the given instant of time

Example 1: If the standard deviation $\sigma_T = T/2$ is adopted, Eq. (3) becomes

$$E(q) = \frac{\rho^2}{8(1-\rho)}$$

Let $E(n) = 2/\text{hour}$, $T_E = 10 \text{ min}$, $T_A = 2.5 \text{ min}$ (including listing required at each assembly), $T_L = 1/4 \text{ min}$, $T_{EX} = 1/4 \text{ min}$ (very short programs), $T_A^{call} = T_E^{call} = 1/4 \text{ min}$, $n = 3$, $T_{C_1} = 4 \text{ min}$, $T_{C_2} = 2 \text{ min}$. For these values, the average number of programmers waiting for the terminal is $E(q) = 3$.

New assembler construction

To improve the efficiency of the program realization, we propose an assembler construction as given in Fig. 1. By this conception, the assembler can accept a source program either directly from the terminal (eliminating in this way the need for the separate editing phase) or from already created source program file (created by the editor or the assembler). When the assembler accepts a source program directly from the terminal, it creates the source and intermediary program file, and outputs the error diagnostics (if any) immediately upon an instruction line is entered. If the assembler reads the source program from the file then it

creates a new source file declaring the existing one as a back-up and deleting the existing back-up, Fig. 1. The assembler discards each incorrect instruction line, thus giving the programmer an opportunity to enter the correct one. Because this error diagnostics is given in the first pass, some errors can slip away. By analyzing the diagnostic error message summaries for several assembly languages, it was found that almost all errors can be caught in the first pass with the exception of those ones which could be correlated with the forward reference to symbols (for instance, the violation of the relocation rule). In any case, the number of such errors is small and could be further reduced by an appropriate language construction. When the assembler encounters the END directive it searches for all undefined symbols and, if any, requires from the programmer to define them or to abandon the assembly if he wants it. Upon the successful processing of the END directive, the assembler terminates the first pass and starts the second one. Eq. (2) takes now the following form (no errors in the second pass):

$$T = T_E^A + NT_C^L + T_A + T_L + T_{EX} + T_A^{call} \quad (4)$$

where

T_E^A - average time of editing in assembly phase (approximately of the same value as T_E)

T_C^L - average time for correction of one instruction line

N - average number of incorrect instruction lines detected in the first pass

NT_C^L is generally smaller than $\sum_{i=1}^{n-1} T_{C_i}$ because the

corrections in the editing phase require the positioning of the pointer to an instruction line and verification that the right instruction line is chosen for correction. Thus, we have

$$NT_C^L = \alpha \sum_{i=1}^{n-1} T_{C_i}, \quad \alpha < 1.$$

Example II: If we suppose that all errors are caught in the first pass and that there are no informal errors, then for the same values of parameters as given in Example I (only n is changed to 1) and for $\alpha = 0.8$ and $\alpha = 0.5$, one obtains $E(q) = 0.56$ and $E(q) = 0.4$ respectively, which represents a significant improvement in the program realization.

However, there is a probability that the assembler will not catch all errors in the first pass in which case the editor must be called to correct the file created by the assembler. Also, the editor will be called when a program does not function properly due to informal errors.

Lexical analyzer

Owing to the nonstandardized structure of the assembly languages it is difficult to present an assembler construction which could be applicable up to details in all assembly languages¹. Nevertheless, a general conception can be elaborated in such a way to give the maximum flexibility in the design in order to meet the requirements of a specific assembly language. In this context, we shall present first the lexical analyzer. An address space is assigned to the lexical analyzer for the construction of a temporary table which we shall call the transformation table.

Let be given an instruction line L which is transformed by the lexical analyzer in a set of items,

$$T(L) = \{\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_i, \dots, \alpha_n\} \quad (5)$$

where

T - applied transformation

α_i - item in the set, i.e. in L

n - number of items in the set (variable, depending on the particular instruction line)

The lexical analyzer builds up the transformation table based on the set of items $\alpha_1, \alpha_2, \dots, \alpha_n$. For each item in the set, α_i , an entry in the transformation table is created with an associated set of information, $\beta_1, \beta_2, \dots, \beta_{i-1}, \dots, \beta_n$. Generally, this set could contain the following:

β_1 - field indicator (FI): 0-label field, 1-operation code field, 2-address field. FI designates the position of an item in the instruction line.

β_2 - item internal code (IIC): 1-symbol, 2-fixed-point constant (integer), 3-fixed-point constant (fractional), 4-floating-point constant, 5-alphanumeric string, 6-alphanumeric constant, 7-arithmetic operators, 8-relational operators, 9-boolean operators, 10-separators, 11-special characters, 12-line terminator. IIC is needed for syntax analysis.

β_3 - table type (TT): 0-symbol table, 1-operations code table, 3-transformation table. TT is needed to locate the position of a relevant item in the assembler tables.

β_4 - table index (TI): index in a table designated by TT. TI eliminates repeated search for a relevant datum.

β_5 - number precision (NP): 0-standard precision, 1-double precision. This is not a general characteristic of items in the set but, for the sake of uniformity, it is a part of each entry in the transformation table.

If α_i is a number, alphanumeric string or alphanumeric constant, its binary equivalent is stored at the bottom of the transformation table, Fig. 2.

The transformation table is preceded by a header which contains the current value of the program counter (PC).

Besides the construction of the transformation table, the lexical analyzer prepares a variable length vector

$$SRS, IIC_1, IIC_2, \dots, IIC_k, \dots, IIC_m$$

as an input to the syntax analyzer. IIC_k corresponds to k -th entry in the transformation table. The first component of the vector, SRS (syntax rule selector), serves as a selector of the syntax rule which must be obeyed for a particular operation code or a group of operation codes. In fact, SRS is the address of the corresponding program submodule for the syntax analysis. Thus, each entry in the operations code table should be slightly modified to comprise the address of the program submodule for a corresponding syntax rule.

Upon the completion of the lexical analysis the control is passed to the syntax analyzer having as input the vector created by the lexical analyzer.

Syntax analyzer

The syntax analyzer uses the variable length vector and the transformation table for its work. A sequence of states $S = 1, 2, \dots, i, \dots, n$ is assigned to each group of operation codes with the same syntax and semantics. The initial state is 1. The operation of the syntax analyzer is based on the decision table given in Fig. 3. The elements of this table are numbers d_{iIIC_k} , $i = 1, 2, \dots, n$, $k = 1, 2, \dots, m$, identifying the

program segments which perform the particular tasks of the syntax analyzer. A program segment is determined by Eq. (6):

$$d_{iIIC_k} = D(i, IIC_k) \quad (6)$$

where D denotes the decision table. Upon the execution of the program segment d_{iIIC_k} , the syntax state remains

the same or can be changed depending on the specific conditions for the given program segment.

Owing to an interactive conception of the assembler requiring the maximum error diagnostics in the first pass, the semantic analyzer is integrated into the syntax analyzer. For instance, the evaluation of the expressions, the verification of their relocation correctness and the generation of the intermediary form of an instruction line are the tasks of the semantic analyzer which are performed in the first pass concurrently with the syntax analysis.

Semantic analyzer

The central problem in this conception is the complete analysis of the expressions because of one simple reason: the expressions can contain forward references which disable their evaluation and verification in the first pass.

Let an expression is presented in the following canonical form:

$$R + A + S_1 \pm S_2 \pm \dots \pm S_j \pm \dots \pm S_p \pm P_1 \pm P_2 \pm \dots \pm P_i \pm \dots \pm P_q \quad (7)$$

where

R - relocatable value

A - absolute value

$S_1, S_2, \dots, S_j, \dots, S_p$ - forward references

$P_1, P_2, \dots, P_i, \dots, P_q$ - terms containing multiplication and/or division of forward references having the form $P_i = A_i \Pi_i / DS_i$ where A_i - factor with absolute value, S_i - forward or external reference (at least one reference must be forward), Π_i / D - multiplication and/or division. S_i can belong to $S_1, S_2, \dots, S_j, \dots, S_p$ or not. Finally, if P_i is preceded by a minus sign its absolute value can be complemented, so we need only consider addition of P_i .

The external references and terms containing exclusively multiplication and/or division of external references are not included in Eq. (7). They can be resolved by an algorithm similar to that one described by Barron².

Because forward references disable the complete analysis of the expressions, their evaluation and verification must be started in the second pass. To avoid this situation, we apply to forward references and terms an algorithm similar to that one described for the case of external references and products². This has for consequence the creation of two additional tables:

- forward reference table (FRT)
- table of terms with forward references (TTFR)

The entries in these tables are depicted in Fig. 4a and 4b, respectively. In Fig. 4a PC denotes the place in the program where a symbol value S_j is to be added or subtracted. P is a pointer to an entry in the symbol table containing S_j , and AS is an indicator which designates whether a symbol value should be added or subtracted. In Fig. 4b PC has a similar interpretation as in Fig. 4a. $P_1^2, P_2^2, \dots, P_i^2$ are the pointers to entries in the symbol table containing the elements of a term P_i , excluding the external references for which an entry in a separate table of terms with external references (TTER) is created². MD is an indicator which denotes whether a symbol is a multiplier or a divisor, and MM whether P_i is a mix of

forward and external references or consists exclusively of forward references. P is a pointer to the next entry in TTFR. For the case of the mixed mode of P_i one proceeds in the following way:

- set MM indicator
- create an entry in TTER with its absolute multiplier set to value A_i
- create an entry in TTFR without external references with absolute multiplier replaced by a pointer to A_i in TTER.

Upon encountering END directive in the first pass, all forward references are resolved so that the assembler can start evaluating P_i . The procedure is as follows:

- check if MM is set. If not, perform successively the operation designated by MD between absolute multiplier A_i and the symbol values denoted by the pointers $P_1^2, P_2^2, \dots, P_i^2$ and store the result in A_i 's location.

Otherwise, perform successive operation designated by the pointers $P_1^2, P_2^2, \dots, P_i^2$. Upon the completion of this calculation A_i in TTER is multiplied by the value obtained in the evaluation of the corresponding entry in TTFR. Furthermore, it must be verified in both cases that there is no multiplication or division by a relocatable value.

The evaluation of an expression (not completely processed in the first pass) by using FRT and TTFR in the second pass is very simple and efficient. Before we explain this procedure, it is necessary to describe the intermediary form of an instruction line because the assembler deals only with it in the second pass. This intermediary form is illustrated in Fig. 5a. PC is mapped directly from the transformation table.

NW - number of words generated by an instruction line. NW can have zero value for those instructions which do not generate machine word (s) (for instance EQU directive)

GSP - general state of processing. An instruction line can be completely processed in the first pass, GSP = 1, or not, GSP = 0. In the latter case, the indicator PSP (partial state of processing) for the corresponding generated machine word must be examined in the second pass to find those words with PSP = 0 (not completely processed). For instance, the instruction line COPY B,A generates three machine words. If A is forward and B backward reference, the intermediary form for this instruction is given in Fig. 5b.

REL - relocation indicator (absolute, program relocation, common relocation, etc.) assigned to each generated machine word

END - denotes the end of the intermediary program file

GMW - generated machine word.

It should be noted that the mentioned indicators are the basic ones. For each assembly language new indicators can be introduced to meet the requirements imposed by the language construction.

The processing of the intermediary form of an instruction line in the context of FRT and TTFR is as follows:

1. Check GSP. If GSP = 1 then goto 5..
2. Start searching for the next word with PSP = 0. If there is no such a word then goto 5..
3. For this word, take the next entry from FRT to check whether it has the corresponding value of PC or not. If not then goto 4.. Otherwise, add or subtract (according to AS) the symbol value to the word and verify relocation correctness according to the following rules:

$A + A \rightarrow A, A + R \rightarrow R, R + A \rightarrow R, R + R \rightarrow \text{FAULT}$
 $A - A \rightarrow A, R - A \rightarrow R, R - R \rightarrow A, A - R \rightarrow \text{FAULT}$
 Set the pointer to the next entry in FRT and goto 3..

4. For this word, take the next entry from TTFR to check

whether it has the corresponding value of PC or not. If not then goto 2. Otherwise, check if MM is set. If set, change the pointer to the next entry in TTFR and goto 4. If not set, add A_i to the word (the relocation correctness is not verified because A_i is an absolute value). Change the pointer to the next entry in TTFR and goto 4.

5. Include the generated machine word (s) in the object module.

By introducing FRT, TTFR and intermediary form of an instruction line the second pass is considerably simplified. The handling of FRT and TTFR is very simple and does not require any searching.

If other types of operators (logical, relational) are permitted in expressions, FRT and TTFR can be easily adapted to comprise that case by establishing new operation indicators in addition to AS and MD. If the evaluation of an expression follows the precedence rule, all entries in FRT or TTFR with the same PC must be ordered by respecting that rule.

Further improvement

The assembler construction proposed so far is very flexible and does not introduce any problems in the realization. In addition, it contributes to a faster program realization and improves the system performance. The improvement is especially significant in the case of formal errors appearing only in the first pass because there is no need to call the editor. In the case of formal errors detected during the second pass and informal errors, the editor must be called.

If we suppose that it is necessary to call the editor m times to correct the source program due to informal errors until the program starts to function properly, Eq. (4) becomes

$$T = T_E^A + NT_C^L + (m+1)T_A + \sum_{i=1}^m T_{EX_i} + \sum_{i=1}^m T_{C_i} + mT_L + mT_E^{call} + (m+1)T_A^{call} \quad (8)$$

Example III: For the same values of parameters as in *Example II* and for $m = 3$, $T_{C_1} = T_{C_2} = 4$ min, and

$T_{EX_1} = T_{EX_2} = T_{EX_3} = 1/4$ min, one obtains $\rho = 1,33$ (for $\alpha = 0.8$) and $\rho = 1.28$ ($\alpha = 0.5$), which indicates that the terminal station service is failed due to overloading. If we add one terminal station at the observed point thus reducing $E(n)$ to unity for each station, we obtain $\rho = 0.67$ (for $\alpha = 0.8$) and $\alpha = 0.5$. The values for $E(q)$ are 0.85 and 0.7 respectively.

The overloading which appeared in *Example III* could be decreased by the same assembler construction as presented in Fig. 1, with addition of editing facility consisting of the possibility to replace an instruction line by an arbitrary number of instruction lines. To perform the corrections of a source file it is necessary to give to the assembler in advance all instruction line numbers (known from previous assembler listing) where the corrections are to be made, for instance

```
ASSEM/R: ILN1, ILN2, .. ILNM/NAME
```

where

ASSEM - assembler call

R - request for replacement

ILN1, ILN2, ... ILNM - list of instruction line numbers

NAME - source file name

The assembler will stop at each indicated instruction line number waiting for the correction. Upon its completion the assembler will continue with assembling until the next instruction line designated for correction is encountered. This conception makes possible

the corrections of all formal and informal errors without calling the editor. Eq. (8) becomes now

$$T = T_E^A + NT_C^L + (m+1)T_A + mT_L + \sum_{i=1}^m T_{EX_i} + M^*T_C^L + (m+1)T_A^{call} \quad (9)$$

where M^* is total number of lines to be entered for correction.

Example IV: Under the same condition as given in *Example III* and

$$M^*T_C^L = \alpha \sum_{i=1}^m T_{C_i}$$

one obtains $\rho = 1.23$ (for $\alpha = 0.8$) or $\rho = 1.05$ (for $\alpha = 0.5$). Adding one terminal station at the observed point we obtain $\rho = 0.62$ (for $\alpha = 0.8$) or $\rho = 0.53$ (for $\alpha = 0.5$). The values for $E(q)$ are 0.64 and 0.37 respectively. The improvement is evident.

Conclusion

The proposed interactive assembler construction has several advantages over the standard conception. It improves the program realization especially in the case of short (student) programs, thus reducing the queues at the terminal stations. Further, it eliminates or reduces a very annoying ping-pong game between the assembler and the editor improving in this way the system performance. The structure of the assembler is very flexible and has a general form applicable not only to an interactive environment. The packing density of the intermediary form is high so that, for instance, 30 - 60 instruction lines in average (depending on the particular assembly language) could be stored onto one disk sector of 256 words. The editing facilities which should be built into the assembler are very elementary and do not complicate substantially its realization. And finally, this conception has a very favourable psychological influence on programmers and makes the learning process faster.

References

1. Gear C.W.: Computer Organization and Programming, McGraw-Hill, Kogakusha, 1974.
2. Barron D.W.: Assemblers and Loaders, Macdonald, London, 1969.

O POSTAVLJANJU JEDNOSTAVNIH HIPOTEZA POMOĆU RAČUNARA (O LANCIMA ATOMA)

D. JEMUOVIĆ

UDK: 681.3 : 51

MATEMATIČKI INSTITUT, BEOGRAD

U radu su date neke izmene definicije relacije "+" i algoritma za generisanje totalnog lanca izmedju dva atoma, koji se u prvobitnom obliku mogu naći u članku J.Reynoldsa. Takodje je dat aritmetički izraz za računanje razlike u veličini izmedju dva susedna atoma (povezana sa "+").

SOME NOTICES ABOUT SIMPLE HYPOTHESIS FORMATION ON COMPUTER (ABOUT CHAINS OF ATOMS). The paper is based on the work of J.Reynolds and gives a modifications of his definition of relation "+" and of algorithm for generating total chains between two atomic formulas. Also, the paper contains an expression for calculating the difference in value of two atoms connected with "+"

Do sada je konstruisan veliki broj algoritama koji dokazuju teoreme pomoću računara i primenjuju se kad god je potrebno ispitati neprotivurečnost nekog stava sa već prihvaćenim činjenicama, ili uopšte, znanjem iz neke oblasti. Bilo da su to programi koji se mogu primenjivati u okviru pojedinih teorija (specijalizovani programi) ili imaju opšti karakter i bez obzira na razne metode koje koriste svi oni ukazuju na mogućnosti mehanizacije deduktivnog mišljenja.

Poslednjih godina u žiži interesovanja su se našli i komplementarni problemi, jer je pored provere teorije pomoću računara, od velikog značaja i njeno eventualno, generisanje od strane računara. Posebno je važno u slučaju kada se raspolaže sa velikim brojem eksperimentalnih podataka na osnovu kojih treba formirati hipoteze-generalizacije koje će objединiti više pojedinačnih slučajeva. Pri tome se hipoteza postavlja samo na osnovu dela podataka (posebno ako ih ima mnogo), a ostali se koriste za njenu proveru - svaki od njih povećava ili smanjuje uverenje o njenoj tačnosti. Istovremeno se mora ispitivati da li postoji protivurečnost dobijene hipoteze sa ostalim znanjem koje posedujemo, pa ako se protivurečnost ne nadje, hipoteza se prihvata i, možda, još više uopštava. Ako kontradikcija postoji, moraju se izvršiti neke korekcije: vrši se povratak na slabiju generalizaciju, dozvoljava se mogućnost postojanja izuzetaka od pravila i td.

G.Polya [5] smatra da se svaka generalizacija sastoji iz zamene konstanti promenljivim ili predstavlja uklanjanje nekih ograničenja. Prvi pokušaji induktivnog zaključivanja na računaru zasnivali su se na pretpostavci da je indukcija "inverzna dedukcija", pa su mnogi algoritmi korišćeni u dedukciji predstavljali inspiraciju za konstruisanje postupaka koji bi obavljali komplementarne zadatke.

Ovde će biti izložena neka zapažanja o formiranju jednostavnih hipoteza (radi se isključivo sa najjednostavnijim formulama - atomima), koja se baziraju na radovima J.Reynoldsa i G.Plotkina. Upoznajmo se, prethodno, sa nekim osnovnim pojmovima:

Posmatraćemo skup atoma generisanih nad prebrojivim skupovima konstanti i promenljivih, nepraznim skupovima funkcijskih i relacijskih simbola. Svi atomi su zatvoreni univerzalnim kvantifikatorima, koje ne pišemo, jer ćemo ih, prema dogovoru, podrazumevati, a njihov međusobni poredak je nebitan. U taj skup se uvodi sledeća operacija:
DEF.1: SUPSTITUCIJA je konačan skup uredjenih parova oblika (term, promenljiva), (t_i, x_i) , $i=1, \dots, n$, takvih da je $T_i \neq x_i$ za $i=1, \dots, n$ i sve druge komponente u parovima su međusobno različite. //

Primena supstitucije na atom je operacija zamene u atomu svih pojavljivanja promenljive koja je druga komponenta nekog para, termom koji je prva

komponenta tog para u supstituciji. Očigledno je da prazna supstitucija ne vrši nikakve izmene nad atomom na koji se primenjuje, kao i da je rezultat primene supstitucije na atom opet atom. U skup svih supstitucija definisanih nad posmatranom azbukom uvodi se operacija kompozicije:

DEF. 2: KOMPOZICIJA supstitucija

$Q = \{(t_1, x_1), \dots, (t_n, x_n)\}$ i $\sigma = \{(s_1, y_1), \dots, (s_m, y_m)\}$ u oznaci $\theta \cdot \sigma$, je skup $\theta \cdot \sigma$ takav da su u σ samo oni parovi iz σ takvi da je $x_i \neq y_j$ za sve $i=1, \dots, n$ i $j=1, \dots, m$, a u θ su parovi oblika (t_i, x_i) za koje je $t_i \neq x_i$ za $i=1, \dots, n$.

Skup supstitucija sa kompozicijom predstavlja semigrupu sa jedinicom (prazna supstitucija), a njegov netrivialan podskup (supstitucije koje su permutacije) čini grupu.

DEF. 3: SLUČAJ atoma A u odnosu na supstituciju σ je atom $B = A\sigma$ //

DEF. 4: Ako je atom A slučaj atoma B i atom B slučaj atoma A, onda su A i B VARIJANTE. //

Na ovaj način se u skup atoma uvodi jedna relacija ekvivalencije koja ga razbija na klase u kojima su atomi koji se razlikuju do permutacije. U svakoj klasi se može izabrati neki predstavnik i operacije sa klasama se svode na operacije sa njihovim predstavnicima:

DEF. 5: Klasa A je SLUČAJ klase B akko u A postoji atom koji je slučaj nekog atoma iz B. //

Binarna relacija nad atomima, definisana sa $A \leq B \stackrel{\text{def}}{=} (\exists \sigma) A\sigma = B$ je relacija kvazi uredjenja u skupu atoma, ali nije parcijalno uredjenje jer nije antisimetrična. Ta smetnja se uklanja prelaskom na klase ekvivalencije i relaciju $[A] \leq [B] \stackrel{\text{def}}{=} (\exists A \in [A]) (\exists B \in [B]) (A \leq B)$ koja je parcijalno uredjenje u skupu klasa.

Važna je osobina da iz $A \leq B$ proizlazi da $A \rightarrow B$ (A i B su atomi).

Postoje, dakle, totalno uredjeni skupovi (lanci) atoma (klasa ekvivalencije atoma), ali postoje i neuporedivi atomi (recimo: $P(A, f(b), x)$ i $P(y, a, x)$ su neuporedivi iako imaju isti relacijski simbol).

Minimalni element svakog lanca klasa atoma, je klasa koja sadrži atom u kome su svi termi međusobno različite promenljive, a maksimalni elementi su klase u kojima su jedini članovi atomi bez promenljivih (jednočlane klase). U skupu A/\equiv

minimalnih elemenata ima onoliko, koliko se različitih relacijskih simbola iz azbuke koristi, a maksimalnih onoliko, koliko je nad azbukom moguće formirati atoma u kojima se pojavljuju samo neke konstante (termi sa konstantnim argumentima) Zato se uvode jedinstven minimalni element-klasa u kojoj je univerzalni atom α (onaj čiji su slučajevi svi otali atomi) i jedinstven maksimalni element-klasa u kojoj je nulti atom Ω (atom koji nema nijedan svoj slučaj).

DEF. 6: Atom B je NEPOSREDAN SLUČAJ atoma A, u oznaci $A \rightarrow B$, akko je zadovoljen jedan od sledećih slučajeva:

- (1) $A = \alpha$, a B je atom čiji su svi termi međusobno različite promenljive,
- (2) $B = A\{(f(z_1, \dots, z_n), x)\}$, gde se x pojavljuje u A, a z_1, \dots, z_n se ne pojavljuju u A,
- (3) $B = A\{(y, x)\}$, x i y se pojavljuju u A,
- (4) $B = A\{(a, x)\}$, a je konstanta, a x se pojavljuje u A,
- (5) A je atom čiji termi imaju konstantne argumente, a $B = \Omega$.

Ako je $A \rightarrow B$, onda je i $A < B$ i B je slučaj od A dobijen primenom najjednostavnije supstitucije određene vrste, tako da se između A i B ne može umetnuti nijedan atom C bitno različit od A i B, tj.:

$$\neg (\exists C) (C \notin [A] \wedge C \notin [B] \wedge A < C < B)$$

DEF. 7: Niz atoma A_0, \dots, A_n je rastući (opadajući) lanac dužine n od A do B, akko je $A = A_0 < A_1 < \dots < A_n = B$ ($A = A_0 > A_1 > \dots > A_n = B$). Kada umesto $<$ stoji \rightarrow , lanac se naziva TOTALAN. //

Da za svaka dva atoma A i B takva da je $A \leq B$ ($B = A\theta$) postoji totalan lanac od A do B dužine $n \geq 0$, dokazuje se ispitivanjem korektnosti algoritma koji navodimo:

PRVI ALGORITAM ZA FORMIRANJE TOTALNOG LANCA IZMEDJU ATOMA A I B = $A\theta$ ($A < B$)

ULAZ: ATOM A I SUPSTITUCIJA θ

IZLAZ: TOTALAN LANAC OD A DO $A\theta$

I KORAK: Neka je $A_0 = A, \theta_0 = \theta$, $i = 1$.

II KORAK: Ako je svaka prva komponenta parova u θ_i konstanta ili promenljiva, preći na III, a ako postoji par $(f(t_1, \dots, t_k), x)$ u θ_i (za proizvoljne simbole f, t_i, x iz azbuke), staviti $A_{i+1} = A_i\{(f(z_1, \dots, z_k), x)\}$, $\theta_{i+1} = (\theta_i\{(f(t_1, \dots, t_k), x)\}) \cup \{(t_1, z_1), \dots, (t_k, z_k)\}$, gde su $z_i, i=1, \dots, k$ međusobno različite promenljive koje se ne pojavljuju u A_i , i zameniti sa $i+1$ i ponoviti II.

III. KORAK: Ako ne postoje promenljive x i y u A_i , takve da je $x\theta_i = y\theta_i$, preći na IV, a ako postoje, staviti $A_{i+1} = A_i\{(y,x)\}$, $\theta_{i+1} = \theta_i \setminus \{(x\theta_i, x)\}$, i zameniti sa $i+1$ pa ponoviti III,

IV KORAK: Ako je u nekom paru iz θ_i prva komponenta konstanta, recimo a , staviti $A_{i+1} = A_i\{(a,x)\}$, $\theta_{i+1} = \theta_i \setminus \{(a,x)\}$, i zameniti sa $i+1$, pa ponoviti IV, a ako takav par ne postoji, preći na V,

V KORAK: Staviti $n=i$, pa završiti sa radom.

Ponavljanjem drugog koraka algoritma, iz θ_i se uklanjaju svi parovi čije su prve komponente termi sa funkcijskim simbolima dužine veće od 0, na četvrti korak se prelazi kada u θ_i više nema parova sa istim prvim komponentama, da bi se, na kraju uklonili i parovi koji zamenjuju promenljive konstantama. Tako se dobijaju niz atoma i niz supstitucija koji zadovoljavaju sledeće jednakosti: $A_0\theta_0 = A_1\theta_1 = \dots = A_n\theta_n = B$, a kako se θ_n dobija nizom navedenih transformacija iz θ_0 , to je ona ili prazna, ili na obostrano jednoznačan način preslikava skup nekih promenljivih iz A_n u skup promenljivih koje se ne pojavljuju u A_n pa je $A_n = B$. Iz oblika dobijenih supstitucija je jasno da algoritam daje totalan lanac dužine $n \geq 0$ od A do B .

Navedeni algoritam sa definicijom 6 se razlikuje od algoritma J.Reynoldsa utoliko što je slučaj (4) izdvojen od (2), a u algoritmu, IV korak nije postojao, već se obavljao u okviru II koraka u kome nije zahtevano da term $f(t_1, \dots, t_k)$ ne bude konstanta. Jedan od razloga za takve izmene je bio taj da raniji algoritam ne bi dobro funkcionisao u slučaju da je, recimo $B = A\{(f(a,a), x)\}$, a osim toga se ovako bolje vidi da je ovim način konstrukcije samo jednog od totalnih lanaca izmedju A i B i da njih može imati više. Promenom redosleda izvršavanja koraka algoritma, uz izvesne korekcije, dobijaju se novi algoritmi koji daju nove totalne lance izmedju A i B . Takođe, od toga kojim redom uklanjamo parove u okviru ciklusa obavljanja jednog algoritamskog koraka zavisi koji će se totalni lanac dobiti.

DRUGI ALGORITAM ZA FORMIRANJE TOTALNOG LANCA OD ATOMA A DO $B=A\theta$

ULAZ: ATOM A I SUPSTITUCIJA θ

IZLAZ: TOTALAN LANAC OD A DO $A\theta$

I KORAK: Neka je $A_0 = A$, $\theta_0 = \theta$, $i=0$,

II KORAK: Ako ne postoje promenljive x, y u A_i takve da je $x\theta_i = y\theta_i$, preći na III, a

ako postoje, staviti $A_{i+1} = A_i\{(y,x)\}$, $\theta_{i+1} = \theta_i \setminus \{(x\theta_i, y)\}$, i zameniti sa $i+1$, pa ponoviti II,

III KORAK: Ako ne postoji u θ_i par oblika $(f(t_1, \dots, t_k), x)$, preći na IV, a ako postoji, staviti $A_{i+1} = A_i\{(f(z_1, \dots, z_k), x)\}$, $\theta_{i+1} = (\theta_i \setminus \{(f(t_1, \dots, t_k), x)\}) \cup \{(t_1, z_1), \dots, (t_k, z_k)\}$ (uz ista ograničenja za z_i), pa i zameniti sa $i+1$ i preći na II,

IV KORAK: Ako u θ_i ne postoji par čija je prva komponenta konstanta preći na V, a ako postoji (a) , staviti $A_{i+1} = A_i\{(a,x)\}$, $\theta_{i+1} = \theta_i \setminus \{(a,x)\}$, i zameniti sa $i+1$, pa ponoviti IV,

V KORAK: Ostaviti $n=i$ pa završiti sa radom. Ovaj algoritam daje totalan lanac, u opštem slučaju, različit i nezavisan od lanca dobijenog prvim algoritmom. Interesantna je i promena redosleda izvršavanja koraka II i IV prvog algoritma.

Na ovaj način se mogu formirati totalni lanci izmedju dva nesusedna atoma, takvi da zadovoljavaju određene uslove: zamena promenljivih konstantama (funkcijama dužine veće od 1, jednačene promenljivih) se obavlja na određenom mestu u algoritmu i to prema zadatom redosledu uklanjanja istorodnih parova iz supstitucija.

Primer 1.: Jedan od slučajeva kada se generalizacija može uspešno primeniti je otkrivanje zakona koji važe u nekoj algebarskoj strukturi zadatoj tablicaom.

Ako raspolažemo konačnim skupom jednakosti i nejednakosti oblika $ab=cd(ab \neq dc)$, gde su a, b, c, d elementi te konačne ili beskonačne strukture, onda jedan od načina konstrukcije generalizacija može biti sledeći:

Polazeći od samo jedne jednakosti, recimo $ae=ea$, kada bi mogla da se predstavi preko predikata $E(f(a,e), f(e,a))$, gde se E, f, a, e u ovom slučaju interpretiraju kao jednakost, operacija zadate strukture i njena dva elementa, mogu se dobiti sledeće generalizacije:

$$\begin{aligned} A_0 &= E(f(a,e), f(e,a)) \\ A_1 &= E(f(x,e), f(e,x)) \quad , \quad A_0 = A_1\{(a,x)\} \\ A_2 &= E(f(x,y), f(y,x)) \quad , \quad A_1 = A_2\{(e,y)\} \\ A_3 &= E(f(x_1,y), f(y,x)) \quad , \quad A_2 = A_3\{(x,x_1)\} \end{aligned}$$

$$\begin{aligned} A_4 &= E(f(x_1,y_1), f(y,x)) \quad , \quad A_3 = A_4\{(y,y_1)\}, \\ A_5 &= E(z_1, f(y,x)) \quad , \quad A_4 = A_5\{(f(x_1,y_1), z_1)\} \\ A_6 &= E(z_1, z_2) \quad , \quad A_5 = A_6\{(f(y,x), z_2)\} \end{aligned}$$

Ovo je jedan od totalnih lanaca od A_0 do A_6 i svi posmatrani atomi su zatvoreni univerzalnim kvantifikatorima, pa A_1 predstavlja zamenu za $(\forall x)(x=e_x)$ (tj. e je jedinica strukture), a A_2 zamenu za $(\forall x)(\forall y)(xy=yx)$ (komutativnost u strukturi).

Svaka sledeća jednakost ili nejednakost koja je zadata potvrđuje ili negira (kao kontraprimer) neku od dobijenih generalizacija. Ako je, recimo, A_4 netačno, onda se i A_5 i A_6 ne uzimaju više u obzir. Pretraživši jedan deo tablice i ustanovivši da je A_1 njome potvrđen, možemo reći da je e verovatno jedinica strukture u odnosu na navedenu operaciju. Ako ni A_1 nije tačno, mora se formirati neki drugi totalan lanac, pa ispitivati tačnost njegovih generalizacija. Ako nijedna od generacija ne važi, atom A_0 se ne može iskoristiti ni za jednu od generalizacija.

Primer 2 : $A=P(x,a,f(y),g(x,f(z))), B=P(g(f(x),y), a, f(f(a)), g(g(f(x),y), f(f(a))))$
 $\phi = \{(g(f(x),y),x), (f(a),y), (f(a),z))\}$

Prvi algoritam daje sledeći totalan lanac:

$A_0=A, \phi_0=\phi, i=0$
 $A_1=P(g(z_1,z_2), a, f(y), g(g(z_1,z_2), f(z)))$
 $\phi_1=\{(f(x),z_1), (y,z_2), (f(a),y), (f(a),z))\}$
 $A_2=P(g(z_1,z_2), a, f(f(z_3)), g(g(z_1,z_2), f(z_3)))$
 $\phi_2=\{(f(x),z_1), (y,z_2), (a,z_3), (f(a),z))\}$
 $A_3=P(g(z_1,z_2), a, f(f(z_3)), g(g(z_1,z_2), f(f(z_4))))$
 $\phi_3=\{(f(x),z_1), (y,z_2), (a,z_3), (a,z_4))\}$
 $A_4=P(g(f(z_5),z_2), a, f(f(z_3)), g(g(f(z_5),z_2), f(f(z_4))))$
 $\phi_4=\{(x,z_5), (y,z_2), (a,z_3), (a,z_4))\}$
 $A_5=P(g(f(z_5),z_2), a, f(f(z_4)), g(g(f(z_5),z_2), f(f(z_4))))$
 $\phi_5=\{(x,z_5), (y,z_2), (a,z_4))\}$
 $A_6=P(g(f(z_5),z_2), a, f(f(a)), g(g(f(z_5),z_3), f(f(a))))$
 $\phi_6=\{(x,z_5), (y,z_2)\}, n=6, A_6=B$

Drugi algoritam daje sledeći totalan lanac:

$A_0=A, \phi_0=\phi, i=0$
 $A_1=P(x,a,f(z),g(x,f(z)))$
 $\phi_1=\{(g(f(x),y),x), (f(a),z)\}$
 $A_2=P(g(z_1,z_2), a, f(z), g(g(z_1,z_2), f(z)))$
 $\phi_2=\{(f(x),z_1), (y,z_2), (f(a),z)\}$
 $A_3=P(g(z_1,z_2), a, f(f(z_3)), g(g(z_1,z_2), f(f(z_3))))$
 $\phi_3=\{(f(x),z_1), (y,z_2), (a,z_3)\}$
 $A_4=P(g(f(z_4),z_2), a, f(f(z_3)), g(g(f(z_4),z_2), f(f(z_3))))$
 $\phi_4=\{(y,z_2), (x,z_4), (a,z_3)\}$
 $A_5=P(g(f(z_4),z_2), a, f(f(a)), g(g(f(z_4),z_2), f(f(a))))$
 $\phi_5=\{(y,z_2), (x,z_4)\}, n=5, B=A_5$

DEF.7: VELIČINA terma (atoma) je broj jednak razlici broja pojavljivanja simbola azbuke u njemu

i broja različitih promenljivih koje sadrži, $Vel(\alpha) = 0, Vel(\Omega) = \infty$.

PRIMER: Veličina promenljive je 0, konstante 1, veličina svakog drugog terma je veća od 1, a veličina svakog atoma je najmanje za 1 veća od zbiru veličina terma koji se u njemu pojavljuju. Univerzalni atom ima kao svoje slučajeve sve atome bez promenljivih koji se mogu konstruisati od elemenata azbuke, pa zbog neograničene dubine terma u njima, mogu imati prizvoljnu veličinu, isto kao što promenljiva može imati bilo koji term kao svoj slučaj, pa analogija između promenljive (kao "univerzalnog" terma) i α ima odraza i u njihovoj jednakoj veličini.

Ovaj pojam je vrlo koristan kod nalaženja totalnih lanaca pomoću računara, jer se sintaksnom analizom jednostavno dolazi do veličine svakog terma ili atoma, a to se u daljem radu može iskoristiti zahvaljujući sledećim tvrdjenjima:

Ako su atomi ekvivalentni, njihove veličine su jednake, ako je $A < B$, onda je $Vel(A) < Vel(B)$.

Tako se svakoj klasi ekvivalencije u skupu atoma na jednoznačan način pridružuje ceo broj veći ili jednak 0 i poredak među uporedivim klasama se "slaže" sa poredkom (uobičajenim) između tih brojeva.

Pošto se između svaka dva nesusedna atoma može postaviti netrivialan totalni lanac, interesantan je odnos između veličina susednih atoma u takvom lancu.

Ako su A_i i A_{i+1} ($i > 0$) susedni atomi, takvi da $A_i \rightarrow A_{i+1}$ ($A_i \sigma = A_{i+1}$), gde se supstitucijom σ zamenjuje promenljiva koja se k puta pojavljuje u A_i , termom sa $n+1$ -im simbolom (lako se utvrđuje da term u σ mora imati veličinu 1 sem kada je promenljiva), tada je $Vel(A_{i+1}) = Vel(A_i) + n(k-1) + 1$.

(1) Ako je $B \neq \Omega$ ne postoji lanac od A do B duži od $Vel(B) - Vel(A)$.

(2) Ne postoji bekonačan opadajući lanac koji počinje od bilo kog atoma.

(3) Ako je A atom čiji su svi termi konstante, jedini lanac od A do Ω je $A \rightarrow \Omega$.

(4) Ako A nije Ω , niti atom sa konstantnim termima, postoji bekonačan rastući totalan lanac od A do Ω .

Tačka (1) ukazuje na postojanje gornje granice dužine totalnog lanca između dva atoma ($B \neq \Omega$),

(2) da generalizacijom uvek u konačno mnogo koraka dolazimo do najmanjeg atoma u lancu, a (4) navodi jednu od manifestacija semiodlučivosti procedura (u predikatskom računu I reda) koje obavljaju neku dedukciju.

Do sada je bilo reči samo o lancima atoma (njihovih klasa ekvivalencije) tj. o generalizacijama nad jednim atomom (svaki atom A takav da je $A < B$ je generalizacija od B, a ako je $A + B$ onda je najmanja generalizacija od B). Kao što je na početku rečeno, interesantne su generalizacije nad skupovima atoma koje su, u određenom smislu, hipoteze proistekle iz skupa slučajeva. Takve generalizacije se mogu naći uz pomoć algoritma generalizacije, a zatim se metodima dokazivanja teorema ispituje da li su neprotivurečne sa već prihvaćenim činjenicama. Napomenimo samo to da se uvođenjem operacije najmanje generalizacije (najmanje, jer je vrlo važno da se uopšte samo činjenice stvarno date) i najopštije unifikacije u parcijalno uređen skup klasa ekvivalencije atoma, dobija nemodularna, komplementirana (ali ne i relativno komplementirana) mreža atoma, koja se na računaru može jednostavno formirati. Slični postupci se koriste kada su u pitanju i nešto komplikovanije formule - disjunkcije atoma, dok se pojavljuju vrlo komplikovani problemi kada je

potrebno naći hipoteze nad nekim opštijim formulama. Često je potrebno koristiti i metode statističke analize i verovatnosne logike (2), no ova oblast se odskora intenzivnije proučava, pa rezultate tek treba očekivati.

LITERATURA:

1. Chang, Clee, R., Symbolic Logic and Mechanical Theorem Proving, Academic Press, 1973.
2. Hajek, P., Havranek, T., Mechanising Hypothesis Formation, Mathematical foundations for a general theory, Springer Verlag, 1978.
3. Jemuović, D., Mehaničko dokazivanje teorema sa primenom u mehaničkom formiranju hipoteza, Magistarski rad, Univerzitet u Beogradu 1979.
4. Plotkin, G., A note on inductive generalisation, Machine Intelligence, vol 5.
5. Plotkin, G., A further note on inductive generalisation, Machine Intelligence, vol. 6
6. Polya, . Induction and Analogy in Mathematics; Patterns of plausible inference, Princeton Univ. Press, 1954.
7. Reynolds, J., Transformational systems and the algebraic structure of atomic formulas, Machine Intelligence, vol. 5, (Meltzer, B., Michie, D., eds), American Elsevier, New York.

DOPOLNITEV
PROCESSORJA TEKSTA

A. P. ŽELEZNIKAR

UDK: 681.3-181.4

INSTITUT JOŽEF STEFAN, LJUBLJANA

ČLANEK OPISUJE IZBOLJŠAVO UREJEVALNIKA TEKSTA, KO SE UVEDE PROGRAMSKI MODUL ZA OŠTEVILČEVANJE VRSTIC Z UPORABO MONITORSKE DIREKTIVE 'X' (PREK KONZOLE).

TEXT EDITOR IMPROVEMENT, THIS ARTICLE DESCRIBES A TEXT EDITOR IMPROVEMENT INTRODUCING A PROGRAM MODULE FOR LINE NUMBERING AND THE MONITOR DIRECTIVE 'X' (VIA CONSOLE).

1. UVOD

UREJEVALNIK TEKSTA IMA OŠTEVILČENE VRSTICE, NA KATERE SE LAHKO SKLICUJEMO Z UKAZI UREJEVALNIKA. V VRSTI UREJEVALNIKOV SE OPRAVI OŠTEVILČENJE VRSTIC TEKSTA, KI SE NAMAJA V UREJEVALNIŠKEM POMNILNEM VMESNIKU, Z UPORABO DIREKTIVE 'READ' (KRATKO 'R'). VRSTICE SO OŠTEVILČENE LE, ČE SMO TEKST VČITALI IZ PERIFERNE NAPRAVE (IZ DISKA, TELEPRINTERJA, KASETE) Z UKAZOM 'READ'.

TEKST VPISUJEMO S POMOČJO UREJEVALNIKA TUDI PREK TERMINALSKE KONZOLE. S TASTATURO VSTAVLJAMO V UREJEVALNIŠKI POMNILNI VMESNIK ZNAK ZA ZNAKOM IN OBLIKUJEMO TUDI VRSTICE TEKSTA. PRI PISANJU TEKSTA SAM TEKST VEČKRAT TUDI POPRAVLJAMO, VSTAVLJAMO, IZBOLJŠUJEMO IN PRI TEM KONZOLNEM POSTOPKU ŽELIMO IMETI VRSTICE ZNOVA OŠTEVILČENE.

TA PRISPEVEK OPISUJE KRATEK PROGRAM, KI NAM OŠTEVILČI VRSTICE POMNILNEGA VMESNIKA VSAKOKRAT, KO TO ZAHTEVAMO.

2. PODATKOVNI FORMAT TEKSTA

PODATKI, KI JIH VNAŠAMO V PODATKOVNI VMESNIK (PV) UREJEVALNIKA, SE NALAGAJO V POSEBNO OBDOČJE HITREGA POMNILNIKA TIPA RAM. OSNOVNI PODATKOVNI FORMAT V PV JE VRSTICA IN JE TALE:

<VRSTICA> ::= <ŠTEVILKA VRSTICE> 'CR' <TEKST> 'CR'

TU JE 'CR' TERMINALNI ZNAK IN PREDSTAVLJA POMIK VALJA V LEVO (CARRIAGE RETURN). ZA PODATKOVNI VMESNIK VELJA V VSAKEM TREUTKU

<PV> ::= <PRAZNO> | {<VRSTICA>}

ŠTEVILKA VRSTICE JE 16-BITNA BESEDA IN JE SESTAVLJENA IZ ŠTIRIH BINARNO KODIRANIM DESETIŠKIM ŠTEVILK. TE ŠTEVILKE MORAMO VSTAVLJATI NA DOLOČENA MESTA, IN SICER NA ZAČETEK PV, NATO PA PO VSAKEM ZNAKU 'CR', KI SLEDI VSTAVLJENEMU TEKSTU.

PV JE OMEJEN Z DVEMA KAZALCEMA, IN SICER Z

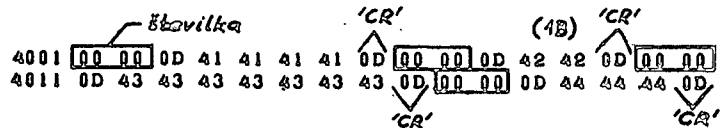
BUFBEG (ZAČETEK PV) IN
BUFEND (KONEC PV)

OGLEJMO SI PRIMER NA SLIKAH 1A, 1B IN 1C. SLIKA 1 NAKAZUJE, KAKO MORAMO ZGRADITI PROGRAM ZA VSTAVLJANJE BCD-ŠTEVILK V USTREZNA POLJA.

3. MODUL ZA OŠTEVILČENJE VRSTIC

PROGRAM ZA UREJEVALNIK TEKSTA SE NAMAJA V POMNILNIKU TIPA ROM ALI NA DISKU. OBSTOJEČEGA UREJEVALNIŠKEGA PROGRAMA NE ŽELIMO SPREMINJATI (ČE JE V POMNILNIKU TIPA ROM, GA TUDI NE MOREMO). ZA OŠTEVILČENJE UPORABIMO T.I. PONOVNI VSTOP V UREJEVALNIK (REENTRY).

<AAAA
<BB
<CCCCCC (1A)
<DDD



0000 AAAA
0000 BB
0000 CCCCCC
0000 DDD
načrtovilec

(1C)

0001 AAAA
0002 BB
0003 CCCCCC
0004 DDD
oštevilčevalc

SLIKA 1. PRIMER VSTAVITVE TEKSTA (1A), PREDSTAVITVE V POMNILNIKU (1B) IN PO ČITANJU Z UREJEVALNIKOM (1C).

Y42INF TEXT LINE NUMBERING
 ADDR OBJECT ST #

```

0002 NAME Y42INF
0003 ; MODULE FOR TEXT LINE NUMBERING
0004 ; *****
0005 ; PROGRAMMED BY ANTON P. ZELEZNIKAR
0006 ; FOR MOSTEK SDB-80 TEXT EDITOR
0007 ; DATE: NOVEMBER 16, 1979
0008 ; *****
0009 ; THIS MODULE CAN BE USED FOR NUMBERING
0010 ; OF LINES IN TEXT EDITOR IN CONNec-
0011 ; TION WITH THE TEXT PROCESSING AND
0012 ; ALSO IN PURE EDITING PROCEDURES.
0013 ; THE MODULE WILL BE ACTIVATED BY
0014 ; JP NUMB COMMAND.
0015 ; *****
0016 GLOBAL NUMB
0017 ; *****
0018 LD HL,(BUFBEQ) ;BEGIN OF BUFFER
0019 LD BC,1 ;LINE COUNTER BC
0020 LD DE,(BUFEND) ;END OF BUFFER
0021 DEC DE
0022 THREE: LD A,H ;END OF BUFFER
0023 CP D ; TEST
0024 JR NZ,ONE-$
0025 LD A,L
0026 CP E
0027 JR NZ,ONE-$
0028 JP 0D4D9H ;EDITOR REENTRY
0029 INC HL ;NUMBER INSERTION
0030 LD (HL),B
0031 INC HL
0032 LD (HL),C
0033 INC HL
0034 LD A,C
0035 ADD A,1
0036 DAA
0037 LD C,A
0038 JR NC,TWO-$
0039 LD A,B
0040 ADD A,1
0041 DAA
0042 LD B,A
0043 JR NC,TWO-$
0044 JP 0D4D9H ;EDITOR REENTRY
0045 TWO: LD A,0DH
0046 FOUR: INC HL
0047 CP (HL)
0048 JR NZ,FOUR-$
0049 JR THREE-$
0050 ; *****
0051 BUFBEQ EQU 0FF02H
0052 BUFEND EQU 0FFA6H
0053 ; *****
0054 END
  
```

Y42INF TEXT LINE NUMBERING
 ADDR OBJECT ST #

```

BUFBEQ FF02 BUFEND 002E NUMB [INT] 0000
ONE 0016 THREE 002C
ERRORS=0000
FFA6 FOUR 000B TWO
  
```

SLIKA 2. LISTA MODULA ZA OŠTEVILČEVANJE.

```

XNUMB X-DIRECTIVE ST #
ADDR OBJECT
0002 NAME XNUMB
0003 GLOBAL NUMB
0004 ; *****
0005 INIT: LD HL,XC ;JUMP 'INIT' AFTER
0006 ; EACH SYSTEM RESET
0007 LD (EXPAND),HL ;EXPAND WITH 'X'
0008 JP MONREN ;GO TO MONITOR
0009 XC: CP 'X' ;DIRECTIVE 'X'
0010 JP NZ,MONREN
0011 JP NUMB ;GO NUMBERING WITH
0012 ; JUMP TO EDITOR
0013 ; REENTRY POINT
0014 ; *****
0015 EXPAND EQU 0FF1FH
0016 MONREN EQU 0E11DH
0017 ; *****
0018 END
  
```

SLIKA 3. PROGRAM ZA DODATNO DIREKTIVO 'X' MONITORJA.

V UREJEVALNIK LAHKO VSTOPIMO ZAČETNO (INICIALNO) ALI PONOVNO. PRI ZAČETNEM VSTOPU SE INICIALIZIRAJO KAZALCI UREJEVALNIKA, IZMERI SE OBSEG RAZPOLOŽLJIVEGA POMNILNIKA ITN. PRI PONOVNEM VSTOPU V UREJEVALNIK PA SO VREDNOSTI KAZALCEV ENAKE VREDNOSTIM PRED IZSTOPOM.

OŠTEVILČENJE VRSTIC SE BO LAHKO OPRAVILO SAMODEJNO PRI VSAKEM PONOVNEM VSTOPU V UREJEVALNIK, KO VSTOPAMO PREK PROGRAMSKEGA MODULA ZA OŠTEVILČENJE. TOČKO (NASLOV) ZA PONOVNI VSTOP V UREJEVALNIK BOMO IMENOVALI 'REENTRY'.

ZA MODULSKI PROGRAM IMAMO TALE PSEVDOKOD:

```

MODUL NUMB
PREČITAJ MEJNA KAZALCA;
NASTAVI VRSTIČNI ŠTEVNIK;
DOWHILE (NISMO NA KONCU PV)
  POIŠČI MESTO ZA VSTAVITEV;
  VSTAVI ŠTEVILKO VRSTICE;
ENDDO
SKOČI V TOČKO PONOVNEGA
VSTOPA UREJEVALNIKA;
ENDMODUL

```

NA SLIKI 2 JE PRIKAZANA LISTA MODULA ZA PROCESOR Z-80 IN ZA PRIMER, KO UPORABLJAMO TOVARNIŠKI UREJEVALNIK SISTEMA SDB-80 (MOSTEK).

4. RAZŠIRITEV MONITORJA

MONITOR (OPERACIJSKI SISTEM) ZA SDB-80 JE ODPRT IN GA LAHKO ŠIRIMO UPORABNIŠKO. MONITOR IMA SPREMENLJIVKO 'EXPAND', KI SE OB INICIALIZACIJI MONITORJA POSTAVI NA VREDNOST 'MONREN' (MONITOR REENTRY); TA VREDNOST JE V PRIMERU TOVARNIŠKEGA MONITORJA OE11DH. TAKO IMAMO PODATKOVNO OZ. NASLOVNO:

(EXPAND) = MONREN OZ.
(OFF1H) = OE11DH

V MONITOR UVEDEMO NOVO DIREKTIVO 'X' ZA PONOVEN VSTOP V UREJEVALNIK. NAJPREJ MORAMO RAZPOZNATI KARAKTER 'X', NATO PA SKOČIMO V MODUL NUMB. TU JE

(OFF20H).(OFF1FH)

NASLOV NADALJEVANJA MONITORJA, PROGRAM ZA TO NADALJEVANJE PA IMAMO NA SLIKI 3.

5. SKLEP

PRIMER IZBOLJŠAVE UREJEVALNIKA KAŽE, KAKO JE MOČ ENOSTAVNO DOPOLNJEVATI DANI MIKRORAČUNALNISKI OPERACIJSKI SISTEM Z NOVIMI FUNKCIJAMI. OPISANI DODATEK JE BISTVEN IN IZBOLJŠUJE FUNKCIONALNOST UREJEVALNIKA ZLASTI PRI VPISOVANJU IN POPRAVLJANJU OBSEŽNIH TEKSTOV PREK KONZOLE.

NOVICE IN ZANIMIVOSTI

NOVA ARITMETIČNA PROCESORJA

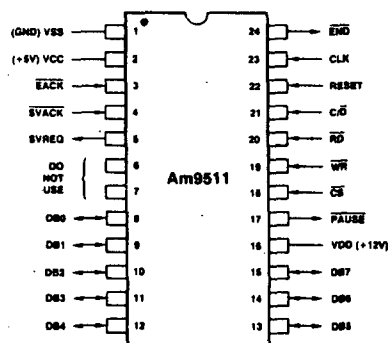
PODJETJE ADVANCED MICRO DEVICES, INC., 901 THOMPSON PLACE, SUNNYVALE, CALIFORNIA 94086 JE DALO NA TRŽIŠČE DVA IZREDNO ZANIMIVA IN UPORABNA ARITMETIČNA PROCESORJA V OBLIKI 24-NOŽIČNIH INTEGRIRANIH VEZIJ. OZNAKI TEH SAMOSTOJNIH DVEH PROCESORJEV STA AM 9511 (ARITHMETIC PROCESSOR) IN AM 9512 (FLOATING POINT PROCESSOR). OGLEDJMO SI PODROBNEJE ZGRADBO IN DELOVANJE OBEH NOVIH PROCESORJEV.

VEZJE AM 9511 (KRATKO 'APU') ZMORE ARITMETIKO S FIKSNO IN PLAVAJOČO VEJICO TER VRSTO DRUGIH TRIGONOMETRIČNIH IN MATEMATIČNIH OPERACIJ. TO VEZJE ZNATNO POVEČUJE ZMOGLJIVOSTI OBSTOJEČIH MIKROPROCESORJEV, S KATERIMI GA JE MOGOČE ENOSTAVNO POVEZATI V SISTEM. PRENOS PODATKOV, KOT SO OPERANDI, REZULTATI, STATUSI IN UKAZI, TEČE PREK 8-BITNEGA DVOSMERNEGA PODATKOVNEGA VODILA. OPERANDI SE NALAGAJO V NOTRANJJI SKLAD PROCESORJA, Z UKAZI PA SE IZVAJAJO OPERACIJE NAD PODATKI V TEM SKLADU. REZULTATE JE MOGOČE POTEM DOBITI IZ SKLADA, LAHKO PA SE VSTAVIJO DODATNI UKAZI.

PRENOS PODATKOV K 'APU' IN IZ 'APU' OPRAVLJA PRIKLJUČENI MIKROPROCESOR, NPR. V OBLIKI PROGRAMIRANEGA V/I ALI S POMOČJO KRMILNIKA ZA NEPOSREDNI POMNILNISKI DOSTOP (DMA) V SISTEMIH Z VIŠJO STOPNJO ORGANIZACIJE. PO IZVEDBI (IZVRŠITVI) VSAKEGA UKAZA V 'APU' SE GENERIRA SIGNAL IZVRŠITVE UKAZA, KI GA LAHKO UPORABIMO KOT PREKINITEV ZA CPU.

KARAKTERISTIKE TEGA ZANIMIVEGA PROCESORJA PA SO:

- OPERACIJE S POMIČNO VEJICO NAD 16 IN 32 BITI;
- OPERACIJE S POMIČNO VEJICO NAD 32 BITI;
- FORMAT (OBLIKA) BINARNIH PODATKOV;
- SEŠTEVANJE, ODŠTEVANJE, MNOŽENJE IN DELENJE;
- TRIGONOMETRIČNE IN INVERZNE TRIGONOMETRIČNE FUNKCIJE;
- KORENENJE, LOGARITMIČNANJE, EKSPONENCIRANJE;
- KONVERZIJA IZ FIKSNE V PLAVAJOČO OBLIKO IN OBRATNO;
- OPERANDNI POMNILNIK S FUNKCIJO SKLADA;
- PRENOS PODATKOV Z DMA ALI PROGRAMIRANIM V/I;
- SIGNAL KONČANJA ZA POENOSTAVITEV PARALELNEGA PROCESIRANJA;
- SPLOŠNI VMESNIK ZA 8-BITNO PODATKOVNO VODILO;
- STANDARDNO 24-NOŽIČNO PODNOŽJE;
- NAPAVALNI NAPETOSTI 5 IN 12 V.



SLIKA 1. IMENA NOŽIČ NA PODNOŽJU ARITMETIČNEGA PROCESORJA 9511

SLIKA 1 KAŽE PRIREDITEV SIGNALOV POSAMEZNIH NOŽIČAM, NA SLIKI 2 PA IMAMO NOTRANJO ZGRADBO 'APU'. VEZJE DELUJE KOT SAMOSTOJNI 16-BITNI PROCESOR. KRATEK OPIS SIGNALOV IZ SLIKE 1 PA JE TALE:

CLK (CLOCK, INPUT)

-- ZUNANJI TAKTNI SIGNAL MORAMO PRIPELJATI NA NOŽICO CLK. TAKTNI SIGNAL JE LAHKO ASINHRON GLEDE NA KRMILNA SIGNALA RD- (READ) IN WR- (WRITE).

RESET (RESET, INPUT)

-- VISOK RESETIRNI SIGNAL POVZROČI INICIALIZACIJO VEZJA. TA SIGNAL KONČA VSAKO OPERACIJO, ZBRIŠE STATUSNI REGISTER IN POSTAVI VEZJE V BREZDELNO STANJE. RESETIRANJE NE VPLIVA NA SKLAD. SIGNAL RESETIRANJA MORA BITI PRISOTEN VSAJ 5 TAKTNIH PERIOD. VEZJE NIMA MOŽNOSTI AVTOMATIČNEGA RESETIRANJA OB VKLOPU NAPETOSTI.

CS- (CHIP SELECT, INPUT)

-- NIZEK SIGNAL CS- OMOGOČI S SIGNALOMA READ IN WRITE KOMUNIKACIJO S PODATKOVNIM VODILOM.

C/D- (COMMAND/DATA, INPUT)

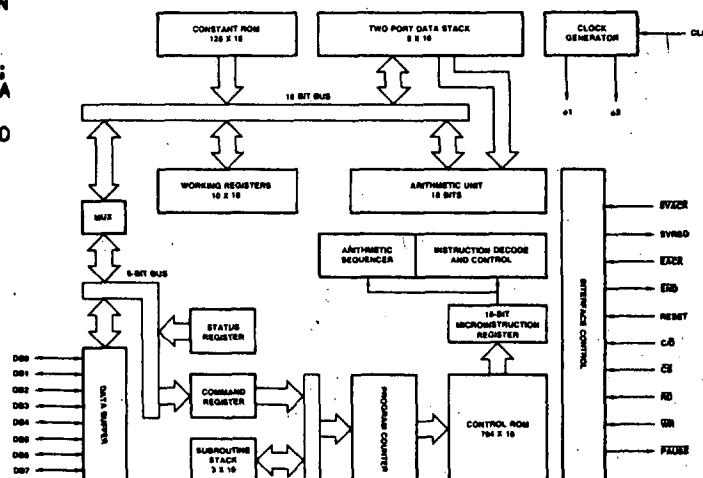
-- V POVEZAVI S SIGNALOMA RD- IN WR- DOLOČA C/D- TIP PRENOSA NA PODATKOVNEM VODILU. IMAMO TOLE TABELO:

C/D-	RD-	WR-	FUNKCIJA
0	1	1	VPIŠI PODAT. ZLOGA V SKLAD
0	0	1	ČITAJ PODAT. ZLOG IZ SKLADA
1	1	0	VPIŠI UKAZ
1	0	1	ČITAJ STATUS

RD- (READ, INPUT)

-- NIZEK READ SIGNAL, KI JE POGOJEN S CS-, DOLOČA, DA BO INFORMACIJA PRENESENA Z NOTRANJE LOKACIJE NA PODATKOVNO VODILO. SIGNALA RD- IN WR- STA PAROMA IZKLJUČUJOČA.

WR- (WRITE, INPUT)



SLIKA 2. ZGRADBA PROCESORJA 9511. VIDEN JE 16-BITNI 8-BESEJNI SKLAD ZA SHRANJEVANJE OPERANDOV, POMNILNIKA TIPA ROM ZA KONSTANTE IN ZA KRMILNI PROGRAM PROCESORJA, ARITMETIČNA ENOTA ITN. IZ ZGRADBE PROCESORJA NA TEJ SLIKI LAHKO SKLEPAMO TUDI NA NJEGOVO DELOVANJE.

COMMAND SUMMARY									
Command Code				Command Mnemonic				Command Description	
FIXED POINT 16 BIT									
SF	0	1	1	0	1	SADD	0	Add TOS to NOS. Result to NOS. Pop Stack.	
SF	0	1	1	0	1	SSUB	0	Subtract TOS from NOS. Result to NOS. Pop Stack.	
SF	0	1	1	1	0	SMUL	0	Multiply NOS by TOS. Lower half of result to NOS. Pop Stack.	
SF	0	1	1	1	1	SMUJ	0	Multiply NOS by TOS. Upper half of result to NOS. Pop Stack.	
SF	0	1	1	0	1	SDIV	1	Divide NOS by TOS. Result to NOS. Pop Stack.	
FIXED POINT 32 BIT									
SF	0	1	0	1	0	DADD	0	Add TOS to NOS. Result to NOS. Pop Stack.	
SF	0	1	0	1	1	DSUB	0	Subtract TOS from NOS. Result to NOS. Pop Stack.	
SF	0	1	1	0	0	DMUL	0	Multiply NOS by TOS. Lower half of result to NOS. Pop Stack.	
SF	0	1	1	0	1	DMUJ	0	Multiply NOS by TOS. Upper half of result to NOS. Pop Stack.	
SF	0	1	1	1	0	DDIV	1	Divide NOS by TOS. Result to NOS. Pop Stack.	
FLOATING POINT 32 BIT									
SF	0	0	0	0	0	FADD	0	Add TOS to NOS. Result to NOS. Pop Stack.	
SF	0	0	0	1	0	FSUB	0	Subtract TOS from NOS. Result to NOS. Pop Stack.	
SF	0	0	1	0	0	FMUL	0	Multiply NOS by TOS. Result to NOS. Pop Stack.	
SF	0	0	1	1	0	FDIV	1	Divide NOS by TOS. Result to NOS. Pop Stack.	
DERIVED FLOATING POINT FUNCTIONS									
SF	0	0	0	0	0	SQRT	1	Square Root of TOS. Result in TOS.	
SF	0	0	0	0	1	SIN	0	Sine of TOS. Result in TOS.	
SF	0	0	0	1	1	COS	0	Cosine of TOS. Result in TOS.	
SF	0	0	0	1	0	TAN	0	Tangent of TOS. Result in TOS.	
SF	0	0	0	1	1	ASIN	0	Inverse Sine of TOS. Result in TOS.	
SF	0	0	0	1	0	ACOS	0	Inverse Cosine of TOS. Result in TOS.	
SF	0	0	0	1	1	ATAN	0	Inverse Tangent of TOS. Result in TOS.	
SF	0	0	0	0	0	LOG	0	Common Logarithm (base 10) of TOS. Result in TOS.	
SF	0	0	0	0	1	LN	0	Natural Logarithm (base e) of TOS. Result in TOS.	
SF	0	0	0	0	1	EXP	0	Exponential (e ^{TOS}) of TOS. Result in TOS.	
SF	0	0	0	0	1	PWR	1	NOS raised to the power in TOS. Result in NOS. Pop Stack.	
DATA MANIPULATION COMMANDS									
SF	0	0	0	0	0	NOP	0	No Operation	
SF	0	0	1	0	0	FIXS	0	Convert TOS from floating point to 16-bit fixed point format.	
SF	0	0	1	1	0	FIXD	0	Convert TOS from floating point to 32-bit fixed point format.	
SF	0	0	1	0	1	FLTD	0	Convert TOS from 16-bit fixed point to floating point format.	
SF	0	0	1	1	1	FLTJ	0	Convert TOS from 32-bit fixed point to floating point format.	
SF	0	0	1	0	0	CHSS	0	Change sign of 16-bit fixed point operand on TOS.	
SF	0	0	1	1	0	CHSD	0	Change sign of 32-bit fixed point operand on TOS.	
SF	0	0	1	0	1	CHSF	1	Change sign of floating point operand on TOS.	
SF	0	0	1	1	1	PTOD	0	Push 16-bit fixed point operand on TOS to NOS (Copy)	
SF	0	0	1	0	0	PTOF	0	Push 32-bit fixed point operand on TOS to NOS (Copy)	
SF	0	0	1	1	1	POPS	0	Pop 16-bit fixed point operand from TOS. NOS becomes TOS.	
SF	0	0	1	0	0	POPJ	0	Pop 32-bit fixed point operand from TOS. NOS becomes TOS.	
SF	0	0	1	1	0	POPF	0	Pop floating point operand from TOS. NOS becomes TOS.	
SF	0	0	1	1	1	XCHS	0	Exchange 16-bit fixed point operands TOS and NOS.	
SF	0	0	1	0	0	XCHD	0	Exchange 32-bit fixed point operands TOS and NOS.	
SF	0	0	1	1	1	XCHF	0	Exchange floating point operands TOS and NOS.	
SF	0	0	1	0	1	PUPI	0	Push floating point constant "r" onto TOS. Previous TOS becomes NOS.	

- NOTES:
1. TOS means Top of Stack. NOS means Next on Stack.
 2. AMD Application Brief "Algorithm Details for the Am9511 APU" provides detailed descriptions of each command function, including data ranges, accuracies, stack configurations, etc.
 3. Many commands destroy one stack location (bottom of stack) during development of the result. The derived functions may destroy several stack locations. See Application Brief for details.
 4. The trigonometric functions handle angles in radians, not degrees.
 5. No remainder is available for the fixed-point divide functions.
 6. Results will be undefined for any combination of command coding bits not specified in this table.

TABELA 1. SPISEK KODOV IN UKAZOV S POJASNILI ZA PROCESOR 9511

-- NIZEK WRITE SIGNAL, KI JE POGOJEN S CS-, DOLOČA, DA BO INFORMACIJA IZ PODATKOVNEGA VODILA PRENEŠENA NA NOTRANJO LOKACIJO.

EACK- (END ACKNOWLEDGE, INPUT)

-- NIZEK EACK SIGNAL ZBRIŠE SIGNAL END- (GLEJ KASNEJE). DOKLER JE EACK NIZEK, JE SIGNAL END- IMPULZ, KI JE KRAJŠI OD PERIODE TAKTA.

SVACK- (SERVICE ACKNOWLEDGE, INPUT)

-- NIZEK SIGNAL SVACK ZBRIŠE SIGNAL SVREQ (GLEJ KASNEJE).

END- (END EXECUTION, OUTPUT)

-- NIZEK SIGNAL (IZHOD Z ODPRTIM KOLEKTORJEM) OZNAČUJE, DA JE PREJŠNJI UKAZ IZVRŠEN. TA SIGNAL SE LAHKO UPORABI ZA PREKINITEV, UKINE PA SE GA Z EACK-, RESET ALI ČITALNIM ALI PISALNIM DOSTOPOM.

SVREQ (SERVICE REQUEST, OUTPUT)

-- VISOK SIGNAL SVREQ OZNAČUJE, DA JE UKAZ IZVRŠEN IN DA JE BILA ZAHTEVANA POIZVRŠITVENA OBDELAVA V PREJŠNJEM UKAZNEM ZLOGU. TA SIGNAL SE UKINJA S SVACK-, RESET ALI S KONCEM NASLEDNJEGA UKAZA, KI NI ZAHTEVAL OBDELAVE.

PAUSE- (PAUSE, OUTPUT)

-- NIZEK SIGNAL PAUSE OZNAČUJE, DA VEZJE 9511 ŠE NI KONČALO PRENOSA PODATKA Z GOSTITELJEM (ALI Z DMA) PREK PODATKOVNEGA VODILA. KADAR SE ZAHTEVA ČITALNA OPERACIJA, POSTANE SIGNAL PAUSE

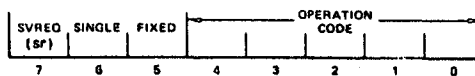
NIZEK. TA SIGNAL POSTANE VISOK, KO JE NA PODATKOVNEM VODILU VELJAVEN IZHODNI PODATEK. KO SE TRENUTNI UKAZ ŠE IZVAJA IN KO SE ZAHTEVA PODATKOVNO PISANJE ALI ČITANJE ALI UKAZNO PISANJE, POSTANE SIGNAL PAUSE NIZEK MED IZVAJANJEM UKAZA IN NIZEK JE ŠE V ČASU, KI JE POTREBN ZA INICIALIZACIJO PODATKOVNEGA ČITANJA. V PRIMERU NIZEKEGA PAUSE NAJ BI GOSTITELJ NE SPREMINJAL PODATKA K VEZJU 9511 ALI ČITAL PODATEK IZ NJEGA.

DB0-DB7 (DVOSEMerno PODATKOVNO VODILO, I/O)

-- PO TEM VODILU SE PRENAŠAJO UKAZI, STATUS IN PODATKI MED 9511 IN GOSTITELJEM.

STRUKTURA UKAZOV:

VSak UKAZ, KI VSTOPI V VEZJE 9511, JE 8-BITNI ZLOG Z NASLEDNJIM FORMATOM:



BITI 0-4 DOLOČAJO OPERACIJO, KOT KAŽE TABELA 1. BIT 5 DOLOČA OPERACIJO S FIKSNO (-1) IN POMIČNO (=0) VEJICO. BIT 6 DOLOČA NATANČNOST PODATKA PRI OPERACIJAH S FIKSNO VEJICO, TODA

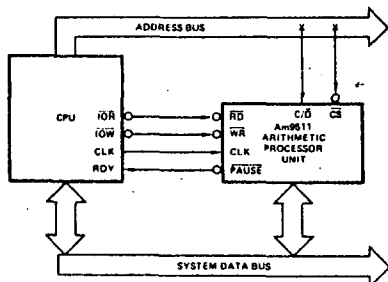
COMMAND EXECUTION TIMES

Command Mnemonic	Clock Cycles	Command Mnemonic	Clock Cycles
ACOS	6304-8284	LOG	4474-7132
ASIN	6230-7938	LN	4298-6956
ATAN	4992-6536	NOP	4
CHSD	26-28	POPD	12
CHSF	18-20	POPF	12
CHSS	22-24	POPS	10
COS	3840-4878	PTOD	20
DADD	20-22	PTOF	20
DDIV	196-210	PTOS	16
DMUL	194-210	PUPI	16
DMUU	182-218	PWR	8290-12032
DSUB	38-40	SADD	16-18
EXP	3794-4878	SDIV	84-94
FADD	54-368	SIN	3796-4808
FDIV	154-184	SMUL	84-94
FIXD	90-336	SMUU	80-98
FIXS	90-214	SQRT	782-870
FLTD	56-342	SSUB	30-32
FLTS	62-156	TAN	4894-5886
FMUL	146-168	XCHD	26
FSUB	70-370	XCHF	26
		XCHS	18

TABELA 2. IZVAJANJE POSAMEZNIH UKAZOV V ODVISNOSTI OD ŠTEVILA TAKTNIH CIKLOV

Command Description	Am9511 (2MHz)	Am9511-1 (3MHz)
32-Bit Floating-Point Cosine (COS)	1920 μ sec	1280 μ sec
32-Bit Floating-Point e ^x (EXP)	1897 μ sec	1265 μ sec
32-Bit Floating Point Multiply (FMUL)	73 μ sec	49 μ sec
16-Bit Fixed-Point Multiply, Lower (SMUL)	42 μ sec	28 μ sec
32-Bit Floating-Point Add (FADD)	27 μ sec	18 μ sec
16-Bit Fixed-Point Add (SADD)	8 μ sec	5 μ sec

TABELA 3. ČASI IZVAJANJA POSAMEZNIH UKAZOV PROCESORJA 9511



SLIKA 3. ARITMETIČNI KRMILNIK LAHKO DIREKTNO PRIKLJUČIMO NA VODILO PROCESORJA 8080 ALI Z-80

MORA VELJATI BIT 5 = 0 IN BIT 6 = 0 (PRI POMIČNI VEJICI). ENOJNO NATANČNOST (16 BITOV) IMAMO PRI BIT 6 = 1, DVOJNO (32 BITOV) PA PRI BIT 6 = 0. BIT 7 DOLOČA SERVISNO ZAHTEVO PO IZVRŠITVI UKAZA. ČE JE BIT 7 = 1, SE BO NA SPONKI SVREQ POJAVIL VISOK SIGNAL V ČASU IZVAJANJA UKAZA IN BO OSTAL VISOK, DOKLER NE BO

RESETIRAN S SIGNALOM SVACK- ALI DO KONCA IZVRŠITVE NASLEDNJEGA UKAZA, PRI KATEREM JE BIT 7 = 0. TABELA 1 KAŽE SPISEK UKAZOV S POJASNILI.

TABELA 2 PRIKAŽUJE IZVAJANJE UKAZOV S ŠTEVILI TAKTNIH CIKLOV, TABELA 3 PA ČASOVNO PRI TAKTNIH 2 IN 3 MHZ.

NA SLIKI 3 JE PRIKAZANA PRIKLJUČITEV ARITMETIČNEGA KRMILNIKA 9511 NA MIKRORAČUNALNIŠKI SISTEM.

PROCESOR 9512 JE TIPIČEN ARITMETIČEN PROCESOR ZA PLAVAJOČO VEJICO IN NE VSEBUJE TRIGONOMETRIČNIH IN OSTALIH DODATNIH FUNKCIJ PROCESORJA 9511. NJEGOVA PREDNOST JE V VEČJI NATANČNOSTI, SAJ OPRAVLJA TUDI OPERACIJE V PLAVAJOČI VEJICI S 64 BITI. RAZEN SEŠTEVANJA, ODŠTEVANJA, MNOŽENJA IN DELJENJA PREMORA SE POMOŽNE OPERACIJE, KOT SO SPREMEMBA PREZNAKA, NALAGANJE IN IZLAGANJE V SKLAD, ZAMENJAVA VSEBIN MED POSAMEZNIH REGISTRI, ČIŠČENJE STATUSA ITN. TA PROCESOR JE TAKO UPORABEN PREDVSEM TAKRAT, KO ŽELIMO IMETI VELIKO NATANČNOST PRI ARITMETIČNIH OPERACIJAH IN SI PREOSTALE DODATNE FUNKCIJE ZGRADIMO S PROGRAMI Z UPORABO TEH OPERACIJ.

OBA NOVA PROCESORJA PREDSTAVLJATA POHITRITEV DOLOČENIH OPERACIJ IN STANDARDNE SUBRUTINE V VISOKOH PROGRAMIRNIH JEZIKIH LAHKO Z UPORABO TEH PROCESORJEV NADOMESTIMO S KRAJŠIMI RUTINAMI, KI UPORABLJAJO FUNKCIJE ARITMETIČNIH PROCESORJEV. V NAČRTIH NOVIH MIKRORAČUNALNIŠKIH SISTEMOV SE BRZKONE NE BO VEČ MOGOČE OGNITI UPORABI TEH ARITMETIČNIH PROCESORJEV.

A.P.ŽELEZNIKAR

CENTER ZA VLSI

STANFORDSKA UNIVERZA USTANAVLJA CENTER, KI NAJ BI ZAGOTOVIL ZDRUŽENIM DRŽAVAM AMERIKE VODILNO VLOGO V NASTOJAJOČI ELEKTRONSKI REVOLUCIJI, VODILA PA NAJ BI GA SKUPINA PROFESORJEV TEHNIKE TE UNIVERZE. CENTER ZA INTEGRIRANE SISTEME V PALO ALTO BO RAZISKOVAL TEHNOLOGIJO ZELO VELIKIH INTEGRIRANIH VEZIJ IN NJIHOVO UPORABO TER BO NAJVEČJI CENTER TE VRSTE V ZDA. MED DRUGIM BO LETNO PRODUCIRAL 100 MAGISTROV IN 30 DOKTORJEV ZNANOSTI, PODPIRAL RAZVOJ AVTOMATIZACIJE PROIZVODNJE VEZIJ TER PRIREJAL DOPOLNILNE TEČAJE IN KONFERENCE.

A.P.ŽELEZNIKAR

OPERACIJSKI SISTEM UNIX

OPERACIJSKI SISTEM UNIX, KI GA JE IZDELALO PODJETJE BELL LABORATORIES, VSE BOLJ PRODIRA TUDI V MIKRORAČUNALNIŠKE SISTEME. TA OPERACIJSKI SISTEM OMOGOČA NAMREČ UPORABO VRSTE PREVAJALNIKOV ZA VISOKE PROGRAMIRNE JEZIKE NA MIKRORAČUNALNIŠKIH. ZILOG DOBAVLJA SVOJ SISTEM S PROCESORJEM Z8000, KI UPORABLJA POSEBNO RAZLIČICO SISTEMA UNIX, IMENOVANO ONIX. NA RAZPOLAGO JE TUDI PREČNA PROGRAMSKA OPREMA TEGA PODJETJA, KI OMOGOČA RAZVOJ KODA ZA Z8000 NA PDP-11. DVE NADALJNI RAZLIČICI SISTEMA UNIX STA OMNIX (YOURDON SOFTWARE PRODUCTS GROUP, N.Y.) ZA Z80 IN IDRIS (WHITESMITHS LTD.) ZA 8080.

A.P.ŽELEZNIKAR

NAKUP VISOKE TEHNOLOGIJE V ZDA

VLADNA RAZISKOVALNA ORGANIZACIJA VELIKE BRITANIJE JE NAROČILA PRI STANFORDSKEM RAZISKOVALNEM INSTITUTU V PALO ALTO IN PRI KONZULTANTSKI ORGANIZACIJI ARTHUR D. LITTLE INC. ISKANJE PODJETIJ Z VISOKO TEHNOLOGIJO, KI BI PRODALE SVOJE LICENCE VSAJ VELIKI BRITANIJI, PO MOŽNOSTI PA TUDI ZAPADNI EVROPI. TE LICENCE NAJ BI SE NANAŠALE NA RAČUNALNIŠKO PERIFERIJU, ELEKTRONSKO INSTRUMENTACIJO, ROBOTIKO, IZDELOVALNO TEHNOLOGIJO, NAPRAVE ZA AVTOMATIČNO TESTIRANJE IN NA DRUGA PODROČJA ELEKTRONIKE. ZINTERESIRANA PODJETJA SE BODO MORALA OBRNITI NA BRITANSKO VLADNO RAZISKOVALNO ORGANIZACIJO, KI BO SKLEPALA POGODBE.

A.P. ŽELEZNIKAR

FRANCIJA UVAJA TERMINALE ZA DOM

FRANCOSKA POŠTNA IN TELEKOMUNIKACIJSKA AGENCIJA JE SPROŽILA PROJEKT, S KATERIM NAJ BI SE V VSAK FRANCOSKI DOM UVEDEL RAČUNALNIŠKI TERMINAL. TA VLADNA AGENCIJA SI PRIZADEVA, DA BI DOBIL VSAK TELEFONSKI NAROČNIK BREZPLAČNO SVOJ ZASLONSKI TERMINAL NAMESTO DOSEDANJH TISKANIH IMENIKOV. NAPRAVA, KI LAHKO ODDA ALI SPREJME ENO STRAN TEKSTA V DVEH MINUTAH, NAJ BI STALA MANJ KOT 500 \$. V NASLEDNJEM LETU NAJ BI POSKUSNO INSTALIRALI 1000 TAKIH TERMINALOV. RAČUNAJO, DA BO AGENCIJO STAL TAK TERMINAL MANJ KOT 100 \$.

A.P. ŽELEZNIKAR

VISOKOZANESLJIVI PROCESOR Z-80

ČEPRAV SE POJAVLJAJO VISOKOŠPOSOBNI 16-BITNI MIKROPROCESORJI, PREDVIDEVA MOSTEK ŠE DOLGOLETNO TRŽIŠČE ZA TAKŠNE 8-BITNE MIKROPROCESORJE, KOT JE Z-80. LETOS BO PRIŠEL NA TRŽIŠČE VISOKOZANESLJIVI PROCESOR Z-80, KI BO IMEL TAKTNO FREKVENCO DO 6 MHZ. ŠE NADALJE SE BOSTA PROIZVAJALI 5-ELEMENTNI DRUŽINI Z-80 ZA 4 IN 2,5 MHZ.

A.P. ŽELEZNIKAR

NAČRTOVANJE MIKRORAČUNALNIKOV

MIKRORAČUNALNIKE LAHKO NAČRTOJEMO IN RAZVIJAMO TUDI SAMI, ZLASTI DOMA. ČE O TEM NIMAMO POSEBNEGA ZNANJA, SE LAHKO MARSIKAJ NAUČIMO TUDI SAMI. AMERIŠKO PODJETJE ADVANCED MICRO DEVICES INC. NUDI TEČAJ V SEDMIH KNJIZICAH Z NASLOVOM 'BUILD A MICROCOMPUTER'. TE KNJIZICE OBRAVNAVAJO RAČUNALNIŠKO ZGRADBO, MIKROPROGRAMIRNO NAČRTOVANJE, POTEK PODATKOV (V DVEH KNJIZICAH), PROGRAMIRANE KRMILNE ENOTE, PREKINITVE IN NEPOSREDEN POMNILNIŠKI DOSTOP. TA TEČAJ SEZNANJA BRALCA S KOMPONENTAMI AMD. NASLOV: BOB GROSSMAN, AMD, 901 THOMPSON PLACE, SUNNYVALE, CA 94086.

A.P. ŽELEZNIKAR

OPERACIJSKI SISTEM CP/M

CP/M JE IME SPLOŠNEGA KRMILNEGA (OPERACIJSKEGA) PROGRAMA ZA PROCESORJE 8080,

8085 IN Z-80, KI JE POSTAL STANDARD ZA TE PROCESORJE. CP/M SE UPORABLJA Z IZBOLJŠAVAMI ŽE VEČ KOT 5 LET IN JE DOŽIVEL TEMELJITO PREIZKUŠNJO V PRAKSI V VEČDESETTISOČ INSTALACIJAH. PRAVKAR JE IZŠLA NJEGOVA PETA REVIZIJA CP/M 2.0, KI JE INTERAKTIVNI PROGRAMSKI RAZVOJNI SISTEM IN PREDSTAVLJA OSNOVO ZA VISOKE PROGRAMIRNE JEZIKE, PROCESIRANJE TEKSTA IN POSLOVNE UPORABE. OSNOVNE LASTNOSTI PROGRAMA CP/M OBSEGAJO DINAMIČNO UPRAVLJANJE ZBIRK, HITRI ZBIRNIK, SPLOŠNI UREJEVALNIK IN VISOKOŠPOSOBEN ISKALNIK NAPAK. DODATNA PROGRAMSKA OPREMA K TEMU OSNOVNEMU PAKETU PA ZAJEMA ŠE MAKROJSKI ZBIRNIK, SIMBOLICNI ISKALNIK NAPAK, OBLIKOVALNIK TEKSTOV TER KRMILNIK ZA ISTOČASNO TISKANJE IZHODA IN UPORABO SISTEMA ZA IZVAJANJE UPORABNIŠKIH PROGRAMOV. K TEMU JE MOČ DODATI ŠE VRSTO PREVAJALNIKOV ZA VISOKE PROGRAMSKE JEZIKE IN ZA POSLOVNE NALOGE RAZLIČNIH PROIZVAJALCEV PROGRAMSKE OPREME.

PROGRAM CP/M SESTAVLJAJO TILE DELI:

BDOS (BASIC DISK OPERATING SYSTEM) UPRAVLJA SISTEM POIMENOVANIH ZBIRK, KO LAHKO IMAMO 16 DISKOVNIH POGONOV. VSAKA ZBIRKA LAHKO VSEBUJE DO 8 MILIJONOV ZLOGOV, KO SE PROSTOR DINAMIČNO DODELUJE IS SPOŠČA. SISTEMSKI POZIVI OMOGOČAJO KREIRANJE, PREIMENOVANJE, BRISANJE, ČITANJE IN PISANJE V ZBIRKE NA AKTIVNEM DISKOVNEM POGONU, IN SICER S SEKVENČNIM IN RELATIVNONAKLJUČNIM DOSTOPOM DO ZAPISOV. STANDARDNI PAKET CP/M JE PRIREJEN ZA UPORABO MEHKO SEKTORIRANIH IBM DISKET, KO DOPUŠČAMO 64 ZBIRK NA DISKETO, ŠTIRI DISKOVNE ENOTE IN 240 ZLOGOV NA DISKETO. PAKET IMA PRIROČNIK, KI OMOGOČI PRAKTIČNO VSAKO MODIFIKACIJO SISTEMA NA UPORABNIŠKI MIKRORAČUNALNIK. BDOS PODPIRA VEČ PERIFERNIH NAPRAV, KOT SO KONZOLA, LINIJSKI TISKALNIK TER PAPIRNI IN MAGNETNI TRAK. ZA VSE TE FUNKCIJE POTREBUJE BDOS 4K ZLOGOV POMNILNIKA.

CCP (CONSOLE COMMAND PROCESSOR) KOMUNICIRA Z UPORABNIKOM S POMOČJO NASLEDNJIH UKAZOV:

DIR - LISTA SE IMENIK
 TYPE - TISKA SE VSEBINA ASCII ZBIRKE
 REN - PREIMENOVANJE ZBIRKE
 ERA - ZBRIŠI ZBIRKO ALI MOŽIČO ZBIRK
 SAVE - REŠI VSEBINO NA DISK
 USER - SPREMEMI UPORABNIŠKO ŠTEVILKO

CCP OMOGOČA SISTEMU CP/M IN UPORABNIKU NALAGANJE IN IZVRŠEVANJE PROGRAMOV Z VGRAJENIMI UKAZI.

PIP (PERIPHERAL INTERCHANGE PROGRAM) SKRBI ZA PRENOS ZBIRK MED NAPRAVAMI IN DISKOM TER OPRAVLJA PREOBLIKOVANJE IN STIKANJE. FORMATIRNE MOŽNOSTI OBSEGAJO IZKLJUČITEB BITA PARNOSTI, KONVERZIJO PRIMEROV, UPORABO INTELLOVIH HEX ZBIRK, EKSTRAKCIJO PODZBIRK, TABULIRANJE, GENERIRANJE VRSTIČNIH ŠTEVILK IN STRANENJE.

ED (EDITOR) IMA UKAZE, KOT SO SUBSTITUCIJA NIZOV, ISKANJE NIZOV, VSTAVLJANJE, BRISANJE IN POMIK BLOKA. UREJEVALNIK JE DOKAJ MOČAN.

ASM (ASSEMBLER) ZA 8080 UPORABLJA STANDARDNO MNEMONIKO ZA INTELLOV PROCESOR IN PSEVDOOPERACIJE.

DDT (DYNAMIC DEBUGGING TOOL) OMOGOČA POPRAVLJANJE PROGRAMOV, IMA INTEGRALNI MODUL ZA ZBIRANJE IN INVERZNO ZBIRANJE TER VRSTO DIREKTIV.

SUBMIT OMOGOČA POVEZOVANJE (BATCH) UKAZNIH SKUPIN V OBLIKI ZBIRKE TER POŠILJA TO OPERACIJSKEMU SISTEMU Z ENIM SAMIM UKAZOM.

STAT SPREMINJA IN PRIKAŽUJE V/I TER STATUS ZBIRKE VKLJUČNO Z IZRAČUNOM PROSTEGA

PROSTORA, STATUS DISKET TER DODELJEVANJE NAPRAV.

LOAD PRETVORI INTEL OV HEX FORMAT V ABSOLUTNO BINARNO OBLIKO, KI JE PRIPRAVLJENA ZA IZVRŠITEV.

SYSGEN OBLIKUJE NOVE DISKETE IZ OBSTOJEČIH.

MOVCPM REGENERIRA CP/M SISTEM ZA RAZLIČNE POMNILNIŠKE KONFIGURACIJE IN DELUJE SKUPAJ S SYSGEN, KO SE GENERIRAJO KOPIJE CP/M.

CELOTEN OPERACIJSKI SISTEM CP/M JE PORAZDELJEN NA DISKETI IN JE V IZVIRNI OBLIKI OSPOSOBLJEN ZA DELOVANJE NA INTEL MDS-800 MICROCOMPUTER DEVELOPMENT SYSTEM, IN SICER ZA DISKOVNE POGONE Z ENOJNO GOSTOTO ZAPISA. TA, STANDARDNI CP/M SISTEM PA LAHKO UPORABNIK PREOBLIKUJE TAKO, DA DELUJE Z VSAKIM MIKRORAČUNALNIKOM, KI IZPOLNJUJE NASLEDNJE POGOJE:

- IMA 8080 ALI Z-80 CPU
- IMA NAJMANJ 20K ZLOGOV POMNILNIKA TIPA RAM, ZAČENŠI Z LOKACIJO 0000H
- IMA ENEGA DO ŠTIRI MEHKO SEKTORIRANE DISKOVNE POGONE IN FDC KRMILNIK
- IMA ASCII KONZOLO (NAVADNO CRT)

VRSTA ZNANIH PROIZVAJALCEV
MIKRORAČUNALNIKOV PRODAJA CP/M SKUPAJ S
KRMILNIMI SUBRUTINAMI V SVOJIH NAPRAVAH.

CP/M SISTEM SE PRODAJA V OBLIKI STROJNEGA KODA, IZVIRNA OBLIKA PA IMA ZNATNO VIŠJO CENO. DOKUMENTACIJA JE PRILOŽENA. TA PROGRAMSKA OPREMA JE LICENCIрана IN JO LAHKO UPORABLJAJO LE POSAMEZNIKI, KI SO KUPCI SISTEMA CP/M. OPREMA JE REGISTRIRANA IN VSAK PRODANI PAKET IMA SERIJSKO ŠTEVILKO, TAKO DA STA KOPIRANJE IN NADALJNA PRODAJA ONEMOGOČENI. REGISTRIRANI KUPEC DOBIVA OPOZORILA O IZBOLJŠAVAH IN SPREMENBAH OBSTOJEČE PROGRAMSKE OPREME.

SISTEMSKO DOKUMENTACIJO SESTAVLJA SEDEM PRIROČNIKOV:

- CP/M 2.0 USER'S GUIDE: TA PRIROČNIK DAJE PREGLED SISTEMA CP/M 2.0 IN POJASNJUJE LASTNOSTI TEGA RAZŠIRLJIVEGA OPERACIJSKEGA SISTEMA.

- AN INTRODUCTION TO CP/M FEATURES AND FACILITIES: TA PRIROČNIK PREDSTAVLJA ORGANIZACIJO CP/M SISTEMA Z OBLIKAMI NAVAJANJA ZBIRK, VGRAJENIH UKAZOV, DELOVANJA UREJEVALNIKA (ED), ZBIRNIKA (ASM), POPRAVLJALNIKA (DDT), PERIFERNEGA PROGRAMA (PIP) IN PROCESIRANJA POSLOV (SUBMIT).

- CP/M SYSTEM INTERFACE GUIDE: TA PRIROČNIK PODAJA PODROBNOSTI ZA PROGRAMIRANJE V OKOLICI SISTEMA CP/M. OPISANI SO VSI SISTEMSKI POZIVI S PODROBNOSTMI ORGANIZACIJE ZBIRK SISTEMA CP/M, DA SE TAKO LAHKO NAPIŠEJO PROGRAMI, KI DELUJEJO V MEHANIZMU ZBIRK SISTEMA CP/M.

- NADALJNI TRIJE PRIROČNIKI
CP/M ASSEMBLER GUIDE,
CP/M DEBUGGER GUIDE,
CP/M CONTEXT EDITOR GUIDE

OPISUJEJO PODROBNOSTI DELOVANJA TREH GLAVNIH PODSISTEMOV V CP/M, IN SICER UREJEVALNIKA, ZBIRNIKA IN POPRAVLJALNIKA (ED, ASM, DDT).

- CP/M SYSTEM ALTERNATION GUIDE: TA PRIROČNIK OPISUJE KORAKOMA POSTOPEK SPREMENBE SISTEMA CP/M ZA NESTANDARDNO MATERIALNO OPREMO.

KOT VIDIMO, GRE PRI SISTEMU CP/M ZA PROGRAMSKO OPREMO, KI JE PRENOSLJIVA, SPREMENLJIVA IN RAZŠIRLJIVA IN PRAV ZARADI TEH LASTNOSTI JE TA PROGRAMSKI PAKET POSTAL NEKE VRSTE STABDARD ZA OPERACIJSKE SISTEME MIKRORAČUNALNIKOV.

CENA OPISANEGA PAKETA ZNAŠA 150 \$, CP/M ZA DVOJNO GOSTOTO ZAPISA PA IMA CENO 200 \$. NASLOV PROIZVAJALCA JE: DIGITAL RESEARCH, P.O. BOX 579, PACIFIC GROVE, CA 93950.

A.P. ŽELEZNIKAR

GOVOREČI JAPONSKI KALKULATOR

JAPONSKO PODJETJE SHARP CORP. JE RAZVIL GOVOREČI KALKULATOR, KI SE BO V LETU 1980 PRODAJAL ZA 250 \$. KALKULATOR IMA GOVORNI SESTAVLJALNIK, KI OZVOČI VRSTO FUNKCIJ, KO PRITISKAMO NA TIPKE. KO PRITISNEMO TIPKO ZA ZNAK ENAKOSTI, SE IZVRŠI IZRAČUN, KI JE LAHKO SESTAVLJEN IZ STO KORAKOV. MED TIPKANJEM SLIŠIMO IMENA ŠTEVILK, SLIŠIMO PA TUDI REZULTAT PA TUDI PREJŠNJI IZRAČUN. OPERATOR LAHKO IZBERE POSAMEZNE ŠTEVILKE ALI ŠTEVILA. GOVORNA SINTEZA SE OPRAVLJA S HITRIM 4-BITNIM MIKRORAČUNALNIKOM. 60 BESED, KI JIH KALKULATOR LAHKO OZVOČI, JE SHRANJENIH V 6K-ZLOŽNEM POMNILNIKU ROM V OBLIKI IPULZNO KODNO MODULIRANIH KOMPRESIRANIH SIGNALOV, TAKO DA JE POTREBNIH SAMO PETDESETINA ŠTEVILA NORMALNIH BITOV. OZNAKA TEGA KALKULATORJA JE CS-6500 IN IMA 16-ŠTEVILČNI PRIKAZOVALNIK.

A.P. ŽELEZNIKAR

MILIJONBITNI MEHURČNI POMNILNIK

PODJETJE INTEL JE DALO V PRODAJO PRVI MEGABITNI MEHURČNI POMNILNIK, KI SE GA LAHKO KUPI TUDI V OBLIKI SESTAVLJENKE. MEHURČNI POMNILNIK PREDSTAVLJA NOV IZZIV ZA NAČRTOVALCE MIKRORAČUNALNIŠKIH SISTEMOV, SAJ JE DOVOLJ KOMPAKTEN, TROŠI MALO ENERGIJE IN JE IZREDNO ZANESLJIV. DOSLEJ SO BILI MEHURČNI POMNILNIKI DOKAJ DRAGI IN NEKOLIKO ODROČNI ZA UPORABO, KER SO POTREBOVALI ZAPLETENA NASLOVNA, KRMILNA IN POVEZOVALNA VEZJA. INTEL JE RAZVIL TUDI VSA TA SPREMLJAJOČA VEZJA, KI SO PRILAGOJENA UPORABI MEHURČNIH POMNILNIKOV V MIKRORAČUNALNIŠKIH. TA POMNILNIK VSEBUJE 128K ZLOGOV TER GA JE MOČ Z UPORABO POVEZOVALNIH VEZIJ VGRADITI V SISTEME Z MIKROPROCESORJI 8080, 8085, 8086 IN 8088.

INTELOV MEHURČNO POMNILNIK JE SESTAVLJEN IZ 7 KOMPONENT, KI JIH JE MOGOČE POVEZATI NA POVRŠINI 10 KRAT 10 CM. BISTVENI SESTAVNI DEL JE VEZJE Z OZNAKO 7110, KI JE MEHURČNI POMNILNIK. PREOSTALA INTEGRIRANA VEZJA PA SO ŠE: 7242 (OJAČEVALNIK ZA FORMATIRANJE IN SPREJEM), 7230 (GENERATOR TOKOVNIH IMPULZOV), 7250 (OJAČEVALNIK ZA NAVITJA) TER DVOJE 7254 VEZIJ (ŠTIRJE TRANZISTORJI V VEZJU). VEZJE 7220 JE KRMILNIK, KI DAJE TAKT IN SISTEMSKE FUNKCIJE, SESTAVLJENKA PA NOSI OZNAKO CPK-72.

IZ TEH INTEGRIRANIH KOMPONENT LAHKO SESTAVIMO TUDI VEČMEGABITNI POMNILNIK, SAJ OMOGOČA KRMILNIK 7220 VEZJA Z 8M BITI POMNILNIKA. MEHURČNI POMNILNIK JE POLPREVODNIŠKO VEZJO, ZANESLJIVO IN PODATKI OSTANEJO OHRANJENI TUDI PRI IZKLJUČITVI SISTEMA. DODATNE BATERIJE ZA OHRANJANJE PODATKOV V POMNILNIKU TAKO NISO POTREBNE. VEZJE 7242 LAHKO OPRAVLJA TUDI FUNKCIJO ZAZNAVANJA IN KOREKCIJE NAPAK.

INTELOV MEGABITNI MEHURČNI POMNILNIK PREDSTAVLJA IDEALNO REŠITEV ZA MIKRORAČUNALNIŠKO INSTRUMENTACIJO, TERMINALE, PROCESNO KONTROLO IN TELEKOMUNIKACIJE. KER SO VSI ELEMENTI NA RAZPOLAGO, JE MOČ ZGRADITI LASTEN IN UPORABI PRIKROJEN POMNILNIŠKI SISTEM. INTEL DOBAVLJA TUDI TEHNIČNO IN UPORABNIŠKO DOKUMENTACIJO PA TUDI GOTOVE SISTEME Z USTREZNO PROGRAMSKO PODPORO.

A.P. ŽELEZNIKAR

MIKRO RAČUNALNIK Z VOJAŠKIMI ZAHTEVAMI

Tvrdba INTEL je dala na tržišče prvi uporabniško krmiljen eno-čipni i mikroračunalnik, ki zadovoljuje vojaškim zahtevam. Računalnik M8748 je 8-bitni mikroračunalnik in je bil razvit za aplikacije, kjer se zahteva visoka zanesljivost; odgovarja vojaškimi standardom MIL-STD-883B. Temperaturno območje je od -55°C do $+100^{\circ}\text{C}$.

Na EPROM pomnilniku osnovan mikroračunalnik je zlasti primeren za krmilne in aritmetične aplikacije in je zamenljiv s prej razvitim M8048. Medtem, ko vsebuje M8748 1-Kby EPROM (1024x8 bitov) za shranjevanje uporabnikovih programov, ima M8048 1Kby maskirni ROM pomnilnik.

M8748 vsebuje še 8-bitni centralni procesor, 64 bytni RAM, 27 vhodno/izhodnih linij, 8-bitni intervalni časovnik / števnik in oscilatorsko-urino vezje. Sistem se lahko razširi z dopolnilnimi pomnilniki in perifernimi enotami.

Značilno za računalnik je, da ima več kot 70% od 90 instrukcij, ki so eno-besedne in ne daljše od dvo-besednih. Pakiran je v 40-kontaktne podnožje in potrebuje samo eno (5V) napajalno napetost. Minimalni instrukcijski cikel je dolg 4,17 μs .

R.Murn

DMA - KRMILNIK

Podjetje INTEL je dalo pred kratkim na tržišče DMA krmilni vezji z oznako 8237 in 8237-2; namenjeno je zlasti za skupno delovanje z mikroprocesorjema 8085A in 8088. 8237 je uporabno za frekvence do 3MHz in 8237-2 za 5MHz. PIN kompatibilni sta s počasnejšimi izvedbami kot so AMD 9517, 9517-1 in 9517-4.

R.Murn

VEZJE Z DVANAJSTIMI FREKVENCAMI

Japonsko podjetje Suwa Seikoska proizvaja hibridna integrirana vezja z oznako SS 8600, kjer so v 16-polnem DIL ohišju vsebovani naslednji funkcionalni deli: 600KHz kristalni oscilator, programski del in izhodna stopnja. S pomočjo štirih preklopnih kontaktov lahko izbiramo 12 različnih stabilnih frekvenc (1Hz do 200KHz).

Frekvence 1.10^0 in 2.10^0 do 1.10^5 Hz so takoj dosegljive, dočim dobimo s pomočjo enostavnega delilnika frekvence 5Hz, 50Hz, 500Hz, 5KHz in 50KHz. Poraba toka (v odvisnosti od izbrane frekvence) je med 0,5mA⁴ in 1,5mA. Izveden je v CMOS tehnologiji, točnost 1.10^{-4} .

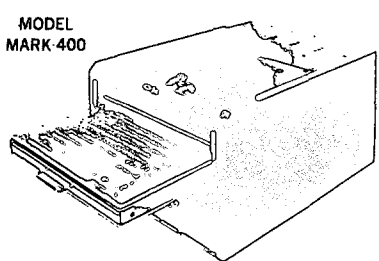
R.Murn

Q: HOW DO I CONNECT MY PRODUCTS TO MY TEST SYSTEM?

A: DATAMASTER

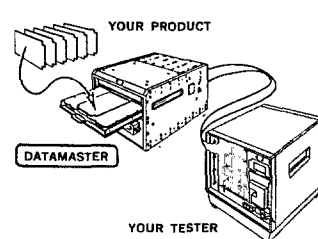
INTERFACING FIXTURES

MODEL MARK 400




- FAST
- SIMPLE
- ACCURATE
- NO MAINTENANCE
- REPLACEABLE PROBE PINS
- INTERCHANGEABLE PROBE HEADS
- FOUR SIZES
- EIGHT PROBE PIN STYLES
- FIELD-PROVEN
- COST-EFFECTIVE
- COMPLETE USER SUPPORT

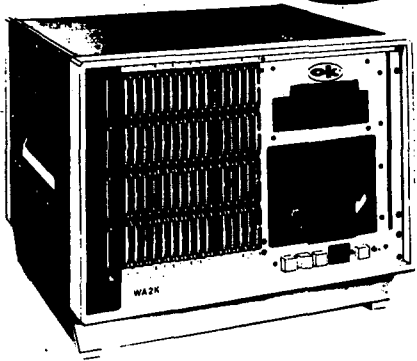
COMPATIBLE WITH ANY TESTING SYSTEM USED FOR MODERATE OR LOW VOLTAGE VERIFICATION OF BACK PANELS, PC BOARDS, MULTILAYER, HYBRID LOGIC ASSEMBLIES, FLAT ELECTRICAL ASSEMBLIES AND MORE.



OK MACHINE & TOOL CORPORATION



the NEW LOW COST answer for AUTOMATIC CIRCUIT TESTING from **OK**



Revolutionary SELF-PROGRAMMING SYSTEMS
for TESTING all types of Electronic Circuitry

■ **Model WA2K** FOR TESTS UP TO 1024 POINTS,
EXPANDABLE TO 2048 POINTS TEST CAPACITY.

■ **Model WA6K** FOR TESTS UP TO 2176 POINTS,
EXPANDABLE TO 6144 POINTS TEST CAPACITY.

Features:

- SELF-PROGRAMMING
- LOW COST PER TEST
- EASY TO OPERATE
- FAST
- RELIABLE
- ADVANCED ELECTRONIC DESIGN
- CAPACITY EASILY EXPANDED
- MONITORS OWN INTERNAL FAILURES
- CLEAR ERROR PRINT OUT
- SIMPLE INTERFACING WITH TEST OBJECTS

OK MACHINE AND TOOL CORPORATION

8485 CONNER STREET, BRONX, NEW YORK, N.Y. 10478 U.S.A.
• PHONE (212) 994-8800
TELEX NO. 17 9811 TELEFAX NO. 72796

LED DIODE ZA MAJHNE TOKOVE

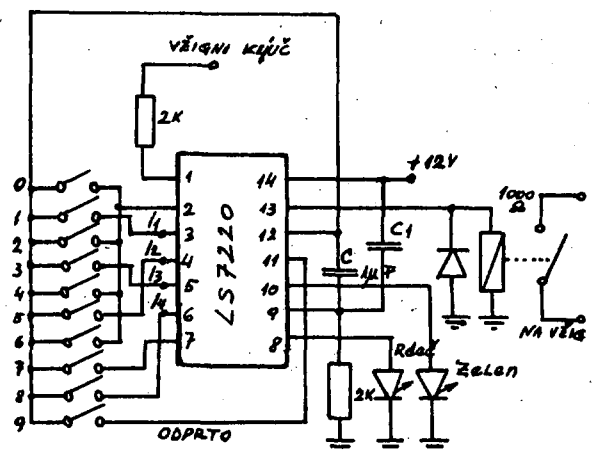
Tvrdba Xciton je izdelala LED-diode z rdečo svetlobo z oznako SP-848-5569. Značilnost le-teh je, da dosežemo značilno svetlobno jakost 1,5 mcd (zagotovljen minimum je 1 mcd) že pri toku 1mA, medtem ko pri dosedanjih diodah potrebujemo za enako svetlobno jakost 20 mA. Majhna poraba toka omogoča sedaj priključitev na baterijsko napajane naprave in za direktno izkrmiljenje standardnih tehnologij. Dioda je v standardni izvedbi s 5 mm prenosom in dolžine 8,8 mm.

R.Murn

VARNOSTNO VEZJE V ČIPU

Za varnost v poslopih, avtomobilih in napravah je izdelala tvrdka LSI Computer Systems elektronsko ključavnico LS 7220 v obliki čipa. Bistvo vezja predstavlja detektor zaporedij, ki preizkuša pravilnost zaporedij, ki jih zakodiramo s pomočjo štirih vhodnih signalov. V primeru pravilne kode lahko aktiviramo relejno vezje. Npr. pri avtomobilih moramo pred zavrtitvijo vžignega ključa s pomočjo ustrezne tastature pritisniti pravilno štiri mestno številko (tastatura ima deset tipk). Pri predaji avtomobila v servis, lahko ključavnico izključimo; to stanje označuje LED dioda zelene barve. (Z LED-om rdeče barve je označeno stanje, ko je ključavnica zaprta).

S pomočjo dodajanja kondenzatorjev, lahko dosežemo različno dolgo odprta ali zaprta stanja ključavnice. Čip z oznako LS 7220 potrebuje napajalno napetost od 5-18V. Mirovni tok je pri 12V samo 40uA, obremenitveni pa 30mA. Natančnejše informacijo lahko dobimo npr. pri Tvrdki Scantec GmbH, D-8 München 19, Landshuter Allee 49.



R.Murn

OSEBNI RAČUNALNIKI V LETU 1980

V LETU 1980 SE PRIČAKUJE BISTVENO VEČJA PRODAJA OSEBNIH RAČUNALNIKOV KOT V PREJŠNJEM LETU. INDUSTRIJA OSEBNIH RAČUNALNIKOV JE STARA KOMAJ PET LET IN PRAV SEDAJ PREHAJA V SVOJO ZRELO DOBO. TA INDUSTRIJA IMA PERSPEKTIVNO PRIHODNOST, SAJ NAJ BI POKRIVALA ŠIROKO POTROŠNIŠKO PODROČJE, KI STOPNJUJE SVOJE ZAHTEVE PO KOLIČINI IN KAKOVOSTI.

VENDAR SE TUDI V TEJ INDUSTRIJI ŽE POJAVLJAJO DOLOČENE ZAPREKE, KI ONEMOGOČAJO NJENO NORMALNO RAST. PROIZVODNJE TEŽAVE IZVIRAJO NPR. IZ POMANKANJA INTEGRIRANIH VEZIJ IN DOBAVITELJI TEH VEZIJ KASNIJO. V LETU 1978 JE PRODAJA TE INDUSTRIJE DOSEGLA POVEČANJE ZA 67 ODSOTKOV. VZROKI ZA TA PORAST SO PREDVSEM V UPORABI OSEBNIH RAČUNALNIKOV ZA MALE POSLOVNE OBDELAVE. V LETU 1979 PA SO SE POJAVILI ŠE NOVI PROIZVAJALCI, KOT SO DEC, TEXAS INSTRUMENTS IN ATARI. POJAVILE SO SE TUDI PRODAJALNE PROIZVAJALCEV, KO JE DEC ODPRL VERIGO SVOJIM TRGOVIN ZA UPORABNIKE MAJHNH RAČUNALNISKIH SISTEMOV.

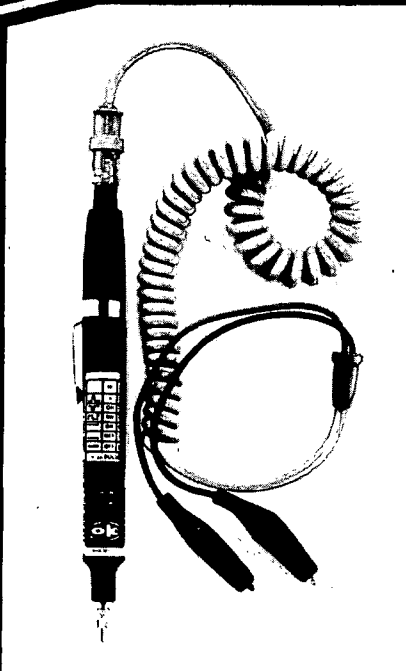
ČEPRAV JE INDUSTRIJA OSEBNIH RAČUNALNIKOV ŽIVA IN ZNAŠA NJENA RAST VSAJ 50 ODSOTKOV NA LETO, JE MOČ PRIČAKOVATI TUDI NEUSPEHE NEKATERIH PODJETIJ. V PRETEKLEM LETU SO NPR. NEKATERE ZNANE TVRDE PRENEHALE S PROIZVODNJO OSEBNIH RAČUNALNIKOV, KOT NPR. IMSAI, POLYMORPHIC IN XITAN; VZROKI ZA IZSTOP S TRZIŠČA SO BILI PREDVSEM V NESPOSOBNOSTI PRENOSA LJUBITELJSKEGA TRZIŠČA NA PODROČJI MALIH POSLOVNIH OBDELAV IN VISOKOTEHNOLOŠKIG SEGMENTOV. ZGODILO SE JE TUDI TO, DA JE NPR. BILO PODJETJE MITS/ALTAIR KUPLJENO OD PODJETJA PERTEC COMPUTER, TO PODJETJE PA SO KASNEJE KUPILI ZAPADNI NEMCI, TJ. VOLKSWAGEN AG.

NAPOVEDI KAŽEJO, DA BO V LETU 1983 TRZIŠČE OSEBNIH RAČUNALNIKOV SPREJELO VSAJ 2 MILIJONA RAČUNALNIKOV Z VREDNOSTJO DVEH MILIJARD DOLARJEV. ZA LETO 1985 NAPOVEDUJEJO VSAJ PET MILIJARD DOLARJEV ZA TRZIŠČE OSEBNIH RAČUNALNIKOV. KLJUB TAKIM NAPOVEDIM PA SE JE PODJETJE HEWLETT-PACKARD ODLOČILO, DA NUDI SVOJ OSEBNI RAČUNALNIK HP-85 PREDVSEM ZA POSLOVNE NALOGE IN TEHNIČNIM STROKOVNJAKOM, KOT SO INŽENIRJI, RAZISKOVALCI, KNJIGOVODJE IN NALOŽBENI ANALITIKI.

PODJETJE APPLE COMPUTERS MENI, DA JE DOM POTENCIALNO NAJVEČJE TRZIŠČE ZA OSEBNE RAČUNALNIKE IN PRIČAKUJE, DA BO IMEL OSEBNI RAČUNALNIK PODOBNO VLOGO DOMA, KOT JO IMA TELEVIZOR, STEREOSISTEM ALI MIKROVALOVNA PEČ. SEVEDA PA JE TRZIŠČE TUDI DRUGJE, NPR. V POSLOVANJU, ZNANOSTI IN POKLICNIH DEJAVNOSTIH. POMEMBNA PODROČJA PA SO OBDELAVA BESEDIL, KNJIGOVODSTVO IN RAČUNOVODSTVO. SKRATKA, VEČINA PROIZVAJALCEV OSEBNIH RAČUNALNIKOV PRIČAKUJE PLASMAN SVOJIH IZDELKOV ZLASTI NA POSLOVNEM PODROČJU IN ZA TEHNIČNE STROKOVNJAKE.

A.P.ŽELEZNIKAR

NEW!



PRB-1 DIGITAL LOGIC PROBE

Compatible with DTL, TTL, CMOS, MOS and Microprocessors using a 4 to 15V power supply. Thresholds automatically programmed. Automatic resetting memory. No adjustment required. Visual indication of logic levels, using LED's to show high, low, bad level or open circuit logic and pulses. Highly sophisticated, shirt pocket portable (protective tip cap and removable coil cord)

- DC to > 50 MHz
- 10 Nsec. pulse response
- 120 K Ω impedance
- Automatic pulse stretching to 50 Msec.
- Automatic resetting memory
- Open circuit detection
- Automatic threshold resetting
- Compatible with all logic families 4-15 VDC
- Range extended to 15-25 VDC with optional PA-1 adapter
- Supply O.V.P. to ± 70 VDC
- No switches/no calibration

OK MACHINE & TOOL CORPORATION

3455 Conner St., Bronx, N.Y. 10475 (212) 994 6600 Telex 125091



VSEBINA LETNIKA 1979

Babić G.A.:

Primena AKK MR metode u analizi kružne mreže računara za podršku distribuirane baze podataka (št.3, str.58)

Barber D.L.A., Kalin T.:

Scenarij za raziskave na području računalniških mreži (št.2, str. 29)

Barber D.L.A.:

Vloga računalniških mreži se spreminja (št.3, str.28)

Barić A.:

Grafički terminal s prikazom slike na TV monitoru (št.1, str. 82)

Barlić B.:

Prečni prevajalnik za Pascal (št.1, str. 63)

Budin L., Nožica Ž., Peruško V., Vuković D.:

Program za proračun pouzdanosti različitih memorijskih sistema (št. 2, str. 34)

Čop R., Kovačević M.:

Tehnologija ožičavanja I (št. 3, str. 64)

Čeramilac S., Glušac D.:

Dijalog računarskih sistema sa periodičnim odzivom (št. 1, str. 55)

Gams M., Lavrač N., Bratko I.:

Osnovni koncepti in struktura ekspertnih sistema (št. 4, str.11)

Guid N.:

Simulacija strežnih sistema z GPSS (št. 1, str. 72)

Hadžina N.:

Principi izgradnje sistemskih programskih komponenti u multi-mikroprocesorskim sistemima (št. 4, str.17)

Hadžina N.:

Postupci detekcije stanja potpunog zastoja računarskog sistema na modelu grafa sistema (št. 2, str. 14)

Hodžić M., Vrhovac S.:

Primjena mikroprocesora MPR-52B na jedan primjer praćenja ciljeva u realnom vremenu (št.3, str.18)

Han S.:

Neke društvene implikacije informatike (št.4, str. 4)

Jemuović D.:

O postavljanju jednostavnih hipoteza pomoću računara (o lancima atoma) (št. 4, str. 53)

Juričić-Kette W.:

Iskustva u obradi podataka na malim poslovnim sistemima (št.3, str. 41)

Kastelic B., Kovačević M., Novak D.:

Mali diskovni operacijski sistem (št. 3, str. 9)

Kastelic B., Kovačević M., Hadži A.:

Krmilno vezje za gibki disk (št.1, str. 32)

Kapus M., Devide G., Horvat B.:

Konvalucijski kodirni in dekodirni postopek pri mikro računalniku ISKRA DATA 1680 (št.2, str. 47)

Kapus M., Železnikar A.P.:

Komunikacijski protokoli (št. 2, str. 50)

Kandus G., Špegel M.:

Računalniški vid: Obravnava vizualnih informacij in analiza slik (št. 4, str.30)

Knop J., Speth R.:

Prenos daljinskih paketnih obdelav med različnimi računalniki (št.1, str. 51)

Kovačević M., Novak D., Kastelic B., Železnikar A.P.:

DMA procesori u u mikro računarskim sistemima (št. 2, str. 58)

Leskóvar S.:

Spektralne črte informacijskih sistema in kritični pogledi nanje (št. 1, str. 3)

Miletić M., Komunjer M.:

Sistem obrade podataka na svetskom prvenstvu u skijaškim skokovima na Planici 1979 godine (št.1, str.68)

Miljan D., Kolbezen P.:

Nekateri pripomočki načrtovanja mikroručunalniških sistema (št. 4, str. 25)

Murn R., Peček D.:

Verifikacija zapisa serijskih podatkov (št.2, str. 25)

Novak D., Železnikar A.P.:

Mikroručunalniška kriptografija II (št. 3, str. 22)

Novak D., Exel M., Kovačević M., Kastelic B.:

Modula-Programski jezik prihodnosti za mikroručunalniške aplikacije? (št. 2, str. 18)

Otović V.:

Softverski sistemi za simulaciju multi-mikro-procesorskih računara (št. 2, str. 43)

Popovski D.B.:

Jedan hibridan algoritam za nalaženje korena (št.3, str. 16)

Popovski D.B.:

Jedan Fortran IV potprogram za nalaženje izdavanog korena (št. 4, str. 23)

Prešeren S., Špegel M., Dacar F.:

Senzorski sistemi robotov (št. 4, str. 42)

Raić D.:

Integrirani mikroručunalnik ISKRA-EMZ 1001 (št.1, str. 12)

Reinhardt R., Martinec M., Dorn R.:

Tretjerepubliško tekmovanje srednješolcev iz područja računalništva (št. 1, str. 76)

Rozman I.:

Programska rešitev izvajanja programa po korakih za M 6800 (št. 1, str. 42)

Ružić F.:

Prilog definiciji informacijskih sistema za javno komuniciranje podacima i informacijama (št.4, str. 35)

Šubelj M., Korenini J., Novak F., Trobec R.:

Kodiranje in dekodiranje korekturnega koda z mikroracionalnikom (št.3, str. 36)

Velašević D.M.:

Konstrukcija asemblera za interaktivni razvoj programa (št. 4, str.49)

Vrsalović D., Filipović N.:

Mikroprocesorski terminalski koncentrador (št.1, str.47)

Žečević M.:

Sistemska predstroženost u implementaciji formalno izvedenih determinističkih programa (št.3, str.3)

Železnikar A.P.:

Mikroracionalniška kriptografija I (št.1, str.24)

Železnikar A.P., Novak D.:

Mikroracionalnik IKE-1 s procesorjem I 8086 (št. 2, str.3)

Železnikar A.P.:

Procesiranje tekstov z mikroracionalniki I (št. 3, str. 48)

Železnikar A.P.:

Dopolnitev procesorja teksta (št. 4, str. 58)

CENIK OGLASOV

Ovitek - notranja stran (za letnik 1979)	
2 stran -----	20.000 din
3 stran -----	15.000 din
Vmesne strani (za letnik 1979)	
1/1 stran -----	9.600 din
1/2 strani -----	6.000 din
Vmesne strani za posamezno številko	
1/1 stran -----	3.600 din
1/2 strani -----	2.400 din
Oglasi o potrebah po kadrih (za posamezno številko)	
	1.200 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 1979)	
2nd page -----	1100 ₯
3rd page -----	880 ₯
Inside pages (for all issues of 1979)	
1/1 page -----	660 ₯
1/2 page -----	440 ₯
Inside pages (individual issues)	
1/1 page -----	220 ₯
1/2 page -----	165 ₯
Rates for classified advertizing:	
each ad -----	55 ₯

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

SREČANJA

23-25 junij, Norfolk, Velika Britanija

EDUCATIONAL COMPUTING CONFERENCE

Organizator: CCUC, ACM, SIGSCE, ACM, SIGCUE, ACM SIGUCC, ADCIS, AEDS, AERA/SIGCAL, ABEE/COED, ECMI, IEEE-CS, AFIPS
 Informacije: Gerald, L. Engel, Computer Science Dept., Christopher Newport, College, Newport News, VA 23606; 804 599-7065

23-27 junij, Roma, Italija

IBI WORLD CONFERENCE ON TRANSBORDER DATA FLOW POLICIES

Organizator: Intergovernmental Bureau for Informatics
 Informacije: IBI, Viale Civiltà del Lavoro 23, POB 10253, 00144 Roma, Italy

23-27 junij, Brussels, Belgija

WORLD FORUM OF INTERNATIONAL TRANSNATIONAL ASSOCIATIONS

Organizator: Union of International Associations (UAI)
 Informacije: UAI, rue Aux Laines 1, 1000 Brussels, Belgium

23-27 junij, Cambridge, Velika Britanija

SYMPOSIUM ON RESEARCH AND DEVELOPMENT IN INFORMATION RETRIEVAL

Organizator: Joint British Society in cooperation with ACM, SIGIR
 Informacije: Michael J. McGill, School of Information Studies, Syracuse University, Syracuse, NY 13210; 315 423-2911

24-26 junij, Nordwijkerhout, Nizozemska

APL 80 INTERNATIONAL CONFERENCE

Organizator: Dutch Society for Informatics
 Informacije: J. Mulder, c/o CRI, Postbus 9312, 2300 RA Leiden, The Netherlands

25-27 junij, Interlaken, Švica

SIMULATION 80

Organizator: IASTED IMACS, Swiss Association for Operations Research,
 Informacije: Simulation 80, Box 354, CH-8053 Zürich, Switzerland

30 junij-4 julij, Bratislava, ČSSR

INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND INFORMATION CONTROL SYSTEMS OF ROBOTS

Organizator: The Scientific Board for Artificial Intelligence of the Systems Analysis Committee of the Presidium of the Academy of Sciences of the USSR and the Institute of Technical Cybernetic of the Slovak Academy of Sciences
 Informacije: Institute of Technical Cybernetic of the Slovak Academy of Sciences, 80931 Bratislava, Dubravska 9, ČSSR

2-4 julij, Univerza Aberdeen, Aberdeen, Škotska

CONFERENCE ON DATABASES

Organizator: University of Aberdeen
 Informacije: S.M. Deen, Dept. of Computing Science, University of Aberdeen, Old Aberdeen AB9 2UB, Scotland, U.K.

7-11 julij, Stanford, ZDA

TUTORIAL CONFERENCE ON PRACTICAL OPTIMIZATION

Organizator: Stanford University Systems, Optimization Laboratory
 Informacije: Philip E. Gill, Systems Optimization Laboratory, Dept. of Operations Research, Stanford University, Stanford, CA 94305

8-11 julij, Les Arc, Francija

5th CONFERENCE ON AUTOMATED DEDUCTION

Organizator: IRIA
 Informacije: IRIA, Service des Relations Exterieures, Domaine de Voluceau, 78150 Le Chesnay, France

14 julij, Madison, Wis., ZDA

NONLINEAR PROGRAMMING SYMPOSIUM

Organizator: Mathematical Programming Society
 Informacije: Nonlinear Programming Symposium 4, Computer Sciences Dept. University of Wisconsin, 1210 W. Dayton St., Madison, WI 53706; 608 262-1204

16-21 julij, Hong Kong

THE 5th BIENNIAL MEETINGS OF THE SOUTHEAST ASIAN MATHEMATICAL SOCIETY RECENT ADVANCES IN MATHEMATICS AND ITS APPLICATIONS

Organizator: Southeast Assian Mathematical Society, Hong Kong
 Informacije: Southeast Asian Mathematical Society, Hong Kong

14-18 julij, Seattle, ZDA

SIGGRAPH 80, SEVENTH ANNUAL CONFERENCE ON COMPUTER GRAPHICS AND INTEZACTIVE TECHNIQUES

Organizator: ACM, SIGGRAPH
 Informacije: SIGGRAPH 80, Box 88203, Seattle, WA 98188; 206 453-0599

21-23 julij, Durban, Južnoafriška Republika

SIXTH SOUTH AFRICAN SIMPOSIUM ON NUMERICAL MATHEMATICS

Organizator: University of Natal
 Informacije: Chairman, Computer Science Dept., University of Natal, King George V Ave., Durban 4001, Republic of South Africa

22-25 julij, Cambridge, Velika Britanija

FORTH EUROPEAN CONFERENCE ON OPERATION RESEARCH

Organizator: AF CET
 Informacije: Dr. J.P.Brans, Euro IV; University of Brussels, V.U.B. - CSOO, Pleinlaan 2-B 1050 Bruxelles, Belgium

18-22 august, Edinburgh, Velika Britanija

4th SYMPOSIUM ON COMPUTATIONAL STATISTICS

Organizator: International Association for Statistical Computing
 Informacije: COMPSTAT 1980, director Program Library limit, Edinburgh University 18, Buccleuch Place, Edinburgh EH8 9LN, Scotland

24-27 august, Stanford, ZDA

1980 LISP CONFERENCE

Organizator: Stanford University
 Informacije: John R.Allen, Stanford Artificial intelligence I ab. Stanford University, Stanford, CA 94305; 414 497-4971

25-27 august, San Francisco, ZDA

NATIONAL MEETING OF AMERICAN CHEMICAL SOCIETY, SYMPOSIUM ON SUPER COMPUTERS IN CHEMISTRY

Organizator: ACM, Division of Computers in Chemistry
 Informacije: Peter Lykos, Illinois Institute of Technology, Chichago, IL 60616; 312 567-3430

25-27 avgust, Seattle, ZDA

1980 SUMMER COMPUTER SIMULATION CONFERENCE

Organizator: SCS, AMS, ISA, AGU, AIAA, BMES, IMACS, AICLE, IEEE-CS, IEEE Systems
 Informacije: David R.S.McCall Boeing Aerospace Co., Mail Stop, 84-16, Box 3999, Seattle, WA 98124

27-28 avgust, Asilomaz, Calif., ZDA

SECOND WORKSHOP ON PICTURE DATA DESCRIPTION AND MANAGEMENT

Organizator: IEEE-CS
 Informacije: Harry Hayman, Box 639, Silver Spring, MD 20901; 301 439-7007

1-5 september, Rydzyna, Poljska

9th INTERNATIONAL SYMPOSIUM ON MATHEMATICAL FOUNDATIONS OF COMPUTER SCIENCE

Organizator: Polish Academy of Sciences, Institute of Computer Science
 Informacije: Maluszynski, Institute of Computer Science, Polish Academy of Sciences, Box 22,00-901 Warsaw PKiN, Poland

September, Katowice, Poljska

THIRD INTERNATIONAL CONFERENCE ON FAULT-TOLERANT SYSTEMS AND DIAGNOSTICS

Organizator: Silesian University, Department of Applied Electronics and Electrotechnics
 Informacije: doc.dr.eng. Jan Piecha, Dept. of Applied Elektronics and Electrotechnics, Silesian University, Katowice 40-955, Ul. Podgorna 4, Poland

9-11 september, Cracovie, Poljska

EIGHT CONFERENCE OF THE IMECO TECHNICAL COMMITTEE TG. 3

Organizator: IMEKO
 Informacije: Dr. A.Gizmajer, PKNiM, ul.Elektoralna 2, 00950, Varsovie, P.O.B. P10, Poland

16-18 september, London, Velika Britanija

EUROMICRO 80, SIXTH SYMPOSIUM ON MICROPROCESSING AND MICROPROGRAMMING

Organizator: European Association for Microprocessing and Microprogramming
 Informacije: Lionel R.Tompson, H.S.D.E., Hatfield AL 109 LP, England

16-19, september, Lausanne, Switzerland

EUSIPCO - 80

FIRST EUROPEAN SIGNAL PROCESSING CONFERENCE
 Organizator: Swiss Federal Institute of Technology
 Informacije: Mrs. C.Stehle - EUSIPCO-80 - Dept. of Electrical Engineering, Swiss Federal Institute of Technology, 16 Chemin de Bellerive, CH-1007 Lausanne, Switzerland. Tel. (21)47 26 24 Telex 24478 EPFVD CH

17-19 September, Palo Alto, Calif., ZDA

JOINT ACM SIGSMALL/SIGPC SYMPOSIUM ON SMALL SYSTEMS

Organizator: ACM, SIGSMALL, SIGPC
 Informacije: Nancy Grosch, 1383 Zürich Terrace, Sunnyvale, CA 94087; 408 732-0619

22-26 september, Washington, ZDA

COMPCON FALL 80

Organizator: IEEE-CS
 Informacije: Compcon Fall 80, Box 639, Silver Spring MD 20901; 301 439-7007

23-25 september, Pariz, Francija

AUTOMATING TESTING 80, CONFERENCE AND EXHIBITION

Organizator: Network, Printers Mews, Market Hill, Buckingham, MK18 1 JX, England
 Informacije: Brigadier R. Knowies, Institute of Quality Assurance, 54 Princes Gate, Exhibition Road, London, SW7, U.K.

29 september-4 oktober, Tokyo, Japonska

COLING 80, INTERNATIONAL CONFERENCE ON COMPUTATIONAL LINGUISTICS

Organizator: International Committee on Computational Linguistics
Informacije: David G. Hayes, 5048 Lake Shore Road, Hamburg, NY 14075

2-4 oktober, Tokyo, Japonska

IFIP WORKING CONFERENCE ON MAN-MACHINE COMMUNICATION IN CAD/CAM

Organizator: Department of Precision Machinery Engineering, Faculty of Engineering, The University of Tokyo, 7-3 Hongo, Bunkyo-ku, Tokyo 113, Japan
Informacije: prof. Toshio Sata, Dept. of Precision Machinery Engineering, Faculty of Engineering, the University of Tokyo, 7-3 Hongo, Bunkyo-ku, Tokyo 113, Japan

29 september-4 oktober Tokyo, Japonska

MEDINFO 80: 3rd WORLD CONFERENCE ON MEDICAL INFORMATICS

Organizator: IFIP TC4
Informacije: IFIP Foundation, 40 Paulus Potterstraat, 1071 DB Amsterdam, Netherlands

1-3 oktober, Kyoto, Japonska

TENTH INTERNATIONAL SYMPOSIUM ON FAULT-TOLERANT COMPUTING

Organizator: IEEE-CS Tech. Comm. on Fault-Tolerant Computing and IECE of Japan Tech. Groups on Electronic Computers and on Reliability
Informacije: Shuzo Yajiwa, Dept. of Information Science, Kyoto University, Kyoto 606, Japan

3-5 oktober, Kyoto, Japonska

CONFERENCE ON MAN-MACHINE COMMUNICATIONS IN CAD AND CAM

Organizator: IFIP WG5.2, 5.3
Informacije: IFIP Secretariat, 3rue du Marche, CH-1204, Geneva, Switzerland

6-9 oktober, Tokyo, Japonska in

14-17 oktober, Melbourne, Australia

IFIP CONGRESS 80

Organizator: IFIP, Information Processing Society of Japan, Australian Computer Society
Informacije: 8th World Computer Congress, c/o AFIPS, 210 Summit Ave., Montvale, NJ 07645; or IFIP Foundation, 40. Paulus Potterstraat, 1071 DB Amsterdam, The Netherlands

8-11 oktober, Kyoto, Japonska

SEVENTH INTERNATIONAL CODATA CONFERENCE

Organizator: International Council of Scientific Unions Comm. on Data for Science and Technology
Informacije: CODATA Secretariat, 51 Boulevard de Montmorency, 75016 Paris, France

14-17 oktober, Düsseldorf, ZRN

6th IFAC/IFIP INTERNATIONAL CONFERENCE ON DIGITAL COMPUTER APPLICATIONS TO PROCESS CONTROL

Organizator: VDI/VDE-Gesellschaft Mess- und Regelungstechnik
Informacije: VDI/VDE, Postfach 1139, D-4000 Düsseldorf 1, Germany

21-24 oktober, Jakarta, Indonesia

SEARCC 80

Organizator: South East Asia Regional Computer Confederation
Informacije: SEARCC 80 Conference Implementation Committee, P.O.Box 4487, Jakarta, Indonesia

27-29 oktober, Nashville, ZDA

ACM 80

Organizator: ACM
Informacije: Charles Bradshaw, Computer Center, Box 1577, Vanderbilt University Station, Nashville, TN 37203

16-19 november, Morgantown, ZDA

ACM SIGUCC EIGHTH ANNUAL USER SERVICES CONFERENCE

Organizator: ACM, SIGUCC (University-Computing Centers)
Informacije: Rita Sepowitz Saltz, Supervisor, Information Services, WUNET, 837 Chestnut Ridge Road, Morgantown, WV 26505; 304 293-5192



Vabimo vas na

**XIV. JUGOSLOVANSKI SIMPOZIJ
O TELEKOMUNIKACIJAH**

ki ga v organizaciji

Elektrotehniške zveze Slovenije
prirejata

**Zajednica jugoslovenskih PTT in
Stručna sekcija za telekomunikacije ETAN**

v času mednarodne razstave SODOBNA ELEKTRONIKA
na Gospodarskem razstavišču v Ljubljani.

**Usmerjena tema simpozija je
UPRAVLJANJE IN NADZOR TELEKOMUNIKACIJSKIH
SISTEMOV**

Splošna problematika s področja telekomunikacij zajema
naslednje skupine:

- a) Komutacijska tehnika
- b) Tehnika prenosnih sistemov
- c) Brezžična prenosna tehnika
- d) Telekomunikacijska omrežja in naprave
- e) Tehnologija
- f) Izvori napetosti
- g) Telekomunikacijski terminali
- h) Zaščita in zanesljivost

Simpozij naj bi s konkretnimi predlogi za ukrepanje dal
odgovor na številne probleme, vezane na usmerjeno temo
ter na mnoga druga strokovna vprašanja, s katerimi se
srečujejo naši strokovnjaki pri raziskavah in razvoju, v
proizvodnji, eksploataciji in pri vzdrževanju, prikazal naj
bi tudi stanje razvoja telekomunikacij v svetu in dosežke
naših strokovnjakov in organizacij.

Splošne informacije

Za prijavo in predajo referatov so postavljeni naslednji
roki:

- prijava referata s povzetkom 30. april 1980
- predaja referata 15. julij 1980

Povzetek naj vsebuje izčrpen pregled referata in naj ne
presega 250 besed.

Posebna strokovna komisija bo prijavljene referate pre-
gledala in ocenila ter bo najkasneje mesec dni po spre-
jetju prijave obvestila referenta, ali je prijavljeni referat
sprejet v program simpozija. Istočasno z obvestilom bodo
referenti prejeli navodilo o pisanju referata.

Prijavnina za simpozij je:

1.200 din za referente, fakultete in znanstvene ustanove,
1.600 din za druge udeležence
in je plačljiva vnaprej na žiro račun 50101-678-48748.

Vsi, ki bodo v prijavnici za simpozij označili, da žele pre-
nočišče, bodo prejeli posebno obvestilo z vsemi potreb-
nimi podatki in prijavnico za rezervacijo prenočišča.

Vse nadaljnje informacije daje:

ELEKTROTEHNIŠKA ZVEZA SLOVENIJE

61000 Ljubljana, Erjavčeva 15, telefon (061) 22 263

Predsednik
organizacijskega odbora
Alfonz Medved, dipl. ing.

Usmerjena tema

**UPRAVLJANJE IN NADZOR TELEKOMUNIKACIJSKIH
SISTEMOV**

Usmerjena tema zajema nadzor in upravljanje telekomuni-
kacijskih sistemov in naprav, da bi z ekonomizacijo stro-
škov upravljanja sistemov povečali njihovo zanesljivost in
dosegljivost. Pozornost je usmerjena na oceno problema-
tike v jugoslovenskem in širšem prostoru, na merilne me-
tode, probleme projektiranja merilne opreme in nadzora
nad delovanjem in upravljanjem telekomunikacijskih siste-
mov.

Usmerjeno temo razčlenimo na:

1. **Zajemanje podatkov**, ki obsega razne merilne postopke,
merilne naprave in opremo, shranjevanje izmerjenih
podatkov ter eventualni prenos v obdelovalne centre.
2. **Obdelavo podatkov**, kjer gre predvsem za vrednotenje
zajetih podatkov s stališča statistike, vzdrževanja, upra-
vljanja, planiranja in projektiranja telekomunikacijskih
omrežij.
3. **Ukrepanja**, ki zajema različne strategije upravljanja in
vzdrževanja telekomunikacijskih omrežij na osnovi zbra-
nih in obdelanih podatkov.

Področja obravnav so naslednja:

- promet in kvaliteta storitev
- digitalni sistemi
- kabli
- optične komunikacije
- mikrovalovni sistemi
- antene in prenosne poti
- govorni signal
- slikovni signal
- omrežja in posredovalni sistemi
- frekvence in čas

NAVODILO ZA PRIPRAVO ČLANKA

Avtorje 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 vršite na papir.

Bodite natančni pri tipkanju in temeljiti pri kori giranju. 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 presledki in brez zamikanja prve vrstice novega odstavka.

Prva stran članka:

- v sredino zgornjega okvira na prvi strani napišite naslov članka z velikimi črkami;
- v sredino pod naslov članka napišite imena avtorjev, ime podjetja, mesto, državo;
- na označenem mestu čez oba stolpca napišite povzetek članka v jeziku, v katerem je napisan članek. Povzetek naj ne bo daljši od 10 vrst.
- če članek ni v angleščini, ampak v katerem od jugoslovanskih jezikov izpusite 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 napišite povzetek tudi v enem od jugoslovanskih jezikov;
- izpusite 2 cm in pričnite v levo kolbno 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 vpišite 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.

Časopis INFORMATICA

Uredništvo, Institut Jožef Stefan, Jamova 39, Ljubljana

Naročam se na časopis INFORMATICA. Predplačilo bom izvršil po prejemu vaše položnice.

Cenik: letna naročnina za delovne organizacije 300,00 din, za posameznika 100,00 din.

Časopis 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.

Časopis INFORMATICA
Uredništvo, Institut Jožef Stefan, Jamova 39, Ljubljana

Please enter my subscription to INFORMATICA 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

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:

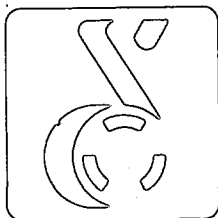
Headings are separated from text by double spacing.

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.



delta sistemi

ELEKTROTEHNA LJUBLJANA, TOZD za računalništvo Digital proizvaja in prodaja naslednje standardne računalniške konfiguracije:

DELTA 700/80

- DELTA 700 centralna procesna enota
- 512 KByte centralni pomnilnik s paritetno kontrolo, ki se lahko razširi do 4 MBytev
- 2 KByte vmesni pomnilnik spomina (cache)
- ura realnega časa
- konzolni terminal s kontrolno enoto
- dve diskovni enoti s kapaciteto po 80 MByte s kontrolno enoto
- dve magnetni tračni enoti 800/1600 b/i, 45.1/2, 9 kanalni zapis s kontrolno enoto
- asinhroni komunikacijski vmesnik (8 linij: EIA/CCITT modemska izhod) (8 linij: 20 mA tokovna zanka)
- 600 linijski tiskalnik
- KOPA 1000 alfanumerični video display terminal (2 kom)

DELTA 340/80

- DELTA 340 centralna procesna enota
- 256 KByte centralni pomnilnik s paritetno kontrolo
- 2 KByte vmesni pomnilnik (cache)
- ura realnega časa
- konzolni terminal s kontrolno enoto
- enota za baterijsko napajanje pomnilnika
- procesor s plavajočo vejico (floating point processor)
- dve diskovni enoti s kapaciteto po 80 MByte s kontrolno enoto
- dve magnetni tračni enoti (1600 b/i, 75 i/s, 9 kanalni zapis), s kontrolno enoto
- asinhroni komunikacijski vmesnik (8 linij EIA/CCITT modemska izhod) (8 linij 20 mA tokovne zanke)
- 600 linijski tiskalnik
- KOPA 1000 alfanumerični video display terminal (2 kom.)

DELTA 340/5

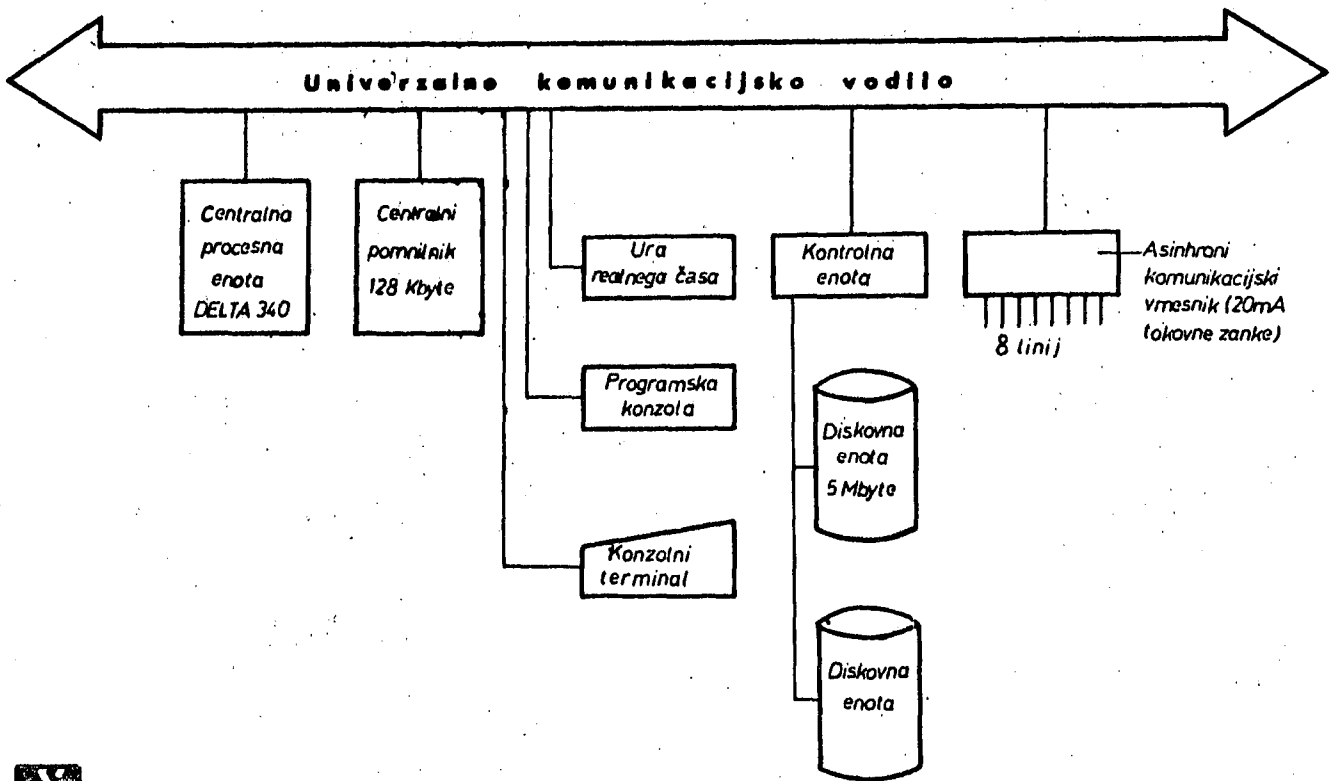
- DELTA 340 centralna procesna enota
- 128 KByte centralni pomnilnik s paritetno kontrolo, ki se lahko razširi do 256 KBytev
- ure realnega časa
- konzolni terminal s kontrolno enoto
- dve diskovni enoti s kapaciteto po 5 MByte s kontrolno enoto
- asinhroni komunikacijski vmesnik (8 linij: 20 mA tokovne zanke)

DELTA 340/40

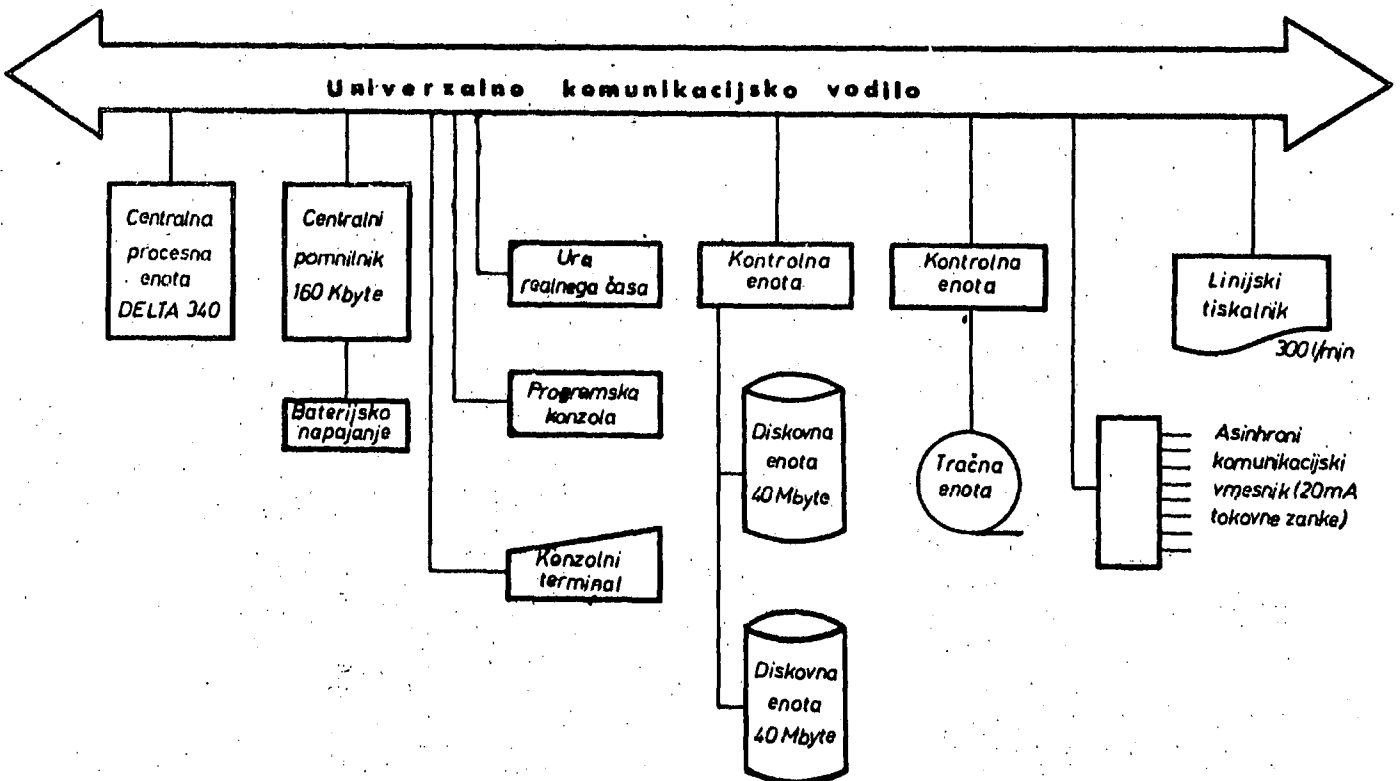
- DELTA 340 centralna procesna enota
- 160 KByte centralni pomnilnik s paritetno kontrolo do 256 KBytev
- ure realnega časa
- konzolni terminal s kontrolno enoto
- enota za baterijsko napajanje pomnilnika
- dve diskovni enoti s kapaciteto po 40 MByte s kontrolno enoto
- ena magnetna tračna enota (1600 b/i, 75 i/s, 9 kanalni zapis) s kontrolno enoto
- asinhroni komunikacijski vmesnik (8 linij: 20 mA tokovne zanke)
- 300 linijski tiskalnik

NAŠTETE STANDARDNE KONFIGURACIJE LAHKO RAZŠIRITE S PRIKLJUČEVANJEM NOVIH VHODNO-IZHODNO ENOT, POVEČANJEM POMNILNIKA IPD.

SISTEMSKI PAKETI DELTA 700/80, 340/80, 340/40 IN 340/5 VKLJUČUJEJO TUDI: OPERACIJSKI SISTEM DELTA/M S PREVAJALNIKI IN APLIKATIVNIMI PROGRAMI, ŠOLANJE V LASTNEM IZOBRAŽEVALNEM CENTRU, POMOČ PRI UVAJANJU PROGRAMSKE OPREME, INSTALACIJO RAČUNALNIŠKEGA SISTEMA IN ENOLETNO GARANCIJO ZA STROJNO IN PROGRAMSKO OPREMO.

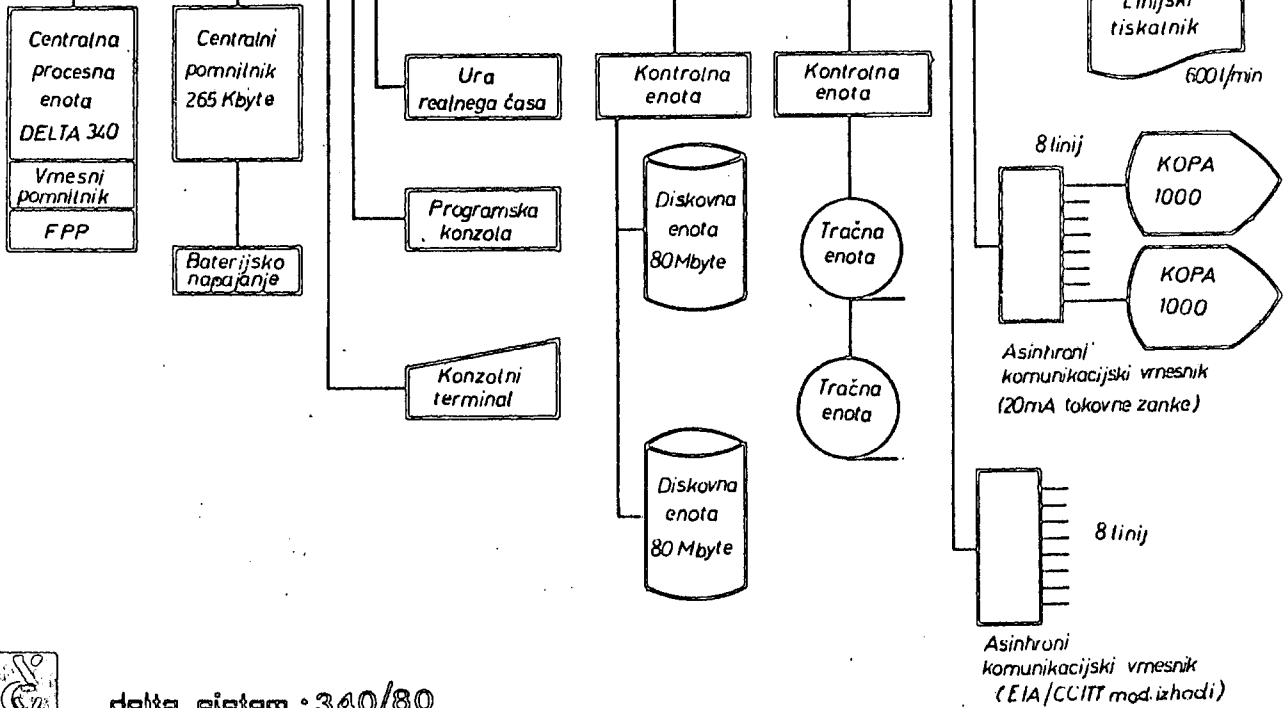


delta sistem : 340/5



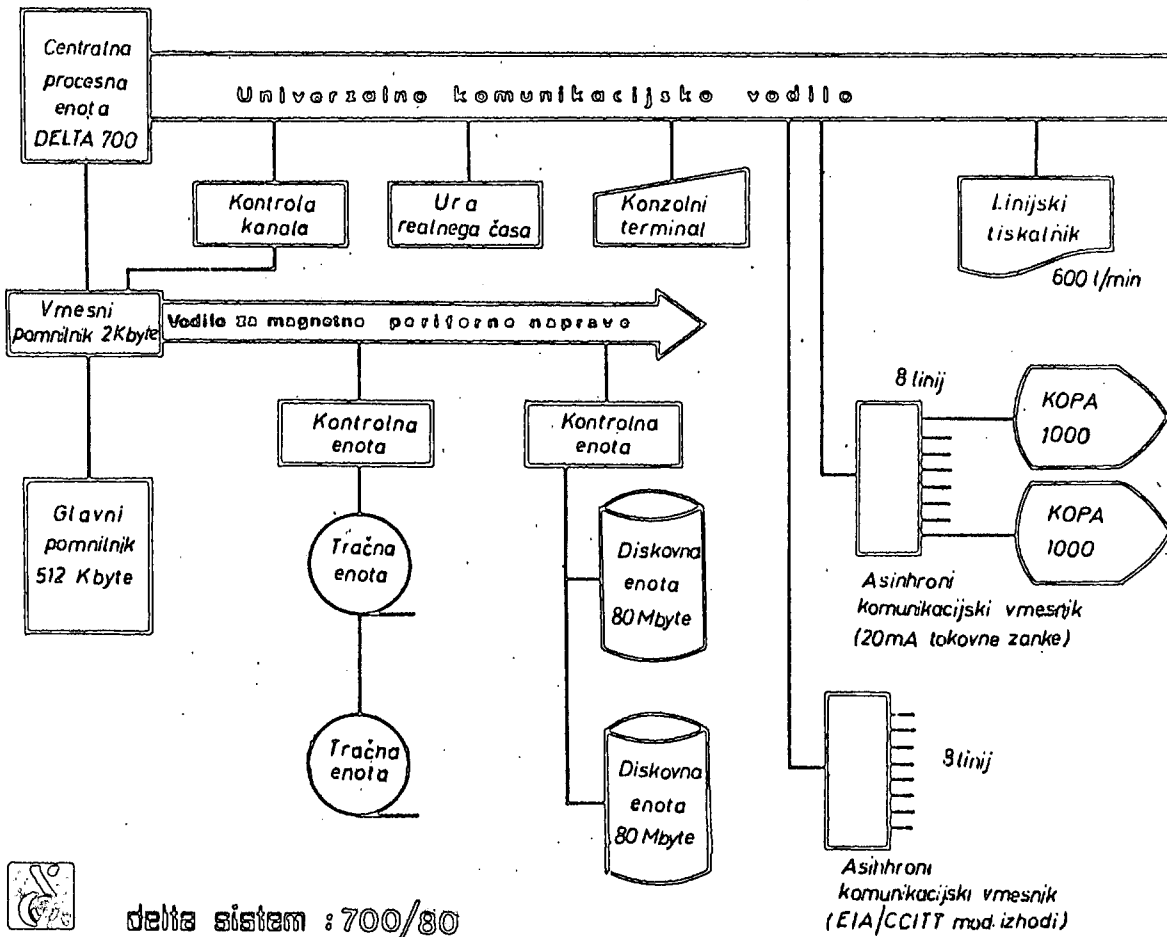
delta sistem : 340/10

Univerzalno komunikacijsko vodilo



delta sistem : 340/80

Univerzalno komunikacijsko vodilo



delta sistem : 700/80

PROGRAMSKA OPREMA DELTA SISTEMOV

Osnovo systemske programske opreme predstavlja DELTA/M operacijski sistem, ki je namenjen za delo v realnem času in časovno dodeljevanje resursov do 256 uporabnikom, ki lahko istočasno uporabljajo sistem. Glavna karakteristika DELTA/M sistema je interaktivnost. Človek in sistem komunicirata preko posebne enote, ki je običajno video terminal. Vsak monitorski ukaz se lahko vnese preko poljubnega terminala, če le uporabnikovo geslo zadošča ustrezni stopnji tajnosti. To pomeni z vidika uporabnika enake možnosti, kot da bi delal sam na sistemu.

Večuporabniško okolje zahteva zaščito med uporabniki samimi, saj bi lahko napaka enega uporabnika povzročila težave vsem drugim. Zaradi tega obstaja med uporabniki zaščita na nivoju programske opreme in na nivoju strojne opreme. Vsak disk je razdeljen v več logičnih področij, od katerih jih vsak uporabnik lahko nekaj uporablja. Praktično to pomeni, da lahko briše samo svoje nize in bere nize drugih uporabnikov, če mu le-ti to dovolijo. Elektronsko pa je zaščiten adresni prostor programov in uporaba instrukcij, ki bi lahko porušile integriteto sistema. Te lahko uporablja samo izvajalni sistem.

Multiprogramiranje je realizirano na nivoju sistema kot celote in na nivoju posameznega terminala. Tako ima lahko vsak uporabnik lastni multiprograming. To je važno predvsem za programerje, saj lahko istočasno razvijajo (prevajalnik, povezovalnik) in testirajo (izvajajo) programe.

Velika hitrost procesorja in perifernih enot ter učinkovito oblikovana programska oprema omogočata gospodarno uporabo vseh komponent DELTA računalnika.

Sistem lahko

istočasno upravlja industrijski proces (visoka prioriteta - realni čas), interaktivne poslovne aplikacije (srednja prioriteta), razvoj novih programov v poljubnih programskih jezikih (standardna prioriteta) in paketne obdelave (nizka prioriteta).

Aplikacijski programi se lahko pišejo v MACRO zbirnem ali enem od višjih programskih jezikov:

- FORTRAN IV
- FORTRAN IV PLUS
- BASIC 11
- RPG II
- COBOL (ANSI 74 standard)
- BASIC-PLUS-2
- PASCAL
- DATARETRIEVE 11

Na tržišču ugotavljamo velike potrebe po kvalitetni komunikacijski opremi, zato posvečamo veliko pozornost prav temu področju. Komunikacijska programska oprema na DELTA/M je eden od poslov, ki se odvija v multiprogramingu in omogoča povezavo z računalniki: DELTA, PDP-11, VAX, DEC-10, DEC-20, CDC-6600, IBM 360/370, UNIVAC-11.

DELTA sistemi so namenjeni splošni uporabi. Zato je v osnovni paket vedno vključena samo tista programska oprema, ki je potrebna vsem uporabnikom. Vsak pa si lahko izbere dodatno systemsko ali aplikativno programsko opremo. DELTA/M namreč ohranja popolno kompatibilnost navzdol z RSX-11/M operacijskim sistemom firme DEC. Ta operacijski sistem je zelo razširjen, zato je tudi ponudba ELEKTROTEHNE, DEC-a in drugih proizvajalcev zelo velika.



PODROBNE INFORMACIJE O NAKUPU DELTA SISTEMOV NUDI ELEKTROTEHNA LJUBLJANA, TOZD ZA RAČUNALNIŠTVO DIGITAL:

LJUBLJANA
Linhartova 62a
tel. (061) 323-585

ZAGREB
Aleja Borisa Kidriča 2
tel. (041) 516-690

BEOGRAD
Karadordev trg 13
tel. (011) 694-537