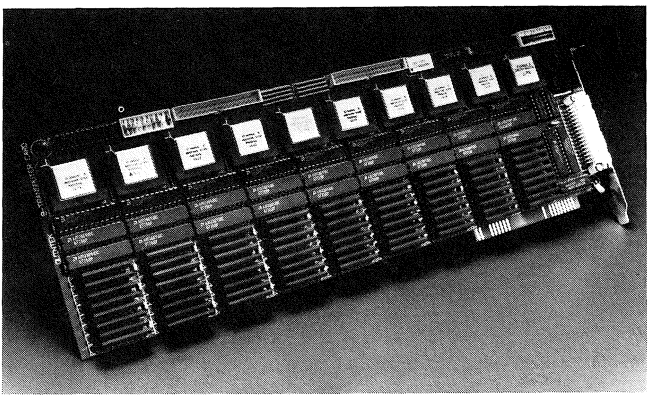
·O·U·G·

occam® user group · newsletter

For all users of occam and the transputer

No 10 January 1989

CONTENTS



Quintek Fast9-T800 Transputer Expansion Board for PCs. See page 55.

NEWS

Change of Occam User Group Chairman

The OUG was formed in 1984 and has grown from the original 7 founders to well over one thousand members. As OUG chairman since those pre-transputer days of 1984 it has been extremely satisfying to see occam evolve from proto-occam, through occam-1 to occam-2 whilst the transputer has progressed from the laboratory to an established component of many commercial products.

I now believe that it is time for change in the OUG committee and consequently Peter Welch will be taking over as chairman from November 1988.

I would like to take this opportunity of thanking the committee members, and in particular Michael Poole, for their support over the last 4 years and to wish Peter all the best in his new role.

Gordon Harp

From the (new) Chairman

From the point of view of its members, the Occam User Group has an infinite productivity index! We charge no fees, the meetings and published proceedings are self-financing, we have no paid administrators, no expenses for committee members, no bank account and no legal existence!!

Nevertheless, we have grown from a gathering of about 60 enthusiasts in September, 1984, into a lively international organisation with a current membership of over 2000. Nine technical conferences have been held, three books published and a regular newsletter distributed. The next two conferences have been planned (book early, or you won't get in!), along with technical workshops emerging from some of our Special Interest Groups (e.g. Artificial Intelligence and UNIX(TM)) As with the design of occam itself and the transputer, the principle of Occam's Razor was firmly applied to the OUG. Only those structures needed to make things happen were created ... and things have certainly happened. Enormous thanks are owed to our founding Chairman, Gordon Harp, for keeping the administration so lightweight, for providing sound guidance and for all the energy expended in enabling the OUG to grow and thrive so well. On behalf of the whole membership, many many thanks, Gordon.

Through unwritten mechanisms, I have become Chairman. [One of these days, when the history of this organisation is written, the grizzly details of this coup will emerge. Basically, I got mugged by the rest of the committee ... and I thought they were my friends!] Anyway, we hope to run the OUG as before. On the other hand, if the membership continues to grow at its present rate, a more formal structure will be needed. At present, the membership list, newsletter and various mailings are funded by INMOS. Eventually, we may have to take this over ourselves and we shall then need a bank account, fees, constitution, auditing,

etc...!! [To manage these overheads, a second generation OUG will be required, with its own "administration processor" cast, of course, in silicon on the same chip. Formal methods will be applied for the specification and implementation of a formal constitution - would the INMOS design team please note.]

Finally and seriously, it seems a good idea to re-state the purpose of the Occam User Group. We are simply the forum for all users of the transputer. We are the only such forum (together with our sibling organisations in Europe, North America, Australasia and Japan) that is officially recognised by INMOS. The word "Occam" in our title does not imply an exclusive interest in the occam processing language. "Occam" refers to the goals of simplicity, security and performance that are derived from the principle of Occam's Razor. "Occam", therefore, encompasses not only the transputer but also all its best applications, languages, support tools and methods.

Peter Welch

From the Editor

I would like to record my appreciation of the work done by Gordon Harp on behalf of the OUG during the last four years. Gordon has been particularly helpful to myself and the previous editors of the newsletter. However, we are not losing Gordon totally as he will continue as a member of the OUG committee.

The past six months has been quite eventful with some companies collapsing, many being created and the most important of all, Inmos, being involved in new ownership negotiations. New products and innovative uses of the Transputer and Occam continue at an ever increasing pace. We can look forward to another interesting year.

Please continue to supply the editor with information on any Occam or Transputer topic. The next copy date is 19th May, 1989.

Derek Paddon, January 1989.

Contributions to the Newsletter

To allow the efficient processing of submitted articles, news, etc., please use the following guidelines.

- Follow the style in this Newsletter.
- Camera ready material is acceptable, provided that it follows the general style of this Newsletter. Pay particular attention to page width and page length. The editor may have to cut and paste your layout to fit part-pages; please bear this in mind.

- email (derek@uk.ac.bristol.compsci) and PC floppy disc material: LaTeX preferred, following the style of this Newsletter, otherwise unformatted ASCII, again being sympathetic to the general style of this Newsletter.

Occam User Group Publications

Michael Poole, Inmos Ltd.

Proceedings of OUG technical meetings 7, 8 and 9 have been published and are available direct from the publisher: IOS, Van Diemenstraat 94, 1013 CN, Amsterdam, The Netherlands. or, in America: IOS, PO Box 2848, Springfield VA, 22152-2848, USA. OUG Newsletters 1-5 are now out of print. Any requests for copies of Newsletters 6-9 should be sent to Claire Williams, INMOS Limited, 1000 Aztec West, Almondsbury, Bristol BS12 4SQ. A directory of members is in preparation and a copy will be sent to all members when it bocomes available. A bibliography is also being put together by Information Services staff at INMOS. Any queries on this should be addressed to Zena Woodley at the INMOS office.

Future of INMOS

Colin Whitby-Strevens, Manager Special Projects, INMOS Ltd.

The following announcement was release by THORN EMI on 13th December 1988:

"THORN EMI and SGS THOMSON MICROELECTRONICS announce that they have reached a preliminary agreement regarding the proposed acquisition of INMOS Limited and its Subsidiaries by the Holding Company of the THOMSON SGS MICROELECTRONICS Group associated with a participation of THORN EMI plc in such holding company. This agreement is subject to approval of the respective Boards of the Parties, their Shareholders and various regulatory authorities."

For information, SGS Thomson Microelectronics (STM) is one of the 'big three' European semiconductor companies, with a turnover of about $1 billion. It was formed in April 1987 by merging SGS and the microelectronics part of Thomson. It is owned 50% by the Italian Government through STET, and 50% by the French Government through Thomson.

The Transputer Development System

Michael Poole, Inmos Ltd.

The product version of the Transputer Development System IMS D700D has now been available for about six months and is currently the most widely used development system for transputer software. Its (nearly) complete implementation of the Occam 2 language,

enhanced interactive user interface and postmortem debugger have been particularly welcomed.

Since the completion of the D700D the INMOS software products development team have been concentrating their efforts on the "stand alone" or "disintegrated" tools. The principal design aims for these have been to enable users to use their own favourite editors, configuration control tools, etc and to be able to have the same tools available on a variety of host workstations including the Sun 3 and MicroVax. Meanwhile design has been proceeding on future generations of development software, and in particular the establishment of interface standards to facilitate the mixing of tools designed and built by a wide variety of people.

The purpose of this note is to encourage readers to let INMOS know what they feel should be the priorities in such developments. In particular, I should like to know what improvements they would like to see in the next release of the IMS D700, the design of which will be frozen very soon. The probable developments include: using a common server interface with the toolset products; allowing the editing of text files in ordinary host text format; allowing attachment of, and execution of programs from, host binary format code files; mixed language programming; enhancement of search and replace operations. Do you want all these features? If so, which would you consider the most important if resource limitations force a partial implementation? Are there other improvements which you consider equally important? Please communicate your feelings to me. (address on the back cover)

Electronic Grapevines

If you are an electronic mail user, you may want to know about two electronic mailing lists, carrying discussions on occam and the transputer. These offer you a mechanism rapidly to distribute information, short papers, programs, problems, even gossip about INMOS, to the sort of people who may be interested. You may even want to read this sort of thing. We even have subscribers from INMOS who can be sometimes be goaded into authoritative declarations.

Each list has distribution points both in the UK and the USA. To join try making contact with the appropriate address:

for the occam mailing list contact		
	occam-request@uk.ac.oxford.prg	(in the UK)
or	occam-request@sutcase.case.syr.edu	(in the USA);
for the transputer list contact		
	transputer-request@tcgould.tn.cornell.edu	(in the USA)
or	transputer-request@uk.ac.oxford.prg	(in the UK).

In case of (extreme) difficulty, resort to Geraint Jones
Programming Research Group
11 Keble Road
Oxford OX1 3QD

Future Technical Meetings

Michael Poole, Inmos Ltd.

The 9th Technical meeting of the Occam User Group will be held at Enschede 3rd - 5th April 1989. Registration forms for this meeting were sent to all members in December 1988. A leaflet accompanying this newsletter gives further information. The organiser is Andy Bakkers, Twente University, PB 217, 7500 AE Enschede, The Netherlands. Tel +31-53-892790.

The 10th Technical meeting of the Occam User Group will take place at the University of Edinburgh on 25th and 26th September 1989. All enquiries about this meeting should be addressed to John Wexler at Edinburgh University Computing Service, The King's Buildings, Mayfield Road, Edinburgh, EH9 3JZ, Scotland.

The OUG Unix and Operating Systems SIGs are holding a one day meeting at Canterbury on 21st February 1989. The OUG AI SIG is holding a two day meeting at Imperial College, London on 17th and 18th July 1989. These SIG meetings have been separately advertised; any queries should be addressed to the SIG chairmen.

Offers from members to organise future meetings are always welcomed by the committee. The OUG itself does not have any funds to subsidise these meetings but can help to put potential organisers in touch with organisations who might be prepared to help with costs. All enquiries should be addressed to the secretary.

North American Transputer Users Group Meeting

Revised Announcement and Call for Papers
April 5 and 6, 1989, Salt Lake City, Utah.

The North American Transputer Users Group will hold its Spring 1989 conference in Salt Lake City, Utah, on April 5 and 6. The conference is tentatively scheduled to begin at noon on the 5th and run through the 6th. The conference will be accompanied by a display of products from several vendors. The sponsors are Brigham Young University, the University of Utah, and Utah State University.

Professor Hoare will be speaking at the University of Utah at the same time as our meeting and has agreed to participate in the users group gathering.

Presentations will consist of contributed long (30 min.) and short (15 min.) papers. Long papers should emphasize solid results of experimental or theoretical work. Short papers may discuss work in progress. Those wishing to contribute should submit an abstract of no more than 600 words (long papers) or no more than 300 words (short papers) by February 1, 1989, to:

G. S. Stiles, Department of Electrical Engineering, Utah State University, Logan UT 84322-4120. stiles@cc.usu.edu, stiles@USU.bitnet.

All presented papers will be published in the conference proceedings.

(Please note the change in the due date for abstracts. Email submissions would be preferred, if at all possible.)

Vendors interested in displaying their products should contact:

Kris Sikorski, Department of Computer Science, University of Utah, Salt Lake City, Utah, 84112. Tel: (801) 581-8579, sikorski@cs.utah.edu.

Information on accommodations and travel will be forthcoming.

Program committee:

Dyke Stiles (Chair), Utah State Univ.

Aurel Cornell, Brigham Young Univ.

Kris Sikorski, Univ. of Utah.

Jean Michel Favre, Tektronix, Inc.

OUG Artificial Intelligence, Special Interest Group

Call for Papers, 1st technical meeting of the OUG AISIG
ARTIFICIAL INTELLIGENCE AND COMMUNICATING PROCESS ARCHITECTURE
17th and 18th of July 1989, at Imperial College, London UK
Keynote speakers will include, Prof. Iann Barron.

The conference organising committee includes:

- Dr. Atsuhiro Goto Institute for New Generation Computer Technology (ICOT), Japan.
- Dr.med.Ulrich Jobst Ostertal - Klinik fur Neurologie und klinische Neurophysiologie
- Dr. Heather Liddell, Queen Mary College, London.
- Prof. Dr. Y. Paker, Polytechnic of Central London
- Prof. Dr. L. F. Pau, Technical University of Denmark.
- Prof. Dr. Bernd Radig, Institut Fur Informatik, Munchen.
- Prof. Dr. Alan Robinson Syracuse University, U.S.A.
- Prof. Dr. David Warren Bristol University, U.K.

Conference chairmen:

- Dr. Mike Reeve, Imperial College, London
- Steven Ericsson Zenith, INMOS Limited, Bristol (Chairman OUG AISIG)

Topics include:

- The transputer and a.i.
- Real time a.i
- Applications for a.i.
- Implementation languages
- Underlying kernel support
- Underlying infrastructure
- Toolkits/environments
- Neural networks

Papers must be original and of high quality. Submitted papers should be about 20 to 30 pages in length, double spaced and single column, with an abstract of 200-300 words. All papers will be refereed and will be assessed with regard to their quality and relevance.

A volume is being planned to coincide with this conference to be published by John Wiley and Sons as a part of their book series on Communicating Process Architecture.

Papers must be submitted by the 1st of February 1989. Notification of acceptance or rejection will be given by March 1st 1989. Final papers (as camera ready copy) must be provided by April 1st 1989.

Submissions to be made to either of the following:

Steven Ericsson Zenith, INMOS Limited, 1000 Aztec West, Almondsbury, Bristol BS12 4SQ, UNITED KINGDOM. Tel. 0454 616616 x513, email: zenith@inmos.co.uk

Mike Reeve, Dept. of Computing, Imperial College, 180 Queens Gate, London, SW7 2BZ, United Kingdom. Tel. 01 589 5111 x5033. email: mjr@doc.ic.ac.uk. Regional Organisers:

- J.T Amenyo Ctr. Telecoms Research, Columbia University, Rm 1220 S.W. Mudd, New York, NY 10027-6699.
- Jean-Jacques Codani INRIA, Domaine de Voluceau - Rocquencourt, B.P.105-78153 Le Chesnay Cedex, France.

- Pasi Koikkalainen Lappeenranta University of Technology, Information Technology Laboratory, P.O.BOX 20, 53851 Lappeenrantra, Finland.
- Kai Ming Shea Dept. of Computer Science, University of Hong Kong, Hong Kong.
- Dr. Peter Kacsuk Multilogic Computing, 11-1015 Budapest, Csalogaiy u. 30-32. Hungary.

Transputers for Industrial Applications II

Belgian Institute for Automatic Control

Due to the success of the 1988 issue, the Belgian Institute for Automatic Control (abbreviated as BIRA) will organize a "Transputers for Industrial Applications II" conference in October 1989. We will show a number of new and revolutionary applications of transputers in signal processing, real-time control, image processing, simulation, telecommunications, computer aided design, functional languages, artificial intelligence, supercomputers, operating systems, ...

If you think you have a useful application of transputer systems, that can be or has been used in industry (or know someone who has), you are invited to present this at the BIRA seminar, which will be held in Antwerp, Belgium during the month of October 1989. Purely academic presentations (I don't deny their value), as well as purely commercial talks will not be considered.

Send an abstract of your presentation to me, either by email, fax or by postal mail BEFORE MARCH 31, 1989.

There will also be a large exhibition of transputer technology (last time we had 14 companies) that can be visited by all participants. Interested companies can already book space in the exhibition room.

We hope to make this seminar even bigger and better than the last one.

Patrick Van Renterghem, R&D Assistant,

Bitnet: pvr@bgerug51.bitnet,

EDU: pvr%bgerug51.bitnet@cunyvm.cuny.edu,

UUCP: mcvax!bgerug51.bitnet!pvr.

Automatic Control Lab/The Transputer Lab, State University of Ghent, Grotesteenweg Noord 2, B-9710 Ghent-Zwijnaarde, Belgium.

Tel: +32 91 22 57 55 ext. 313, Fax: +32 91 22 85 91

TRANSPUTERS†, UNIX† and FUTURE SUPPORT ENVIRONMENTS

An Occam User Group Workshop

The University of Kent at Canterbury
(Tuesday, 21st February, 1989)
(10.00 — 17.00)

Aims: To hear from leading suppliers of software support tools for *transputer* systems about their current and future plans. The technical backgrounds to the various compromises that have been chosen will be presented. It will be an opportunity for users to question those decisions and influence future ones. *INMOS*, *MEiKO*, *Parsys*, and *Perihelion* (plus some others) will be attending.

Questions: How close to "industry standard" UNIX can we get with *transputers*? How important is this? How "inter-operable" will be *transputer* support environments from different sources? How "inter-operable" will be these environments with existing services on SUNs, VAXen, etc ...? What "industry standard" tools (like X-windows) will be available? Can the *INMOS TDS* be opened up to other suppliers of tools? Should multiple *transputers* be used to support the computational load required by sophisticated user-interfaces and tools in the next generation of support environments? ...???

Programme: Formal presentations will be made in the morning session. Participants will be divided into smaller "working groups" in the afternoon. Informal short presentations may be made there, but the aim will be to thrash out the issues on a particular topic and produce an outline report. The reports from each working group will be written up later (by the chair — volunteers are sought!) and published in some form.

Participants: *Active* participants in this workshop are sought. Do not just come to listen! The size of the workshop will be limited to about 40. A fee of £15 will be charged to cover lunch and administration.

"Sponsors": This workshop is organised by the UNIX SIG and Operating System SIG of the Occam User Group. It is also part of a series of workshops and courses being planned through the EEC *COMETT* programme entitled "Training for Transputer Technologies". (*COMETT* = Community Action Programme for Education and Training for Technology.)

Contact: For further details, accommodation, etc please contact :-

Dr P.H. Welch (Tel: 0227-764000 extn. 7695)
Computing Laboratory (Fax: 0227-762811)
University of Kent (Telex: 965449 UKCLIB)
Canterbury (email: phw@uk.ac.ukc)
Kent, CT2 7NF

To apply: Please write to Dr P. Welch and enclose either a cheque for the fee (made payable to UNIKENT) or an official order form.

†TRANSPUTER is a trademark of the INMOS Group of Companies.
†UNIX is a trademark of AT&T Bell Laboratories in the USA and other countries.

MEETING REPORTS

Report on the 9th OCCAM User Group Technical Meeting

The 9th technical meeting was held at the University of Southampton, UK from the 19 to 21st, September, 1988.

The overall impression of the delegates was that the metting was very well organised, enjoyable and technically beneficial. A variety of activities attracted large audiences: produce update report from INMOS; SIG meetings - well attended with quite heated debate at times; some light hearted moments - such as the joke session (reported by Geraint Jones); and the comprehensive technical programme as follows:-

- Virtual Memory Management for the Transputer - P.J. Bakkes
- CDL - A Distribution Language for HELIOS - C.H.R. Grimsdale
- A Concurrent Approach to the Towers of Hanoi - W.D. Crowe and P.E.D. Strain-Clark
- OCCAM2 Implementation of Prolog and its Preliminary Performance - K. Zhang
- The Meaning and Implementation of PRI ALT in Occam - G. Barrett, M. Goldsmith, G. Jones and A. Kay
- Simulation of Gas Pipline Networks - N. Patel, P. Bentley and C. Hughes
- An Extension of the Processor Farm Using a Tree Architecture - S.A. Green and D.J. Paddon
- A Preprocessor to Augment the Description of Occam Processes for Multi-transputer Machines - H. Ohara and H. Iizuka
- Randomised Routing:"Hot Potato" Simulations - X.M. Qiang and S. Turner
- Mapping a Process Network onto a Processor Network - F.C.M. Lau and K.M. Shea
- Support for Occam Channels via Dynamic Switching in Multi-Transputer Machines - P. Jones and A. Murta
- The Computing Tower: A Supercomputer for Real-Time Simulation of Continuous Systems - P.Van Renterghem
- The Application of Transputer and Occam to an Industrial Energy Management System - A. Sinclair and P. Kelly
- Fast Prototyping of Architectural Designs Using Transputers - D. Skillicorn

- GECKO: A Graphical Tool for the Modelling and Manipulation of Occam Software and Transputer Hardware Topologies - M. Stephenson and O. Boudillet
- A State-of-the-Art Radar Pulse Deinterleaver - S.P. Turner, R.D. Betòn and C. Upstill
- Techniques for Rendering Solid Objects on Processor Farm - P.Dew, N. Holliman, D. Morris and A. De Pennington
- A Prototype Simulator Output Movie System Based on Parallel Processing Technology - N. Carmicheal, D. Hewson and J. Van Der Vorst

Report on Southampton meeting of the Hardware SIG

Denis Nicole, University of Southhampton.

As a member of the host University, I was asked to stand in for Tony Gore and chair the Hardware SIG. Several short presentations were made:

Mike Moore (Southampton University) described experiments with transputers chilled in liquid Nitrogen. He obtained T800 processor clock speeds of 45MHz and link speeds of 40MHz with no special precautions. A paper is available from the Southampton transputer centre.

Klaas Wijbrans (Twenty University) described the Twenty LINX backplane system for control and industrial applications of reconfigurable transputer arrays. He has a paper available.

M Woods (Bristol Transputer Centre) discussed the trials and tribulations involved in building a reliable fibre optic transputer link system. He did, however, extol the virtue of using optical fibres to propagate reset signals.

Denis Nicole (Southampton University) described a student project building a transputer X25 interface and JANET software, and outlined modifications to Sension IBM link boards for Inmos compatibility about which a note can be obtained from the Southampton transputer centre.

Adrian Lawrence (Oxford Microprocessor Centre) described a Fast Data Acquisition module including A/D converter and corner turning memory.

Pat Pope (Paradox Systems) described a proposed transputer SCSI controller module.

A Garrett (A G Electronics) described the use of the Transtech IBM PC hard disk system with TDS. The disk is interfaced to the PC as an ordinary MSDOS block device. There is an additional MSDOS file system implemented on the disk board's transputer, accessible through the transputer links. The TDS, running on a B004 or lookalike, is modified by

adding an extra outer layer which directs file accesses across a link to the disk board rather than to the PC. This avoids the PC entirely, and dramatically speeds disk access.

Roger Shepherd (Inmos), in answer to a question by Jon Kerridge, explained some methods of running transputers at microwatt powers. A substantial reduction in power results from running the clock at one tenth speed. A further improvement accrues from operating the component with a 3V power supply. This has the effect of disabling the transputer's TTL level converters.

STOP PRESS

For my sins, I have now been asked to take over the hardware SIG permanently. I will be at the next OUG meeting, but in the mean time can be contacted by Email as DAN@UK.AC.SOTON.ESP.V1 or by telephone on +44-703-559122 ex 3367 or +44-703-787167. I am not on CIX.

UNIX SIG report

Peter Welch, University of Kent.

About 30 people attended this SIG which enabled some useful dialogue to take place. The discussions centred around what features of UNIX were becoming available to support both software development and applications on transputer systems. The MEiKO and Helios approaches were explained in some detail.

It became clear that this was a topic which would sustain a much longer debate and for which there would be wider interest. There are also several other contenders in the field (e.g. IDRIS and TROLLIUS), as well as the issue of the openness of the TDS and the whole question of portable software tools and future transputer support environments.

Accordingly, it was decided to hold a one-day workshop on these matters in conjunction with the Operating System SIG. This will take place on the Tuesday, 21st February, at the University of Kent. Details are announced elsewhere in this newsletter.

Report on the Formal Techniques Special Interest Group

R.P. Stallard, Racal-Milgo Limited, Bartley House, Station Road, Hook, Hampshire, RG27 9PE.

The meeting at Sheffield spent much of its time considering Specification Techniques. Peter Strain-Clark of the Open University gave a preview of his talk on C.A.P. (Communicating Asynchronous Processes) which is a derived version of C.S.P. specifically aimed at easily mapping to occam. There was hope that a mapping of C.S.P. to C.A.P. was straightforward and that the step of generating occam from C.A.P. should be fairly easy too.

Z seems to be growing in favour by the minute, there was news of a fairly mechanical type checker for Z and of the much awaited Z handbook (edited by Mike Smiley, published by Prentice Hall). However, what is really needed is a method of more automation or integration of specification techniques and programming language.

Perhaps the most exciting news was that concerning the MALPAS and SPADE ESPRIT projects, for which RSRE is a major UK participant (contact J. Collier at RSRE). This system will verify that a formal specification is satisfied by a program written in 'C' or Algol or FORTRAN. There are some limitations on the use of these languages in the form of subsets but this is a major step forward. At that point discussion moved onto the subject of whether conformance to a formal specification is what it is all about. The emphasis must shift from getting a program to properly match a spec to making the specification properly match the requirements. There seems to be little prospect of a simple solution to this problem, it is desirable to have a highly flexible specification language which allows an engineer to specify any sort of system. This in itself does not aid an automated system into judging whether a specification describes all possible 'sensible' solutions. Humans make as many mistakes leaving out special conditions in specifications as they do in code. The importance and brevity of a formal spec assure it is more likely to be (manually) a correct statement of the requirements.

There was some news about tools frequently used to implement Formal Techniques. ML has been successfully moved to the Transputer and runs impressively fast (Jon Kerridge, Sheffield University). Edinburgh are in the process of writing a Parallel form of ML. At Inmos David Shepherd is working with HOL (Higher Order Logic) for formal proofs in hardware design. Together with work by Mike Gordon at Cambridge in Theorem Proving, the use of HOL is becoming more widespread.

There was disappointingly little news concerning occam itself. Everybody continues to be happy with the TDS from a formal technique point of view. It was merely mentioned that it takes a cautious view of program constructs when doing rigorous checks and may therefore ban legitimate but dodgy constructions. I hope that somebody out there will take the 'unbundled' TDS and start integrating it with specification languages, transformation systems and the like. Now that 'standard' programming languages are available on the Transputer and 'occam' is far too often relegated to a footnote in articles and documentation concerning it, it is imperative that the specific 'formal method' benefits of using occam should be exploited.

There was concern expressed about the future of this Special Interest Group, my current feeling is that there is not yet sufficient commercial take up of the ideas for a more focussed group to emerge. Most of the people who contribute are from the academic fraternity, and they manage to keep pretty much in touch outside these meetings. Another important role is to get people interested in the area. The closest organization to us is the British Computer Society F.A.C.S. Special Interest Group (Formal Aspects of Computer Science). They hold regular workshops, seminars and conferences and send out a wealth of information in newsletters. You are most welcome to join, the Chairman's address is as follows :

Dr. D. J. Cooke, Department of Computer Studies, Loughborough University, Ashby Road, Loughborough, Leicestershire, LE11 3TU. Tel 0509-222676.

I have recently come across a book giving a broad overview of Specification Techniques : 'Software Specification Techniques' Edited by N. Gehani and A.D. McGettrick, published by Addison- Wesley which could be a useful introduction to the area although not related to parallel systems. Another book which I have found useful, but much more technical is 'Fairness' by Prof. Nissim Francez, published by Springer-Verlag which delves into the difficult areas of non-determinism, priority and other topics embracing 'fairness'.

When is a transputer not a joke?

The transputer joke is an art form developed by Roger Shepherd (he of INMOS) from his old favourite:

> A man with a transputer on his head goes to see his doctor.
>
> "What appears to be the problem?" asks the doctor.
>
> "Well," says the transputer, "it all started with a lump on AD26."

The panel session at the 9th occam user group technical meeting at Southampton was asked whether they knew any good transputer jokes; none being forthcoming, Roger was pressed to illustrate the style:

Q: What goes eighty-three-Thump, eighty-three-Thump?

A: A transputer with a wooden pin.

and Tony Hey suggested prizes of bottles of wine for the best transputer jokes submitted by the end of the meeting. The winners are among the following, which are some of the printable ones submitted:

Q: What tool do you use to debug a transputer?

A: A hammer.

(I think that might be a refined version of 'To him who has only a hammer in his toolkit, everything looks like a transputer.')

Q: What did the big transputer say to the little transputer?

A: Nothing; they were deadlocked.

Q: What has a transputer in common with a Russian submarine?

A: They both go faster under water.

(See published results, with limited circulation, about running transputers under liquid nitrogen.)

Q: What do you call long hair on a transputer?

A: Dreadlocks.

Q: What has four links and flies?

A: A transputer bug.

Q: How many occam programmers does it take to change a lightbulb?

A: You mean, of course, 'how many lightbulbs can an occam programmer change at once?'

Q: How many C programmers does it take to run a 16-bit transputer?

A: 65537: one to write the program, and 65536 to keep a watch on each separate memory location.

a: What is the difference between a transputer and a post box?

b: I don't know, what is the difference between a transputer and a post box?

a: Well, I'm not going to ask you to pass my messages on then.

Q: What is the difference between a null transputer and a Meiko box?

A: One is a chip full of SKIPs, the other is something else.

Q: What do you get if you cross a fruit loaf with a working network of transputers?

A: Concurrent buns.

Q: What is the difference between the Channel Tunnel and a transputer?

A: They both send things across a channel, but the transputer doesn't like running under C.

Perhaps it was the way David May told them.

I seem to remember that there were a number of contributions to *101 Uses for a Dead Transputer*: high-tech thumb-tack; curry-comb for programmer's beard; that sort of thing.

To my shame I forgot to make a note of who won, but then I do not suppose they mind that as much as they would have minded had they not received the wine.

(Recorded by) Geraint Jones
Programming Research Group
11 Keble Road
Oxford OX1 3QD

LETTERS TO THE EDITOR

Members seem to be very reserved about writing to the editor, however this reserve does not extend to open letters and articles posted on email. Therefore, this issue contains articles picked up from email sources in addition to the traditional letters to the editor.

Carefully scheduled selection with ALT

Geraint Jones, Programming Research Group, 11 Keble Road, Oxford OX1 3QD

The Summer 1988 edition of the oug newsletter contains an article[2] by Alan Chalmers which devotes about forty lines of occam and six of transputer assembler code to an obscure (and, as it happens, unsuccessful) attempt to make a 'fair' selection with an ALT. This is a well-studied problem to which there are a number of very much tidier known solutions; the earliest reference to coding a fairish ALT in occam upon which I could readily lay a hand was [5].

What makes an ALT fair?

Briefly, the problem is that the process

```
WHILE busy
  ALT i = 0 FOR n
    c[i] ? x[i]
      P(i)
```

makes potentially 'unfair' selections. Suppose, for simplicity, that the process $P(i)$ uses only the corresponding $c[i]$, and no other $c[j]$. If one of the channels, say $c[0]$, is driven by a process which makes it ready again before $P(0)$ has completed execution, then it is permitted by the semantics that the input from $c[0]$ will be selected every time. (In fact, this happens to be guaranteed by the transputer implementation.)

For those who have not come across the problem before: a frequently asked question is 'why is the occam ALT not a fair one'? Roughly speaking, the problem is that fairness cannot be a static property of the ALT construct. An ALT makes just one selection, and the best that you could hope for in the way of fairness is that the selection is reasonably even-handed. A long sequence of independent executions of an even-handed randomised ALT might yet be unfair, although with a very low probability. Fairness is a dynamic property of a sequence of selections made by the same construct. That, incidentally, is why there have to be state variables *outside* the loop in each of the programs which are suggested in this article.

Chalmers codes up something like

```
INT favourite :
SEQ
  favourite := 0
  WHILE busy
    INT how.many :
    [n]INT available :
    [n] BOOL selected :
    SEQ
      how.many := 0
      SEQ i = 0 FOR n
        IF
          ... c[i] is ready -- this is not coded in occam
            SEQ
              available[how.many] := i
              how.many := how.many + 1
          ... c[i] is not ready
            SKIP
      --
      --  how.many is the number of channels that are ready
      --  [available FROM 0 FOR how.many] are the indexes,
      --           in increasing order, of those that are ready
      --
      IF
        how.many = 0              -- none ready, choose any one
          SEQ i = 0 FOR n
            selected[i] := TRUE
        how.many > 0
          SEQ
            IF
              (favourite + 1) < how.many
                favourite := favourite + 1
              (favourite + 1) >= how.many
                favourite := 0
            SEQ i = 0 FOR n
              selected[i] := (i = favourite)
      ALT
        selected[i] & c[i] ? x[i]
          P(i)
```

(I have left out the assembly code to make it clearer what happens.)

It is difficult to tell what he intended, but it is plain that something has gone wrong here. Suppose that n is at least three, and that the following sequence of events happens:

`favourite := 0`
$c[0]$, $c[1]$, $c[2]$ become ready
`how.many, [available FROM 0 FOR 3] := 3, [0, 1, 2]`
`favourite := 1`

```
c[1] is selected, and is no longer ready
how.many, [available FROM 0 FOR 2] := 2, [0, 2]
favourite := 0
c[0] is selected, and is no longer ready
c[0], c[1] happen to become ready again
how.many, [available FROM 0 FOR 3] := 3, [0, 1, 2]
favourite := 1
...
```

Guess when $c[2]$ is chosen. I would not like to swear that I have got Chalmers' code right: it certainly was not immediately obvious what it did, which is partly the point of this present article.

Coding a fairer ALT

So, how should one code up a fairer selection? A feasible strategy is to rotate the priority of the branches through a fixed sequence:

```
INT favourite :
SEQ
  favourite := 0
  WHILE busy
    SEQ
      PRI ALT index = favourite FOR n
        VAL shifted IS index \ n :
        c[shifted] ? x[shifted]
          P(shifted)
      favourite := (favourite + 1) \ n
```

which guarantees that any of the channels will be serviced after at most n passes through the loop. Even if some of the channels are saturated, this scheme will guarantee about $1/n$th of the bandwidth to any channel that needs it. (That is the solution suggested by [5].)

Of course, if you do not like all those modulo operators inside a loop, you could always do it by indirection through an array that keeps track of a permutation of the branches:

```
[n]INT perm:
SEQ
  SEQ i = 0 FOR n
    perm[i] := i
  WHILE busy
    SEQ
      PRI ALT index = 0 FOR n
```

```
        VAL shifted IS perm[index] :
        c[shifted] ? x[shifted]
          P(shifted)
      INT first:
      SEQ
        first := perm[0]
        SEQ i = 0 FOR n - 1
          perm[i] := perm[i + 1]
        perm[n - 1] := first
```

which has the same effect. Of course, this is rather silly, because if you were worried about how long a few modulo operations were going to take, you ought to be frantic about all that copying of the *perm* array. Roger Peel suggested the following altogether more sensible coding:

```
[2 * n] INT perm :
INT favourite :
SEQ
  SEQ i = 0 FOR n
    SEQ
      perm[i]     := i
      perm[n + i] := i
  --  0 <= i < 2n   ==>  perm[i] = i \ n
  favourite := 0
  WHILE busy
    SEQ
      PRI ALT index = favourite FOR n
        VAL shifted IS perm[index] :
        c[shifted] ? x[shifted]
          P(shifted)
      favourite := perm[favourite + 1]
```

which simply precomputes the results of all of the modulo operations. The cost is in the extra storage required for the array.

Michael Goldsmith suggests yet another way of implementing the same selection strategy, guaranteeing $1/n$th of the bandwidth to any saturated channel. It is essentially the same as Roger Peel's solution, but without the need to keep the precalculated remainders:

```
INT split :
SEQ
  split := 0
  WHILE busy
    SEQ
      PRI ALT
        ALT index = split FOR n - split
          c[index] ? x[index]
```

```
            P(index)
        ALT index = 0 FOR split
          c[index] ? x[index]
            P(index)
      split := (split + 1) \ n
```

These schemes can be adapted to implement different scheduling strategies, for example:

```
INT favourite :
SEQ
  favourite := 0
  WHILE busy
    PRI ALT index = favourite FOR n
      VAL shifted IS index \ n :
      c[shifted] ? x[shifted]
        SEQ
          P(shifted)
          favourite := (shifted + 1) \ n
```

which gives the lowest priority next time around to the branch which was selected this time. That has the advantage over the previous scheme that if p of the n channels are carrying high volumes of traffic, it will tend to give $1/p$th of the bandwidth to each of them.

Harold Curnow suggested another more traditional looking way of getting this stronger guarantee:

```
[n]BOOL mask :
BOOL reset :
SEQ
  SEQ i = 0 FOR n
    mask[i] := TRUE
  reset := FALSE
  WHILE busy
    PRI ALT
      ALT index = 0 FOR n
        mask[index] & c[index] ? x[index]
          SEQ
            P(index)
            mask[index] := FALSE
            reset := TRUE
      reset & SKIP
        SEQ
          reset := FALSE
          SEQ i = 0 FOR n
            mask[i] := TRUE
```

which is much more in the spirit of what Chalmers was trying to do. It guarantees no more than $p-1$ missed opportunities between successive selections of a given one of p saturated channels.

If all you want is non-starvation, and the $1/n$th bandwidth guarantee will do, my favourite coding is simpler:

```
INT favourite :
SEQ
  favourite := 0
  WHILE busy
    SEQ
      PRI ALT
        c[favourite] ? x[favourite]
          P(favourite)
        ALT index = 0 FOR n
          c[index] ? x[index]
            P(index)
      favourite := (favourite + 1) \ n
```

This keeps the time overheads down to a single modulo operation, and an extra branch in the ALT; and the space to a single INT variable. It relies on the idempotence of ALT – the fact that it does not matter much if a branch appears twice. (Bill Roscoe[6] uses this one, but attributes it to a message from me to an electronic mailing list; you will also find it in [4].)

The moral of the tale

Well, first of all, do not start from the assumption that you need to write your own scheduler; start from the required behaviour of the program, and you will probably find that it can be done cleanly with the existing scheduler.

Secondly, the longer and more complicated your program and the woollier your idea of what it is meant to do, the less likely you are to get it right. This is particularly true of schedulers, as witness the embarrassing confusion which is the transputer's own scheduler, a sorry tale related in part in [1].

One of my colleagues (a year or two older than me, perhaps) said that there was a time so far back I would hardly remember it (he meant the sixties, I think) when the people building compilers and operating systems would insist on writing their systems in assembler. It was one thing for computing scientists to write toy systems in high level languages like BCPL (well, this was the sixties) but real systems had to be written in assembler, or even better in signed binary. My colleague thinks that that battle has been won, but then he does not know about Helios, and transputer-code schedulers.

Acknowledgement

I am grateful to a number of electronic-mail correspondents who contributed to this article, although since they have not all had an opportunity to correct my mistakes I lay a personal claim to those.

References

[1] Geoff Barrett, Michael Goldsmith, Geraint Jones, Andrew Kay, *The meaning and implementation of PRI ALT in occam*, in "occam and the transputer – research and applications" (Proc 9th technical meeting of the occam user group), ed. Charlie Askew, Southampton, 1988.

[2] Alan Chalmers, *Useful Titbits*, oug newsletter, pp.15–19, Summer 1988.

[3] INMOS Ltd., *Communicating Process Architecture*, Prentice Hall International, 1988.

[4] Geraint Jones, Michael Goldsmith, *Programming in occam2*, Prentice Hall International Series in Computer Science, 1988.

[5] David May, *Communicating Processes and occam*, draft of April 1986, subsequently INMOS technical note 20, subsequently in [3].

[6] Bill Roscoe, *Routing Messages Through Networks: An Exercise in Deadlock Avoidance*, in "Parallel Programming of Transputer Based Machines" (Proc 7th technical meeting of the occam user group), ed. Traian Muntean, Grenoble, 1987.

The following is a more detailed version of Harold Curnow's solution, as out-lined by Geraint Jones.

Deadlock and Round-robin:- More useful titbits

Harold J Curnow, Advanced Technology Group, CCTA

This is in response to Alan Chambers' article in the Occam User Group Newsletter 9. I have already responded to a reponse from Geraint Jones on the 'Fair ALT' problem on the occam net, and a version of this article has been posted to the transputer net.

My TEDIOUS demonstration system, running under TDS, consists of a keyboard fan-out process (1 in, n out), a screen-driver multiplexor (n in, 1 out), and a floating population of intermediate processes, some of which display in windows on the screen through the screen-driver. The following are the rules I have devised for the intermediate processes to ensure that the whole system will terminate on a single keystroke command.

1. Every channel type has a defined 'abort' signal.
2. Within each process, every input is tested for 'abort'.
3. When a process detects 'abort' on any input channel:
   (a) it does not read that channel again
   (b) as soon as it conveniently can it ceases all activity, except for a routine in which IN PARALLEL it: sends 'abort' on all output channels (except to screen) reads all input channels until 'abort' has been read
   (c) it sends 'abort' to the screen and terminates.

Making the screen a special case helps diagnosis. The screen multiplexor follows a different plan, simply counting the number of 'abort's received, and then terminating.

An Example

The following is a process which can communicate with another copy of itself, if necessary:

```
PROC active(CHAN OF INT inchan, outchan, up, down)
            -- inchan from kbd, outchan towards screen)
            -- up from, down to, other active procs
  BOOL run :
  INT key, char :
  VAL abort IS ft.refresh :  -- 'ESC' key
  SEQ
    ... initialise run:=TRUE; key:= char:= 0;
    WHILE run
      SEQ
        ALT
          inchan ? key
            SKIP           -- it's actually more complex of course
          up ? char
            SKIP
        -- end ALT
        IF
          (key = abort) OR (char = abort)
            run := FALSE
          (key <> abort) AND (char <> abort)
            SEQ
              ... rest of action
      -- end SEQ
    -- end WHILE


    PAR  -- start of termination routine
      down ! abort
```

```
      WHILE char <> abort
        up ? char
      WHILE key <> abort
        inchan ? key
    -- end PAR
    outchan ! abort  -- close window
  --  end SEQ
: ---- end of PROC
```

If rule 2 seems to impose too great an overhead, it may be possible to treat groups of inputs as "messages", and only test each message.

Round Robin Scheduling

The trouble with an 'obvious' numerical rotation round robin is that if channel B follows channel A in simple sequence, then if there are 10 channels, it will follow it in 9 out of 10 rotated sequences. If A is 100busy, B will get only 1/10th of the cake. The following is a simplified version of the multiplexor from my screen windowing system. It has been posted to the occam net and has found some favour there. [Close-down mechanism not shown].

```
VAL n IS number-of-channels :
[n] CHAN OF BYTE c :
[n] BOOL mask :
[n] BYTE x :
BOOL reset, busy :
SEQ
  ... busy := TRUE; reset := FALSE; mask[] := TRUE
  WHILE busy
    PRI ALT
      ALT i = 0 FOR n
        mask[i] & c[i] ? x[i]
          SEQ
            mask[i] := FALSE    -- incoming message closes
            reset := TRUE       -- the door behind itself.
            P(i)                -- appropriate processing.
      -- end ALT
      reset & SKIP              -- if any doors are closed
        SEQ                     -- and no messages waiting.
          reset := FALSE
          SEQ i = 0 FOR n
            mask[i] := TRUE     -- reopen all doors.
    -- end PRI ALT
```

This will not let any channel have a second go until there are no other channels waiting for a first go - sort of round robin but with some (inevitable?) quantum uncertainty of exact

sequencing.

Harold J Curnow, Advanced Technology Group, Central Computer & Telecommunications Agency, Riverwalk House, 157-161 Millbank, London SW1P 4RT. Tel +44 1 217 3209

email: CCTA452%GB.GOLD-400.hmg.idem.ccta452@uk.ac.ucl.cs

or : CCTA452@GB.GOLD-400.hmg.idem.ccta452

What makes Transputer interesting

From: Jerry DeLapp <jxdl@gov.lanl>, Los Alamos National Laboratory

In article <5255@cbmvax.UUCP>, daveh@cbmvax.UUCP (Dave Haynie) writes: in article <6048@lanl.gov>, jxdl@lanl.gov (Jerry DeLapp) says: The transputer is fast (about the equivalent of a vax or sun 3 now, and getting faster).

DH:-

The comparison was with a 68030 or 80386, not a vanilla 68000. For integer processing speed, a 68000 isn't quite a VAX 750. The only T800's I've seen benchmarked IN A SYSTEM process integers in the VAX 780-785 range, like a medium performance 68020 system (which is just what a smaller Sun 3 is). 68030 systems outperform 8xxx VAXen in most integer benchmarks.

JD:-

No, the question was what made it interesting, not whether it was faster than the 68030. The 68030 was just used as an example. The transputer's speed (I agree, about equal to a 780 on integer ops, but much dependent on memory technology used with the transputer) is plenty fast enough to be "interesting" (at least to me). Besides, try to plug a 68030 into the hole for a 68000. You'll get smoke. Most transputer systems can be upgraded by just plugging in newer, faster chips.

DH:-

The communications are very fast, have very low startup overheads, and operate without any need of the CPU after setup.

Any DMA driven communications channel will operate without CPU intervention after setup.

JD:-

I agree, but I haven't seen one discrete implementation that can comes close to the transputer in terms of low setup time. Also, why implement it yourself if you don't have to, and can't beat the setup times to boot.

[stuff about FP unit and context switching deleted].

DH:-

The transputer is RISC technology. The small instruction set means that it's fairly easy to port compilers to it (although INMOS seems to be real stodgy about realizing that the real world wants C and FORTRAN).

The Transputer isn't RISC at all. About the only thing it has in common with true RISC-methods CPUs is that it's rather small. There aren't any registers; RISC chips typically have a minimum of 32, some close to 200. A good portion of Transputer instructions are slow, microcoded instructions; RISC chips use hardwired instructions that execute in or near single cycles. Transputers don't have cache memory; all RISC chip COUNT on caches. Transputers don't have memory management, which is crucial to running protected operating systems.

JD:-

Excuse me, but I think you're confusing implementation with theory. RISC means: Reduced Instruction Set Computer. In that context, the transputer certainly qualifies as RISC. Register counts, caches and memory management have nothing to do with whether a chip is "true RISC." They are just implementation details that might make one chip more attractive than another. I agree that the absence of virtual memory management support makes the transputer less attractive for implementing large multi-user systems, but it certainly doesn't eliminate it completely.

(BTW: No flames please. I know the RISC/noRISC debate belongs in comp.arch)

[remaining inane comment about transputers only being good as sattelites to 68xxx systems or as loosely coupled systems deleted]

Actually, my experience has been that the transputer works best in tightly coupled systems, not loosely coupled. Still, it's brought loosely coupled systems into the realm of affordable reality. Hence, most of the discussion in this newsgroup centers on how to make loosely coupled systems work well.

Remember when drawing comparisons between the transputer and the 68xxx 803xx and others, that you're discussing differences between chip series that have been around for many years vs. something "new and different". In terms of progress and improvements, I

think that INMOS is much faster than either Motorola or INTEL in addressing problems with and making improvements in their product. (It took INTEL 8 years to finally build a CPU worthy of respect. I've always respected the 68000 series).

C compilers

From: J.Wexler@uk.ac.edinburgh

Here follows the first (and perhaps only) report on my investigations into C compilers. Many thanks to all who responded to my request for information.

I should have asked, but I didn't, what kind of model of concurrency is provided by each implementation: occ¡am-like, Unix fork-and-join, Ada tasking, ... ?

I will summarise below what I think I have learned. I am not sure that I always understood correctly what people were telling me, so I will append extracts from the original messages that I received.

Patrick van Renterghem has already published much of the information through this group, and I cannot summarise it or present it better than he has done.

Much of my own information is incomplete or unconfirmed. This is indicated by phrases such as "believed to be" and "I suppose". I'd be grateful for any more details or corrections which you can supply.

John Wexler,

Edinburgh Concurrent Supercomputer Project.

Penguin, Pentasoft and Unidot are the same. They have only a very few "compiler escapes" for low-level use of Transputer features - i.e., below the level at which occam operates, and more like the things one could do in assembler. These are not the sort of things that would normally be recognised as "language extensions". You think and program in "ordinary" C. You aren't expected to use the "compiler escapes" either, because MACROS are provided which use the "escapes" to do all the useful occam-like things.

Cornell's Trollius uses this compiler. They have written a library of functions to handle concurrency, and these functions use the macros which in turn use the compiler escapes.

Logical Systems and Definicon compilers are the same. It has a very good reputation for performance. It uses a library of functions to support concurrency; the language itself is standard and unextended. However, the compiler recognises many of the special functions as intrinsics, and generates in-line code for them. There are actually TWO sets of functions, to support two different models of concurrency. One gives an occam-like style (PAR, PRI PAR, ALT, etc.), and the other is more like forking and joining with semaphores and so forth. I don't know what happens if you mix calls on the two sets of functions. Another source told me that "the Definicon compiler does have language extensions", but I suppose

that he was referring to the special intrinsic functions. I believe this compiler is marketed by Micropar (who?). It is a cross-compiler, supplied as C source code to run on a system of your choice. It does not generate modules compatible with Inmos' linker.

3L parallel C has no syntactic extensions, so programs are lint-checkable. It does have a special type for channels, "but it's really just a pointer". Concurrency and message-passing are handled by a library of special functions. The mechanism distinguishes two kinds of process, light- and heavy-weight. Light-weight processes share code and data space (but not stack), and I suppose that they are roughly the same as the things you would group under PAR in occam. Heavy-weight processes seem to be more like the things you control with PLACED PAR: they have to be configured by a special language outside C. The compiler comes with a nice "flood-fill loader" which will rapidly load a copy of a piece of code onto every processor in any network, automatically supplying all the necessary code to implement a task farm. This means that you don't have to program a task farm yourself, and your compiled code is usable without modification on any configuration on Transputers.

This compiler seems to be used by lots of people, and is marketed by Microway. Microway also sell the 3L(ex Lattice Logic) straight C compiler (see below), which causes some confusion. I have heard that they even sell yet another version of C, but I don't know whether that is true, and I certainly don't know which one.

3L also do a straightforward non-parallel C compiler for Transputers. This is quite distinct from their parallel C. Originally it was a Lattice Logic product. Inmos' C is this one.

Renishaw Controls have written a C compiler which is used on Meiko systems. This is another more or less standard C which uses a library of functions to provide control of channels and concurrency. It is also usable in a style favoured by Meiko in which one writes straightforward sequential procedures in C, and one calls them from a main program written in occam and called a harness. You can write your own harnesses, but the good news is that you can get pre-written harnesses which implement useful general styles of concurrency such as task-farming.

Parsec is a Dutch company who have a parallel C compiler which is marketed by Parsytec (who are not the same people at all). The language is called Parallel C, and it really does have language extensions. It includes all of K&R C, with H&S extensions, but it also embodies the occam concurrency features in the language. This seems to be the only implementation which takes such a bold line. This compiler comes with a loader which automatically analyses any network of Transputers while loading is going on, and leaves a complete "map" of the network in the memory of every processor where the application program can find it.

I have no information about Norcroft.

I have not heard of any other C compilers. I haven't followed up the story of who uses or sells whose compiler and what names or badges they attach to the compilers. There is still a lot of confusion to be cleared up in that area.

Several contributors included a disclaimer such as "Any statement appearing here repre-

sents my own opinion and should not be construed as any sort of official position or opinion of my employer."

Steve Otto (Southampton) OTTO@UK.AC.SOTON.ESP.V1 writes: I think that the Logical Systems C compiler and the Definicon "parallel" C compiler are one and the same....Parasoft at Caltech would know for sure. This compiler gives one the comm channels via subroutine calls.

Moshe Braner, Cornell Theory Center, 265 Olin Hall, Cornell University, Ithaca, NY 14853. (607) 255-9401 (Work) <braner@tcgould.tn.cornell.edu> (INTERNET), <braner@crnlthry> or <braner@crnlcam> (BITNET).

writes: Penguin, Pentasoft, and (now) Unidot are all the same: This compiler is based upon the Unidot modular compiler setup, with a transputer code generator, etc added by Pentasoft (formerly called Penguin). Unidot recently bought it back from Pentasoft. This compiler does not support the transputer features in any way that comprises language extensions except for a few that may be called 'compiler escapes':

ABCregs(a,b,c) (b&c optional) – fill in the CPU regs with the value of vars.

asm("blah blah") – inline assembly.

Together these allow for neat macros like:

#define in(c,b,n) – ABCregs(c,b,n);asm("in")

Supplied macros (?) include in/out, ldtimer, stpri, etc.

Of course, we at Cornell have implemented the Trollius OS, with its support for parallel processing, as library functions. They are almost entirely in C, with occasional usage of in(), etc.

Chris Brown, A.I. Vision Research Unit, Sheffield University, (chris@aivru.sheffield.ac.uk)

writes: For the 3L Parallel C compiler:

1. has a special type for channels? Yes there's a typedef for CHAN but they're just pointers really.
2. does input by
   (a) evaluating the name of the channel as if it were a variable
   (b) by a function call <—- YES
   (c) some other way
3. does output by
   (a) assigning to the name of the channel as if it were a variable
   (b) by a function call <—- YES

(c) some other way

4. has special constructs for PAR and ALT, or uses special functions? ... It has special functions for creating "threads" - i.e. lightweight processes which share code and data space (but not stack!). There are functions to implement sempahores to allow threads to safely share buffers, channels, etc. ... It has heavyweight processes, more like unix processes. These are bound together by a separate configuration language (not C).

3L Parallel C has NO new syntactic constructs, so you can use programs like lint on the code.

Kirk Bailey <bailey@edu.orst.cs.mist> writes:

Several notes: the Pentasoft and "Peng[Buin" compilers are one and the same! Pentasoft is the current name of the company (Dwight VandenBurghe is the owner/ programmer).

I also have some anwsers to your questions about the Logical Systems "C" compiler (which I'm a co-author of).

1. It uses functions calls for channel I/O and par/alt concurrency. It does however treat many of the functions as "intrinsics" and generates basicallythe same in-line code sequences that OCCAM uses (ignoring differencies in activation record layout, etc.).

2. In addition to the OCCAM-style PAR, PRIPAR, ALT, etc. The "C" library also includes a UN*X like concurrency paradigm based on the "fork"/"join" model (plus semaphores, priority switch routines, etc.)

Jerry DeLapp <jxdl@gov.lanl> writes:

I'm not sure, but I think that the Penguin and Pentasoft are the same thing...

I don't think the Penguin handles channels at all... At least, we had to write our own libraries to handle them in Tcode (from C) and occam.

Robert Virding, UUCP: rv@erix.se or decvax,philabs,seismo!mcvax!enea!erix!rv,

Computer Science Lab. EUA/SU, Ellemtel Utvecklingsaktiebolag, Box 1505, S-125 25 Alvsjo, SWEDEN.

writes: I'm afraid I can't help you with information about concurrency in Transputer C compilers. the only one I've used is 3L which has no such capabilites at all, though it can read and write channels through special library functions. Unfortunately there is no way to detect empty channel.

[I suppose this refers to the straight C compiler, not to parallel C. JW]

Occam 2 syntax

From: Geraint Jones <geraint@uk.ac.oxford.prg>, Original-From: Pete Jinks <pjj@uk.ac.man.cs.r2>

From pjj:-

I am attempting to create a parser for occam 2, using YACC. In the course of this, I came across the following unlikely construction.

In the occam 2 reference manual, the following syntax definitions are given:

```
      primitive.type  = CHAN of protocol | INT | ...
      protocol        = simple.protocol | ...
      simple.protocol = type | ...
      type            = primitive.type | ...
Thus the following would appear to be legal:
      CHAN OF CHAN OF INT snafu :
```

In your book [Jones & Goldsmith: Programming in occam 2] in chapter 15, you seem to avoid this construction by defining base.type and data.type etc. You also say "With the qualifications in the accompanying prose in each [of the two books], both this chapter and the reference manual describe the same language." However, although in this instance I much prefer your version of the syntax, I am unable to find any "accompanying prose" in either book which explains this apparent difference.

Could you comment on the authority and completeness of the two syntaxes ?

Thanks in anticipation: Peter J. Jinks, Room 1.11, Department of Computer Science Annexe, University of Manchester, Oxford Road, Manchester, M13 9PL, U.K. Tel: 061-275 6186 Email:pjj@uk.ac.man.cs.ux

And my answer:

I think the "accompanying prose" I had in mind when I wrote that sentence was the bit at the bottom of page 30 of the Reference Manual, which reads:

A simple protocol is either a data type or ...

and which qualifies the production

simple.protocol = type — ...

quoted above by Pete Jinks. I reckon it is all a bit too subtle: there is a distinction being made between 'data types' and things like 'CHAN OF phu' and 'PORT OF phu' and timer types and things.

My version of the grammar was written to be read by human readers, so I was prepared to have (for example) ambiguous parses that could only be resolved by semantic information. I also wanted to differentiate between things that a reader would think of as being qualitatively different, like for example data types and channel types.

The version in the Reference Manual was almost certainly written with a mechanical parser in mind. There is of course a certain syntactic similarity between the various things which are 'type's in the Reference Manual grammar. One of the benefits of writing a parser along the lines of the Reference Manual is that it is probably easier to generate intelligent error messages: the parser would read the 'type' 'CHAN OF CHAN OF INT', and the semantic check – whether CHAN OF INT was a 'data type' – would fail. A parser written to my grammar would almost certainly just object that 'CHAN' could not appear in 'CHAN OF CHAN ...'.

As for authority, well I suppose that in the absence of a 'Standard', the authoritative definition of what constitutes occam2 is the Reference Manual. The Reference Manual is undoubtedly 'correct' with respect to itself; it is my hope that the language described in chapter 15 of Jones and Goldsmith is the same, and that the description is easier to understand.

gj

What makes Transputers interesting

From: J.Wexler@uk.ac.edinburgh

I have made a compendium of the recent correspondence on this topic. If anyone is interested in seeing it all in one place, here it is. My thanks to all those contributors whose words are quoted here.

John Wexler

On one of the electronic bulletin boards much frequented by Transputer users, an extended discussion was recently (November 1988) provoked by the question "What distinguishes a Transputer from any other processor, especially if I take a, let's say 68030 or 32532, add 4 communication channels and write software to do processor-processor communication? What makes a Transputer so interesting?"

Many of the answers concentrated on the integration of the design and the consequences of having many features built into a single chip: speed, low chip count, and the potential simplicity of systems which use Transputers instead of other processors. Some correspondents mentioned particularly the ability to build useful single-chip systems (e.g., embedded controllers), and the advantage of being able to bootstrap one Transputer (or several) from another.

The features themselves make up a long list. As well as the instruction processor, a single Transputer includes a memory controller so that it can drive DRAM with no external

circuitry; it includes a small amount of on-chip memory; it includes the DMA control for four independent fast "links" for external and inter-processor data transfers, with access scheduling to prevent the links and the processor from locking one another out; it provides hardware and microcode to support inter-processor and inter-process communications; it includes a microcoded multitasking kernel, which recognises two priority levels of processes; it includes an elapsed-time clock; and the T800 model includes a floating-point processor.

The performance of the individual components is also important. As far as the instruction and floating-point processors are concerned, a full discussion of speeds and comparisons with other systems would be impossible in this article, but one can say, at least, that they are reasonably fast. However, it is possible to be much more definite about other aspects. Process switching, for instance, is extremely fast. Communications are fast and have low start-up overheads.

Coherent design is another virtue. All the built-in features are mutually compatible and make up an overall structure which is simple and easy to use. For instance, inter-processor and inter-process (within a single processor) communications are handled by quite different mechanisms, but they are controlled by identical instructions so that they can be handled in the same way by software. Again, the multi-tasking and communication facilities are fully integrated so that the necessary synchronisation of processes (e.g., a receiver waiting until a message has arrived) is obtained without software intervention or "busy-waiting".

Scalability is a major advantage of Transputer-based systems: it is much easier to enhance a system by adding further processors to it than would be the case with other microprocessors. Compatibility - the ease of replacing one model of Transputer by another without major design changes in a system - is also valuable. This extends to mixing models of Transputers within one system (including processors running at different speeds, or using different word lengths).

Whether or not the Transputer is a RISC processor is debatable. It certainly has many of the virtues which one expects to derive from RISC-architecture, such as simplicity and speed.

The principal complaints which came to light in the course of the discussion were, on the hardware side, that the Transputer should use its on-chip memory as a cache store, and that it should provide at least some support for memory management and protection. Software clearly annoyed a much larger number of people, who called for something more like an operating system, better support for standard languages, and a better software development environment.

Various collected comments from contributors:

Integration

Its integration (communication, multitasking, floating-point on the same chip) ==> speed

First of all, I agree that it's the level of integration that makes a transputer interesting.

They can do useful work with no external components - just feed it power, ground, clock, reset line, and hook up a link or two. You can boot it, download programs, and run them.

Low chip count

Needs no memory interface chips

its built-in memory controller: The Transputer can drive DRAM with no additionnal circuitry.

Another hardware facility provided is that of a DRAM controller built right into the chip. This simplifies DRAM system design considerably. Memory management of the channels vs processor requirements are done on chip.

Very fast process switch

The context-switch time on a transputer makes the 68xxx look like a pig.

As far as the multitasking is concerned, all instructions use a register stack (on chip) which is valid only for the duration of the instruction. This makes context switching extremely fast.

ONLY IF you can use the hardware defined task model. In that case, it's pretty nice, since it'll actually wait for a minimum of task state before swapping. If you wanted to run a standard operating system on the thing, you'd be in trouble.

Scheduling

The transputer provides a multitasking kernel built right in the microcode.

"Nearly free" inter-process communications

The speed and simplicity of its multitasking and communication due to the fact that they are integrated at the processor-instruction level.

The communications are very fast, have very low startup overheads, and operate without any need of the CPU after setup. This is not easy to accomplish in discrete silicon with software. In addition, the technology used for the communications allows for long (about 30-40 feet) cable runs.

Lets start with your 68030 alongwith its four communication channels - to match the transputer these links need to operate at 10 Mbps and contending with these communication

devices is no cakewalk - both in terms of H/W and S/W The multitasking processes use channels for interprocess communication - and these channels can be implemented either with memory exchanges or over the serial links. This can be made transparent to the application programmer.

RISC(ish)

its RISC-like architecture (few simple short and fast instructions)

The transputer is RISC technology. The small instruction set means that it's fairly easy to port compilers to it (although INMOS seems to be real stodgy about realizing that the real world wants C and FORTRAN).

Far as the software is concerned - Inmos claims that a high percentage (approximately 70% ?) of the instructions can be coded in one byte. I have looked at the instruction encoding philosophy and found it to be impressive. If you are at all interested in CPU architectures you really should look at it. It is, to say the very least, 'Interesting'.

I haven't really had any experience with the software but soon will have some. But probably not with Occam.

Well, they have got a patent on it. I think it's closer to 50%, but still the code is highly compact. (There are a few rearrangements I'd like to make, but that's another story.) Having programmed it in assembler a fair bit, I'll avoid "impressive" and stick to "interesting". There are things they could have done better. (Have cj pop the 0? Unsigned gt?)

Scalability

What this does is that it allows initial development and use over a lesser number of transputer and at a later time, if so desired, performance can be enhanced and almost linear speedup achieved, by increasing the number of transputers in the system and redistributing processes.

On-chip memory

Another hardware goody provided is on-chip memory. This is either 2k or 4k depending on the CPU (T414 or T800 resp). While not much in itself it can be used for code optimization as instructions running out of this on chip RAM run a lot faster than from external RAM.

Range compatibility

Most transputer systems can be upgraded by just plugging in newer, faster chips.

No virtual memory support

One of the main things I reproach to the Transputer is that it does not support virtual memory (vital to build any reasonnable stand-alone machine). And the 68030's interface to memory makes the T800's "look like a pig", to coin a phrase. Also the on-board memory should be organised as a cache. This makes programming much easier.

Miscellaneous

This philosophy of integrating the links right into the kernel pays dividends in another manner. The transputer is capable of booting itself right from the links. This implies that in a multiple processor system only one transputer is required to have a ROM. The others will be perfectly content with a simple RAM subsystem.

And the final hardware goody provided is an on chip frequency multiplier. This means that the different speed versions all take in 5 MHz clocks and multiply it appropriately to generate 20/25/30 Mhz. Thus these high frequencies are restricted to within the chip.

Meanwhile, occam is designed for people who dream distributed systems as opposed to others who dream von Neumann and then have to coerce their one thread onto multiple processors. In fact, that's the key transputer characteristic as well. It's designed for the way I think.

To sum-up, the Transputer is great because it is ONE Transputer instead of being MANY circuits + software.

INMOS has good plans for the future growth and enhancement of the chip series. (Now if they'd just do the same with the software).

OPINION

The transputer will probably define the future of parallel computing for the next 5 years or so IF IF IF INMOS will wake up and realize that the OCCAM language is a significant hindrance to acceptance of their product in the US market. OCCAM is a language best suited to CS weenies (BTW IR1, so I can say that :-).

P.S. I am not an INMOS employee. I have had significant experience with the transputer in a large scale parallel machine. The transputer hardware works well, the software sucks rocks. OCCAM is the single biggest roadblock to general acceptance of transputer based systems. Most people that I introduced to the OCCAM language system said "Come back

when you have 'real' languages". I am certain that we will not solve the problems of broad acceptance and understanding of parallel processing's capabilities as long as OCCAM is the context.

I hope members of this group realize that, in spite of Inmos's best efforts (in the past), there ARE several compilers available right now for the transputer, especially C compilers. There are also assemblers. Thus, one does NOT need to use the Occam language to utilize the interesting hardware features of the transputer, even though Occam has some nice features too. One can use the third party software, and Inmos themselves now offer C and FORTRAN too! So why doesn't the transputer take over the world? Lack of decent SYSTEM software. Of course, we at Cornell are trying to fix that with the Trollius OS... (and one should also mention Helios). There are already at least two vendors of transputer-based hardware (for UNIX hosts) offering Trollius.

'Interesting' is a subjective characteristic. I personally feel that the transputer is interesting because, atleast from a hardware developer's viewpoint it offers a lot of bang for the design effort.

Well, several C compilers are available. I reccomennd Logical Systems' C compiler, $6xx.xx with full source last time I looked. Kirk, are you still out there to correct me? They're based in Corvallis, Oregon. We had a couple of problems writing our OS in it, but it can handle serious work.

I haven't done any work in Occam, but C works fine. No, Occam isn't mandatory, although it makes communications-rich code a bit more legible. Kirk's compiler has #pragma asm and #pragma endasm so you can escape to assembler and get at anything the machine provides.

Actually, my experience has been that the transputer works best in tightly coupled systems, not loosely coupled. Still, it's brought loosely coupled systems into the realm of affordable reality. Hence, most of the discussion in this newsgroup centers on how to make loosely coupled systems work well.

Channel Utilization

From: transputer-request@edu.cornell.tn.tcgould

I wonder if anybody has done research on measuring the level of utilization of the channel, the link, and the CPUs?

Reply: Charlie Askew

Yes, various pieces of work have been done at Southampton University, and the Transputer Support Centre, on measuring processor and link activity.

Link and Processor activity can be estimated by using bits of hardware to monitor the

links and processor activity. In software, there was a posting on this mailing list some while back of the source of a process which would estimate the utilisation of a processor. We used this for quite a while and also modified it so that it would compile under D700D with the checking on.

Recently however, a process was written which would poll the links and low priority processes, to assess the activity of the processor. This provides a more complete picture of what is going on, but it does not deal with high priority processes. It can only assess the time that a link is waiting for a transfer, not the time spent actually transferring data.

The code is written in occam, but I am sure that it could be written in any language where you are allowed to access memory locations.

Charlie Askew,

JANET: CHAS@UK.AC.soton.esp.v1

BITNET: CHAS%UK.AC.soton.esp.v1@AC.UK

Transputer Support Centre, Unit 2, Chilworth Research Centre, SOUTHAMPTON, SO1 7NP, UK.

Tel: +44 703 760834/5

BOOK AND ARTICLE REVIEWS

The Editor welcome the contribution of books and articles for consideration in this section.

Communicating Process Architecture

Communicating Process Architecture (author) INMOS Limited, Prentice Hall, pp. xii+170, (UK) price £19.95, ISBN 0 13 629320 4

This is one of the books in the 'Prentice Hall Series of INMOS Technical Publications on Transputer Technology', and consists of a collection of INMOS technical reports roughly bullied into the gap between two covers. (Not to be confused with a companion volume which is confusingly called Transputer Technical Notes.)

The notes in this volume are:

- Communicating processes and occam; note 20, which is a chapter about why occam is the way it is
- The transputer implementation of occam, note 21 an outline of what a transputer does to implement occam

- Communicating process computers, note 22 is really about designing lage-grain parallel programs
- Compiling occam into silicon, note 23 a wonderfully off-the-wall paper about using occam as a chip design language (no really, I like it)
- The development of occam2, note 32 does a note 20 for protocols and types and such like
- IMS T800 architecture, note 6 partly an outine of the architecture, partly an outline of the following note:
- The role of occam in the design of the IMS T800, note 47 a promotion for the formal techniques used in the design of INMOS silicon
- Simpler real-time programming with the transputer, note 51 an explanation of the timer abstraction of occam, for the unconvinced practitioner who thinks he needs clock interrupts and scheduler queues and what have you to write real-time programs
- Exploiting concurrency: a ray tracing example, note 7 outline of how to ray-trace on a process farm
- High performance graphics with the IMS T800, note 37 the same again but for polygon rendering and such
- The transputer based multi-user flight simulator, note 36 the same again, but ... well, it might be the only documentation
- on this demonstrator, an essential feature of all good parallel computing meetings

The principal weakness of this book is that it is clearly a sequence of papers rather than a coherent book. There is much duplication between chapters: I lost count of the number of block diagrams of the transputer there were in it, and there are a number of chapters that could profitably be shortened by omitting yet another outline of the occam language. (Perhaps I am just annoyed because I find it difficult to read very wide lines of small print sans-serif characters.) Still, here it is, between covers and on your bookshop's shelves, complete with a glossy colour pictures of the backs of the heads of two people using the flight simulator.

(Geraint Jones, University of Oxford.)

Parallel Algorithms and Matrix Computation

by: J.J.Modi. Published by Oxford University Press, 1988, at £15.00

Parallel computers present the computer user with a new intellectual dimension - a dimension which at first may seem confusing, since many of the old guidelines of serial computing are no longer relevant. While the evaluation of serial machines can be based on the simple criteria of store size and basic instruction rate, the prospective user of a parallel machine

is faced with a bewildering range of architectures, from the FPS T/40000 with its 16,384 processors to the Cray 3, which has only 16. And with this choice of architecture comes an even more disturbing demand - the demand that the user abandons the familiar habits of serial programming for new parallel algorithms and new programming languages in which to express them.

This book provides a guide to this intellectual jungle. It begins with a general account of the various architectures which have been employed in the design of parallel machines. The major part of the book, however, is concerned with the effect of these architectures on numerical techniques and programming strategy. It is intended for serious computer users: people with large problems to solve who are willing to seek for algorithms, and programs based on those algorithms, which make effective use of the various forms of parallel machines now available. The treatment covers both the basic operations of data manipulation, sorting, etc., and the standard numerical procedures of linear algebra such as finding eigenvalues, singular value decomposition and the solution of linear equations.

Parallel computing is in very much the state that serial computing was in during the 50's and 60's. Hardware developments, research into numerical algorithms and language developments are all proceeding simultaneously. Current users of serial machines who are happy with Fortran 77 and the standard procedures of the NAG library may well hope that the computer specialists will find a way to shield them from all the disturbing changes which are taking place. But anyone planning to become an efficient user of the next generation of super-computers will need at some stage to become familiar with the ideas and techniques presented in this book.

(R.K.Livesley, Cambridge University Engineering Department)

Neural and Massively Parallel Computers - The Sixth Generation

by: Branko Soucek and Maria Soucek, pp.460, Wiley Interscience 1988, ISBN 0 471-63533-2, price £45

In the last few years there have been important advances in connectionist modelling and in the neurosciences. The research is often based on simulations of connectionist and neural models. Because the simulations are usually computationally-intensive, there has been much interest in the design of "massively-parallel" multi-processor systems such as the Connection Machine, for simulating these network models.

In their book, the Souceks discuss "Sixth Generation" computers, which they define as "computers that seek to emulate the logical and intelligent functions of biological systems". They start by outlining the ways in which they anticipate computer designs will evolve into the Sixth Generation, if computer designers are guided by knowledge of brain structure and function. After brief forays into psychophysics and neurophysiology the authors conclude the first section of the book with a rather brief discussion of communication processes within and between organisms. They unfortunately provide no more than a cursory discussion of nerve impulse generation and transmission, which is curious as

they cannot, therefore, cover the simulation of neural models with sufficient depth or clarity in the second section of the book, which is concerned with simulation and modelling. Nonetheless, the second section of the book provides a good introduction to simulation techniques and contains several interesting examples of models of communication in birds and insects.

The third section of the book covers some connectionist models, examples of "neurocomputers", event-train processing systems, and, rather incongruously, production systems. The authors offer a largely uncritical account of several connectionist models; there is, for example, no mention of Minsky and Papert's analysis of the abilities and limitations of Perceptron-like networks. When discussing the use of connectionist networks for solving the Travelling Salesman problem, the authors, apparently unaware of Lin and Kernighan's algorithm (which outperforms networks), state that typical solutions "could easily take hours ... on a classical computer and by comparison, a neural network computer finds the answer in a fraction of a second" (p.230). They go on to state (p.232) that the answer is found with "... the minimum of hardware. By comparison, a typical classical computer has about 10^4 times as many devices as a neural net". This comparison surely ignores the point that a computer is a general-purpose computing device, whereas a single neural network is not.

The last section of the book describes associative and parallel computers, and concurrent systems. Here, one finds a curious allocation of space to the different subjects. For example, hash coding is allocated 12 pages while pipelining is not mentioned at all. There is almost no discussion of the theoretical issues and problems involved in designing parallel computers. The last two chapters describe the transputer and the OCCAM language; each chapter is a summary of the INMOS company's own documentation.

Perusal of the glossary reveals some carelessness or serious misunderstanding by the authors. For example, decomposition is defined as "the *degree* to which an application and its tasks are divided into subtasks between processors for simultaneous execution" (my italics). The authors appear to hold unorthodox views about computers : "... the main purpose of the computer is to model the world ..." (p.437). Throwing caution to the winds they announce elsewhere that "brain-like computers are here to stay" (p.vii) and that "the best investment that we can make is ... in our bodies and our minds." (p.65)

Unfortunately, the book is not sufficiently rigorous or comprehensive for it to be able to serve as "a reference for practising engineers and scientists", as the preface suggests. However, the book is not without merit; it contains many interesting and unusual examples of neural models and it is the first book-length exposition of subjects associated with the so-called Sixth Generation; it might, therefore, be of some use and interest to undergraduates as part of their general reading.

(William Mackeown, Department of Computer Science, University of Bristol)

Parallel Computers 2

by: R.W. Hockney and C.R. Jesshope, Adam Hilger, 1988, ISBN 0-85274-812-4, price £19.50

In 1980, I had the good fortune to be shown the manuscript on Parallel Computers by Hockney and Jesshope. I was sufficiently impressed by this manuscript that I decided to offer a final year under graduate course on parallel computers. The recommended text was, of course, by Hockney and Jesshope. I continued to recommend this book in successive years until, in 1985, it took second place to Computer Architecture and Parallel Processing by Hwang and Briggs. This year, my recommended texts have been revised with Hwang and Briggs taking second place to the extensively written second edition of Hockney and Jesshope's Parallel Computers.

Most of the theoretical sections are very close to the first edition but it differs from that edition by three major revisions. In 1980, Japanese manufacturers had not started to manufacture vector computers, today things are very different. As well as the latest Cray and CYBER sytems, the Fujitsu, Hitachi and NEC vector computers are described and analysed. A new section on Multi-processor Systems, addresses a subject which hardly existed ten years ago. The Transputer and Occam are described, albeit, briefly. The final section on Technology and the Future makes an interesting first read on this topic.

Although, I would recommend this book to students and professional computer scientists and engineers, I believe it does have one serious imbalance. Outside of the large companies, such as Cray and Fujitsu, most research and development work will concentrate on multi-processors rather than vector architectures, therefore, I would like to have seen a much broader section on MIMD systems. However, I recognise that the authors have described parallelism in such a way that the principles they develop are applicable across the total spectrum of the theory and practice of parallel computation.

(Derek Paddon, Department of Computer Science, University of Bristol.)

PRODUCTS, SERVICES AND ANNOUNCEMENTS

Update from Transputer Marketing

Mark Jones, INMOS Ltd

Since the last newsletter, there has been plenty of positive press coverage on Transputer products and an increasing number of design-ins worldwide. There has been a major reduction in the price of silicon products and some important new product announcements, the highlights being:

- T222, a T212 replacement with 4K bytes of SRAM and faster serial links is now qualified and in full production in both 17 and 20MHz versions.

- T800, the 25MHz version is now available for sampling at our distributors giving a 25% increase in performance over the 20MHz device.
- T425 an enhanced T414 with 4K bytes of SRAM faster links and enhanced instruction set will be sampling in January.

A number of new books written by INMOS staff and published by Prentice Hall are available from all good bookshops now including:

- Transputer Reference Manual (Data Sheets and hardware description)
- Communicating Process Architecture (a selection of technical notes on the theory and practical implications of Occam and the Transputer)
- Transputer Technical Notes - (A selection of technical notes on implementing the Transputer in real applications - of interest to S/W and H/W engineers)

On the board front The Motherboard and Module strategy is moving ahead despite the worldwide shortage of DRAM's. New modules that have been released are:

- B407 - A size 8 ethernet TRAM meeting IEE802.3.
- B408 - Graphics TRAM with 1Mbytes DRAM and 1.25Mbytes VRAM for use with screens up to 1024 x 768 pixels. (Size 8)
- B409 - Video TRAM, for use with B408, incorporates 3 x G176 and supports 8 or 18 bits per pixed. (Size 8)

A new addition to the Motherboard is:

- B014 - A VME bus card for use with Sun based systems, can accommodate up to 8 Modules.

The major news in software has been the full product release of the new TDS D700D, a fully integrated compiler and debugger for the PC.

Over the next few months we are looking forward to the 30MHz T800, a new occam toolset release, a fully qualified Mil Std 883C T800-17 and continued profits!

The International Journal Microprocessors and Microsystems

Announces: Applying the Transputer

- Two authoritative special issues
- 20 fully refereed papers

Special issue Guest Editors:

- Dr Chris Jesshope, Southampton University, UK
- Dr Derek Paddon, Bristol University, UK
- Mr Ian Whitworth, RMCS, UK

The transputer was launched in 1985 as the first commercially-available microprocessor to embody the principles of parallel processing. These special issues are timed to report the growing body of international work that implements transputers and to reflect the innovative design practices the device has encouraged. The first issue will highlight papers on transputer hardware, software and development tools. Applications to be covered in the second part include robotics, vision and image processing, defence applications and networks.

Examples of papers to appear:

Issue 1 (March 1989) Transputer enabling technology

- Solving problems with transputers
- Helios: a distributed operating system for transputers
- Multitransputer graphic subsystem
- Peripheral handling techniques for the transputer
- Support for the OCCAM objects on transputers

Issue 2 (April 1989) Transputer applications

- Transputer-based digital waveform generator
- Hybrid architecture paradigms in a radar data processing application
- Automatic compilation of parallelism in visual object recognition

- Transputer-based simulation tool for evaluating wide area networks
- Distributed algorithm for a robotics application on transputers

For more information and an order form contact: Sales Promotion, Microprocessors and Microsystems, PO Box 63, Guildford, GU2 5BH, UK. Tel: (0483) 300966. Fax: (0483) 301563

John Wiley & Sons Ltd announce a new International Journal

Concurrency Practice and Experience

Editor: Geoffrey C. Fox, California Institute of Technology, USA.

Associate Editor: Paul Messina, California Institute of Technology, USA.

Associate Editor: Tony Hey, University of Southampton, UK.

Aims and Goals

Recent developments in technology have stimulated the development of concurrent computers. These machines consist of a collection of processors connected in a network - or alternatively a collection of processors sharing access to a common memory. These include both general purpose MIMD and SIMD architectures and special purpose systems such as neural networks. Optical and dataflow hardware can be expected in future. There are now several commercially available concurrent computers and an increasing number of microprocessor chips specifically designed to permit the construction of parallel computers varying in size from PC add-in boards with a few processors up to 64000 processor supercomputers.

These machines are being successfully applied in a wide range of application areas especially in science and engineering. This is producing a substantial amount of practical experience in those problems which parallelise well and the features of hardware and systems software needed to use concurrency effectively. There are also new computational methods, such as cellular automata and massively parallel neural networks, which are particularly suited to concurrent execution. At present, there is no journal that brings this work together. Results, if published at all, are scattered through specialized technical journals.

This journal will therefore, focus on practical experience with concurrent machines, especially:

- Concurrent solutions to specific problems

- Concurrent algorithms and computational methods
- Programming environments, operating systems and tools
- New languages
- Performance design, analysis, models and results

The papers will all have a practical or phenomonological emphasis.

Authors wishing to submit a paper to the journal should contact the editor directly at the address given above.

Call for papers

Papers are solicited for the second issue for which the deadline is April 30, 1989.

We encourage papers from a broad range of authors and point of view and will attempt to choose referees that will judge papers on quality and not approach. If you are interested in submitting a paper for these first two or later issues please contact an editor or send email to cpando@hamlet,caltech.edu;cpando@caltech.bitnct, The FAX number is 818-584-5917.

Publishing Information

G. Redver-Mutton, John Wiley & Sons Ltd. Baffins Lane, Chichester, West Sussex,PO19 1UD, England. Telephone: (0243) 779777 Telefax: (0243) 775878. Telex: 86290.

Subscription Details

Concurrency: Practice and Experience is to be published quarterly; Volume 1 (1989) 2 issues UK £40.00 Elsewhere US$72.00

Occam and the MEiKO Computing Surface

Short courses, 12 - 14 April, 1989,
28 - 30 June, 1989, 20 - 22 September, 1989

These courses are aimed at those with little or no previous experience of occam and the transputer. Contact: Edith Field, Unived Technologies Ltd., 16 Buccleuch Place, Edinburgh. EH8 9LN.

OCCAM 2† AND TRANSPUTER† ENGINEERING

Computing Laboratory, University of Kent at Canterbury

Course Objectives: To acquire technical knowledge, insight and practical experience of *parallel system design* using occam and transputer networks.

Further Details: Harnessing the potential processing power of Transputer networks requires the development of a fluency in parallel systems design equal to our traditional skills for sequential logic. Occam is a simple, small but powerful language which enables such fluency. Software engineering principles, load-balancing techniques, real-time applications and various embedded and super-computing issues will be covered. The strengths, weaknesses and likely future developments of occam and transputer technologies will be discussed.

Course Members: Engineers with *some* experience of a traditional "high-level" language. *[Note: we have found that hardware engineers, with only a modest knowledge of software, find the occam concepts for parallelism particularly easy to master.]* Since September 1986, this course has attracted over 160 participants from Industry and Academia worldwide.

Course Methods: Informal lectures with a large proportion of "hands-on" experience being provided through practical exercises and a "mini-project". Practical work will be on the MEiKO† Computing Surface† and will be supervised at the ratio of one tutor for every six attendees. The MEiKO provides a multi-user multi-transputer development and applications environment. Our system will support up to 20 simultaneous users, each with dedicated access to a private network of transputers including at least two T800s. The full system comprises over 80 transputers (including 56 T800s) with a gigabyte distributed file store and three high resolution graphics workstations.

Length: **Five days**

Dates:
Course No. 10 : 10 - 14 April, 1989.
Course No. 11 : 26 - 30 June, 1989.
Course No. 12 : 10 - 14 July, 1989.
Course No. 13 : 4 - 8 September, 1989.

Contact: For a full syllabus, application forms, fees, special arrangements and accommodation, please contact :–

Dr P H Welch (*Tel*: +44 227 764000 ext. 7695)
Computing Laboratory (*Fax*: *+44 227 762811*)
The University (*Telex: 965449 UKCLIB*)
Canterbury (*email: phw@uk.ac.ukc*)
Kent — CT2 7NF
ENGLAND

EEC Recognition: This course is one of the foundations for a series of courses and technical workshops entitled "*Training for Transputer Technologies*". This is being developed under contract with the EEC as part of the Communities Action Programme for Education and Training for Technology (COMETT). Details of this programme are given elsewhere in this Newsletter.

---

† Occam and the Transputer are trademarks of the INMOS group of Companies;
MEiKO and the Computing Surface and trademarks of Meiko Limited.

TRAINING FOR TRANSPUTER TECHNOLOGIES

EEC Community Action Programme for Education and Training for Technology (COMETT)

Introduction

This award comes from the third round of the £32 million, three-year COMETT programme. It links together Higher Education and Industrial Organisations across Europe to define and implement an integrated series of courses and technical workshops. The partners in this project include :-

The University of Kent, UK (Prime Contractor)
(Contact: Dr P H Welch — back of this newsletter)

Twente University, The Netherlands
(Contact: Dr A W P Bakkers, El. Eus Department, P O Box 217, 7500 AE Enschede, The Netherlands, Tel: +31-53-893794; Fax: +31-53-354003.)

Brainware GmbH, West Germany
(Contact: Mr J Stender, Gustav-Mayer-Allee 25, 1000 Berlin 65, W. Germany, Tel: +49-30-4633058; Fax: +49-30-4694649.)

Intelligent Systems International, Belgium
(Contact: Dr E Verhulst, Zavelstraat 142, 3200 Kessel-Lo, Belgium, Tel: +32-16-259586; Fax: +32-16-207710.)

MEiKO Limited, UK
(Contact: Mr D Barr, The Hague Business Centre, Parkweg 2, 2585 JJ The Hague, Tel: +31-70-520956; Fax: +31-70-503075.) Tel: +44-454-616171; Fax: +44-454-618188.)

GEC Avionics Limited, UK
(Contact: Mr J Rees, Airport Works, Rochester, Kent, ME1 2XX, Tel: +44-634-44400; Fax: +44-634-827332.)

CAP (Scientific) Limited, UK
(Contact: Mr C Hedges 40-44 Coombe Road, New Malden, Surrey, KT3 4QF, Tel: +44-1-942-9661; Fax: +44-1-949-8067.)

Aims

In recent years there has been an immense amount of research in the field of transputers. The EEC, in particular through the ESPRIT programme, has contributed considerably through the SUPERNODE, SPAN and PADMAVATI projects. As a result, there have been major developments in transputer-based parallel architectures which have, in turn, led to the construction of a large variety of machines.

The floating-point T800 transputer (a direct product from the ESPRIT SUPERNODE project) has established Europe, for the time being, at the leading edge of research into practical massively-parallel high-performance computing. However, in terms of marketed transputer-based products, vigorous centres of exploitation already exist in Japan and North America. There is likely to be only a limited window of opportunity for Europeans to convert their role in developing the technology into commercial success. Rapid transfer of knowledge from the developers to the consumers of the technology is urgently required.

Mechanisms

This programme is to enable the partners to build on the success of an existing course at the University of Kent (in which they have all been involved) and make a large contribution to the establishment of the necessary further training across the EEC member states. It is proposed to :-

(i) develop further the existing short course on basic transputer engineering skills;

(ii) establish a coherent programme of short course training aimed at middle and senior management;

(iii) develop the training materials (including some video) for both these purposes;

(iv) package the course materials and make them available (under licence) for other trainers to present at other sites;

(v) determine the requirements for advanced courses on techniques for applying transputer technology to specific application areas (e.g. real-time systems, robotics, databases, large-scale simulations, numerical algorithms, image processing, ...);

(vi) within the COMETT network, offer our expertise as a focal point for basic and advanced training in transputer technology. This will take the form of short courses and workshops.

The academic partners will contribute to the teaching of the training courses and the development of training materials. The industrial partners will help to establish the precise areas of training need, assist in the monitoring of the training in their industries, and ensure that the whole project is kept in close contact with the on-going needs of industry and commerce within the Community.

Further Information

For further information, please contact one of the partners listed above. Two activities from this programme (the 'Transputer Engineering' course and 'UNIX and Future Support Environments' workshop) are announced elsewhere in this newsletter.

Optimizing Supercompilers for Supercomputers

Michael Wolfe, Kuck & Associates, USA

Effective use of a supercomputer requires users to have a good algorithm and to express this algorithm in an appropriate language, and requires compilers to generate efficient code. This book investigates several problems facing compilers for supercomputers. One problem is building a comprehensive data dependence graph; the book proposes data dependence testing methods and, by labelling the arcs with direction vectors that show the flow of data within the loop structure of the program, this labelled data dependence graph is used in several high-level compiler loop transformations, namely vectorization, concurrentization, loop fusion, and loop interchanging. It is shown how to apply these optimizations for different supercomputer architectures.

The book also investigates some problems associated with reductions and recurrence relations, including conditional recurrences and mixed control-data dependence cycles. A new formulation for the wavefront method of executing a loop is also given. Two other problems studied are the management of compiler temporary arrays and vectorizing WHILE loops.

176 pages/ISBN 0 273 08801 7 (Pitman)/£17.95

Reconfigurable Processor Array: a bit-sliced parallel computer

Andrew Rushton, University of Southampton

This monograph describes the development of an SIMD-class parallel computer based on processor-arrays. Surprisingly, for many problems a large array of bit-serial processing elements is a better source of processing power than a small array of complex processors, and the Reconfigurable Processor-Array described here is enhanced with floating-point, multiplication and data cache facilities to improve the operation of such arrays.The RPA also has features which allow clusters of processing elements to operate on each data item so that hardware parallelism can be matched with data parallelism. The implementation of the architecture as a chip design, for possible VLSI realization, is described, and an appendix contains a high-level formal description of the processing element in a register-transfer language.

192 pages/ISBN 0 273 08799 1 (Pitman)/£17.95

Partitioning and Scheduling Parallel Programs for Multiprocessing

Vivek Sarkar, IBM T. J. Watson Research Center

This work is one of the first studies in depth into the questions posed by partitioning potential parallelism into useful parallelism and is an attempt at a general solution. Given that it is desirable for the partitioning and scheduling to be performed automatically, so that the same parallel program can execute efficiently on different multiprocessors, this work presents two solutions to the partitioning and scheduling problems. The first approach is based on a macro-dataflow model, where the program is partitioned into tasks at compile-time and the tasks are scheduled on processors at run-time. The second approach is based on a compile-time scheduling model, where the partitioning of the program and the scheduling of tasks on processors are both performed at compile-time.

Both approaches have been implemented to partition programs written in the single-assignment language SISAL. The inputs to the partitioning and scheduling algorithms are a graphical representation of the program and a list of parameters describing the target multiprocessor. Execution profile information is used to derive compile-time estimates of execution times and data sizes in the program. Both the macro-dataflow and compile-time scheduling problems are expressed as optimisation problems, which are proved to be NP-complete in the strong sense. This work presents approximation algorithms for these problems. The effectiveness of the partitioning and scheduling algorithms is studied by multiprocessor simulations of various SISAL benchmark programs for different target multiprocessor parameters.

208 pages/ISBN 0 273 08802 5 (Pitman)/£17.95

Functional Programming for Loosely-coupled Multiprocessors

Paul Kelly, Imperial College of Science and Technology

In parallel programming, the critical issues for performance are communications efficiency and "grain size": good parallel algorithms use only local communications paths, and do a substantial amount of computation between communications.

This book studies how very high-level functional programming languages can be used to specify, reason about and implement parallel programs for a variety of multiprocessor systems, but in particular a class of loosely-coupled multiprocessors whose operation can be described by a process network. In these networks the nodes correspond to processes and the arcs to communications channels.

A simple language is described, called Caliban, in which the functional program text is augmented with a declarative description of how processes are partitioned and mapped onto a network of processing elements. The notation gains much expressive power by allowing these annotations to be generated by predicates defined in the functional language. Thus, common communications structures have simple and concise definitions as "network-forming operators". The main objective of these annotations is to provide an abstract description of the process network specified by the program so that an efficient mapping of processes to processors can be carried out by the compiler.

176 pages/ISBN 0 273 08804 1 (Pitman)/£17.95

Anticipated titles in 1989 include:

Algorithmic Skeletons: a Structured Approach to the Management of Parallel Computation/Murray Cole

Implementation of Logic Programming Languages on Parallel Computers/Peter Kacsuk

The Butterfly™ GP1000 Parallel Processor/ Pam Waterman (Ed.)

Massively Parallel Computing with the DAP/Dennis Parkinson (Ed.)

Editorial Board

Marketing and Distribution

The originating publisher for the series is Pitman Publishing. MIT Press will market and distribute the series in the USA, Canada and Israel; Pitman Publishing in the rest of the world.

Both publishers have extensive experience of such operations. The Pitman experience has been built up from their two existing series of Research Notes: in Artificial Intelligence and in Theoretical Computer Science.

Product news from Transtech

Unit 3, St. Johns Estate, Penn, Bucks., HP10 8HR.

Transtech offers more trams in less space

Transtech Devices have continued development of their TRAM module and motherboard range. The lastest TRAM to become available is the TTM3, a size 1 1MByte TRAM. The TTM3 has either a T414 or T800 running at 20MHz and 1MByte of 100ns 4 cycle RAM.

Upto 10 TTM3's can be mounted onto the Transtech TMB08 PC motherboard in a single PC slot, and upto 240 can be accommodated in the Transtech TOWER development system.

Transtech have also been expanding the software support for their TSB05 transputer hard disk board. A 55MByte disk option is now available with data transfer rates of upto 400Bytes/second to the root transputer. As well as the TDS, 3L's Parallel C and FORTRAN are now supported. A special fast FORTRAN development kit is available comprising a TSB05-55 hard disk board, a TTM10-8 TRAM module with a T800-20 and 8MBytes of RAM and 3L's Parallel FORTRAN.

Copyright of the Transputer Development System:

Transtech Devices Limited recently released a software product comprising a computer program and related documentation under its own name and denoted Transputer Development System IMSD700D.

That program was developed and produced by INMOS Limited who own all the copyright in it.

Transtech Devices Limited confirm that all rights to the IMSD700D are owned by INMOS Limited, that Transtech does not own any such rights and that Transtech was not authorised to make or distribute copies. Transtech Devices Limited apologises to INMOS Limited for any misunderstanding that may have arisen.

Product news from Niche Technology

Many of us had become impressed by the approach Niche Technology was taking to supporting multiuser access to Transputer systems via Sun workstation. Therefore, the news in November 1988 that Niche had gone into receivership caused shock and regret among the user community. The commercial survival of such an innovative technology is important, therefore, the following announcement is made with great relief and pleasure. (Editor)

Announcement on Niche Technology

Transtech Devices have stepped into rescue Niche Technology by acquiring all of Niche's software and hardware assets. Transtech have assured the Niche customer base of ongoing software and hardware support and will be upgrading all existing customers as new releases of software become available.

If you would like any further information about the takeover please contact Transtech (0494) 464303

Mike Cahill, Managing Director, Transtech Devices Ltd.

News from Quintek

Southfield House, 2 Southfield Road, Westbury-on-Trym, Bristol, BS9 3BH, UK. Tel: 0270 622196

Quintek Launch their Poppy PS/2 Transputer Card

Quintek Ltd have launched a Four-Megabyte single transputer card for IBM's new generation PS/2, called Poppy, it will increase the processing power of PS/2 machines with 80286/80287 processors bby a factor of 10, and extend access to even greater power through the connection of multiple Poppy cards, of Quintek's "Fast" family og transputer cards currently offering combinations of four and nine porcessors per card.

It provides customers who have sophisticated or scientific computing applications, with an effective, reliable means of boosting processor power on a single card.

Quintek's latest innovation will run a wide range of software from such companies as Rockfield, TopExpress and PS1, as well as the established 3L Fortran and C compilers. The card and software provide complete applications packages in finite element, commercial and financial analysis and scientific computing.

With the PS/2 range leading the new generation in personal computing, the Quintek card is likely to be a best seller with both direct users on OEM manufacturers.

Quintek sign Asia/Pacific Distribution Deal

Quintek's parallel processing products will shortly be making a bigger impact in the Asia Pacific region following distribution deals just signed in Japan, Australia and Singapore.

Managing Director, Patrick Mills, undertook an extensive Far East tour in search of the right distributors, and returned having reached agreement with NPS in Tokyo, Hawk Electronics PTY in Edgecliffe, New South Wales, and Multisync, part of the Singapore based Hong Leong Corporation.

Quintek Launch Big Memory Transputer

Quintek's latest addition to its successful "Fast" family of transputer boards, the Fast1XL, is launched this week. With up to 16 Megabytes of RAM memory on a single card, the 1XL offers access to significant extra power and memory, and provides the ideal front end for multiple transputer systems.

The single transputer alone will outperform 80836/387 machines by a factor of around three times, with corresponding increases when linked to additional Fast boards.

The 1XL can be connected to any of the other Fast family members, which include a range of boards with multiple transputer configurations.

The memory capacity of the 1XL - from 2 - 16 Megabytes - is especially valuable in complex programmes, where large problems can be completely held in RAM, giving further increases in speed.

For further information please contact Tony Harrison on (0270) 741198.

EXPANSION BOARDS FOR THE IBM PC/AT

Fast1S	1 x T800C-20 plus 1 MByte of memory
Fast1XL	1 x T800C-20 available with a range of memory options:
Fast1XL/16	with 16 MBytes of memory
Fast1XL/8	with 8 MBytes of memory
Fast1XL/4	with 4 MBytes of memory
Fast1XL/2	with 2 MBytes of memory
Fast4	4 x T800C-20 plus 4 MBytes of memory
Fast9	9 x T800C-20 plus 9 MBytes memory and C004A reconfiguration switch
Fast17	17 x T800C-20 plus 4 MBytes memory and 2 x C004A reconfiguration switches
Harlequin	1 x T800C-20 plus 1 MByte memory and 2 x 1/4MByte image buffers Includes image acquisition and graphics software

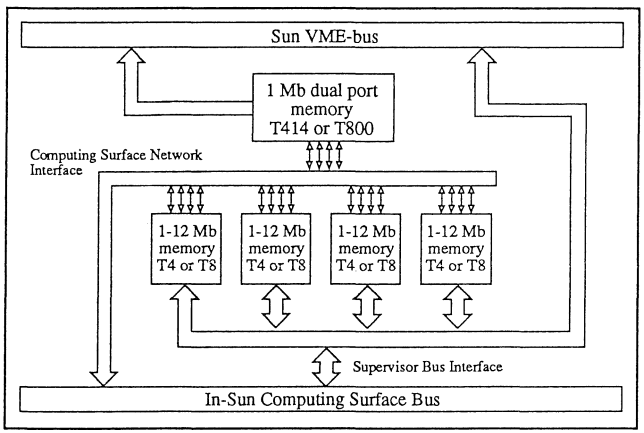
News from MEiKO

650 Aztec West, Bristol BS12 4SD. Tel 0454 616171

MK200 • SUN Computing Element

Meiko's In-Sun products allow massively parallel Computing Surface technology to be closely integrated with new and existing Sun installations.

Up to four MK200 In-Sun Quad Compute Elements can reside in a Sun chassis depending on the number of free slots and power supply capacity. The amount of memory available to each Compute Element can be configured between 1Mb and 12Mb and communication between any two Elements can be established dynamically under software control using Meiko's Computing Surface Network (CSN).



MK200 In-SUN Computing Surface Schematic

Hotlines:
USA (415) 952 9900
UK (0454) 616171

Each MK200 communicates with the Sun host using a fifth, dedicated Compute Element with either 1Mb or 4Mb of memory which is dual-ported with the Sun VME-bus. This allows full-speed transfers to take place between programs running on the MK200 and system services and other programs running on the Sun. Significantly, programs on the MK200 written in C, Fortran 77, occam or Pascal can achieve 450Kbyte per second file I/O speeds using the Sun disc cache.

A full implementation of Meiko's Supervisor Bus and CSN is provided at a connector on the rear of the MK200 so that multiple boards can be interconnected with the Computing Surface Bus. Alternatively, the MK200 can be used to provide a very high performance interface between the Sun and an externally attached Computing Surface.

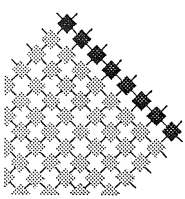
meiko

The most significant integration of the MK200 with Sun is, however, provided by its Sun Virtual Computing Surface (SVCS) software support. Access from Sun user processes is by a standard SunOS Transport Control Protocol (TCP) Socket interface for requesting and using MK200 resources. Two modes of operation are possible: either one or more conventional (sequential) programs run on individual Computing Elements, or a single parallel application runs on a domain of several Computing Elements. In the first case SVCS determines, transparently to the user, that a program is to run on the MK200 rather than on the Sun. By evaluating the program header information, SVCS also automatically decides what domain resources (processor type, memory capacity, etc.) are necessary to run the program. Where an application program can be partitioned into a number of communicating concurrent processes SVCS allows some of these to reside on the Sun host and others to run on one – or many – MK200 Computing Elements.

Because multiple Computing Elements are available in MK200 installations, a number of users can access Computing Elements at the same time, whether remotely over a workstation network or from the local Sun. SVCS allows requests for Computing Surface resource to be scheduled as soon as the necessary facilities are available, or provides an option for a request to return immediately when facilities are currently allocated to another user.

Each of the four Computing Elements on each MK200 runs integer programs at about the same speed as a dedicated Sun-3, while floating-point programs are run at 1MFlops: rather faster than a Sun-3 with floating-point accelerator. A full configuration of four MK200s hosted by a Sun3/160 therefore provides a 16MFlops computing service.

The SVCS software support for the MK200 is designed to be highly consistent with SunOS, Sun Microsystem's implementation of Berkeley Unix 4.3, and will consequently 'intersect' with MeikOS, the second generation system software family for the Computing Surface. MeikOS provides a Network File Server (NFS)-compatible environment that allows the Computing Surface to reside on a network of Sun workstations. The MK200 further integrates the Computing Surface Architecture into existing Sun installations - Seamlessly.

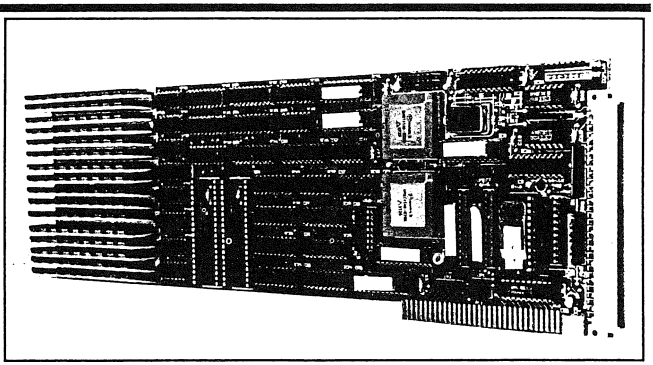


Systems West

Desktop Supercomputer Series

T4010 HOST INTERFACE ADAPTOR

The T4010 is a two transputer, high performance computer system for IBM PC/XT,AT or compatible machines. It forms the basis of a very fast and powerful parallel processor computer capable of out-performing almost any other micro-based system currently available.



FEATURES

- Two 32-bit transputers
- T414 or floating point T800
- Up to 30 MIPS (Million Instructions Per Second)
- 1,2,4, or 8 megabytes of parity-checked dynamic RAM on each transputer.
- Up to 16 megabytes on the board
- XT form factor card, compatible with PC/XT/AT bus
- Programmable interrupt-driven communications with host
- Programmable DMA to host RAM
- Selectable interrupt and DMA lines
- Full hardware and software compatibility with Inmos B004
- Optional 256k of EPROM
- 4 or 8 high speed serial links
- Fully configurable transputer-transputer communications
- All links fully buffered to Inmos specifications
- Can operate as master or slave in transputer network
- Up to 3 per XT or AT
- Fully compatible with other boards in System West Desktop Supercomputer series
- Development software (assembler and utilities) included
- FORTRAN, Pascal, occam, and C available.

TYPICAL APPLICATIONS

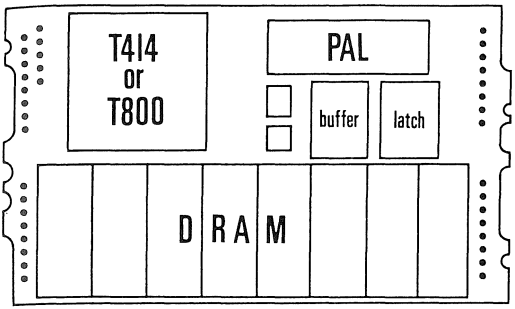
- Financial modelling
- Scientific and engineering calculations
- Graphics
- Image Processing
- CAD
- Artificial intelligence
- High performance transputer development system
- Research and education
- Programmer's workbench: high-speed compilation engine
- Data cache
- Transputer network controller
- Encyrption/decryption

TECHNICAL DESCRIPTION

The T4010 Host Interface Adaptor (HIA) contains two independent 32-bit computer systems on a single IBM PC/XT form factor card. These can be used as a high-performance coprocessors for an XT, AT, or 386-based system; they can be used as an interface to a larger network of transputers; or the card can be used in conjunction with other special-purpose cards in the Desktop Supercomputer series to build very high-performance workstations.

The T4010 can be supplied with the second circuit unpopulated. Systems West can upgrade the board when it is required.

TM8012S MASTER TRAM



TM8012S, actual size

The TM8012S is one of a series of Systems West TRAnsputer Modules conforming to published TRAM specs. It is a 16-pin dual in-line package compatible with TRAM motherboards available from Systems West, Inmos, and others. TRAMs represent the current state of the art in high performance computing. They can be used to build personal supercomputers or to add enormous processing power to conventional PCs or workstations; they are the ICs of the massively parallel processors of the 1990s.

FEATURES

- 10 MIPS T414 or T800 32-bit transputer
- one megabyte 100 ns DRAM
- subsystem and service lines, four high speed links
- stackable Size2 TRAM

TECHNICAL DESCRIPTION

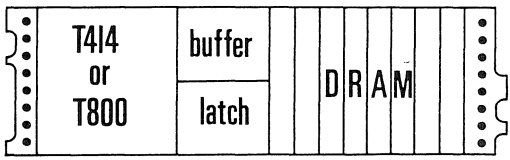
The TM8012S is a TRAnsputer Module intended for use as a master in transputer networks. It is a Size2 TRAM conforming to published TRAM specifications: a 16-pin dual in-line package with 5 additional pins. Three of these are used to control a transputer subsystem; the other two allow it to request attention from a system controller in a network configured dynamically by Inmos C004 crossbar switches. Low profile DIL DRAM packages are used, so the TRAM is stackable.

ORDERING INFORMATION

TM4012S, T414-20 version
TM8012S, with T800-20
TM8012S/17, with T800-17

TM8011 SLAVE TRAM

The TM8011 represents the highest MIPS*megabytes density currently available from any manufacturer in the world. This TRAM delivers 10 million instructions per second, 1.5 megaflops, and 1 megabyte of fast DRAM in approximately 1.5 cubic inches.



FEATURES

- T414-20 or T800 transputer
- 1 MB 100 ns DRAM
- four high speed serial links
- Size1 TRAM

TECHNICAL DESCRIPTION

The TM8011 is a 16-pin dual in-line transputer module conforming to published TRAM specs. The 1.05" wide package occupies one slot in a TRAM motherboard. It incorporates a 32-bit transputer and one megabyte of fast DRAM.The transputer communicates via four 20 MHz serial links, and is controlled via standard Reset/Analyse/Error lines.

ORDERING INFORMATION

TM4011, T414-20 version
TM8011, T800-20 processor
TM8011/17, T800-17 processor

Systems West
Whitefriars Southgate
Lewins Mead
Bristol BS1 2NT

TEL (0272) 273 990
TELEX 449 731 SGEX G
FAX (0272) 221 450

Gemini
Computer Systems Limited

News Release

ENHANCED GM8101 TRANSPUTER BOARD EX-STOCK

The Gemini GM8101 Transputer Board - plug-in 8Mb co-processor for the IBM PC, AT or clones - can now be supplied ex-stock with a 25MHz IMS T800-G25 64-bit floating point transputer.

Twenty-five per cent faster than the G20S, the G25S offers 25MIPS, 3.5 MFLOPS peak performance and 2.9 MFLOPS sustainable floating point performance.

This enhanced version of the GM8101 transputer board continues the performance advancement of British parallel processing systems which have been developed by Gemini in conjunction with the Computational Chemistry Group of Glasgow University.

Gemini offers an upgrade to the faster board. This simple task can be carried out by users themselves, or the boards can be sent to Gemini for upgrading.

Benchmark:

These tests are only for a small program, to give a feeling of performance of the GM8101 board. In general, we expect an 8101 to be between 2-3 1/2 times faster than the fastest available Intel processors.

Machine	Int Math (time in secs)	Real Math (time in secs)
IBM XT 4.77 MHz 8088	1.00	3.9
Mission 25386 25 MHz 80386/387	0.066	0.015
Gemini AT with GM8101 (20 MHz)	0.0040	0.0026
Gemini AT with GM8101 (25 MHz)	0.0032	0.00208

For further information, contact Flemming Christensen of Gemini Computer Systems Ltd. Chesham (0494) 791010.

Issued for Gemini Computer Systems by Stocker Hocknell Ltd. Amersham (0494) 433075. Contact Richard Salt.

Springfield Road, Chesham, Bucks HP5 1PW.
Telephone: (0494) 791010. Telex: 837788

BIBLIOGRAPHY UPDATE

We are in the final stages of creating a consolidated bibliography of items relevant to the transputer and occam. If any members know of material that has not been included in these lists in the Newsletter please send details to: Zena Woodley, Information Services, INMOS Limited, 1000 Aztec West, Almondsbury, Bristol BS12 4SQ, UK.

Articles authored by INMOS staff

Chris Followell "Staying afloat with the transputer" Australian Computing, June 1988, pp30-32

Articles authored outside INMOS

K.Adamson et al "Simple transformation rules in the application of transputer to the physiological processing of speech" Microprocessing and Microprogramming 24(1988) 397-402

E. Arnould and JP. Dugre "Real time discrete cosine transform: an original architecture" Proceedings - ICCASP '84, San Diego; March 1984, pp48.6.1-48.6.4

David Bannister "Cause for applause" Personal Computer Magazine October, 1988, pp92-96

Donna Bergamark "Programming the FPS T-Series" Cornell Computer Services, Ithaca, New York 14853

O. Boudillet et al "The implementation of a functional machine on a transputer network" Microprocessing and Microprogramming 24 (1988) pp389-396

G.F. Carpenter et al "Analysis and Protection of Interprocess communications in real-time systems" International Conference on Software Engineering for Real-Time Systems Cirencester,UK; 28-30 September, 1987, pp135-143

Alan Chalmers "OCCAM - The language for educating future parallel programmers?" Microprocessing and Microprogramming 24 (1988) pp757-760

P.K. Das and D.Q.M. Fay "Performance studies of multi-transputer architectures with static and dynamic links" Microprocessing and Microprogramming 24 (1988) pp281-290

M.V. Garcia and A.F. Jienez "OCCAM: Description of the language and implementations" Mundo Electronico, 1988; pp39-45 (In Spanish)

S. Geffin et al "A Massively Parallel Architecture for Robot Arm Control" Miami Technicon '87 (IEEE) Miami, FL; 28-30 October 1987, pp417-421

W. Hahn et al "A multi-transputer-net as a hardware simulation environment" Microprocessing and Microprogramming 24 (1988) pp291-298

G. Haussler "The assembler language of the transputer" Mikrocomputer ,Zeitschrift, 6, 1988, pp80-87 (in German)

D.I. Jones and P.M. Entwistle "Parallel computation of an algorithm in robotic control" Internatioonal Conference on Control 88 Oxford, UK; 13-15 April, 1988 pp438-443

H. Jorge and R. Gonclaves "A new cellular VLSI architecture based on transputers with application to circuit simulation" EURASIP - Signal Processing IV: Theories and Applications, 1988 pp919-922

Ir. P. Knoppers et al "Transputer network with flexible topology" Microprocessing and Microprogramming 24 (1988) pp275-280

A. Leger et al "Distributed arithmetic implementation of the D.C.T for real-time photovideotex on ISDN" Proceedings of SPIE: Advances in Image Processing (1987) pp364-370 The Hague, Holland; 31 March-3 April, 1987 (Vol.804)

Douglas Miles, Paul Kinney, Judson Groshong, Rodney Fazzari "Specification and Performance Analysis of Six Benchmark Programs for the FPS T Series" Abstract, published by Floating Point Systems, Inc. P.O.Box 23489 Portland, OR 97223

John Poplett "Development of a parallel C compiler" EXE 3(3), August 1988, pp34-39

Franz J. Rammig et al "A transputer-based accelerator for multilevel digital simulation" Microprocessing and Microprogramming 24 (1988) pp299-306

Simon Roberts "Applying the transputer" Program Now, August 1988, pp18-20

M.B. Sandler and S. Eghtesadi "Transputer based implementations of the Hough transform for computer vision" Microprocessing and Microprogramming 24 (1988) pp403-408

N.S. Scott et al "A comparison of programming paradigms for the parallel computation of Racah coefficients: an application of transputers to computational atomic physics" Microprocessing and Microprogramming 24 (1988), pp535-540

A.M. Tyrrell "Increasing Software Reliability of Distributed Systems with Occam" 2nd International Conference on Computers and Applications Beijing, China; pp23-27 June,1988

Toyokazu Uda, et al "An image processing system using parallel processing" 30th Anniversary Conference of the Society of Electrophotography of Japan (SEPJ) Tokyo,Japan; 16-18 May, 1988 pp211-214

Transputer (Numerous Articles in German) Chip Plus, Nr.8. August 1988 pp3-22

NEW MEMBERS

J E Ambrose, Nieuwstraat 134, 3011 GM Rotterdam, Holland.

Kevin Ambrose, 7631 Leesburg Pike, Falls Church, VA 22043, USA.

G Authie, LAAS, 7 Avenue du Colonel Roche, 31077 Toulouse Cedex, France.

Umar Bin Baba, Dept of Comp. Science, University of Hull, Hull, HU6 7RX.

Mr Baccaglione, Celdis Italiana Spa, Div. Componenti, Via Filli Gracchi 36, 20092 Cinisello B.S.A (Hilano), Italy.

Andrew E Bailey, University of Surrey, Elect. Eng. Dept., 129 Avondale, Ashvale, Aldershot, Hampshire GU12 5NF.

D El Baz, Laboratoire d'Automatique, et d'Analyse des Systemes du CNR, 7 Avenue du Colonel Roche, Toulouse, France.

Harold Bender, Muehldorfstr. 15, c/o fa. Rohde, 8000 Muenchen 80, West Germany.

Dr R J Best, Dept. of Chemical Eng., South Bank Polytechnic, Borough Road, London, SE1 0AA.

Denise Bland, Redifon Limited, Unit 9, Mole Business Park, Randalls Road, Leatherhead, Surrey KT22 7BA.

Robert D Bloor, Royal Ornance (Mardi Project), VT4 Parde (ch), Chobham Lane, Chertsey, Surrey, KT16 0EE.

D C de Boer, Electronic Building Elements, P O Box 4609, Pretoria 0181, Pine Square 18th Street, Hazelwood, S Africa.

Mehmet Bor, King Saud University, K.S.U. Coll of Computer Sci, P O Box 51178, Riyadh, S Arabia.

Dr M Bozyigit, King Faud University of, Petroleum and Minerals, KFUPM Box 1619, Dhahran 31261, Saudi Arabia.

Dr Allen H Brady, University of Nevada, Dept of Mathematics, Reno, NV 89557, USA.

Albert Brandl, Inst. f. Angew. Physik III, Universitat, Universitatsstr. 31, D-8400 Regensburg, W-Germany.

J N van de Break, St Paulusstraat 4, room C134 - DNL TWS, 2260 AK Leidschenclam, Netherlands.

M Buckwell, 4 Balaclava Road, Surbiton, Surrey, KT6 5PN.

John Burnell, Applied Maths Division, P O Box 1335, Wellington, New Zealand.

Ian Caldwell, Alsys Limited, Partridge House, Newtown Road, Henley-on-Thames, RG9 1EN.

Ugo Cei, Universita di Pavia, Dip Informatica e Sistemistica, Via Abbiategrasso, 209 Pavia 27100, Italy.

Stefano Ceriani, Agusta Sistemi SPC, vie Ison +0 33, 21049 9 Tradate (VA), Italy.

Mr K S Chana, Trent Polytechnic, 52 Lansdowne Road, Handsworth, Birmingham, B21 9AT.

N Chiment, c/o Brusa Engineering, Bodedracher str.99, 8121 Benglen, Germany.

C Chow, Rutherford Appleton Labs, Chilton, Didcot, Oxon, OX11 0QX.

Gianfranco Ciccarella, Scuola Superiore G.Reiss Romoli, S.P. DiCoppilo, KM0.300, 67010 L'Aquila, Italy.

Michael A Cipollo, Applied Mathematics & Mechanics , Research Division, Benet Laboratories, CCAC, Waterveliet, NY 12189-4050, USA.

Charles Clayton, 4A Berners Road, London, N22 5NE.

Edward H Coghran, Computer Science Corporation, P O Box 518, Kailua, HI96734, Hawaii.

P M Conlan, Systems Programmer, Imperial Chemical Industries PLC, Room R203, Topfloor

Block R, P O Box 14, The Heath, Runcorn, Cheshire, WA7 4QG.

Chris Craig, Mimansa Systems, Lower Tytherleigh Farm, Tytherleigh, Axminster, Devon, EX13 7AZ.

Patrick Daegelen, I.N.B.T., 1 rue du Parc, 92190 Meudon, France.

Dr R W Daniel, Department of Eng Science, Oxford University, Parks Road, Oxford, OX1 3PJ.

Raja B Daoud, The Ohio State University, 205 Dreese Labs, 2015 Neil Avenue, Columbus - OH 43210, USA.

C E Dawe, Easams Ltd, Lyon Way, Frimley, Camberley, Surrey, GU16 5EX.

Charles L Dawson, Bendix Communications, 1300 East Jappa Road, DPT 465, Baltimore, MD 21204, USA.

Russell Deich, General Dynamics, Electronics Division, P O Box 231395, San Diego, California, USA.

J C Delattre, Leanord R & D, Groupe In 2, 236 rue Sadi-Carnot, 59320 Haubourdin, France.

T B Dinesh, CCAD/University of Iowa, 310 MLH, Iowa City, IA 52242, USA.

Fraser Donachie, Marconi Defence Systems Ltd, The Grove, Warren Lane, Stanmore, Middlesex, HA7 4LY.

Jacqui Dowsett, Artificial Intelligence Ltd, Grey Caine Rd, Watford, Herts, WD2 4JP.

M Duranton, LEP, B P 15, 3 Avenue Descartes, 94 451 Limeil Brevannes, Cedex, France.

Neil Edwards, Dexter Electronics, Ltd, 162 High Street, Stevenage, Herts, SG1 3LL.

E Eid, Pixellar Imaging Systems Corp, 21 Antares Drive Suite 123, Nepean, Ontario, Canada, K2E 7T8.

Mehmet Rasit Eskicioglu, University of Alberta, Dept of Computer Science, 615 General Service Bldg, Edmonton, Alberta, Canada, T6G 2H1.

Antonio Ferrari, Universidade de Aveiro, Dept Electronica, 3800 Aveiro, Portugal.

Roger Feyereisen, 49 rue Pierre Wiser, I-9092 Ettelbruck, Gd Duchy of Luxembourg.

Mairead Flanagan, Trinity College Dublin, Computer Science Dept, 201 Pearse Street, Trinity College, Dublin, Ireland.

Carlos Couros Frias, Instituto De Astrofisica De Canarias, C Hornera S/N, LaLaguna, Tenerife, Islas Canarias, Spain.

D Frieauff, Stokerholzstr 27, 7990 Friedrichshafen 1, West Germany.

Mr H E George, George Consultants Ltd, International House, 2-4 Wendell Road, London, W12 9RT.

Hans Peter Gisiger, ETH-Zurich, Institut fur ElectronikSystemtec, ETH-Zentrum, CH-8092 Zurich, Switzerland.

Martin Gorrod, Astronomy Group, Dept of Physics, The University, Highfield, Southampton, SO9 5NH.

S Grogg, ERNI & Co AG, Stationsstr. 31, CH-8306 Bruttisellen, Switzerland.

Qinping Gua, School of Computer Science, Polytechnic of Central London, New Cavendish Street, London, W1M 8JS.

J B G Gurney, Hatfield Polytechnic, Dept of Elec Eng, College Lane, Hatfield, Herts, AL10 9AB.

Erik Hagersten, Swedish Inst. for Comp. Sci, Box 1263, S-16428Kista, Sweden.

Nobuyuki Haginoya, Fuji Xerox Inf Systems Co. Ltd, 3-16-6 Nishishinjuku Shinjuku-ku, Tokyo 106, Japan.

Gaetan Hains, Oxford University, Computing Laboratory, 11 Keble Road, Oxford, OX1 3QD.

P M Hallam-Baker, Department of Nuclear Physics, Oxford University, Oxford.

Juha Hamalainen, Instrumentarium Imaging, Teollisuuskatu 27, SF 00510 Helsinki, Finland.

Ken Hansen, Information Systems Labs (ISL), 8130 Boone Blvd, Va 22180, Vienna, Austria.

George Hartung, ABB-CRHL (Asea Brown Boveri), Eppelheimer Str 82, D-6900 Heidelberg, Germany.

R N Hayes, Plessey Avionics, Martin Road, Westleigh, Havant, Hampshire, PO55DH.

Paul Healy, Trinity Vision Group, Dept Computer Science, Trinity College Dublin, Dublin 2.

Karl Heesch, ETALON GmbH, Rahlau 30, D2000 Hamburg 70, West Germany.

B W Henderson, Rutherford Appleton Labs/SERC, Chilton, Didcot, Oxon, OX11 0QX.

Hans Heurer, Technische Universitaet, Otto von Guericke, Postschliessfach 124, Magdeburg/GDR, DDR-3010, Germany.

Brian Higgins, Computer Unit, Central College of Commerce, Hanson Street, Glasgow, G32 2HF.

J P Hulskamp, Royal Melbourne Institute, of Technology, GPO Box 2476V, Melbourne 3001, Australia.

PCM van der Jagt, Electronic Development Engineer, Oldelft B.V., vanMiereveltlaan 9, 2612 XE Delft, The Netherlands.

Tjerk Jansma, Nederlandse Philips Bedryven B.V, P O Box 20000, Eindhoven 5600 JA, The Netherlands.

Victor Jayakaran, Wipro Systems Ltd, 40/1A Lavelle Road, Bangalore 560001, India.

Jonny Johansen, Informasjonskontroll A/S, Solbra V.32, P O Box 265, 1377 Asker, Norway.

Mark A Johnson, Applied Mathematics & Mechanics , Research Division, Benet Laboratories, CCAC, Waterveliet, NY 12189-4050, USA.

Dr Jong-Hoon, Pohang Inst of Science & Tech, Dept of Physics, Postech, P O Box 125, Pohang, Repubic of Korea.

Karri Kaksonen, UNDA OY, Ahventie 4A, 02170 Espoo, Finland.

Y A Kasasbeh, School of Engineering, Exeter University, Exeter.

Martin J Keefe, C.E.G.B. Research Division, Kelvin Avenue, Leatherhead, Surrey, KT22 7SE.

D P Kelly, Information Storage Group, Dept of Elect and Elect Eng, Universityof Manchester, Dover Street, Manchester.

Brendan Kenny, Custom Technology Corp, Nikko Palace Higashiyama Bldg, 2-2-6 Higashiyama, Meguro-ku, Tokyo 153, Japan.

Thomas Kern, Ersonomics AG, Nordstrasse 198, GH-8031 Zurich, Switzerland.

James Kidd, ALTIS, 9 Brackley Street, Stockton Heath, Warrington, Cheshire, WA4 6DY.

Dr Michael Kiel, IBMB, Tu Bizaunschweig, Beethovenstr.52, D-3300 Braunschweig, Germany.

Hasizume Kimiaki, Holonics Laboratory, 9-8-50, Hagiwata 2-chome, Yahatanishi-ku, Kitakyushu-shi, Fukuoka 806, Japan.

Prof St Clair A King, University of West Indies, Dept of Elect and Computer Eng, St Augustine, Trinidad, West Indies.

Hideo Kitagawa, Dept Electronic-Mechanical Eng., Nagoya University, Furo-choChikusa-ku, Nagoya, Japan.

Dr Ing S Kowalik, Politechnika Slaska, Wydziak Gorniczy, Inst. Organ. Iekonomiki, Gornictwa, ul Pstrowskiego 2 44-100 Gliwice, Poland.

Prof A Krauss, Universitat der Bundeswehr, Informatik, Werner-Heisenberg-Weg 39, D-8014 Neubiberg, West Germany.

Patrick Kuchler, S.R.T.P. Department R.V.A., 10 rue de l'Ile Nabon, 44038 Nantes Cedex, France.

Mike Lambert, S.G.I.L, 115 Residence Elysee 2, 78170 La Celle St Cloud, France.

Larry Lawson, US Department of Energy, Morganstown Energy Tech Center, P O Box 880,

Mail Stop H04, Morganstown, USA 26507-0880.

Mr Allan S C Lee, Northern Ireland, Regional Transputer Centre, Queens University of Belfast, David Bates Building, College Park, Belfast BT7 1NN.

W F Lee, City Polytechnic of Hong Kong, Arygle Center (II), Mongkok, Hong Kong.

Banaid L'Hostis, Division ODSA, Celar, 35170 Bruz, France.

Roger Lock, P O Box 9703, Terminal Ottawa, K1G 3ZA, Canada.

S N McCorrie, G P T Telecommunication Ltd, 3rd Floor NLB Box 3, P O Box 53, Coventry, CV3 1HJ.

Lindsay MacDonald, Crosfield Electronics Ltd, Three Cherry Trees Lane, Hemel Hempstead, Herts, HP2 7RH.

Ian McGilray, Litton Systems Canada, 25 City View Drive, Etobicoke, Ontario, Canada, M9W 5AY.

Ahmed I Mahbaly, Queens University, Elect. Eng. Dept., Kingston, Ontario, Canada, K7L 3N6.

Mr Yrjo Makinen, Instrumentarium Imaging, Teollisuuskatu 27, SF 00510 Helsinki, Finland.

C J Martin, Radio Introduction Unit, Royal Signals & Radar Est, St Andrews Road, Malvern, Worcs, WR14 3PS.

Kazuto Matsui, INMOS Japan KK, 4th Floor No. 1 Kowa Bldg, Akasaka 1-chome, Minato-ku, Tokyo 107, Japan.

George Milakovich, Loral Defence Systems, P O Box 85, Litchfield Park, AZ85340-0085, Mail stop: 5011, USA.

P J Miller, Electrical Engineering, Aston University, Aston Triangle, Birmingham, B4 7ET.

Miss D N Mishra, PCL, 105 Charlmont Road, London, SW17 9AB.

Yasu Mochizuki, INMOS Int, Room 308 Kowa No 16 Annex, 9-20 Akasaka 1-chome, Minato-ku, Tokyo, Japan.

Gary P Moore, Marconi Underwater Systems Ltd, Elettra Ave, Waterlooville, Hants, PO7 7XS.

G Morgan, Anglepen Analysis Ltd, Flat 2, 11, Cotham Side, Cotham, Bristol, BS6 5TP.

Toshiyuki Morigami, INMOS Japan KK, 4th Floor No. 1 Kowa Bldg, Akasaka 1-chome, Minato-ku, Tokyo 107, Japan.

Bill Moses, Code 7051, Naval Air Development Centre, Warminster, PA 18974-5000, USA.

Gilberto F Mota, Brazilian Navy, Joracy Camargo 600/108, Rio de Janeiro, RJ21941, Brazil.

Prof. Dr. Albrecht Muller, c/o Fachhochschule, E1, Wiesenstr.14, D6300 Giessen, Germany.

Miss P Mullis, British Maritime Technology, Wallsend Research Station, Wallsend, Tyne, NE28 6UY.

Dr Martin Neumann, University of Vienna, Institut Fur Experimentalphysik, Der Universitat Wien, Strudlhofgasse 4, A-1090 Wien, Austria.

S J Newby, Marconi, The Grove, Warren Lane, Stanmore, Middx, HA7 4LY.

S P Norris, British Telecom Research Labs, RT5333, Room MLB 3/11b, Martlesham Heath, Ipswich, IP5 7RE.

Dr Ing W L Nowinski, Int. Lab. of Comp. Archit., Polish Academy of Sciences, 00-625 Warsaw, Polna 18/20, Poland.

Mr R J Oates, Dept of Computer Science, The University, Sheffield.

Antonello Olla, Scuola Superiore G.Reiss Romoli, S.P. DiCoppilo, KM0.300, 67010 L'Aquila, Italy.

Robert Ott, Grossenbacher Electronik AG, Spinnereistrasse 10, CH-9008 St. Gallen, Switzerland.

Dr Richard Overill, Dept of Computing, Kings College London, Strand, London, WC2R2LS.

R Parkes, SERC, RAL, Chilton, Didcot, Oxfordshire, OX11 0QX.

Giancarlo Parodi, c/o DIBE - Universita GE, via All'Opera pia 11/A, 16145 Genova, Italia.

Donatella Pascoli, Dipartmento di Fisica 6.6 Alile, Via Marzolo 8, 35100 Padova, Italy.

Stephen Pattenden, GEC Computers Ltd, Eyncourt Road, Woodside Estate, Dunstable, Bedfordshire, LU5 4TS.

Artur Jose Carneiro Pereira, Universidade de Aveiro, 3800 Aveiro, Portugal.

M Periasamy, Centre for Dev of Telematics, Sneha Corporation, 71/1 Millers Road, Bangalore 560 052, India.

J L Philippe, Ensaat, 6 rue de Kocoompont, BP 447, Lammion, France.

Mr C J Phillips, IBA, BG, E+D, Crawley Court, Winchester, Hants, SO21 2QA.

Dr M K Pidcock, Dept of Computing and, Mathematical Sciences, Oxford Polytechnic, Oxford.

Wilfred R Pinfold, M Rosenblatt & Son, 5952 N 10th Rd, Arlington, VA 22205, USA.

Prof S S S P Rao, Indian Institute Technology, Bombay, Dept of Comp Sci and Eng, I.I.T., Powai, Bombay India 400 076.

Kyosti Rautiola, Technical Res Centre of Finland, Computer Technology Laboratory, P O Box 181, SF-90101, Oulu 10, Finland.

Keith Rawlings, Smiths Industries, Aerospace & Defence Systems, Bishops Cleeve, Cheltenham, Gloucestershire, GL52 4SF.

Leonardo M Reyneri, Politecnico Di Torino, CSO Duca Degli Abrjzz 24, Torino10128, Italy.

Rick Ricart, USAF, AFWAL/AAAF-2, Wright Patterson AFB, OH 45433, USA.

Mr S P Richards, Citibank NA, Electronic Banking Division, 7 Savoy Court East, Strand, London, WC2R 0EA.

M D Robbins, Applied Maths Division, DSIR, P O Box 1335, Wellington, New Zealand.

Scott Rose, Link Flight Simulation, P O Box 208, Higley, AZ 85236, USA.

Prof. Alberto Rovetta, Dip, to Miccanica Politecnico, Piazza L. Da Vinci 32, 20133 Milano, Italy.

Mr Juss-Pekka Sairanen, Helsinki Univ of Technology, Otakaari 5A, 02150Espoo, Finland.

Aamod A Sane, Indian Institute of Tech, Bombay Hostel 3, Room 123, I.T.T. Powai, Bombay India 400076.

Raymond D Scanlon, Applied Mathematics & Mechanics , Research Division, Benet Laboratories, CCAC, Waterveliet, NY 12189-4050, USA.

A M Schuilenburg, Division of Building Technology, CSIR, P O Box 395, Pretoria0001, South Africa.

Albert Schuler, Technical University Vienna, Institut fuer Allg., Elektrotechnik, Gusshausstrasse 27/359, A-1040 Vienna, Austria.

Aillinh Scotti, Agusta Sistemi SPC, vie Ison +0 33, 21049 9 Tradate (VA), Italy.

Stephen B Seidman, George Mason University, 4400 University Drive, Fairfax, Virginia 22030-4444, USA.

Koh Liang Seng, Institute of Systems Science, National University of Singapore, Heng Mui Keng Terrace, Kent Ridge, Singapore 0511.

P L Smith, Mullard Space Science Lab, Holmbury House, Holmbury St Mary, Dorking, Surrey, RG5 6NT.

Michael B Sparks, I.B.A., Crawley Court, Winchester, Hants, SO21 2QA.

Dr A E F Spink, Dept. Civil Engineering, University of Birmingham, P O Box 363, Birmingham, B15 2TT.

Mr J Stender, Development Director, Brainware GmbH, Gustav-Meyer. Allee 25, 1000 Berlin

65, W Germany.

Nicholas M G Stephen, INMOS Ltd/Southampton Uni, 23 Woodstock Road, Redland Green, Bristol, BS6 7EL.

R S Stephens, 9 Cosin Court, Trumpington Street, Cambridge, CB2 1QU.

Dr K Straughan, Imperial College, Dept of Electrical Engineering, Exhibition Road, London, SW7 2BT.

Andrew Sykes, GEC Avionics Ltd, Sonar Systems Division, 26-28 Hydeway, Welwyn Garden City, Herts.

C Tavernier, DCAN-CAPCA, Les Oursinieres, 83220 Le Pradet, France.

Dr Murat Tayli, King Saud University, College Comp and Info Sci, P O Box 51178, Riyadh 11543, Saudi Arabia.

Murat Tayli, King Saud University, College of Comp & Info Sci, P O Box 51178, Saudi Arabia.

M Touman, 34 Bambrogh Close, Ox Close, Washington, T & W, NE38 0HN.

R H A H Tribe, Lucas Research Centre, Dog Kennel Lane, Shirley, Solihull, West Midlands, B90 4JJ.

Mr Simon Turner, Meiko Ltd, 650 Aztec West, Almondsbury, Bristol, BS12 4SD.

M Ukigai, Dept of Elect and Elect Eng, Kings College, Strand, London, WC2R 2LF.

Brigitte Vachon, University of British Columbia, Dept of Electrical Engineering, 2356 Main Mall, General Office, Vancouver B.C., Canada V6T 1Z3.

Jacques Vautherin, L R I, bat. 490 Universite Paris-sud, 91405 Osray, France.

Paul R M C Verzele, State University Gent, Sem en Lab voor Informatica, Campus Ardoyen, Grote Steenweg noord 2, B-9710 Gent, Belgium.

H W de Waard, Dewa Soft, Welnahorst 5, Enschede, Holland.

A Wagner, University of British Columbia, Dept of Computer Science, 6356 Agricultural Rd, Vancouver B.C., Canada, V6T 1W5.

Mark Ware, Mark Ware Associates, 96 Queens Road, Bristol, BS8 1NS.

Etsuo Watanabe, Matsushita Electric Industrial, Components Department, 26th Floor World Trading Center, 4-1 Hamamatsu-cho, 2-chome, Minato-ku, Tokyo, Japan.

Prof Dr E Wehrhahn, PKI, Thurn-und-Taxis-Str. 10, D-8500, Nurnberg, Germany.

Tom Wiley, ESA/ESTEC, ESTEC WDP, Keplerlaan 1, Postbus 299, 2200 AG Noordwijk, Netherlands.

H R Wipf, Brutsch Elektronik AG, Neusatz str., CH-8448 Uhwieseu, Switzerland.

Dr Alistair S Wood, University of Bradford, Dept Mathematics, Richmond Road, Bradford, W Yorks, BD7 1DP.

Mr Juha Yliolliterru, Instrumentarium Imaging, Teollisuuskatu 27, SF 00510 Helsinki, Finland.

Gisela Ziegler, Steinhofstr.17, 7500 Karlsruhe 41, W Germany.

Karel Zuiderveld, Dept of Radiology, University Hospital Utrecht, Catharjnesinge 101, 3511 GV Utrecht, Netherlands.

STOP PRESS

OCCAM AND TRANSPUTERS - A course at Coventry Polytechnic
20-22 March 1989. Contact Dr A M Tyrrell on 0203 631313 Ext 7669

occam® user group · enrolment form

·O·U·G·

To enrol a new member in the Occam user group and/or to report a change of address please copy and complete this form and return it to the Occam User Group Secretary, INMOS Ltd, 1000 Aztec West, Almondsbury, Bristol BS12 4SQ, England.

Name (must be an individual not a company)
Address (up to 6 lines, 32 chars each)

New member ☐
Address change ☐

The Occam User Group Membership list is held on a computer at the INMOS Bristol office. As this is a computer file holding personal information INMOS are obliged to follow the requirements of the Data Protection Act concerning this file. It is therefore necessary to get the written permission of all members for their data to be included in this file.

Please insert name and address in the box above. Please include a postcode if possible.

The additional information requested below may be of use to the OUG committee. Please ensure that you answer the final question and sign the form.

Telephone number:

Electronic mail address:

Please indicate what type of organisation you belong to by ticking one of the following boxes:

Electronics industry ☐ Software industry ☐ Other industry ☐ Academic ☐ Government ☐ Other (describe) ☐

Please give a brief statement of the nature of your interest in occam and the transputer.

The OUG has several special interest groups (SIGs). Please indicate if you are interested in one or more of these subject areas. If you would help to establish or would join a new group please indicate the subject area(s) of interest:

Artificial intelligence . .	☐	Formal aspects	☐
Graphics	☐	Hardware	☐
Learning	☐	Networks	☐
Numerical methods . .	☐	Operating systems . .	☐
Unix	☐	*other*.....	☐

The mailing list is now becoming a potentially valuable commodity, but we cannot give everyone access without your approval. At present the names and addresses are known by the OUG administration and are provided to INMOS marketing. They are also published in the Newsletter unless a member specifically requests confidentiality. We are also producing a directory of members which could also include telephone and EMAIL numbers and SIG interests if you give permission by ticking appropriate box below.

My name and address may be published in Newsletters/Directory ☐
My answers to all the questions above may also be published ☐
Lists including my name and address may be passed
to third parties offering relevant products or services ☐
I do not mind who sees the information provided here ☐

Signed ..... Date .....

NAMES, ADDRESSES AND TELEPHONE NUMBERS FOR RELATED GROUPS ELSEWHERE
++

North American Transputer Users Group
---

contact:

Mark Hopkins
INMOS Corporation
PO Box 16000
Colorado Springs
CO 80935
USA

(719)630-4000

Next meeting: Salt Lake City
5th-6th April 1989

Deutschen Occam-Interessengemeinschaft der Transputeranwender (DOIT e.V)
---

contact:

Hr J Stender
Brainware Gmbh
Gustav-Meyer Allee 25
1000 Berlin 65
West Germany

or:

Peter Eckelmann
INMOS Gmbh
Danziger Strasse 2
8057 Eching
West Germany

(089) 319-1028

Australian Transputer and Occam User Group
---

contact:

John Hulskamp
Dept of Communication and Electronic Engineering
Royal Melbourne Institute of Technology
GPO Box 2476V
Melbourne 3001
Australia

(03) 660-2453/2090

Next meeting: Melbourne 6th-7th July 1989

Japanese Occam User Group
---

contact:

Prof Tosiyasu L Kunii
Dept of Information Science
University of Tokyo
7-3-1 Hongo, Bunkyo-ku
Tokyo 113
Japan

(03) 505-2840

or:

Kazuto Matsui
INMOS Japan K.K.
4th Floor, No 1 Kowa Building
11-41 Akasaka 1-Chome
Minato-Ku
Tokyo 107
Japan

(03) 815-6460

Next meeting: 24th-25th April 1989

OUG Special Interest Group Chairmen

Artificial Intelligence

Steven Ericsson Zenith, INMOS Limited,
1000 Aztec West, Almondsbury,
Bristol BS12 4SQ.

Tel: 0454 616616
EMAIL: zenith@uk.co.inmos

Formal techniques

Bob Stallard, Racal Milgo Ltd,
Bartley House, Station Road, Hook, Hants RG29 9PE

Tel: 025672 3911

Graphics

George Staniewicz, Weirlord Ltd,
The Meeting House, High Street, South Harting, Hants GU31 5QB

Tel: 0730 85876

Hardware

Denis Nicole, University of Southampton,
Dept of Electronics, Highfield, Southampton, SO9 5NH

Tel: 0703 559122 (x3367)
EMAIL: dan@uk.ac.soton.esp.v1

Learning

Sandy Riach has resigned.

A volunteer to lead this group is required

Networks

Simon Turner, Meiko Limited,
Tel: 0454 616171
650 Aztec West, Almondsbury, Bristol BS12 4SD

Numerical methods

Derek Paddon, Department of Computer Science,
University of Bristol, University Walk,
Bristol BS8 1TR

Tel: 0272 303030 (x4336)
EMAIL: derek@uk.ac.bristol.compsci

Operating Systems

Gordon Manson, Department of Computer Science,
University of Sheffield,
Sheffield, S10 2TN

Tel: 0742 768555 (x5580)

Unix

Peter Welch, Computing Laboratory,
The University,
Canterbury, Kent CT2 7NF

Tel: 0227 764000 (x3629)
EMAIL: phw@uk.ac.ukc

The recent questionnaire sent to all members revealed that there are sufficient members interested in the following subjects for a Special Interest Group to be started. If anyone would like to volunteer to lead such a group would they please discuss what is involved with the OUG secretary.

Image analysis and vision
Digital signal processing
Programming languages (other than occam)
Robots and other control systems
Simulation.

occam® user group · committee

Informal OUG Committee

Dr Peter Welch, (CHAIRMAN)
Computing Laboratory, The University, *Tel:* 0227 764000 (x3629)
CANTERBURY, Kent CT2 7NF *EMAIL:* phw@uk.ac.ukc

Mr Charles Askew,
Transputer Support Centre,
Unit 2, Chilworth Research Park, *Tel:* 0703 760834
SOUTHAMPTON, SO1 2TN. *EMAIL* chas@uk.ac.soton.esp.v1

Ir Andy Bakkers,
Twente University,
PB 217, 7500 AE ENSCHEDE, *Tel:* +31-53-892790.
The Netherlands. *EMAIL:* elbscbk@henut5.earn

Mr Gordon Harp,
RSRE, St Andrews Road, *Tel:* 068489 2733 (x2824)
GREAT MALVERN, Worcs WR14 3PS *EMAIL:* jgh%rsre.mod.uk

Dr Jon Kerridge,
Department of Computer Science,
University of Sheffield, *Tel:* 0742 768555 (x5580)
SHEFFIELD, S10 2TN, *EMAIL:*

Dr Geraint Jones,
Programming Research Group, University of Oxford,
8-11 Keble Road, *Tel:* 0865 273851
OXFORD OX1 3QD *EMAIL:* geraint@uk.ac.oxford.prg

Dr Derek Paddon,
Department of Computer Science, University of Bristol, (EDITOR)
University Walk, *Tel:* 0272 303030 (x4336)
BRISTOL BS8 1TR. *EMAIL:* derek@uk.ac.bristol.compsci

Mr Roger Peel,
Department of Electrical Engineering,
University of Surrey, *Tel:* 0483 509284
GUILDFORD, Surrey GU2 5XH *EMAIL:* roger@uk.ac.surrey.ee

Dr Michael Poole,
Software Support, INMOS Limited, (SECRETARY)
1000 Aztec West, Almondsbury, *Tel:* 0454 616616
BRISTOL BS12 4SQ. *EMAIL:* oug@uk.co.inmos

Mr Simon Turner,
Meiko Limited,
650 Aztec West, *Tel:* 0454 616171
Almondsbury, Bristol BS12 4SD.

Dr Colin Upstill,
Plessey Research, Roke Manor, *Tel:* 0794 515222
ROMSEY, Hants SO51 0ZN

Mr Hugh Webber, (PROGRAM EXCHANGE)
RSRE, St Andrews Road, *Tel:* 068489 2733 (x2728)