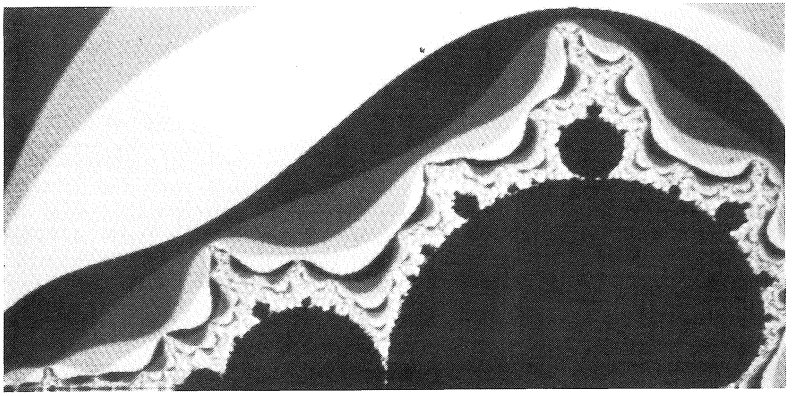
·O·U·G·

# occam™ user group · newsletter

No. 5 July 1986

CONTENTS





**Part of the Mandelbrot set (see L Pegrum, p 17)**

I am sure you will find this newsletter interesting reading once again. There has much occam and transputer activity recently, with a lot more material coming my way than previously. Please keep it up.

Martin Bolton Editor

## OUG NEWS

### Letter to the Editor

From Jack Huisinga, St Louis, MO: 21st Feb 1986

Dear Dr Bolton,

Thanks for your work on the occam User Group Newsletter. I believe the newsletter is crucial to the effective growth and use of occam.

Because occam is sufficiently different from other languages, and it is closely associated with the architecture of a particular microprocessor, there is much for us to learn on a number of fronts. I am delighted to read (Newsletter No. 4) that SIGs are beginning to form. But, it would be a disservice to occam and the user group if the SIGs began their own newsletters. We can all learn more by listening (reading) in on others' converations and the entire group will be better informed. Separate newsletters will fragment the group.

DECUS-US (Digital Equipment Computer Users Society) tecently addressed the newsletter issue. They consolidated all 23 SIG newsletters into one monthly. Each SIG has a section in each issue even if there are no articles, only the title page and officer list, but all together it results in a book the size of a thick magazine. Now I can have access to all of the SIG newsletters more easily and cheaply.

In light of the DECUS-US newsletter experience, I strongly recommend that any oug newsletters be combined with the oug newsletter. I hope you and the SIG organisers agree.

Sincerely; Jack Huisinga

### How the OCCAM USER GROUP works

by Michael Poole, INMOS

Now that our mailing list has reached over 700, I thought it was time I gave some information of how the group has evolved over the three or so years since it was started.

The origin of the group was a suggestion made by Peter Wilkinson from the National Physical Laborarory at a seminar hosted by INMOS in Bristol in September 1983. Everyone who had by then bought an Occam Evaluation Kit (do you remember the slow running, or the wonderful syntax-directed editor?) was invited to an inaugural meeting at NPL in November 1983 and 18 customers and two INMOS people turned up.

Six people volunteered to form a committee, but unfortunately there was no

further activity until a meeting of four of these people with three from INMOS was held at INMOS in June 1984.

At this meeting the pattern of activities which we have now adopted was laid down, and I was nominated by INMOS to act as the company's representative on the committee, of which Gordon Harp assumed the chair. This pattern is described in the "Grey leaflet" which is now included in all INMOS software products, and has been reproduced as the back pages of the newsletter.

Three principal activities were established - the Newsletter, Technical Meetings, and Program Exchange. These are described below.

As INMOS were, and are still, keen to see the group flourish and are prepared to finance the administration of the group and the production of the Newsletter it was decided to be very informal with no membership fee, or even an independent financial existence. The membership list consists of a file on a VAX/VMS system at INMOS, supported by a couple of ring binders containing enrolment forms sent in by applicants. Applicants may ask that any information they provide be kept secret from other members. We have a few members who have asked that not even their names should be made known to other members. For the purpose of the Data Protection Act the file is registered as an address list of existing and potential customers of INMOS.

With a few exceptions everyone on the list has got there by their own request, either by sending in a green enrolment form (the preferred way), by telephoning me or one of my colleagues, or as a side effect of having attended one of our technical meetings, as a lecturer or ordinary participant. The green form asks people to give an indication of the nature of their interest in OCCAM, but we have not transferred this information to the computer file.

The newsletter goes to a small number of non-members also, including such "great and good" as the Alvey Directorate and the INMOS Board of Directors.

The geographical distribution of the membership is very wide and now includes (in alphabetical order) Australia, Austria, Belgium, Bulgaria, Canada, China, Czechoslovakia, Denmark, England, Finland, France, Germany(W), Greece, Ireland (N and S), Israel, Italy, Japan, Jugoslavia, Korea(S), Netherlands, New Zealand, Norway, Papua/New Guinea, Poland, Saudi Arabia, Scotland, South Africa, Spain, Sweden, Switzerland, Turkey, Wales and USA.

**The Newsletter**

The Newsletter has been edited by Martin Bolton of Bristol University since it was started and has appeared at six-monthly intervals in January and July each year. An A5 style was chosen to make it different from the majority of other similar documents and to keep it reasonably small. The first issue had 16 pages, and subsequent ones have had 32 each. The printing is done by Pheon Press of Bristol.

The first issue used the INMOS "wafer" logo as a distinctive mark. This was considered inappropriate and so a logo for the OUG itself was designed for us by Roger Proctor of Bristol. Roger is a professional graphic

designer who cannot be expected to appreciate what occam is all about - any failure of the logo to represent the group must therefore be mine, as I was the person who tried to brief him!

In order to ensure that the various issues are distinctive on the shelf we have tried to find an appropriate picture to put on the front of each issue. The cartoon on No 1 is taken from some early publicity on occam designed for our USA office. It formed part of a comic-strip discussion on languages featuring Blaise Pascal, William of Occam and Ada Lovelace (not used in the excerpt).

The newsletter is very much dependent on the membership providing material for inclusion. Any tendency for material created by INMOS to exceed other material is a result of shortages of other material and hopefully will not go on for ever. Please support Martin Bolton in his efforts to make the newsletter flourish.

**Technical Meetings**

These have been held at approximately six month intervals since September 1984 and apart from the first held in Bristol have all been hosted by academic institutions. The first two meetings were one-day events, the next two lasted two days, and in future we expect to extend over three.

We have been very lucky in finding willing volunteers who have proposed themselves as hosts and programme organisers. The speakers have mostly been invited by the organisers, but some people have volunteered to speak without any prompting from the committee. Offers to speak are always welcome.

As there have been few adverse comments about the structure of these meetings we are likely to continue in the same style, unless someone makes a good case for doing something different. The committee are always willing to receive suggestions from people prepared to have good ideas and to help us put them into practice.

The attendance at the meetings has gone steadily (if not monotonically) upwards (Bristol 96, Oxford 150, Canterbury 135, Manchester 190). It is surprisingly difficult to be certain of these figures, even when we charge people to come as there are always several people who say they are coming and fail to arrive, and others who slip in at the last minute when the administration has produced its "final list" of people.

We are approaching the stage where meetings will be limited by the sizes of lecture rooms available, or will have to be redesigned to encourage parallel activities (perhaps the latter is more appropriate for us!).

By using academic accommodation we can keep the cost down to a reasonable level that does not discourage research students and others of limited means from coming. Some people may hanker for a hotel in Switzerland, but I feel that we are not the kind of people who ought to be doing that sort of thing at our employers' expense. (I wish my employer would pay! ed.)

**Program exchange**

Program exchange has not really got going yet. The major problem is the variety of different media needed for different systems and the slow

convergence of the various occam implementations.

## Back Numbers

Copies of Issues 1, 2, 3 and 4 of the Occam User Group Newsletter are available while stocks last on application to the secretary at INMOS.

**Issue 1** included 6 pages of bibliography on CSP, OCCAM and the Transputer.
**Issue 2** included a the first list of members, and a bibliography update.
**Issues 3 and 4** included supplements to both the above.
**Issue 4** included an inside story of the launch of the transputer and the evaluation boards. All included a variety of other articles of general interest to members of the group.

## Special Interest Groups

### Artificial Intelligence

A quick survey of members' interests suggests that the major interest of the SIG is the implementation of functional and logic programming languages on parallel architectures, with a radical faction interested in continuosly running or 'instigated' systems - i.e. systems which are never switched off.

We propose to hold a meeting of this group at the next OUG meeting. If you have a major topic that you wish to be discussed at this meeting, or if you wish to join the Special Interest Group, please contact Sean Martin or Mike Bell at Cambridge Consultants Ltd., Science Park, Milton Road, Cambridge CB4 4DW. Tel: 0223 358855

### Formal Techniques

From the Secretary: Mr R.P.Stallard, Dept of Computer Studies, Loughborough University, Ashby Road, Loughborough, Leics LE11 3TU

A circular was sent out to the members who expressed an interest in this group at the Canterbury meeting. From the limited response that resulted, there was an interest in the group acting as a forum for meeting people with related interests and for the exchange of information about relevant papers.

There was a joint SIG meeting held in Manchester with the UNIX group. About 40 people attended but at such an initial stage there was little discussion and there was very little news to report. At this meeting a questionnaire was distributed and the results form this particular survey showed that the principal interests were in the areas of program validation and verifiation, deadlock/livelock detection and the speciication of occam programs. The role of the group was seen as organising fringe SIG meetings at OUG meetings and to possibly circulate a separate quarterly newsletter. (It was agreed at the last committee meeting to keep SIG newsletters joint with the main one. ed.) Backgrounds and types of hardware varied amongst the group but UNIX was the most common operating system used for occam program development.

If you know of any interesting papers or software systems available please let me know so that I can distribute them; short comments about the papers would be appreciated.

The principal active group is located at the Programming Research Group at Oxford, and I hope the group will keep us informed about their work.

I am arranging a fringe meeting at Loughborough at the September meeting on Sunday evening. If anyone has any short presentations (5 minutes) or papers to distribute and hopefully provide some comments, I will be very grateful.

Operating Systems

From the secretary: Walter Hicks, Hicks Software Services, 45 Whittaker Lane, Prestwick, Manchester M25 5HA.

G. Manson of Sheffield University chaired the inaugural meeting of the operating system SIG held on 24th March during the 4th oug Technical Meeting. He described a filing system being developed at his university. Another gentleman mentioned his efforts towards building a database system based on an existing filing system. All agreed that occam 1 was not a convenient tool for constructing a multitasking operating system with dynamic allocation of resources.

Gerhard Peise of Parsytech GmbH mentioned that his firm was demonstrating a transputer based system at the Implementors SIG. They have used occam 2, an object which some attendees had not even heard of. (See later. ed.)

A sheet of paper was circulated and most attendees wrote their name and affiliation. J. Seymour of Inmos, who helps out with oug activities, expanded this from her membership information, to give full postal addresses of current members. (This was sent to all on the list). After the meeting Walter Hicks volunteered to act as secretary for the SIG, at least until we all get a little better organised.

The address list allows direct contacts between members and personal mail shots to the group. Many, but not all of us have access to JANET. Would a computer bulletin board be a useful means of contact? If so are there any volunteer system operators?

It is not unknown for SIGs to have their own meetings more frequently than full user group meetings. If you have any strong opinions on the subject, or offers of venue, please communicate them.

UNIX

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

The major interest of those (31) who attended the meeting at OUG-4 on 24th March was to discover the liklihood of obtaining an OPS for the <various>IX systems to which they had access. I have produced an information sheet giving technical details of what is on offer together with contact addresses. To obtain this, please send me a stamped addressed envelope.

Briefly, there are some "Portakit" interpreters (in C or 68000 assembler for BSD 4.2, Version 7 and System V - very slow), a "Portakit" native-code generator (for the Whitechapel MG-1 - quick), a "Portakit" microcode

emulator (for the HLH Orion - very quick), an OPS 2.1 for the SUN (native code, very quick, no transputer target software), the "Loughborough" compiler (in C, currently on VAX BSD 4.2 but potentially very portable) and an OPS 2.0 for the VAX (extensively revised form the original release).

I know that some other systems are being developed. If those responsible are willing to distribute them, please will they send me details.

The above systems have all come from the University/Polytechnic community and are available (for a zero or modest fee) without formal support. It is likely that INMOS will be marketing a commercial UNIX version of its TDS with full support for SUNs and VAXes in the near future.

Some discussion took place as to whether the "standard" Inmos OPS would be very efficient in a large multiple user UNIX environment (such as a big VAX), or whether the tools needed to be "unbundled" and made to conform more to UNIX concepts. We shall have to wait and see, but I personally hope for the former. As occam becomes more available under UNIX, the need for special tools to communicate with the UNIX environment may emerge - this group should certainly monitor this.

Because of lack of time at oug Technical Meetings, it has been decided to merge some of the SIGs. We share a common interest with the Implementors SIG (certainly more than with the Formal Methods SIG with whom we met last time!) so I hope the membership will not mind this union.

## DOIT - German User Group Established

by Joachim Stender, BRAINWARE GmbH

The German OCCAM Interest Group of Transputer Users (DOIT) has finally been founded. The inaugural meeting on April 18th, 1986 took place in West Berlin and assembled nearly 50 people. Until then more than 130 individuals had already expressed their interest in such a group, which has to be considered a local extension rather than as competition. The name DOIT clearly indicates that occam and transputers will be given the same attention, and hardware and software experiences will be exchanged mutually between the members.

Members will be charged a single fee of DM 50,- upon inscription (DM 100,- for companies). This will enable them to receive the DOIT newsletters, to participate in the meetings (twice a year), to receive brochures at a moderate charge, and to participate in the program exchange scheme. To facilitate general communication with the oug, all members of DOIT are anticipated to become oug members too. Without restricting anybody from participation in the DOIT, the German language is assumed to be the carrier of the DOIT communication links.

Those interested in joining DOIT are invited to write to the following address:

DOIT e.V
c/o Brainware GmbH
Herrn Stender
Kirchgasse 24
D-6200 Wiesbaden

## Fifth Technical Meeting

Loughborough University will be holding the Fifth Tehnical Meeting on Sunday 21st to Tuesday 23rd September 1986. Registration details are included with this newsletter. Sunday evening will be devoted to special interest groups and informal meetings. Monday and Tuesday will include lectures and a panel discussion. There will be a conference dinner on Monday evening. Accommodation will be in student study/bedrooms on Campus.

For more details contact:

Mr R.P.Stallard
Dept of Computer Studies
Loughborough University
Ashby Road
Loughborough
Leicestershire LE11 3TU

Tel: 0509 222679 or 0509 263171 ext. 2679

## Call for Papers - Sixth Technical Meeting

Roger Peel will be organising the 6th oug Technical Meeting which will be held at the University of Surrey, Guildford, on 13th-16th April 1987.

Since this will be the first full 3-day meeting, he is hoping to accommodate your views on timetabling, and would particularly appreciate suggestions on the Special Interest Group sessions and the use of the second evening.

Although rather early yet, he is happy to receive requests to present papers and to book exhibition space. Roger's address is:

Roger M.A.Peel
Dept of Electronic & Electrical Engineering
University of Surrey
Guildford
Surrey GU2 5XH

Tel: 0483 571281 ext 2278
JANET: roger @ uk.ac.surrey.syse
UUCP : ...ukc!reading!uoseev!roger

**Did you know.....**

that there is link between occam and Ada? William of Occam (c.1300-1350), Franciscan philosopher, is said to have been born in Ockham, Surrey, which was the location of the King family estate since 1707. William King, first Earl of Lovelace (1805-1893) married Augusta Ada (1805-1852), daughter of Lord Byron, in 1835 who became Countess of Lovelace. For a time she collaborated with Charles Babbage by translating and editing a memoir on the Analytical Engine by Menebrea. (It is probably a myth that she was the "first programmer"). Their eledest son took the title Viscount Ockham.

For more read: Ada: A Life and a Legacy, by Dorothy Stein. MIT Press, 1985.

## The Electronics Times/Inmos Transputer Design Competition

by Mick McLean, Electronics Times

A.C.Tan and Simon Arridge, project researchers at University College Hospital, have won the Electronics Times/Inmos transouter design competition with their design for a low cost system for modelling the effects of facial plastic surgery. Traditional methods of predicting the outcome of cranio-facial operations, like cutting up photographs or making wax and plastic models, have been unable to cope with recent advances in anaesthesia and surgical techniques.

Three dimensional computer modelling promises to be of great help to surgeons, but can not be used at present because of the prohibitive cost of the specialised harware needed. But new algorithms based on a recursive, presorted data structure, called Octree representation, and the advent of the transputer, mean practical systems can now be built. The interactive 3D surgery system proposed by Tan and Arridge uses four transputers to build a dedicated graphics processor interfaced to the conventional host computer already in use at UCH.

A single transputer development system would be sufficient to simulate and evaluate the proposed design, and one of these, worth about £10,000 is what Tan and Arridge received as their prize from Colin Southgate, managing director of Thorn EMI. Tan and Arridge say a "substantial market" exists for their system if it is approved for use in the health service.

A close runner up in the competition was another medical entry, from Alastair Mutch, of Reynolds Medical, Edinburgh. Mutch proposed a six transputer system for for a 24 hour analysis of electrocardiogram data in four minutes.

High quality entries came from Steven Parkes of Wimpol in Wiltshire and Peter Sewell of the Atomic Energy research Establishment. Both entrants proposed using multi transputer systems to build low cost, high performance music synthesisers.

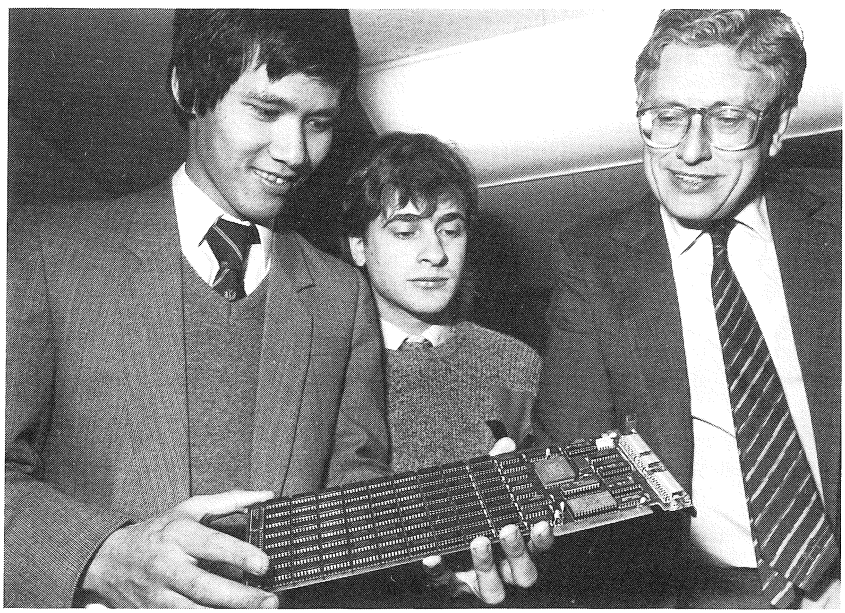
Other entrants proposed using transputer networks to model steam turbine feedheating systems, to build message switching system testers, six axis robot control systems, and for speech synthesis and recognition.

The 20 runners up all received as a consolation prize "The Inmos Saga" book and the "IC Saga" book of cartoons.
(Electronics Times, 23rd Jan 1986)

## Award for Inmos

At the British Electronics Week Ball this spring, Inmos received the Tobie "Component of the Year Award" for the transputer. The award was collected by Ian Barron, Inmos' chief strategic officer, who claimed that, in volume terms, the transputer was the most widely used 32-bit microprocessor. Barron said it was now up to Inmos to capture next year's award for Exporter of the Year. But Barron is concerned that although there is "considerable interest" in the transputer in the UK, "the first products to use the transputer will come from the US and Japan."

Barron criticised the main UK electronics companies for being "too conservative" and said Inmos had not had an "an adequate response from the established companies." He admitted that, in this respect, Inmos had failed to achieve one of its goals - to stimulate the UK electronics industry. "Far too many ideas are developed in the UK to the benefit of other countries. IBM exploited the Manchester developed idea of virtual memory, for example." Senior managers in electronics companies were aware that the transputer was an outstanding component, reported Barron, but they lacked the confidence to exploit it.
(Electronics Times)



Colin Southgate, presents A.C.Tan (left) and Simon Arridge (centre) with a transputer development system. Southgate said: "We need lots more applications like this one and Inmos will do everything possible to encourage them." (Electronics Times)

## MEETING REPORTS

### Fourth Technical Meeting

by Nigel Edwards, University of Bristol

The fourth technical meeting of the group was held at Manchester Polytechnic on 24th and 25th of March. People attending the meeting came from all over Britain as well as from France,West Germany, the U.S.A., Sweden, Ireland and Yugoslavia. The total attendance was just under 200 which represents a growth rate of about 100% a year. Of these about half came from academic and research establishments, and the rest came from a range of hardware and software companies.

The lecture program was organised into four main sessions chaired by Gordon Harp, Peter Welch, Chris Nettleton and Hugh Webber. During the breaks there were demonstrations by Meiko of transputers generating graphic displays using Mandelbrot series and by Parsystec of their new Megaframe development station.

The conference started with meetings of the Special Interest Groups which had been formed on the basis of interests expressed by members at the last meeting.

David May of Inmos gave the opening address of the meeting. He described the additional facilities provided by "occam 2" compared with those available in "occam 1". He spoke of the need for "occam 2": the writing of numerical and control programs in "occam 1" can be "a bit tedious". He then described some of the new features and pointed out that although the language has been extended, an attempt has been made to preserve the laws enshrined in the original version.

Richard Taylor of Inmos spoke of the encouraging upsurge in interest in occam in industry recently. He put the case for teaching occam in higher education, providing several very good reasons for doing so. Five "Heroes" of the occam Revolution then received citations.

Paul Bentley of Logica described a "Unix Implementation" which currently runs with Unix 4.2 on Vax and Sun. The system, which is written in a mixture of C and occam, uses the unix hierarchical filing system to implement folds and only supports Inmos supported languages.

An overview of ALICE, a research machine developed at Imperial College, was presented by Paul Townsend. ALICE (Applicative Language Idealised Computing Engine) is a graph reduction processor intended to support both functional and logical languages. It incorporates 128 transputers and 60 PCBs. The project illustrates one use of occam : it was used very successfully as the specification language of the system during its development.

John Gurd of Manchester University spoke about a parallel simulation facility. He argued the need for such a resource, saying that simulation is preferable to building parallel hardware in order to gain experience and test out ideas in the field of parallel hardware design. He presented a summary of a project he is involved in to build a parallel simulation facility based on transputers.

A presentation on the transputer instruction set was given by John Thornton of Sheffield University. He is interested in building an operating system using occam. However, there are some things which are difficult or impossible to do in occam, which are desirable to be able to achieve if one is writing an operating system. He sees the instruction set of the transputer as a possible way round this problem.

Tim Richings of UMIST spoke about algorithm partitioning for image processing/graphics applications. His long term aim is to produce a low cost concurrent image processing/graphics system. He is therefore looking for efficient algorithms and ways of partitioning images up into a number of segments which is proportional to the number of processing elements available.

Tom Parke of Inmos gave an entertaining talk entitled "Designing Interfaces Across Channels; Mistakes I have made". He went through each of the three types of channel interface which he had identified, showed some mistakes that could be made and then showed how the interface might be properly designed.

Iann Barron of Inmos gave the last presentation of the first day in which he talked about what might come from Inmos after the transputer. He asked the thought provoking question : 'What would the audience like after the transputer ?' While he was waiting for a answer he described the present state of the range of transputer products being offered by Inmos and what could be expected in the short term future. In the longer term using a new CMOS process being developed by Inmos it would be possible to increase chip complexity and speed. Among the various possibilities proposed, a member of the audience suggested that the opportunity be taken to increase the number of links on a transputer . On a show of hands nobody seem to think they would need more than six.

Tony Fisher of York University described a computer which he had built whilst at Hull which could only be programmed in occam. The system consists of a Z80A at each node, the nodes being connected on a ring via Clearway. The ring is controlled by a Prime 550 which also generates the interpreted code from occam programs to run on each node. He stated his intention to build a mark 2 system based upon 68000 and running occam compiled into 68000 machine code on each node.

Burkard Wordenweber of Shape data spoke of how occam could be used in solid modelling. He is interested in the parallelism that is offered by occam in the hope that it can provide improved speed in this application. He gave an example of an existing program sold by his company and came to the conclusion that it would involve too many man-years of work to rewrite the software, currently written in Fortran, in occam just yet. He proposed the compromise of modularising it so that it can be more easily adapted into a parallel environment at a later date.

Several speakers had been involved in projects which had used occam as a design language. Geoff Collis of UMIST gave an account of how he had used occam to describe the behaviour of digital systems. He concluded that occam is useful for providing a greater understanding of a system and compares well with high level description languages such as Ella when used as a behavioural description language.

Mike Lynch of Thorn EMI described how occam could be used in VLSI design.

He outlined the problems of VLSI design which lead to a need for greater and greater levels of abstraction. Currently there is a move to textual descriptions via silicon compilers and VLSI expert systems. However, he asserted that textual descriptions can be difficult to understand, and described work at Newcastle which had resulted in the ability to take an occam description of the system and produce a schematic which is easier to understand.

Geoff Barret of Oxford Universty PRG spoke of work which he had done in applying formal methods to a floating point number system. His method enables him to take an English language description of the problem, produce a mathematical specification, and by application of mathematical proofs and transformation laws arrive at a hardware implementation.

Gerald Johnson of Colorado State University spoke about a useful modification which could be made to CSP. He called the modification "Bidirectional I/O", as it combined the conventional "!" and "?" of CSP into a single statement: "$". This "$" construct had enabled him to propose more elegant solutions to matrix processing on processor arrays.

Falk-D Kuebler of Parsystec spoke of the aims and reasoning behind his company's Megaframe series of products (which incorporate transputers). The aim had been to produce a basic hardware and software system which would be easy to configure, easy to run, and allow field development beyond the host system. He finished by setting out the aims of his company for the future, one of which was to develop a descriptive language for industrial control, as he felt a procedural language like occam could be improved upon for this application area.

Christian Tricot of IMAG described how occam had been installed onto a Unix machine at Grenoble. He then spoke of certain non-standard extensions which he had built into the language in order to deal with time critical issues and process control.

Ronald Cok of Kodak Eastham gave a presentation on how he had used a transputer ring to carry out a Mersenne Prime calculation. He wanted to compare the performance of a transputer-based system with that of a supercomputer such as a Cray. His results showed that the Cray was fifty times faster than eight 20Mhz 32-bit transputers running in parallel. However, as he pointed out, the problem was in fact heavily biased towards the Cray.

Adrian Cockcroft of CCL gave the penultimate presentation it which he showed how a methodology developed by Tom DeMarco could be used in the specification and design of occam programs. As originally developed the methodology was designed to enable the software designer to go from the conception of a problem to a "front end" specification. However, in the context of occam it has been found that one can go right down to the implementation level.

Finally Miles Chesney of Meiko spoke of the "Computing Surface" which is a flexible computer system formed by a network of transputers. The network is intended to be application specific and it is therefore possible to alter the topology of the system. The system was then demonstated using a Mandlebrot series to generate a graphics display.

At the OUG business session it was announced that the next meeting had

been provisionally arranged at Loughborough for September 22nd and 23rd. However, many people felt that there had been insufficient time devoted to the Special Interest Groups and would have liked to have attended more than one SIG meeting. Also the absence of a panel session in which people could direct "Why didn't you ....." questions at David May and others was missed. To accommodate this future meetings may well be three days long. Hugh Webber also made an appeal for contributions to the occam software library.

The guest speaker at the conference dinner was Colin Skeltern, project manager of Flagship at ICL. He gave an interesting account of the efforts being made worldwide on Fifth Generation machines, referring particulary to the U.K. and ICL.

Our thanks must go to John Ainscough and Malcom Mosely for their on the spot organisation, also to the OUG Committee and to Inmos for their valued support of the meeting.

**Colloquium on: "The Transputer: Applications and Case Studies"**

by Nigel Kingswood, Bristol University

A colloquium organised by the Computing and Control division of the IEE was held at their London headquarters of the IEE on 23rd May. Entitled 'The transputer: applications and case studies' the colloquium varied from descriptions of hardware using transputers, through proposed hardware, to the suitable areas of application for transputers. Although the presentations covered several subjects there seemed to be three main areas which linked most of the speeches and the questions afterwards.

First was the problem of mapping a arbitrary problem onto a given arrangement of transputers. It was pointed out that for specific problems the mapping would be simple and direct because the hardware arrangement reflected the nature of the solution described in terms of occam processes. In a more general arrangement several people raised questions about how easy it would be to automatically map the arbitrary arrangement onto the available network of transputers.

The second area of discussion was on the nature of the connectivity of the transputers. In situations where the nature of the problem is known especially where most communication is on a nearest neighbour basis the optimum connection scheme can be used. It was argued by C Elliott that his experience had led him to believe that in such circumstances the performance was linearly related to the number of transputers. In more general situations two alternative approaches of connecting the transputers were described. The PARSIFAL system at Manchester Univ. described by A Knowles and the Meiko Computing Surface described by R Bottomley are reconfigurable to enable different arrangements to be studied. The FPS T-series computers described by M Baylis are permanently connected in a N-binary hypercube onto which a number of architectures can be mapped for vector computations.

Finally several speakers pointed out possible areas of applications. Richard Taylor of INMOS illustrated ways in which transputers could be used to introduce parallelism. C Jesshope and N Holt on the other hand pointed out some situations where the transputer and occam were not the

best means of creating parallelism. This is especially true where there needs to be global communication or dynamic parallelism.

## North American OCCAM and Transputer Meeting

by David Wolfram, Syracuse University

The first North American OCCAM and the Transputer meeting was held at the Sheraton University Inn, New York, on June 17, 1986. There were 82 attendees who came from 18 US states, 3 Canadian provinces, and Great Britain. Over 300 people were interested in the meeting but could not attend. They were from 36 US states, 4 Canadian provinces, and Australia.

The first session was chaired by John Oldfield from Syracuse University who introduced Bradley Strait, Director of the CASE Center at Syracuse University. Dr Strait made the welcoming address and described the New York State Center of Advanced Technology in Computer Applications and Software Engineering (CASE). It is a consortium of 16 universities and colleges with over 19 corporate sponsors and it aims to promote interdisciplinary and mutually beneficial advanced research in the areas of machine architecture, programming languages, artificial intelligence and software engineering.

The first speaker was Thomas Buckley from Leeds University, who is visiting Syracuse. He spoke on digital hardware simulation with occam and he found that difficulties in a simulation of synchronous hardware components by occam processes, and wires by channels, led to more complex models which used additional variables to represent voltages, and fanout processes to repesent broadcast signals. Two counter circuits were successfully simulated, but cross-coupled AND gates, for example, cannot be modelled because of further difficulties in representing transient voltages. More generally, Dr Buckley said that there is a lack of programming examples in Inmos VAX/VMS occam, the input and output routines are undeveloped, there is no detailed deadlock error report, and PROCs cannot be parameterised for passing arrays of differing lengths to them. However, the editor and especially the "folds" facilitated programming.

A.L.DeCegama, senior scientist at GTE Laboratories ten spoke on an approach to producing economical and accurate simulations for C3I, which is inplemented using occam and transputers. The approach is based on NETSIM, which is an event-table driven simulation with a feedback loop and a library of generic simulation programs, and DPSS (Distributed Process Simulation System), which is used to represent communicationg sinulation proceses by a partitioned undirected graph. Of several algorithms, backtracking to a checkpoint was used when a simulation table became full, because of its reportedly lower overhead. A fast and correct simulation of a computer network was completed with 12 B004 boards. OCCAM was considered an essential tool in this project, and faster transputers with more memory, and a FORTRAN compiler which allows floating point operations would also be useful.

The next speaker was Colin Whitby-Strevens, Manager of Microcomputer Support at Inmos, who outlined the transputer instruction set. It was designed for a von Neumann type architecture with an evaluation stack of five registers. Most instructions are word length independent and a byte long with two four bit fields. The PFIX, load local, and store local

instructions were described. The simulation of concurrency by a single transputer using two stacks for active and inactive processes was then presented. Process scheduling and descheduling were designed to require few processor cycles. Channels are implemented using either a special word in memory or a serial link. The T414-20 which operates at 20MHz is expected to be available in the next 3-6 months, and floating point instructions could be added to the instruction set.

After the coffee break, the second sesion was chaired by Simon Dolan, Microcomputer Field Applications Engineer at Inmos. He introduced Ronald Cok, research physicist at Eastman Kodak Research, who spoke on using the Lucas-Lehmer test for primeness of Mersenne numbers as a benchmark for estimating the performance of a ring of up to 20 T414 transputers, which could be used in image processing applications. The test is iterative and it involves a multiplication operation which can be equally destributed around the ring, and also a modulus, which can be achieved with an addition operation. As the size of the number tested increased, the optimal, predicted and observed performance of the test converged, and the overhead of communication between the processors decreased. For very large numbers however, the communication overhead beacme a significant limiting factor.

Colin Whitby-Strevens described occam 2, which includes multi-dimensional arrays, channel and integer typing, record types, type definitions, variant types (not in the beta release), and abbreviations to reduce array indexing. The beta release of the language is expected in the US very soon from "over the pond", and a beta release of the C compiler should be ready in about 4 weeks. Some new hardware is also under development and it includes a T212 16 bit transputer which operates at either 17.5 or 20 MHz, and a T414 which operates at the same rates and it has a doubled link rate, and should be available in the second half of 1986. There is also a B003 board which is designed to use 4 transputers.

Andy Rabagliatti from Inmos, and Simon Dolan then demonstrated some transputer graphics programs. They included a rapidly changing fractal pattern based on the rate of convergence of an iterative function of complex numbers. Each full screen image would have taken about 15-20 minutes using a VAX 11/780. Another example was a collection of "butterflies" on different monitors, two of which were in synchronous flight and one flew "down a channel" and alternated between screens. A third demonstration was a picture formed with ray tracing. There was an almost linear spped-up when eight transputers were used concurrently, rather than one.

The third session, following lunch, was chaired by Thomas Buckley, who introduced Richard Taylor form Inmos, Bristol, who spoke about research in European universities with occam and transputers. In the formal area, Professor C.A.R.Hoare and the Programming Research Group at Oxford University are investigating an algebraic and denotational semantics for occam, a normal form for occam programs, a high level program transformation system, and also proving the correctness of occam programs in a floating point library. In the area of concurrent programming, occam is taught in 25% of UK universities in courses on parallel languages. Inmos may release a low cost PC/transputer board for use in education. In system design, occam is used as a hardware description language for VLSI, in algorithm design at Strathcyde University, and in industrial control applications at Cranfield College. Transputers are also being used in the

Padmarati project, an Esprit funded project which aims to combine Prolog and LISP, and also in the Alvey funded ALICE project, which is adding unification to the fundamental HOPE language, as well as many other smaller projects. The transputer is also used in the Boltzmann machine, which uses statistical techniques to model neurons, and also ZAPP (Zero Assignment Parallel Processor) which is a project at the University of East Anglia, and it is expected to use 500,000 transputers. (Inmos has donated 4 or 5). Transputers are used in two major supercomputer projects: a $10M Esprit project, and the $4.5M Alvey supported Parsifal or T-rack project. Applications include high-energy physics problems, speech and vision programs, finite element analysis, and also VLSI, medical, molecular and flight simulations.

The next speaker was Michael Whelan, Group Leader of Advanced Design Tools at Siemens Research and Support. He spoke on porting existing code to a transputer based coprocessor for a SUN workstation which is used for schematic design. Of several approaches, the one selected was to modify the code generator of a "Portable C Compiler". The unusual transputer stack architecture and a lack of documentation about floating point instructions complicated this task. Although there are some restrictions on the use of short integers and double precision numbers, the compiler is now operational.

Richard Sheldon from Aerospace Control Systems at General Electric then spoke on an evaluation of the transputer for an autonomous I/O control unit in a main engine controller. Three T414 transputers were interconnected and the unit was programmed so that even with two link failures it remained operational. This was successfully tested by atually pulling out wires. Measurements of the transputer gave interchannel synchronisation of about 10 microseconds, and it required 2.5 milliseconds for a 512 word transfer, which was within the design requirements. The transputer was considered well suited for the application once the device and software matures.

After the tea break, Ronald Cok chaired the fourth session and he introduced Charles Stormon from Syracuse University who spoke on an architecture for logic programming using he transputer. A content addressable memory (CAM) coprocessor with a T414 host processor was described as a means of speeding up a Prolog translator by executing unification operations in the CAM. A simulation gave an expected performance of 500 KLIPS for the naive reverse benchmark. An encoding scheme for binary trees was also described and it can be used to retrieve parts of the tree in parallel. The CAM has been designed as a general purpose unit and it can facilitate the revocation of substitutions on backtracking.

Donna Bergmark, Assistant Director for Academic Computing at Cornell University, then spoke on generating occam from parallel Pascal. The aim was to write a compiler for a high level language with implicit parallelism. Parallel Pascal was selected partly because there are many existing programs in that language. It has parallel array processing with shift and rotate operations. Programs can be compiled in three main ways involving a VAX/UNIX, a STRIDE, or an FPS-T20. It is unnecessary for the programmer to know the number or identities of the machines for which the code is targeted. Input and output was designed to use one peripheral. A 150 line program takes at most about 5 minutes to compile. There were some difficulties with writing the compiler because of lack of floating point

operations and problems with the intermediate code for instance, but it was relatively easy because the target language was occam.

Richard Taylor then discussed the possibilities for the organisation of a US occam users group, and called for volunteers for its committee. The next US meeting is expected in about 6 months. An occam mailing list has been organised and messages for distribution can be sent to occam@syr-sutcase.csnet. Requests for changes to the mailing list can be sent to occreq@syr-sutcase.csnet.

The meeting was organised at Syracuse University by John Oldfield, Thomas Buckley, Charles Stormon and David Wolfram, with the assistance of Simon Dolan and Harold Blomquist form Inmos. The effective administration work was due to Peggy Vanarnam at University College, and Susan Craig at the CASE Center.

## NEW PRODUCTS AND SERVICES

### New transputer products

by Laurie Pegrum, INMOS

We have seen a very successful period for the transputer over the last six months in which a large number of occam programmers have been discovering the benefits of using transputers. In order to support the use of transputers INMOS is introducing a number of new products for transputer evaluation. These include new evaluation boards and new silicon products.

Those members of the user group who went to the well attended meetings of the OUG in Manchester or New York will probably have seen two new boards, IMS B003 and IMS B007.

The IMS B003 is the first multi-transputer board from INMOS that contains four T414 transputers each with 256kbytes of DRAM. This very powerful board has 2 links from each transputer brought to an edge connector and the others are configured in a square array.

The IMS B007 is a graphics board that shows off the power of the transputer in graphics applications. The board contains a single T414 transputer with 256kbytes of DRAM and 256kbytes of dual ported video RAM. The video RAMs pass the memory mapped pixel data to a 6545 CRT controller that drives the IMS G170 colour look-up table. This chip provides the red, green and blue signals to drive a colour monitor. The T414 can control special features of the G170 directly which has helped to generate that most popular of demonstration programs, the Mandelbrot set.

Both the IMS B003 and the IMS B007 are different from other boards from INMOS in that the only interface to the boards is via the INMOS serial links to the transputers. Thus these boards are add-on boards to those who have a development system already. Both these boards are available now in double extended eurocard format.

To facilitate the ease of use of double extended eurocards INMOS is introducing a new product called an ITEM, which stands for INMOS transputer evaluation module, of course. This is a sturdy rack for up to

10 boards with an ample power supply, sockets to hold boards firmly in place and air cooling. For those who wish to do large amounts of work on transputers INMOS is offering generous discounts for anyone buying an ITEM fully populated, a product called ITEM 300. This is aptly named as with ten IMS B003 boards today this small machine is capable of 300 MIPS !.

As well as new boards INMOS has been improving its silicon range. Development work on the T414 has meant that we can now run the links on discrete transputers at 20 MHz as well as at the standard 10 MHz. This also applies to the T212 16 bit transputer that has been available for some weeks now. INMOS is designing a board called the IMS B006 to demonstrate the T212 which can accomodate up to 9 transputers on a single card ! INMOS is already taking orders for the IMS B006 which should be available by August.

Future products include the M212 disk controlling transputer that INMOS has now seen working parts for and will be supplying in volume very shortly. INMOS is also introducing new improved link adaptors the IMS C011 and the IMS C012 that can perform at 20 MHz to keep pace with the latest transputers. These chips with new evaluation boards to come to demonstrate them make the future of the transputer range look very exciting.

As a final word we would just like to mention that INMOS is moving its Bristol operations to a new address just outside the city at a new Industrial park called Aztec West. The move is necessary to maintain the growth of INMOS. The move should be complete by the middle of July so if you have any further enquiries on the INMOS transputer products then please contact myself or my colleagues there for further information. We look forward to hearing from you.

## Floating Point Systems T-Series

Recently, Floating Point Systems of Beaverton, Oregon launched its massively parallel Vector Supercomputer - the FPS T-Series. The T-Series is a family of homogeneous parallel systems made up of nodes. Each node is a near supercomputer in its own right, consisting of a 7.5 MIP control processor - a transputer, a 16MFLOP 64 bit vector processor, a dual port video RAM and high speed communications channels.

The nodes are connected in a binary N-cube (or hypercube) architecture with a minimum configuration of 8 (2**3) computational nodes providing a peak speed of 128 MFLOPS. With this architecture, upgrades are made by doubling the number of nodes and in the case of the T-Series the nodes can be increased to 2**14 (16384) giving a peak performance of 262,144 MFLOPS (or 262 GigaFLOPS).

In such a massively parallel system it is not an easy task to control all the processors and to ensure the correct and consistent functionality of the total system. It is also a serious task to take an existing program written in FORTRAN or other high level languages and modify it to take advantage of the parallel processing capability. In fact even the development of new code split across many processors requires a great deal of extra coding to handle deadlocks between processors and generally keep an eye on every concurrent activity.

What FPS needed fot the T-Series was an implementation language that is a

powerful high level programming language, allowing for formal manipulation and in which parallel evaluation is natural and easy to achieve. The solution to this is course occam. With the transputer as the control processor of each node and occam handling internode communication as well as being an operating environment for the various processes which make up a job, life is made a lot simpler. OCCAM's ability to describe the job as a number of concurrent processes communicating with each other through logical channels easily maps onto the T-Series nodes and serial link architecture. OCCAM allows these processes to operate on a single node or on multiple nodes thus making the code configuration-independent. This simplicity and elegance of occam gives code compatibility across the entire T-Series range (which covers 2 orders of magnitude in power).

Concurrent programs can be, and ideally are, written directly in occam. OCCAM can also be used as a frame to link modules of code written in conventional languages such as FORTRAN, C and Pascal. In other words an occam process may be written in any of these languages.

OCCAM and the T-Series combine to create an elegant, powerful, parallel environment for the future of scientific computing.

Provided by: Bruce Jones, Marketing Manager
Floating Point Systems U.K. Ltd
Apex House
London Road
Bracknell
Berks RG12 2TE

tel: 0344 56921

**Transputer Products from Sension**

Sension Ltd, part of the Northwich based Sension group of companies, will shortly be launching its new range of Transputer Evaluation Systems. The Group is well established as a supplier of data communications equipment and custom microprocessor systems, many of which have been the brainchild of the Group's Technical Director, Andy Graham.

Andy has been interested in paractical applications of the transputer since it was first announced some years ago. At the time of the product launch last October, he started to explore the possibility of developing a product which would give potential transputer users the ability to explore the full capabilities of the device. The idea was to develop an evaluation package which could be supplied either as a single transputer system or as a small transputer network comprising two or more transputers, which was cased, powered and ready to use.

A chance conversation with Dr Graham Brookes (Acting Head of Computer Science at Sheffield University) earlier this year, convinced him that not only is there a market for a Transputer Evaluation System, but that collaboration between the resources of Sheffield and Sension would make good sense both technically and commercially.

From this point on, the ball was rolling. Sheffield University has considerable experience of using occam and were looking at possible applications of the transputer, including a transputer based educational

work station. The availability of occam 2 running on a transputer host opened up the possibility of developing an evaluation product incorporating up to four transputers, which, by using suitable low cost adaptors, could easily support popular computer environments.

For Sheffield, who needed to do occam development work on their Nimbus machines rather than IBM PCs, this was a most helpful development. This need has spurred the introduction of the initial product: a Transputer Evaluation System with a link adaptor board to provide connection to the Nimbus machine, together with occam 2 software and a clear, comprehensive Technical Manual. Sension now have this system in manufacture and plan to make first deliveries during August. A link adaptor board for the IBM PC will follow in September, and for other machines as demand dictates.

Sension have also become exclusive UK agents for **Parsytec GmbH.** Parsytec have recently introduced a range of transputer products which provide the system builder with a modular, high performance, parallel architecture, a free choice of topology and a practically unlimited number of processors.

Called the MEGAFRAME SERIES, the Megaframe itself is a desk top unit, capable of being connected to other Megaframe units to form extremely powerful local or distributed computing systems. Each Megaframe comes powered to cope with up to 10 plug-in modules which are of single height extended Eurocard form actor. Two types of module are currently available: Transputer Modules and I/O Subsystem Modules.

Parsytec can also provide "bus bridges" allowing the transputer to interface to other microprocessor buses and providing the access to the wide range of interfaces already designed for other uses. The occam 2 language and proramming environment is implemented on the Megaframe series, and although floppy disc based Megaframes are available, the hard disc versions are strongly recommended to potential users, particularly during the development phase.

Further Information from: John Brierly or Andy Graham
Sension Ltd
Denton Drive
Northwich
Cheshire CW9 7LU

tel: 0606 44321

**The Meiko Computing Surface: A Configurable Supercomputer**

by Roy Bottomley, Meiko

The computing surface is a flexible, extensible concurrent supercomputer. It exploits the latest VLSI technology including the Inmos transputer to provide virtually limitless computing capabilities.

A Computing Surface is formed by networking many computing elements in an application specific topology. Although it can be effectively argued that the computational core of an application could be mapped onto a regular structure, there will invariably be distortions at the edges of this core; an example would be funnelling transformed data into a Display Element. In addition the choice of which regular structure to use isn't clear and

there are many to choose from: rings, meshes, hypercubes, cylinders, trees, toroids. The best solution is to provide them all, and allow for the peripheral distortions. It is for these reasons that the Computing Surface was given the capability for electronic configuration. This allows a user of the Computing Surface to impose their own personality on it, and develop a truly optimal configuration suited to their application.

A Computing Surface is a self sufficient independent hardware process, with procesor, memory, high performance point to point communications channels and an optional function specific unit. The most primitive computing element implemented is a powerful 32 bit computer in ite own right. It consists of a T414 transputer with 256K bytes of error checked RAM, an interface to the configurable communication network, system error reporting and program debugging support.

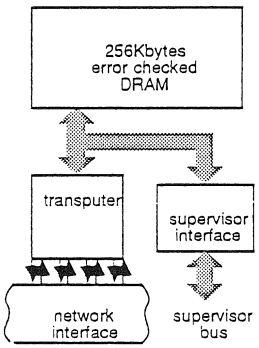
Other computing elements targeted at specific functions have been implemented, allowing the Computing Surface to be tailored to a specific application. Generally these elements perform input or output of some kind, such as a graphics display, although some elements address specific computational problems, such as database switching.

The Display Element is a variable resolution 1.5Mbyte frame buffer that is itself arrayable. The Display Element consists of a 32 bit transputer with 128 Kbytes of local store, 1.5 Mpixels of memory mapped frame store, a custom screen refresh engine, video timing controller, and three colour look-up tables. The display format is flexible and a single element's store can be arranged in any shaped display area of up to 0.5 Mpixels with 24 bits/pixel, or any area of up to 1.5 Mpixels with 8 bits/pixel. These areas can be split into two smaller areas for double buffered applications. In addition the pixel clock can be varied from 15 ns up to 120 ns. The bandwidth between Display Element and the rest of the Computing Surface is .10 Mbytes/s and drawing bandwidth into the frame buffer is 12 Mbytes/s. A 200 Mbyte/s pixelbus is provided for ganging multiple boards. This can be used to create a distributed frame buffer, to support multiple frame buffers, or to achieve the system's maximum resolution of 1300 X 1024 pixels with 24 bit/pixel screen resolution.

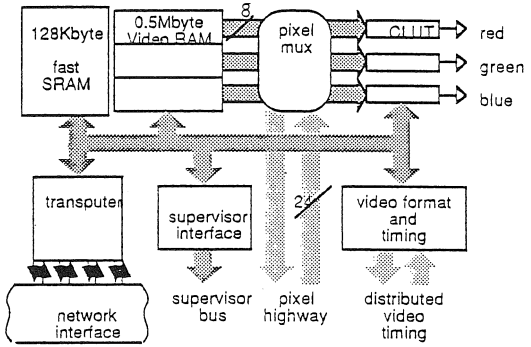
One element, termed the Local Host, has to be included to control the Computing Surface's system functions. In addition to providing I/O for the Computing Surface the Local Host's tasks include monitoring for hardware failures in any of the computing elements, controlling the configuration network, controlling the hardware reset and post-mortem analysis, handling run time errors and providing program debugging support.

The Local Host is capable of building a physical map of the entire Computing Surface. The map contains the gegraphical position and type of each computing element. Using this map and a custom piece of silicon supporting each transputer the Local Host configures the Computing Surface topology to a high level specification derived from the application program which is being loaded. This extends the boundaries of software and blurs the distinctions between software and hardware. As well as specifying the application program the software now specifies the machine upon which the application will run.

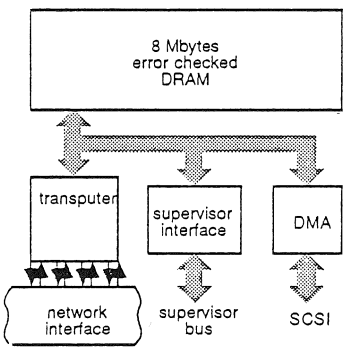
This map also aids hardware or runtime error tracing. An error from any computing element is immediately analysed, problems are pinpointed by positioning the program source editor at the appropriate line of code and



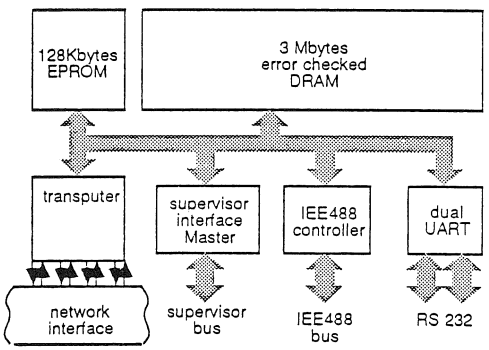

computing element




display element




mass store element




local host & I/O element

naming the process instance in which it occurred. In addition application diagnostic messages are given a guaranteed route to the console using a communication structure which is independent of, and orthogonal to the configurable network.

The last of the initial set of elements is the Mass Storage Element. This provides three levels of memory hierarchy. The 2 Kbytes of internal transputer memory for frequently accessed local variables, 8 Mbytes of 4 cycle, error checked dynamic RAM and a memory mapped 2 Mbyte/s DMA controlled SCSI interface. Multiple Mass Storage Elements may be used to form an intelligent database with local search, match and cache capabilities. Alternatively this element can be viewed as an enhanced memory computing element for use in applications involving large or unpredictable data structures.

The comuting elements are housed in a Computing Surface Module. Two sizes of Modules have been implemented to date, the first having a capacity to support in excess of 35 standard computing elements, the second having a capacity in excess of 150 standard computing elements.

An arbitrary number of Modules form a Computing Surface, providing computing resource ad lib. The electronic configuration applies to computing elements in different modules as if they were in the same module. Error reporting and program debugging support also extends between modules enabling a Computing Surface to be viewed simply as a collection of Computing Elements.

A Computing Surface Module can be populated with an arbitrary mixture of computing element types. Computing Surfaces are thus individually configured to the requirements of their largest applications. If we take just one large module and populate it with one Local Host and 156 Standard Computing Elements we have a machine with an aggregate of 1170 MIPS and 42 Mbytes of concurrently addressed dynamic RAM. The effective bandwidth to this store is 2.5 gigabytes/s, with a peak bandwidth of 9.5 gigabytes/s to the transputer's internal store. Populating the smaller Module in the same fashion yields a 250 MIP personal supercomputer, which is fully compatible and expandable to any scale Computing Surface.

(Reprinted with permission of the IEE)

## IBM PC Software

Users of the transputer board with the IBM PC may be interested in a BCPL toolkit and 8086 assembler. Also available is the Concurrent RunTime System based on monitors and written in BCPL as a ready made user interface for concurrent applications. For details contact:

Charles McLachlan
Graphical Software Ltd
3 Cambridge Place
Cambridge CB2 1NS

tel: 0223 312210/313934

## OCCAM Courses

Courses on occam (1 and 2) and the transputer are run by The Instruction Set. For details contact:

Simon Hawken
The Instruction Set Ltd
152-156 Kentish Town Road
London NW1 9QB

tel: 01 482 2525

## New Monograph - "The Laws of OCCAM Programming"

by A.W.Roscoe & C.A.R.Hoare

OCCAM has been designed with simplicity and elegance as major goals. One way in which this elegance manifests itself is in the large number of algebraic laws which exist between occam programs. The aim of the paper is to investigate the set of laws and to show how they completely chacterise the semantics of a large subset of the language.

The first section lists the majority of the laws we require. We see how each of the laws arises out of our informal understanding of how occam constructors work. We see how algebraic laws allow us to give a precise and succinct description of each operator. The laws given are all congruences in the denotational semantics for occam reported in ["Denotational semantics of occam," A.W.Roscoe, in LNCS 197 (Proceedings of the Pittsburgh Seminar on Concurrency)].

Much of the paper is concerned with the analysis of finite programs, allowing proof by induction. (A finite occam program is one which is WHILE-free; it may, however, contain the racing or diverging process, equivalent to WHILE-TRUE/SKIP). This restriction does not lose us any power, however, because every occam program can be identified with the set of its finite syntactic approximations (a term which is defined precisely in the second section).

The second section shows how the laws introduced in the first section can transform every finite program to a form whose only constructs are IF, ALT, multiple assignment and the diverging process. Particular attention is paid to regularising the use of free and bound variables. We see how this work, together with continuity assumptions, allows us to prove non-trivial laws additional to those of the first section.

The thir section completes this process and develops a normal form for finite programs. Two normal form programs are semantically equivalent if they are syntactically equivalent in a simple way. By showing how every finite program can be transformed to normal form we have thus produced a decision procedure for the equivalence of arbitrary finite programs. An infinitary rule basd on syntactic approximation extends this to general programs, so that the algebraic laws together with this rule give rise to an algebraic semantics for occam.

Finally, we review the relative merits of algebraic, denotational and other forms of semantics, and in particular discuss possible applications

of the algebraic laws to such area as comparing non-finite occam programs, converting sequential programs to parallel ones and to VLSI design.

"The Laws of occam Programming" is published by Oxford University Computing Laboratory as one of series of monographs on topics in computation. Copies are obtainable from the

Programming Research Group (Technical Monographs),
11 Keble Road,
Oxford OX1 3QD

at £2.50 inclusive

## ARTICLES

### An Implementation of OCCAM on a Distributed Computer
(abstract)

by A.J.Fisher, University of York

The programming language occam was designed to allow the specification of programs which comprise many parts, all working concurrently. OCCAM programs can be implemented on a wide variety of hardware, ranging from a conventional uniprocessor machine to a fully distributed machine in which each process runs on its own processor.

OCCAM programs have the property that the set of connections between proceses remains constant throughout the execution of a given program. Because of this, many occam computers contain hardwired connections between processors. This approach produces computers of restricted generality, which cannot readily be applied to more than a limited number of problems. By contrast, the computer I have built contains several "engines" connected in a ring. Although the physical connections between engines are fixed, the logical connections can be varied to suit the problem. An accidental consequence of this architecture is its high degree of fault-tolerance: faulty engines can be unplugged, and extra engines inserted, at will, and the system will reconfigure itself to take account of the changed number of engines.

### OCCAM in Higher Education

by Richard Taylor, Inmos

This is a broad brush survey of how occam and the transputer are being used in higher education in the UK. Its purpose is to show the extent of the occam user community, and the potential of occam to become a standard language over the next few years.

Teaching

One of the most obvious ways in which occam is being used is in teaching. This year a quarter of the universities and at least a sixth of the

polytechnics in the UK teach occam programming. That is, students write and execute occam programs. OCCAM is taught in classes as big as 60-70, and some places have been teching occam for the last 3 years. Interest is growing so that in the next year or so at least half the British higher education institutions will teach occam programming.

Such a large teaching effort immediately raises the question of why teach occam? There are three reasons, of which the most obvious is the desire to teach concurrent programming. However, occam has two other advantages for teaching. Because it is a simple language, the effort required to teach it is reduced. OCCAM also has formal semantics which can be used to teach the formal aspects of computing.

Apart form direct teaching, another form of training is stedent projects. This year about a third of universities have students doing projects in occam. These projects are larger programs which may take several months to write. Through student projects, universities are producing a stream of occam programmers who have more than a passing knowledge of the language.

Inmos will help by providing teching material and suggesting interesting projects. To receive this material write a letter to me asking for it.

Research

Almost every university and most of the polytechnics in the UK have shown a research interest in occam and the transputer. Research falls into three categories, of which straight transputer applications is the largest. The other two areas of research are formal aspects and concurrent programming. A list of research topics which I have been told about is given below. This list gives an impression of the breadth of interest.

| | |
|---|---|
| Artificial Intelligence | CAD Accelerator |
| Computational Physics | Database |
| Declarative Language Processor | Fault Tolerance |
| Flight Control | Flight Simulation |
| G P Parallel Processor | Graphics |
| Hardware Description Language | Hardware Design |
| Image Processing | Image Reconstruction |
| Inference Machine | Information Retrieval |
| Lattice Gauge Theory | Neural Networks |
| Operating Systems | Optimisation |
| Pattern Recognition | Process Control |
| Program Proving | Program Transformation |
| Real Time Control | Reconfigurable Systems |
| Robotics | Semantics of Parallelism |
| Signal Processing | Silicon Compilation |
| Simulation | Teaching |
| Syntax Analysis | Voice Recognition |

OCCAM User Community

The OCCAM User Group now has 700 members of whom only one third are academic. Three occam compilers have been witten apart form those written by Inmos. The occam portakit has been put on computers that range from Macintosh to 370. Many papers and monographs have been written and a number of books will be published this year.

The conclusion is that a real community of occam users exists in the UK and that higher education is training a body of occam programmers. These are signs that occam will become a standard programming language by the 1990s.

## Using De Marco Methodology in the Specification and Design of OCCAM Programs (abstract)

by Adrian Cockcroft, Cambridge Consultants

There have been several talks and papers on occam that have concentrated on coding techniques and design techniques; there have been relatively few talks on specification techniques. Unfortunately requrements analysis and specification of software systems is the hardest area to get right and has the most impact on the eventual success or failure of a software development project. This talk looked at the application of an existing methodology to the specification of an occam program.

The method is called Structured Analysis and was set out in 1978 in the book Structured Analysis and System Specification by Tom De Marco (Yourdon Press). This very readable book gives guidelines that cover the speciication process starting with the client and ending with the production of a structured specification. A traditional specification is likened to a Victorian novel, a solid mass of prose where the structure of the system is hidden beneath layers of unwanted detail. Two main problems exist with traditional specification. Firstly they are hard to read and understand, secondly they contain inconsistencies and ambiguities that are very hard to spot until it is too late.

A structured specification consists of several components. The structure of the system is described using a hierarchy of data flow diagrams using a simple notation so that non-specialists can understand what is happening. The lowest level processes in the data flow digrams are described using minispecifications written in English that encapsulate the details of the system. The data that flows in the diagrams is named and is formed into a data dictionary that shows the hierarchy of the data structure. The two underlying principles are that the specification should be readable and that there should be no redundancy. It is very easy to check a structured specification for inconsistencies, ambiguities and missing components. The important aspect of a structured specification for an occam programmer is that it can be viewed as a maximally parallel set of processes communicating via well defined data flows. This means that an occam program can in pronciple directly implement the specification and is a great improvement on the position with conventional languages, where the first step of Structured Design is to turn data flow diagrams into a sequential module calling hierarchy.

The talk illustrated the methodology using a program to generate Mandelbrot diagrams as an example. A second book by Tom De Marco called Controlling Software Projects (Yourdon Press) contains guidelines on estimating complexity and timescale metrics form structured specification.

## System calls for Occam [1]

C. Hazari, University of Bristol
H. Zedan, Teesside Polytechnic

Existing implementations of occam provide very limited access to system-level functions. Typically, these facilities only cater for parameter-line, screen, keyboard and file-handling. In this article, a methodology is described which can cater for occam-level access to system calls and library subroutines (e.g. GINO graphics calls). The work is based on experience with the Portakit implementation of Occam [2].

In current implementations of Occam, two types of channel are distinguished: 1) soft channels, or ordinary channels used by communicating processes, and 2) hard channels, used for memory-mapped i/o. The semantics of hard channels

```
CHAN hardie AT 100 :
```

instructs the compiler to associate a user-defined identifier (in this case 100) with the hard channel logical name. Subsequent input or output on channel hardie can specifically be trapped within the Occam kernel, since the unique id of 100 is used. For example, in the Portakit machine, with the declaration

```
CHAN screen AT 1 :
```

the statement

```
screen ! 65
```

results in the execution of an outword instruction with the machine's 'a' register containing 65, and the 'b' register word address carrying the unique channel id, 1. The trap in outword for a breg word address of 1 is activated, and this effects the writing of character 'A' to the screen. If a trap did not exist, the output on the channel would be treated in the same way as communication on a soft channel.

The hard channel tehnique used for screen handling can allow the occam programmer access to a larger than existing set of sub-occam-level services. For example, with the appropriate traps within the kernel, and with the declarations:

```
CHAN printer AT 20 :
CHAN HPplotter AT 21 :
CHAN Mouse_key1 AT 22 :
CHAN Mouse_key2 AT 23 :
```

the statements:

```
Mouse_key2 ? x,y
printer ! 'A'
HPplotter ! pen_down;0;0;1;1;pen_up
```

permit access at the Occam-level to the respective physical devices. Facilities for decoding, reformatting, buffering, device initalization and much more may also be provided within the traps to provide a much more 'attractive' interface to the service at the occam-level. The methodolgy can also cater for extension to and enhancement of the services already present. For example, file pointers could be controlled by Occam processes which are reading from, or writing to 100 files concurrently.

The major limitation of the technique lies in the interface between the

occam-level and the system-level: only 32-bit words (& 64 bits in occam II) may be passed. Since in general, system calls and subroutines return data structures which are not necessarily 32-bit (or 64-bit) words, decoding and reformatting of data passing across the interface will be necessary.

By way of illustration of the hard channel technique, and of the limitations imposed by the interface, we outline the implementation of

PAD_$CREATE_WINDOW (pathname,name_length,pad_type,unit,window
stream_id,status)

display system call for Aegis SR9 (an operating system for Apollo DOMAIN workstations), with reference to work on the Portakit implementation. This call creates a new pad and a window to view it.

We first assume that 2 had channels are reserved for use with this resource, e.g. channels 100 and 101. Channel 100 will be used for transferring the input parameters for the calls to the kernel, and the stream_id and status returned will be read back on 101. Next we examine the input parameters to the call. 'Pathname' is an array of up to 256 characters. 'Namelength', 'pad_type' and 'unit' are two-byte integers, with pad_type only allowed values 1, 2 or 3, and unit always set to 1. 'Window' is a record with two field, each 32-bits in length. The output parameters are 'stream_id', a two-byte integer, and 'status', a record with one relevant 32-bit field.

In order to make the call at the Occam level, the procedure

```
PROC PAD.CREATE.WINDOW (VALUE path[],namelen,ptype,w[]: VAR strid,stat) =
      CHAN to AT 100:
      CHAN from AT 101:
      SEQ
            SEQ i=[1 FOR 257]
                  to ! path[i]
            to ! namelen
            to ! ptype
            SEQ i=[1 FOR 3]
                  to ! w[i]
            from ? strid
            from ? stat :
```

is invoked, which returns both the stream_id for the window and the completion status for the call. Since 'unit' is a constant, as a matter of preference, the kernel alone is made responsible for passing it to the system.

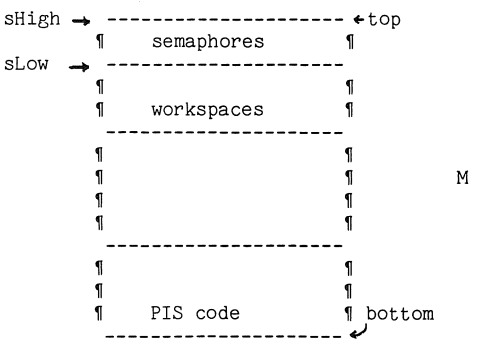
In the kernel, a state machine is implemented to support this system call. The machine repeatedly executes the cycle: fetch parameters; make call; return results. At the start of the cycle, the machine expects 256 32-bit words which are converted to ASCII and stored to 'pathname' (decoding, reformatting, buffering). 'Namelen' and 'ptype' are next assigned, followed by 'window', with type-matching provided across the interface. Once window has been assigned in the kernel, these values along with unit=1 are passed to PAD_$CREATE_WINDOW. A pad and window are created. Note that the implementor is free to determine the level(s) at which

errors (signalled in status, or detected in any parameter checks) will be trapped. Many variations on the use of channels are also possible. At one extreme, all system calls might be multiplexed on the same two channels, while at the other, each call might be available on several pairs of channels (only for resources which may safely be accessed concurrently).

The methodology described in the last section is not merely confined to the provision of system calls. Access can similarly be provided to library subroutines, e.g GINO graphics routines [6], and other programs which accept/return values from/to the caller. Hard channels may be defined and implemented which support communications between occam processes running on physically separated computers connected by a communications subnetwork [3,4]. Much more is possible.

Two important rules should, however, be enforced. These will be outlined with reference to some features of the Portakit implementation.

a) In Portakit, the machine is provided with a large array of 32-bit words which represent the machine's memory M.
b) The executable (PIS) code for the Occam application is loaded into this memory bottom-up.
c) Semaphores are used for soft channel synchronization. The soft channels are identified by the location of their semaphores in M.
d) Each process P is assigned a workspace Wp, the size of which is determined at compile time.
e) When the machine starts to execute the application, first the semaphores and then the process workspaces are allocated in M, form top to bottom.




Rule 1. Hard channel ids should not coincide with semaphores. In other words, for

CHAN <hard> AT x :

and for

M[sLow] =< semaphores =< M[sHigh]

ensure that

x < sLow

or x > sHigh.

Rule 2. For a hard channel which is intended for use as a soft channel (i.e. a hard channel for which no trap is effected within the kernel to prevent soft channel synchronization), the hard channel must neither coincide with the PIS code, nor with the process workspaces.
(Channels implemented in this way can permit run-time monitoring of communications between occam processes [3]. However, because hard channels are treated differently from soft channels, some implementations relax the normal semantic rules applied to channels, resulting in the deficiencies reported in [5].)

REFERENCES:
1. "Implementing system calls in Portakit Occam" C.Hazari, H.Zedan BP project report no.3, Jan. 1985, Univ. of Bristol
2. "Portakit Implementor's Guide" INMOS ltd., 1984
3. "TORIAX_d : User manual" C. Hazari, Computer Centre Report, University of Bristol, May 1986
4. "Implementing a distributed Occam" Barton, Dagless, Edwards, Hazari, Munro, Zedan, ITRC report 1986, Bristol Univ.
5. "Some experiences of Occam implementations" H. Zedan, C. Hazari, OUG newsletter no.3, p13
6. "GINO graphics for Occam" C. Hazari, H. Zedan, (in prep.)

## PARSIFAL: A Parallel Simulation Facility

P C Capon, J R Gurd, A E Knowles, University of Manchester

### Introduction

There is widespread belief in the computer industry that exploitation of software concurrrency in parallel hardware architectures is important for reasons of absolute speed (eg in supercomputers), reliability (eg in modular redundancy systems) and cost-performance (eg in systems constructed from replicated VLSI components). At present, there are no widely accepted criteria for selecting appropriate parallel architectures for particular applications. This is especially unfortunate given the enormous number of proposed parallel architectures, most of which are virtually untested in realistic use. Most work to date has concentrated on the so-called SIMD approach to parallelism, where a single machine instruction is capable of computing over large contiguous data structures. This has led to numerous high-performance vector and array processors, together with vectorisation algorithms to maximise their efficiency. However, the performance of SIMD is limited by technological constraints, and long-term architectural interest is moving towards the so-called MIMD approach, where multiple processors cooperate with one another to achieve high-performance that can be improved continually by expanding the hardware configuration. Such systems are beginning to appear in the commercial market, but techniques for using them are barely in their infancy. In order to make progress towards commerically viable parallel computers, considerable research effort must be expended in trying to improve practical knowledge and understanding of such systems.

This is a problem that is currently receiving great attention (for

example, the Defense Advanced Research Projects Agency in the United States is funding joint industrial and academic research in this area to the tune of $45 million per year). Two approaches to the problem are apparent. Some funds are being used to construct and evaluate specific styles of architecture, such as the BBN Butterly (US), the SUPRENUM supercomputer (West Germany), the ETL SIGMA-1 dataflow machine (Japan), the Flagship declarative system (UK) and the Connecion Machine (US). Other funds are being devoted to less specialised hardware that can be used for flexible architectural experiments that will not be geared to quite such high-performance as the specialised systems but which can be developed much more quickly. Examples of this latter approach are the Multiprocessor Emulation Facility at MIT (US). The Research Prototype Parallel Ptocessor at IBM Yorktown Heights (US) and the Parallel Simulation facility (UK) that is described in the remainder of this paper.

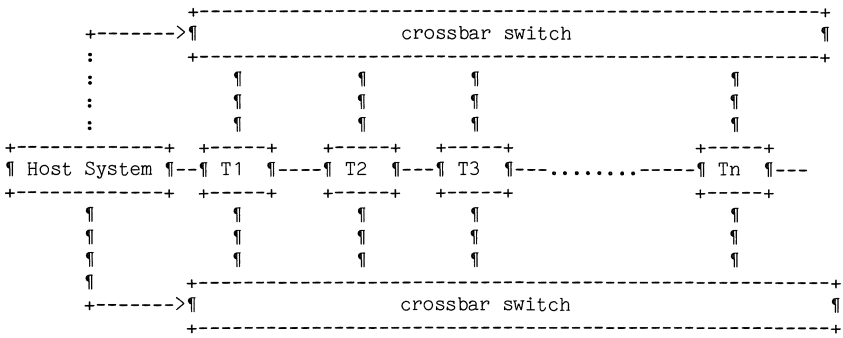
PARSIFAL is a collaborative industrial and academic research project that is being funded under the Alvey Programme in advanced information technology. The partners in the project are Cambridge University (Engineering Department), FEGS Limited, GEC (Marconi Research Laboratory), Inmos Limited, Logica (UK) Limited, the University of Manchester, and the Polytechnic of Central London. The project budget totals £3M and will involve 40 man-years of effort over a three year timespan.

System Characteristics

The important characteristics for an experimental emulation and simulation vehicle are its versatility (ie its ability to cope with a wide range of target architecures), its observability in operation, and its general usability (especially the speed at which new experiments can be performed). It has been decided that the PARSIFAL system will be targetted at message-passing multiprocessors in the first instance. the important features of these machines are their processor functionality and their interconnection topology. Consequently, the project has chosen to implement the simulation system using homogenous processing elements, with funtionality implemented in high-level software, linked together via a flexible, high-connectivity, reconfigurable interconnection network (as in the IBM RPP project). The aim is to provide an extensive set of high-level software tools that will facilitate architectural experimentation by allowing rapid prototyping of simulations and give comprehensive monitoring data on simulated program runs.

Hardware Architecure

To achieve the basic hardware characteristics, it has been decided to take advantage of the concurrent software environment afforded by the occam language implemented on the Inmos transputer. An extensible, reconfigurable array of transputers, known as the T-rack, is to be implemented using transputers (plus local memory) with some of their communication links interconnected via a switchable network, as shown in the following diagram:




The fixed horizontal path connecting all the transputers together is known as the 'necklace', and forms, in graph-theoretic terms, a Hamiltonian path between the computational nodes. The two crossbar switches provide high connectivity beyond that afforded by the Hamiltonian path. Expanded versions of the system can be constructed by replicating the 'n-transputer plus 2-crossbar' configuration the appropriate number of times.

Software Environment

The software environment surrounding the T-rack will provide sophisticated support for parallel program development (principally in occam), including tools for setting up complex emulation and simulation experiements and instrumenting system behaviour at different levels of abstraction. A particular innovation in this area will be the use of user-oriented graphical instrumentation of both abstract occam programs and the T-rack hardware. This interface is known as the Graphical Representation of Activity, Interconnections and Loadings (GRAIL). Less comprehensive behavioural predictions may be possible by coarse stochastic simulation of occam programs running on various hardware configurations. this area will be investigated by developing a specialsed transputer simulator known as TANSIM.

Applications Software

There will be two classes of application on the PARSIFAL system. The first class comprise the parallel architecture emulations and simulations, mentioned in the introduction that form the major justification for the project. The full extent of these experiments has yet to be determined, but work has already started on some candidates. for example, at the University of Manchester, an established research group has been studying the fine-grain dataflow approach to parallel computation for several years. Practical work has been confined to a single hardware processor (albeit with internal parallelism), althougha limited amount of software emulation of up-to-16 processor configurations has been undertaken. the power of the T-rack will allow much larger dataflow multiprocessor configurations to be evaluated without the delays associated with construction of further hardware. It will also be possible to set up entirely new 'soft' architecture experiments without the large initial effort associated with hardware construction. For example, one group at Cambridge University is interested in applying the T-rack to speech

recognition, using a 'Boltzman machine' architecture.

As mentioned before, these emulations and simulations form the raison d'etre for the PARSIFAL project. However, in its own right, the proposed T-rack hardware forms a powerful general-purpose coarse-grain message-passing multiprocessor, and it is envisaged that other applications will be evaluated on the system. For example, another team at Cambridge University will be developing programs for high-speed field analysis, in order to demonstrate the system capabilities. Other interesting possibilities arise when we consider running the programs forming the PARSIFAL software environment (for example, the occam compile and the TRANSIM simulator) on the T-rack, as example parallel applications programs.

Conclusions

High-performance parallel execution environments, together with sophisticated sofware support, are essential for further study of realistic parallel computing techniques. The wealth of conceptual proposals that need investigation militates against constructing specialised systems, favouring the development of 'wide' purpose emulation and simulation tools. the PARSIFAL project aims to exploit the concurrent power of the Inmos transputer plus the occam language to provide a suitable test-bed for future research in this area.

(Reprinted with permission of the IEE)

## OPTIMAL METHODS OF APPLYING Transputers IN LARGE SYSTEMS

Chris Jesshope , University of Southampton

1 Introduction

There are many ways in which large numbers of transputers may be combined into a single ensemble, to accommodate the concurrent processing of a single user job. The techniques used typically include:

i) Single or multiple shared bus systems:

ii) Shared memory with processor - memory switching:

iii) Regular networks with communications channels.

The shared bus approach in method (i) has little to offer unless the bandwidth of the bus is very high (ie many Giga bits per second). Bandwidths of this magnitude can be obtained, although only through using fibre-optic coupling. Even so, any approach which relies on the speed of a single shared device to provide parallel access to resources must be limited in its expansion capabilities.

The shared memory approach has been proposed for a number of multiprocessor systems both under construction and in production in the US, perhaps the most well known being the Denelcor HEP, which comprised up to 16 multitasking pipelined processors connected by a packet switched network to up to 128 memory modules. Each processor has access to the

entire memory space and in this design, synchronisation is provided by full/empty tag bits associated with each word in memory (this bit provide the same protocol as can be found across an INMOS link). However with large numbers of processors networks such as this network either becomes the dominant cost in the system or must further constrict the von Neumann processor-memory interface. For example, in another multi-processor design, the IBM CF11, in order to reduce the cost of the switching network, it has been implemented only eight bits wide, for a 32 bit processor. In general the wiring complexity of a full connection network goows as the product of processors and memories connected. The gate count, however, can be reduced to $n\log_2 n$ for n processors.

Point to point networks based on regular topologies have been used in many multiprocessor systems, including most processor arrays, such as Illiac 4, ICL DAP, Goodyear MPP etc. The advantage of this system of communication is that costs scale linearly with the number of processors in the system. The corresponding disadvantage is that if the implemented algorithms do not map well onto the topology of the architecture , then the communications bandwidth between processors must be shared and hence reduced. This sharing occurs when concurrent transfers are required between distant processors in the network. Also as the size of the array of processors increases the maximum distance through the network will also increase (Jesshope 80). These communications properties in regular networks will now be considered in more detail.

## 2 Communications

For a k-dimensional network the maximum distance through the network will increase as the k th root of the number of processors and in many parallel implementations of problems. (Hockney and Jesshope 1981). In these cases there is a relatively poor growth in performance as more processors are added to the system.

One measure that has been applied to quantify this problem in (Hockney and Jesshope 1981) is a parameter u which measures the ratio of processing rate to communications for problems which have a communications property over a fixed network of processors.

$$u = \frac{D_k \, w \, r_{\langle inf \rangle}}{2 \, m_{\langle inf \rangle}}$$

Where: $D_k$ is the maximum distance through the network
$r_{\langle inf \rangle}$ is the asymptotic processing rate
$m_{\langle inf \rangle}$ is the asymptotic communications rate and
w is the word width of the computational unit

In effect this measure reflects the sharing of bandwidth over a link, where path lengths are assumed to be one half of the maximum distance through the network. This corresponds for example to problems involving the reduction or distribution of data over the whole network and measures the expected degradation in performance to be expected from the communications overheads. This measure has been extracted from a number of array computers and is given in table 1 below.

Table 1 The Processing/Communications Ratio for Past and Future Machines

| Machine | Number of Processors | Network Topology | u | Operation type |
|---|---|---|---|---|
| Illiac 4 | 64 | 2-D NN Mesh | 3 | 64 bit Fl Pt |
| ICL DAP | 4096 | 2-D NN Mesh | 1 | 32 bit Fl Pt |
| ICL DAP | 4096 | 2-D NN Mesh | 10 | 32 bit Int |
| Cosmic Cube | 256 | Hypercube | 4-12 | 32 bit Fl Pt |
| F424 Trp Array | 1024 | 2-D NN Mesh | 32 | 32 bit Fl Pt |
| T414 Trp Array | 1024 | 2-D NN Mesh | 64 | 32 bit Int |

Obviously for problems that require through transport, a figure for u of between 1 to 3 seems reasonable, however, values of 10 and above must be viewed with some alarm.

This figure for transputer nets or indeed any other four connected network may be reduced by implementing a network with logarithmic distances, for example, the nearest-neighbour-shuffle network or butterfly networks. these both have maximum distance measures of log n and would reduce the figures in Table 1 for 2-D nearest-neighbour networks by a factor of $n^{1/2}/2\log$ , or 3 for the 1024 transputer array.

This still gives a communication dominated system. What is more, there will be less problems that can be mapped directly onto the underlying structure, meaning that a greater percentage of applications will be forced to use bucket brigade communications. Any communication on a Transputer will require a finite amount of processor resource to initiate and unless the granularity of the communication is large, communications and processing can not be modelled as concurrent processes. This overhead is equivalent to many operation times.

## 3 Algorithmic vs Geometric

Apart from the batch processing environment, where independent jobs can be distributed to the available resources in a system, in a producer consumer cycle, there are only two ways of exploiting the parallelism in a system: by distributing the data structure across the systems and executing (probably) the same code on subsets of that structure using data flow techniques, use the network to progressively obtain solutions to the problem (Algorithmic).

It can be seen that it requires transputers to be connected in 'Algorithmic Networks', which reflect the data flow of a given algorithm. It is also clear that this form of parallelism may be restricted to a relatively small scale, especially in the light of the grain size required in communications. What is required therefore is a mapping of transputers, which allows the exploitation of this form of concurrency at its lowest levels, but which can also be used to exploit the more conventional data structure sharing. It should be noted here that occam the Transputer's programming language describes both structure and algorithm.

Unfortunately the algorithmic networks required for different problems are likely to be problem specific.

4 The SuperNode an Optimisation

Unfortunately the algorithmic networks described above will vary with different problems. An implementation which requires direct communications between transputers to implement such networks would therefore need to be reconfigurable.

The Supernode is a scheme for utilising large numbers of Transputers which is inexpensive and which optimises communications in a wide range of applications. It achieves this by exploiting local algorithmic parallelism. Loosely speaking a Supernode is a cluster of transputers which can be considered as a Super-Transputer. It obeys an occam programming model by distributing occam code over the cluster of Transputers and realises a network of connections between those transputers using switching circuits. This structure is also obtained from the Occam programs.

Communications are optimised in a number of ways:

**i** Because the supernode contains many transputers it can be interfaced to other supernodes in the system through many more links. For example, if 16 transputers were incoporated into a supernode, then a total of 64 links could be brought to the 'outside'. This, of course, leaves the interior completely unconnected and typically some fraction of these links would be brought out for economy. However, this allows a richer interconnection between nodes in the system. Thus the increase communications bandwidth is balanced against the increase in processing rate, but the networks formed will have smaller maximum distance measures, thus decreasing u.

**ii** Because the supernode contains many transputers, the number of nodes in the network also decreases again decreasing the maximum distance measure. In addition to this there will be a decrease in communications bandwidth requirement between supernodes for many applications, using a surface/volume ratio.

Because the number of Transputers in a cluster is small and because the width of interconnection is narrow, the cost of the additional circuitry to implement the supernode is minimal. Moreover, because code is distributed, less memory is requred on each of the Transputers. Indeed it may be possible to utilise Transputers without external memory, thus reaping the speed benefits of processing totally 'on chip'.

5

R W Hockney and C R Jesshope (1981) "Parallel Computers" Adam Hilger Ltd ISBN 0-85274-422-6

C R Jesshope (1980) "Data routing and transposition in processor arrays" ICL Tech J 2 191-206

## BIBLIOGRAPHY UPDATE

compiled by inmos and the editor

1. Papers by INMOS

P. Wilson, "Highly concurrent systems using the transputer" **Northcon 84,** Mini/Micro Northwest-84 Conference Record, (Seattle, WA, Oct 2-4 1984), pp 13/2 1-11.

P.Wilson, "The Inmos IMS T424 transputer," ibid., pp 24/3 1-11.

R.Taylor, "Transputer communication link," **Microprocessors and** Microsystems, Vol. 10, No. 4, May 1986, pp 211-215.

R.Taylor, "Survey of transputer applications," **IEE Colloquium: The** Transputer: Applications and Case Studies, (London, 23 May 1986) IEE Digest No. 1986/91.

2. Papers by others

A.J.Fisher, **OCCAM on the Prime - a Manual for Users,** Report No. 83/3, Dept of Computer Studies, University of Hull, 1983.

J. Celko, "There's movement, but is there progress?" **Software News,** Vol. 5, No. 5, May 1985, pp 50-53.

S.Ohr, "For hefty computing power, European designers flex multiprocessor muscles," **Electronic Design,** Vol. 33, No. 17, July 25, 1985, pp 57,58.

G.M.Chemla, "The transputer: a high performance microprocessor or a component for fifth generation machines?" **Data Processing: Opportunities** and Drawbacks. Proceedings of Convention Informatique 1985, (Paris, Sept 16-20, 1985), Vol. B, pp 38-44.

"Inmos launches transputer," **Integrated Circuits Int.,** Vol. 9, No. 8, Oct 1985, pp 3,4.

"Chips with everything," **Home Comput. Adv. Course,** No. 93, 1985, pp 1849-1851.

J. Zalewski, "Language selection for modular interface systems. II," **Computer Physics Communications,** Vol. 38, No. 2, October/November 1985, pp 295-300.

A. Kemper, "Programming language OCCAM: introduction and comparison with PEARL," **Automatisierungstechnische Praxis,** Vol. 27, No. 11, 1985, pp 535-539.

J-P Feste, "Microprocessors: complexity with 32 bits, speed with parallelism," **Mesures** (Special Issue), No. 4, November 1985, pp 19,20. (In French)

Nicolas Mokhoff, "Two 32-bit microprocessors set new performance targets," **Computer Design,** Vol. 24, No. 15, 1 November 1985, pp 32,37.

A.J. Fisher, "Practical LL(1)-based parsing of van Wijngaarden grammars," **Acta Informatica**, Vol. 21, No. 6, 1985, pp 559-584.

Geoff Gardiner, "The INMOS transputer and the occam language," **Architectural Concepts**, Open University Press, 1986. ISBN 0 335 10864 4

C.Hazari, H.Zedan, **An Algorithm for Distributed Termination**, Tech. Rept. ITRC 79, University of Bristol, 1986.

A.J. Fisher, "A new algorithm for generating Hilbert curves," **Software - Practice and Experience**, Vol. 16, No. 1, January 1986, pp 5-12.

Nick Anning, "The transputer: out of the air and into action," **Computing** The Magazine, 23 January 1986, pp 3-5. (Reprinted in **Seismic Images: The** GECO Journal, April 1986, pp 4-9)

Harry Atkinson, "Configuring multiple CPUs," **Systems** International, Vol. 14, No. 3, March 1986, pp 48-50.

John Samson, "Simplicity itself," **Datalink**, 3 March 1986, pp 14,15.

Nick Anning & David Hebditch, "New chip displays its powers," **New** Scientist, Vol. 109,No. 1500, 20 March 1986, pp 43-46.

Mark Turner, "Number crunching on a mega scale," **The Engineer**, Vol. 262, No. 6786, 17 April 1986

Caroline Chappell, "Megabangs for minibucks," **Informatics**, Vol. 7. No. 5, May 1986, pp 46,47.

"Transputers in hypercubes beat Cray supercomputers," **Microsystem Design**, No. 10, May 1986, p 3.

N.Kingswood, E.L.Dagless, R.M.Bellchamber, D.Betteridge, T.Lilley, J.D.M.Roberts, "Image reconstruction using the transputer," **IEE** Proceedings, Vol. 133, Pt E, No. 3, May 1986, pp 139-144.

P.C.Capon, J.R.Gurd, A.E.Knowles, "PARSIFAL: A parallel simulation facility," **IEE Colloquium: The Transputer: Applications and Case Studies**, (London 23 May 1986) IEE Digest No. 1986/91.

C.Jesshope, "Optimal methods of applying transputers in large systems," ibid.

R.Bottomley, "The Meiko Computing Surface: a reconfigurable supercomputer," ibid.

N.P.Holt, "A characterisation of algorithms for parallel computation," ibid.

Nuala Morgan, "Transputer stretches the limits," **Computer Weekly**, 5 June 1986, pp 32,33.

"What Clive Sinclair is up to now," **Electronics**, Vol. 59, No. 25, June 23, 1986, p 56.

H.Zedan, "Distributed systems in occam," **Distributed Operating Systems:**

Theory and Practice (NATO Advanced Study Institute), 1986. (To appear in proceedings)

C.Hazari, H.Zedan, "A distributed algorithm for distributed termination," to appear in **Information Processing Letters.**

C.Hazari, H.Zedan, "On distributed termination in a network of communicationg processes," to be presented at 10th Microcomputer Applications Workshop, Strathclyde Univ., Sept 1986.

## MENBERSHIP LIST UPDATE

Members who have joined or moved between 10 Dec 1985 and 10 June 1986.

If anyone notices any errors in this list (or the previous ones) it would be appreciated if they would inform the secretary at INMOS.

S A Abdullah, 26 Melrose Ave, Reading, RG6 2BN.

Stephen Adams, Programmer & Research Student, University of Southampton, Dept of Computer Studies, 131-135 University Road, The University, Southampton S09 5NH.

Janusz Adamson, Thorn-EMI Electronics Ltd, Wookey Hole Road, Wells, Somerset, BA5 1AA.

David Alden, Co Founder, Meiko Limited, PO Box 323, Bristol, BS99 7XD.

Dr P J Allen, Senior Engineer I, British Gas Corporation, On-Line Inspection Centre, Nelson Ind, Est Cramlington, Norhtumberland NE23 9BJ.

Robert Allen, Graduate Student (Ph.D.), University of Manitoba, New Engineering Building, Dept of Electrical Engineering, Winnipeg, Manitoba R3T 2N2 Canada.

Jonathon A D Allin, Consultant Engineer, Syscon Ltd, The Innovation Centre, Cambridge Science Park, Cambridge, CB4 4GF.

Dr Arthur C Allison, Computing Science Dept, The University, Glasgow, G12 8QQ.

Osman D Altan, Senior Reseach Engineer, C M Research Labs, Project TRILBY, 1151 Crooks Road, Troy MI 48084, USA.

Dragan Andonov, Coventry (Lanchester)Polytechnic, Priory Street, Coventry, CV1 5FB.

Herbert S Anger, Research Assistant, University of Passau, Tunstr 27, 8390 Passau, West Germany.

Dr S J Anthony, Smith Associates Ltd, Surrey Research Park, Guildford, Surrey, GU2 5YP.

Arsac, Symbolic Architecture, Cimsa-Sintra, 20 me des Jardins, 92600 Asnieres.

Mr R P Aspinwall, 1 Rosstulla Drive, Jordanstown, Newtownabbey, Co Antrim, BT37 0QJ, N Ireland.

Peter Astbury, Consultant, London New Technology, Network, 86-100 St Pancras Way, London, NW1 9ES.

Sukhvinder Aujla, Fwd Intelligence Unit, British Telecom, Room 601, 12-15 Finsbury Circus, London, EC2M 7DP.

Mick Avis, Dept Computing & Cybernetics, Brighton Polytechnic, Moulsecoomb, Brighton, East Sussex, BN2 4GJ.

Mohammad Reza Bahramparuar, Research Student, University of Salford, Dpet of Electrical and Elec Eng, University of Salford, Salford, M5 4WT.

Mr J C Baker, Senior Software Engineer, Ferranti Computer Systems Ltd,

Bird Hall Lane, Cheadle Heath, Cheshire.
Mr P J Bakkers, Senior Lecturer, University of Stellenbosch, Dept Electronic Engineering, Stellenbosch, 7600, South Africa.
G P Balboni, Researcher, CSELT, Via G Reiss Roholi 274, 10148 Torino, Italy.
Mr Bealy, Opticon Computer Technology, 138 Park Hill Road, Bexley, Kent, DA5 1JA.
Dr W C Beattle, Senior Lecturer, Dept of E & E, Ashby Building, Stranmillis Road, Belfast, BT9 5AH.
Dr Mike Beetham, User Support Manager, Open Unviersity, Academic Computing Service, Walton Hall, Milton Keynes, MK7 6AA.
Steven Bell, 4 Dillons Ave, Whiteabbey, Co Antrim, N Ireland, BT 3705X.
N S Belshaw, Research Associate, Cambridge University, Dept of Earth Sciences, Downing Street, Cambridge, CB2 3EQ.
P Bentley, Logica, 104 Hills Road, Cambridge, CB2 1LQ.
Arnold Berlage, Project Leader, USDA, Dept of Ag Engr, Oregon State University, Corvallis, Oregon 97331 USA.
R Birch, Executive Engineer, Post Office Research, ED1.2.3. Room GE09, Wheatstone Road, Swindon, SN3 4RD.
Richard Bird, School of Behavioural Science, Newcastle Polytechnic, Newcastle Upon Tyne, NE1 8ST.
Michael Biscoe-Taylor, Smith Associates, Surrey Research Park, Guildford, Surrey GU2 5YP.
Prof Judy Bishop, Computer Studies Dept, The University, Southampton, SO9 5NH.
Jacques Boillat, Software-Schule Schweiz, Morgartenstrasse 2, Switzerland.
Mark Bowler, Research Assistant, Leicester Polytechnic, School of Electronic Eng, PO Box 143, Leicester LE1 9BH.
Tord Brabrand, R&D Scientist, Center for Industrial Research, Box 350, Blindern, 0314 Oslo 3, Norway.
John C L Brazier, Consultant, Brazier Computer Systems, 8 Valley Rise, Sarisbury Green, Southampton, SO3 6BN.
Steve Bridges, Dept of E & E Engineering, University of Salford, Salford, M5 4WT.
Michael J Bright, Hewlett Packard, Filton Road, Stoke Bishop, Bristol, BS12 6QZ.
R A Brook, Analyst, A E Joyce (Systems Studies) Ltd, Wolfe Lodge, Farnham Road, Bordon Hants, GU35 0NH.
James Brotton, Managing Director, Syscon Ltd, The Innovation Centre, Cambridge Science Park, Cambridge, CB4 4GF.
Martin Buehler, Graduate Student, Yale University, Yale Station, PO Box 2157, New Haven, CT 06520-2157 USA.
Mike Cahill, Transtech Devices Ltd, Unit 4, Queensmead Road, Loudwater, High Wycombe, Berks HP10 9TY.
Rosemary Candlin, Lecturer, Univ Edinburgh, Kings Buildings, Mayfield Road, Edinburgh.
Nick Carroll, 42 Church Road, Hayes, Middlx, UB3 2LJ.
Brian Carse, Lecturer, Dept of Engineering, Bristol Polytechnic, Coldharbour Lane, Frenchay, Bristol BS16 1QY.
Jack Cawkwell, 4 Yew Tree Gardens, Epsom, Surrey, KT18 7HH.
M Chakradorty, Senior Scientific Officer, National Engineering Laboratory, East Kilbride, Glasgow, G75 0QU.
W G Chen, Data Comm & Computer Networks, Computer Network Dept, Computer Centre, Xian Jiaotong University, Xian Shaanxi Province, The Peoples Republic of China.
M Chesney, Meiko Ltd, Southgate, Whitefriars, Lewins Mead, Bristol.

Kilnam Chon, KAIST, PO Box 150 Chongryang, Seoul, Republic of Korea.
Dr A R Clarke, Director, Instrumentation Software Ltd, 7 Gledhow Wood Avenue, Roundhay, Leeds, LS8 1NY.
Dave Clarke, Building R20, Rutherford Appleton Laboratories, Chilton, Didcot, Oxon, OX11 0QX.
Andrea Clematis, Research Associate, Inst Matematic Applicata, Consig Nazionale delle Ricerche, via L B Alberti 4, 16132 Genova, Italy.
Steven Clifford, Data Products Services Manager, Microlease PLC, Forbes House, Whitefriars Estate, Tudor Road, Harrow Middlesex HA3 5SS.
Dr B A Coghlan, Senior Lecturer, Dept of Civil Engineering, University of Western Australia, Nedlands 6009, Australia.
Marie Coleman, Computer Scientist, BP Research Centre, SPK 30, Chertsey Road, Sunbury-on-Thames, Middlesex.
G Collis, Dept of Computation, UMIST, Sackville Street, Manchester.
Dr Barry M Cook, Lecturer in Software Engineering, University of Surrey, Guildford, Surrey, GU2 5XH.
Dr Dick Cooper, Senior Lecturer, Dept of Computer Science, University of Canterbury, Christchurch, New Zealand.
Barry Corlet, Research Student, Dept Elec Eng & Electronics, University of Liverpool, PO Box 147, Liverpool, L69 3BX.
Jason Cozens, Dept of E & E Eng, University of Surrey, Guildford, Surrey, GU2 5XH.
P Crawford-Smith, Senior Design Engineer, Thorn EMI Electronics Ltd, Albert Drive, Sheerwater, Woking, Surrey GU21 5RU.
D E Crowder, Simulation Methods Manager, Plessey Major Systems Ltd, Sopers Lane, Poole, Dorset BH17 7ER.
David Crowe, Lecturer, Maths Faculty, The Open University, Maths Faculty, Open University Walton HalL, Milton Keynes MK7 6AA.
A Culloch, Lattice Logic, 9 Wemyss Place, Edinburgh, EA3 6DH.
Phil Culverhouse, Lecturer, Plymouth Poly, Plymouth, Devon, PL4 8AA.
Charles Cunningham, Control Intellect Ltd, Grotonwood Cottage, Kersey Tye, Ipswich, IP7 6HD.
Per Arne Dahl, OTE A/S, Postbox 102 Treita, 0617 Oslo 6, Norway.
J Daniels, Senior Research Engineer, Short Brothers Ltd, Montgomery Road, Belfast, BT6 9HN, Northern Ireland.
S G H Daniels, Senior Research Adviser, ACS Open University, Walton Hall, Milton Keynes, MK7 6AA.
Craig Davidson, Gryphon Systems, 922 Grange Hall Road, Cardiff CA, 92007, USA.
A T Davis, Research Engineer, Salford University, Dept of Aero and Mech Eng, The Crescent, Salford, Lancs M5 4WT.
P Dawe, Cambridge Micro Computers, Cambridge Science Park, Milton Road, Cambridge, CB4 4BN.
A Delima, 214 Queens Road, Leicester, LEs 3FT.
S.V. Demiralp, Computer Center, Middle East Tech Univ, Ankara, 06531, Turkey.
M. P H Deslande, Electronic Engineer, CGA/HBS, 3 Rue De Tunis, BP 111 93203, St Denis Cedex 1, France.
Mr Dholiwar, GEC Eng Res Centre, Cambridge Road, Whetstone, Leicester, LE8 3LH.
P Dickinson, Laserscan, Science Park, Cambridge, CB4 4BH.
Dr N A Dodd, Principal Engineer, Systems Studies Group, Smiths Industries, RPT Dept, CH 15 Bishops Cleeve, Cheltenham Glos GL52 4SF.
Richard D'Olejnjk, Universite Des Sciences, Et Techniques De Lille I, U.E.R. d'I.E.E.A., Laboratoire d'Informatique, 59655 Villeneuve D'Asco, Cedex France.
Frank Domoney, Girasol Logic Ltd, 44 Harborough Road, Southampton, SO1

2FY.
Jim Donaldson, System Specialist, Pioneer Technologies, 3100F Northwoods Place, Norcross, Geogia 30071, USA.
Dr Joerg Donandt, AEG Forschungsinstitut, Hollaenderstr, 31-34, D-100, Berlin-51, Germany.
M Dosanjh, 310 Bedfont Lane, Feltham, Middlesex, TW14 9NU.
C F Dunn, The University of Liverpool, Department of Computer Science, Chadwick Building, PO Box 147, Liverpool, L69 3BX.
Steve Dye, Engineer, Syscon Ltd, The Innovation Centre, Cambridge Science Park, Cambridge, CB4 4GF.
W Edmonson, Centre for Computing &, Computer Science, Birmingham University, PO Box 363, Birmingham, B1J 2TT.
M D Edwards, Dept of Computation, UMIST, PO Box 88, Sackville St, Manchester.
Miss S Eghtefadi, King's College London, Latymers, Westmead, Roehampton, London, SW15.
A W Elbourn, PC Specialist, British Aerospace, Brooklands Road, Weybridge, Surrey, KT13 0SF.
Chris Elliot, Smith Associates, Surrey Research Park, Guildford, Surrey, GU2 5YP.
Dr David Ellison, Lecturer, Dundee College of Technology, Bell Street Dundee.
Je H Elman, Phonetics Laboratory C008, University of California, San Diego, La Jolla, California, 92093 USA.
Mary Fagan, Compiler Editor, 14 Manfred Court, Manfred Road, Putney, LONDON, SW15.
A J Fenwick, Principal Systems Engineer, Thorn EMI Electronics Ltd, Sheerwater Industrial Estate, Albert Drive, Woking, Surrey GU21 5RV.
Ion Filotti, Universite de Paris-Sud, Centre D'Orsay, Laboratoire de Recherchi, En Informatique, Bat 490-91405, ORSAY Cedex France.
Dr A.J. Fisher, University of York, Dept of Computer Science, York, Y01 5DD.
P Fitch, Dept Clinical Neuro-Phys, London Nat Hospital, Queen Square, London WC1.
John W Flanigan, Senior Computer Engineer, Queen's University Belfast, Dept of Computer Science, College Park, Belfast, BT7 1NN.
Prof F Foeri, Lab de Signaux et Systems, 41 Boulevard Napoleon III, 06041, Nice Cedex, France.
B Forrest, Dept Physics, Edinburgh Univ, Kings Bldg, Mayfield Road, Edinburgh.
N France, CAP Scientific, 20-26 Lambs Conduit St, London, WC1N 3CF.
Stephen Gaines, Applications Engineer, Rapid Recall, Rapid House Denmark Street, High Wycombe, Bucks, HP11 2ER.
Ken Gale, Ferranti Computer Systems Ltd, PO Box 5, Weymouth, Dorset.
Ronald Garrison, Future Proof Methods, 4301 Raleigh CT.£611, Midland, TX, 79707, USA.
Joh Gilli-Ross, Marconi College, 102 Arbour Lane, Chelmsford, Essex, C M1 5RN.
Professor Claude Girault, Laboratoire MASI, Universite Pierre et Marie Curie, 4 Place Jussieu, 75252, Paris Cedax 05, France.
A Goddard, Centre for Inf Studies, City Univ, Northampton Square, London EC1V 6HB.
Christian De Godzinsky, Electronics Designer, Planmeca OY, Mekaanikonkatu 5, 00810 Helsinki, Finland.
F Gomes-Molinero, Electronic Engineer, STEC/European Space Agency, Kiplerlaan 1, 2201 AZ, Holland.
Howard Goodman, Research and Marketing, Station Box 7157, New Haven, CT

Ave, Toronto, Ontario MeK 2Z1 Canada.
Ian Gorton, Sheffield City Polytechnic, Pond Street, Sheffield, S1 1WB.
A M Graham, Technical Director, Sension Limited, Denton Drive, Northwich, Cheshire, CW9 7LU.
Roger Graham, Senior Lecturer, Manchester Polytechnic, Dept of Computing, John Dalton Faculty, Chester Street, Manchester M1 5GD.
Daniel Grandjean, Project Leader, Institut C.E.R.A.C. SA, Centre Europeen De Recherches, d'Atlas Copco, Chemin des Larges Pieces, CH-1024 ECUBLENS Switzerland.
Dr Green, Engineering Dept, Wigan College of Technology, Parsons Walk, Wigan, Lancs, WN1 1RR.
Dr Martin Green, Senior Software Engineer, Ferranti Computer Systems Ltd, Bird Hall Lane, Cheadle Heath, Stockport, SK3 0XQ.
E Gresset, Thomson CSF, Labs Electronique De Rennes, Avenue De Bell Fontainne, 35510 Ledson, Sevign, France.
M. P Guebey, Project Manager, CGA/HBS, Rue De Tunis, BP 111 93202, St Denis Cedex 1, France.
Barry Gumbert, Engineer, R R Donnelley & Sons, 750 Warrenville Road, Lisk Il, 60532, USA.
J Gurd, Dept of Computer Science, Manchester Univ, Manchester, M13 9PL.
Erik Hagersten, Telefonaktiebolaget LM Ericsson, S126-25, Stockholm, Sweden.
Prof Dr Winfried Hahn, Universitat Passau, Postfach 25 40, D-8390, Innstrasse 27, Passau, Germany.
G Hall, Senior Lecturer, Dept of Elec Eng, Lancashire Polytechnic, Preston, PR1 2TQ.
Ivan Hallworth, Director, Logic Integration, 4 Gloucester Road, Reading, Berkshire, RG3 2TH.
Martin Hancock, Research Assistant, Philips Research Laboratories, Cross Oak Lane, Redhill, Surrey, RH1 5HA.
Martin Hargreaves, Senior Lecturer, Leicester Polytechnic, School of Electrical & Elec Eng, PO Box 143, Leicester, LE1 9BH.
Stephen Harrison, Computer Lab, Cambridge University, Cambridge, CB2 3QG.
Mr L Haton, Managing Director, Programming Research Ltd, 11 Carlton Road, New Maldon, Surrey, KT3 3AJ.
M W Helzle, Entwicklung, Hema, Adlener Str 64, 7080 Adlen, Unterkochen, West Germany.
Ahmed Hemani, ASEA Industry & Electronics AB, Dept ILKHX, S-721 67, Vasteras, Sweden.
John D Hestenes Ph.D, Jet Propulsion Laboratory, California Inst of Tech, 4800 Oak Grove Drive, Pasadena, California 91109, USA.
Dr T R Heywood, IBM UK Scientific Centre, Athelstan House, St Clement Street, Winchester, Hants, SO23 9DR.
H S Hicks, Dept of Computing, Manchester Poly, Chester St, Manchester M1 5GD.
Walter Hicks, Proprietor, Hicks Software Services, 45 Whittaker Lane, Prestwich, Manchester, M25 5HA.
Paul Higham, Senior Lecturer, Bromley College of Technology, Rookery Lane, Bromley, BR2 8HE.
Sachio Hirokawa, Shizuoka University, Faculty of Engineering, 5-1 Johoku 3 Chome, Hamamatsu, 432 Japan.
Nigel Holder, Marconi Research, Room A100, Great Baddow, Chelmsford, Essex, CM2 8HN.
Andrew Holman, Topexpress Ltd, 13-14 Round Church Street, Cambridge, CB5 8AD.
Dr G T J Hooper, Senior Consultant, GEC Avionics, 317 Division, Borehamwood, Herts.

Sinisa Hristov, Development Engineer, W O Elektrouniverzal, Marsala Tita Avenue 147, Yu-16000 Leskovac, Yugoslavia EU.

J C Hughes, Principal Consultant, Logica Cambridge Ltd, Betjeman House, 104 Hills Road, Cambridge, CB2 1LQ.

Ir W Huiskamp, FEL-TNO, Oude Waalsdorperweg 63, Den Haag, 2597 AK, Holland.

Jinho Hur, Research Staff, Dept of CS/KAIST, PO Box 150 Chongryang, Seoul 131, Republic of Korea.

Dr R C Hutty, Principal Lecturer, Leicester Polytechnic, PO Box 143, Leicester, LE1 9BH.

Zvonko A Ilic, Sretena Mladenovica 34/26, 18000 NIS, Yugoslavia.

Peter Ilieve, Memex Information Engines Ltd, 16 Albion Way, Kelvin Industrial Estates, East Kilbride, G75 0YN.

George Irwin, Electrical & Electronic Eng, Ashby Building, Queens University, Belfast, BT9 5AH, Ireland.

J W M Jacobs, OCE Nederland BV, St Urbanusweg 43, Venlo, 5900 MA, Nederland.

P J Jinks, Lecturer, University of Manchester, Dept of Computer Science, Oxford Road, Manchester, M13 9PL.

Christopher Jobson, Research Engineer, Information Systems Dept, Thorn EMI Central Research Labs, Dawley Road, Hayes, Middlesex UB3 1HH.

Mr A W Joines, PTO II, SERC Rutherford Appleton Lab, Chilton, Didcot, Oxon, OX11 0QX.

Jan E Jonak, Sperry Network Services, 45 Crawford Place, London, W1.

K B Jones, Software Section Leader, Thorn EMI Electronics, Mercury House, 252 Blyth Road, Hayes, Middlesex UB3 1HW.

De Joybert, Cimsa-Sintra, 20 me des Jardins, 92600 Asnieres, France.

Blomster Kaj, Field Applications Engineer, Field OY, Veneentekijantie 18, 00210 Helsinki, Finland.

M Karhill, Transtech.

John Karwatzki, 37 Norbiton Avenue, Kingston, Surrey, KT1 3QY.

Eamon Kelly, Dept of Computer Sciences, Trinity College, Dublin 2, EIRE.

Dr R D Kenway, Department of Physics, University of Edinburgh, James Clark Maxwell Building, The King's Building, Mayfield Road, Edinburgh EH9 3JZ.

Michael Kergozou, Software Engineer, Teradyne Ltd, 5 Gipsy Lane, Wokingham, Berkshire.

Tony King, Computer Lab, Cambridge University, Cambridge, CB2 3QG.

Graham Kirsch, Stonefield Electronics Ltd, Lawson Hunt Industrial Park, Guildford Road, Broad Bridge Heath, Horsham, W Sussex RH12 3JR.

Lothar Klein, Sachsenstrasse 3, 6400 Fulda, West Germany.

Toshio Komatu, Processing Equipment Section, NTT Commun & Info Processing, Laboratories 1-2356, Take, Yokosuka, Japan 238-03.

V Korotchenko, AMT Group, Chall Instruments Ltd, Tudor House, Grammar School Road, North Walsham, Norfolk NR28 9JH.

Zdenek Kosina, Nuclear Physics Institute, CS-250-68 REZ, Czechoslovakia.

Goran Kostic, Development Engineer, W O Elektrouniverzal, Marsala Tita Avenue 147, Yu-16000 Leskovac, Yugoslavia EU.

Peter Kropf, Assistant, Inst Informatics & Appl Maths, University of Bern, Langgasstr 51, CH-3013 Bern, Switzerland.

Tom Lake, Technical Director, Intercept Systems Ltd, 59 Alexandra Road, Reading, Berks, RG1 5PG.

Martin Leach, 146a Blythe Road, London, W14 0HD.

W Lennox, 101 Nibley Road, Shirehampton, Bristol.

M G Lequard, Prosee Manage, CGA HBS, Rue de Tunis, BP111 93202, St Denis CEDEX1, FRANCE.

S E Lewis, Section Leader, Thorn EMI Electronics Ltd, Mercury House, 252

Blyth Road, Hayes Middlesex, UB3 1HW.
Terry Lillie, Manager-Mechanical Computational, Accelerator Group, Calma Company, 525 Sycamore Drive, Milpitas, CA 95035 C51-Q USA.
Dr A D Linney, Principal Physicist, University College Hospital, Medical Physics Dept, 11-20 Capper Street, London, WC1E 6AJ.
Adam Lucas, Technical Director, Spectre Graphics Ltd, 189 Freston Road, London, W10 6TH.
M A Lynch, CAD/VLSI Group, Thorn EMI, Dawley Road, Hayes, Middx.
Peter Lynch, Technical Consultant, Prime Computer (R&D) Ltd, 70 Bramley Road, Mount Farm, Industrial Estate, Bletchley Milton Keynes.
David M Lyons, Lecturer, Computing Science Dpet, University of Essex, Wivenhoe Park, Nr Colchester, Essex CO4 3SQ.
Carl McCrosky, Assistant Prof, Dept of Computer Sci, University of Sask, Saskatoon, Canada, S7N 0W0.
Mr McCullagh, School of Info Systems, Univ of East Anglia, Norwich, NR4 7TJ.
Angus McDonald, Software Sciences, Mendon Ave, Farnborough, Hants, GU14 7NB.
Hector McNeil, 15 Avenue Tourists, Woluwe St Pierre, 1150 Brussles, Belgium.
Martin McQuillan, Product Engineer, Floating Point Systems Irl Ltd, Balheary Industrial Park, Swords, Co Dublin, Ireland.
Martin McQuillan, 7 St Columba Heights, Swords, Co Dublin, Eire.
Francis T Marchese, Professor, Pace University, Computer Science Department, New York, 10038, USA.
J Marshall, Lecturer, Dept of Engineering, University of Warwick, Coventry, CV4 7AL.
Richard M Marshall, Post Graduate Student, University of Edinburgh, Dept Computer Science, JCMB, Kings Building, Edinburgh EH9 3J2.
Harchand Singh Matharu, Design Engineer, British Telecom Fulcrum, OS 3.2 "B" Block, Fordrough Lane, Birmingham, B9 5LD.
Bob Mathews, CHASE, Tredean House, Devauden, Gwent.
J Meadows, ICL, Wenlock Way, West Gorton, Manchester.
Mr Meriaux, University of Lille, Bat M3-USTLFA, 59655 Villeneuve, D'Ascq Cedex, FRANCE.
S Metic, Boris Kidric Inst, Vinca, Comp Sys Design Lab, PO Box 522,11001, Belgrade, Yugoslavia.
Riveill Michel, Univ of Gremoble, IMAG - LGI, BP68, F-38402, Saint Martin D'Heres, France.
O J Miles, Bristol University, University Walk, Bristol.
Richard Miller, Programming Research Group, Oxford University, 8-11Keble Road, Oxford.
M Mimshdi, Computer Science Dept, City Univ, Northampton Square, London EC1.
Tomoyuki Minamiyama, Engineer, Tokyo Comm Edquip Co Ltd Lab, 3-484 Tsukagoshi Saiwai-Ku, Kawasaki, Kanagawa-Pref, Japan 210.
Majid Mirmehdi, Computer Science Dept, City University, Northampton Square, London EC1.
Jeff Mock, Pixar, PO Box 2009, San Rafael, California, 94912, USA.
A J Mocking, Thorn EMI, Radar Division, Radar House, Dawley Rd, Hayes Middx, UB3 1HN.
S Mohammed, Managing Director, SLM Associates, 39 Holmewood Gardens, London, SW2 3NA.
M H A Mommers, Programmer, Eurocontrol, Horsteieweg 11, Gig 1 RX Beek (L), 619RX, The Netherlands.
Francois Monnet, CGE Labs, Raute de Nozay, Marcoussis, 9146D, France.
Mr C M Moreton-Smith, Scientific Officer, SERC Rutherford Appleton Lab,

Chilton, Didcot, Oxon, OX11 0QX.
Philip Morrow, Research Officer, The Queens University of Belfast, Department of Computer Science, Belfast, BT7 1NN, Northern Ireland.
Mr M Morse, Elec Eng Dept, Leicester Poly, PO Box 143, Leicester, LE1 9BH.
John Mortimer, Senior Lecturer, Computer Studies & Mathematics, Bristol Poly, Coldharbour Lane, Frenchay, BS16 1QY.
D I W Mullner, Inst Statistik und Informatic, Universitat Wien, Lenaugasse 2/8, A-1080 Wien, Austria.
Dr F Naghdy, Senior Lecturer, Postsmouth Polytechnic, Dept Electronic Eng, Anglesea Building, Portsmouth, PO1 3DJ.
Herr Neumann, Technologieberatungszentrum, Zippelhaus 5, 2000 Hamburg 11, Germany.
David Newton, Hewlett Packard Ltd, South Queensway, West Lothien, EH30 9TG.
T H Normanton, Software Research Engineer, Smith Industries, Bishops Cleeve, Cheltenham, GL52 4SF.
Terry A Nuzzolo, Design Engineer, RCA, 68D, PO Box 588, Mail Stop 9-1, Burlington, MA 01803 USA.
Dr E-R Olderog, Institut Fur Informatik, Und Praktische Mathematik, Christian-Albrechts-Univeritat, 2300 Kiel 1 den, Olshausenstrabe 40 Haus S 21, West Germany.
Stephen J Oldham, Senior Software Engineer, Ferranti Computer Systems, Bird Hall Lane, Cheadle Heath, Stockport, SK3 0XQ.
Pat O'Leary, Technical Director, Micro Marketing, Glenagery Office Park, Glanagery, Co Dublin.
B Oliver, ITCU, 189 Freston Road, London, W10 6TH.
Alex Osadzinski, Sun Microsystems Europe, Berkshire House, High Street, Ascot, Berks, SL5 7HU.
Dr C Osgood, Research Officer, CEGB, Berkeley Nuclear Labs, Berkeley, Glos, GL13 9PB.
Jamie Packer, INMOS Int, Room 308 Kowa No 16 Annex, 9-20 Akosaka 1-chome, Minato-ku, Tokyo, Japan.
G S Panesar, Research Fellow, Southampton University, The University Highfields, Southampton.
Seung-Kyu Park, Senior Engineer, Electronics & Telecom Res Inst, PO Box 8, Dae Dog Dan Ji, Chungnam, Korea (S).
Prof A Parkin, Leader MCS Research Unit, Leichester Polytechnic, PO Box 143, Leicester, LE1 9BH.
Martin Parolow, Manger Instrument Dewmat, AMT Group, Chall Instruments Ltd, Tudor House, Grammer School Road, North Walsham Norfolk NR28 9JH.
Gurudatta Parulkar, Research Assistant, University of Delaware, Dept of Computer Science, Smith Hall, Newark, DE19716 USA.
Mr H Pathak, Analyst, Distributed Systems Planning, American Express, 154 Edward StBr, BN2 1YT.
M. P Patry, Software Engineer, CGA/HBS, Rue De Tunis, BP 111 93203, St Denis Cedex 1, France.
Dr Emil Petriu, Assistant Professor, University of Ottawa, Electrical Engineering Dept, Ottawa, Ontario, K1N 6N5 Canada.
P S Phillips, Manager, Blaenau Gwent Technology Centre, Tafarnaubach Industrial Estate, Tredegar, Gwent, NP2 3AA.
Vincent Pierre, Service Informatique, Inst Nat de Telecomms, 9 Rue Charles Fourier, 91011 Evry, FRANCE.
T S Plews, Group Leader, British Aerospace, FPC 119, PO Box 5, Filton, Bristol BS12 7QW.
Anthony D Pollard, Engineer, Dexter Electronics, 162 High Street, Stevenage, Hertfordshire, SG1 3LL.

Adam Postula, ASEA AB, ILKHX, 72167 Vasteras, Sweden.
Dr J H Pote, Computer Officer, University of Birmingham, Dept E & E Egineering, PO Box 363, Birmingham, B15 2TT.
Dr W C A Pulford, Senior Scientific Officer, SERC Rutherford Appleton Lab, Chilton, Didcot, Oxon, OX11 0QX.
Dr IC Pyle, System Designers (Scientific), Ferneberga House, Farnborough, Hampshire.
Peter Qualey, Jiskoot Auto Control Ltd, Goods Station Road, Tunbridge Wells, Kent, TN1 2DJ.
Dr Art Rabinowitz, Software Engineer, De La Rue Printrak, 1250 N Tustin Ave, Anaheim, California, 92807 USA.
T Raby, BP Research, Chertsey Road, Sunbury-on-Thames, Middx, TW16 7LN.
Simon Rae, Computing Services, Trent Polytechnic, Burton Street, Nottingham, NG1 4BU.
I Ramos, CESELSA - Ctra de Loeches, Torrejon De Ardoz, Madrid, Spain.
O F Ransen, Technical Director, Kikyo Lions Mansion 606, Hirakawa Cho, 1-8-8 Chiyoda-Ku, Tokyo, Japan.
D Rawson-Harris, Ferranti Computers, PO Box 5, Weymouth, Dorset, DT4 9TY.
Paul Rayner, Ferranti Computer Systems Ltd, PO Box 5, Weymouth, Dorset.
Prof G L Reijns, Delft Univ of Technology, PO Box 5031, 2600 GA Delft, The Netherlands.
R W Rennell, Business Manager, Shirley Institute, Wilmslow Road, Didsbury, Manchester, M20 8RX.
Dan Resler, Lecturer, NIHE, Plassey Tech Park, Castletroy, Limerick, Ireland.
T J Reynolds, Dept of Computing Sciences, Essex Univ, Wivenhoe Park, Colchester, Essex CO4 3SQ.
Mr Alexander Riach, Computing Advisor, DTI/ITec Consultancy Unit, 189 Freston Road, London, W10 6TH.
T Richens, Dept of Computation, UMIST, Sackville St, Manchester, M13 9PL.
Miroslav Ristic, Development Engineer, W O Elektrouniverzal, Marsala Tita Avenue 147, Yu-16000 Leskovac, Yugoslavia UE.
M Riveill, IMAG, Lab de Genie Information, de Grenoble, BO 68, St Martin d'Heures F38402, FRANCE.
Ken Robbins, Brown University, Div of Engineering, Box D, Providence, RI 02912, USA.
Dr John Roberts, 27 Clare Street, Cambridge, CB4 3BY.
Graham Robinson, Microtechnic, CSI Building, Corporation Street, Middlesborough, Cleveland, TS1 2RJ.
Laurens Rodrigo, Grupo Occam, 239 Bd Jean Jaures, 92100 Boulogue, France.
Karel Roose, R&D Engineer, Barco Industries, Th. Sevenslaan 106, B-8500 Kortrijk, Belguim.
John Roper, Research Engineer, Information Systems Dept, Thorn EMI Central Research Labs, Danley Road, Hayes, Middx UB3 1HH.
Paul Ryder, Senior Engineer, British Gas Corp, R&D Division, Engineering Research Station, Killingworth, Newcastle-Upon-Tyne NE99 1LN.
Pirooz Saeidi, Research Assistant, University of Sheffield, Flat 2, 9 Louth Road, Sheffield, S11 7AU.
Derek Salkeld, Ferranti Computer Systems Ltd, PO Box 5, Weymouth, Dorset.
Douglas H Sargent, Systems Engineer, ITT Avionics Div, 390 Washington Ave, Nutley, NJ 07110, USA.
S Sawyers, Dept of Electronic Eng, Birmingham Polytechnic, Perry Barr, Birmingham, B42 25U.
Marcel Schelvis, Research Engineer, OCE Nederland, St Urbanusweg 43, Venlo, The Nederlands.
Carl Schofield, Software Engineering, Sycologic Ltd, 20 Conduit Place,

London W2.
John Schuttle, 3M, 3M Center, Building 518-1, St Paul Mn, 55144, USA.
H F Scott, Principal Engineer, British Gas Corporation, Engineering Research Station, Killingworth, Newcastle-Upon-Tyne, NE99 1LH.
Susan Sepney, Marconi Research Ltd, West Hanningfield Ltd, Great Baddow, Chelmsford, CM2 8HN.
Mr G D Sermon, Engineer, British Aerospace, (Army Weapons) Division, PB 122, PO Box 19 Six Hills Way, Stevenage Herts SG1 2DA.
J S Severwright, Thorn EMI CRL, Dawley Road, Hayes, Middx, UB3 1HH.
Mr Sharman, BAE, Millbank Way, Bracknell, Berks.
S N Sheard, Dept of Physics, Edinburgh University, Kings Building, Mayfield Road, Edinburgh.
C A M Shipper, F.E.L.-T.N.O, Oude Waalsdorperweg 63, Den Haag, 2597 AK, Holland.
Hirdaki Shizuya, Univ of Southern California, AHF315, Dept of Biological Sciences, Los Angeles, California 90089-0371, USA.
A W Sibbald, NE London Polytechnic, Dept Elec Engineering, Longbridge Road, Dagenham, Essex, RM8 2AS.
Kia Silverbrook, Director, Integrated Arts PTY Ltd, 61 Manly View Road, Killcare Heights, 2256 N.S.W., Australia.
Dr A.J.Simmons, Smith Associates, Surrey Research Park, Guildford, Surrey, GU2 5YB.
P Simpson, RSRE, St Andrews Road, Malvern, Worcs.
Jan P Smit, F.E.L.-T.N.O., IT Division, Physics & Electronics Lab, PO Box 96064, 2509 JG, Holland.
Michel Sorine, Research Engineer, INRIA, 78153 Le Chesney Cedex, France.
Mr S K Sprague, The London College of, Computer Studies, Meridian House, Royal Hill, Greenwich, London SE10 8RJ.
J Standeven, Dept Computer Science, Essex University, Wivenhoe Park, Colchester.
S Stepney, Marconi Research, West Hanningfield Rd, Gread Baddow, Chelmsford, Essex CM2 8HN.
P F D Strain-Clark, Lecturer, Open University, Mathematics Faculty, Walton Hall, Milton Keynes, MK7 6AA.
Hakon Styri, Research Scientist, SINTEF, Runit, Strindvn 2, N-7034, Trondheim NTH NORWAY.
R Sugimori, Matsushita Electric Co, World Trade Centre Building, 4-1 Hamamatsu-cho, 2 Chome, Minato-Ku, Tokyo AD5 Japan.
Mr A C Tan, Dept Medical Physics &, Bio-Engineering, University College London, Shropshire House, 11-20 Capper Street, London WC1E 6JA.
David Thomas, Room 460, Brunel House, 2 Fitzalan Road, Cardiff, CF2 1YA.
Ian C H Thomas, Southampton University, Dept of Elect & Info Eng, Southampton, SO9 5NH.
W G Thomas, Systems Engineer, Systems Designers Scientific, Pembroke House, Pembroke, Broadway, Camberley GU15 3XD.
J Thompson, Dept Computer Science, Sheffield University, Hicks Bldg, Hounsfield Road, Sheffield 83.
E T Thomson, Staff Engineer, Biddle Instruments, 510 Township Line Road, Blue Bell PA, 19422, USA.
David Thurnton, Systems Engineer, British Aerospace, Air Weapons, Lostock, Bolton, BL6 4BR.
M Tournaux, HBS, Rue De Tunis, BP 111 93203, St Denis Cedex 1, France.
P Di Tria, Staff Engineer, CSELT, Via Reiss Romoli 274, 10148, Torino, Italy.
Tricot, IMAG Univ of Grenoble, IMAG - LGI, BP68, F-38402, Saint Martin D'Heres, France.
Peter D Tully, Project Leader, European Space Agency ESTEC, Keplerlaan 1,

2201 AZ Noordwijk, The Netherlands.
John Turner, Apt 1815, 30 Charles St West, Toronto, Ontario, M4Y 1R5, CANADA.
John Turner, Development Engineer, Marconi Instruments Ltd, Six Hills Way, Stevenage, Herts, SG1 2AU.
A T Vassilakis, Full Speed, 47 Murray Mews, London, NW1.
Dr David Vernon, Lecturer, Trinity College, Dept Computer Science, Dublin 2, Ireland.
Dr G S Virk, Dept of Aero & Astronautics, The University, Southampton, SO9 5NH.
Dr Gianuzzi Vittoria, University Mathematical Inst, Via L B Alberti 4, Genova, 16132, Italy.
John Hugh Warner, Software Design Engineer, Thorn_EMI Electronics Ltd, 252 Blyth Road, Hayes, Middlesex UB3 1HW.
Jerry Wases, Research and Marketing, Limicon Inc, 144 Hampton Ave, Toronto, Ontario, M4K 2ZI Canada.
L J Way, Technical Director, Tompatrow Ltd, 6 Portman Road, Reading, Berkks, RG3 1JO.
Sara Jayne Webb, Ferranti Computer Systems Ltd, PO Box 5, Weymouth, Dorset, DT4 9TY.
A J West, Dept Computer Science, Manchester University, Manchester, M13 9PL.
John White, Research OfficerC.E.G.B., Canal Road, Gravesend, Kent, DA12 2RS.
M G Williams, EWPG - Thorn EMI Electronics Ltd, Mercury House, 252 Blyth Road, Hayes, Middlesex, UB3 1HW.
D B Wilson, Systems Programmer, ACS Open University, Walton Hall, Milton Keynes, MK7 6AA.
G A Wilson, Computer Science Dept, Sheffield University, Hicks Building, Hounsfield Road, Sheffield S3.
Norman Winterbottom, Research Programmer, IBM UK Limted, IBM UK Science Centre, Athelstan House, St Clement Street, Winchester Hampshire SO23 9DR.
Mr David A Wolfram, Research Assistant, Syracuse University, School of Comp & Info Sci, Syracuse, New York, USA.
F Wray, Topexpress Ltd, 13/14 Roundchurch St, Cambridge, CB5 8AD.
Giles Wright, Senior Editor, Prentice-Hall, International (UK) Ltd, 66 Wood Lane End, Hemel Hempstead, Hertfordshire HP2 4RG.
Dr Hanno Wupper, Faculteit Der Wiskunde En, Natuurwetenschappen, Katholieke Universiteit, Nijmegen, Nederland.
Dimitris Zois, Aeronautics Dept, Imperial College, Prince Consort Road, London, SW7 2BY.

·O·U·G·

occam™ user group · newsletter

**Informal OUG Committee**

Dr John Ainscough,
Dept of Electrical and Electronic Engineering,
John Dalton Faculty of Technology,
Manchester Polytechnic, Chester Street,
MANCHESTER, M1 5GD Tel: Ø61 228 6171

Dr Martin Bolton, (EDITOR)
Department of Electrical and Electronic Engineering,
University of Bristol,
Queens Building, University Walk,
BRISTOL BS8 1TR. Tel: Ø272 3Ø3Ø3Ø (x33Ø1)

Mr Gordon Harp, (CHAIRMAN)
RSRE, St Andrews Road,
GREAT MALVERN,
Worcs WR14 3PS Tel: Ø6845 2733 (x2824)

Dr Geraint Jones,
Programming Research Group, University of Oxford,
8-11 Keble Road,
OXFORD OX1 3QD Tel: Ø865 54141

Mr Chris Nettleton,
System Designers Scientific,
Pembroke House, Pembroke Broadway,
CAMBERLEY, Surrey GU15 3XD Tel:Ø276 686200

Dr Michael Poole, (SECRETARY)
Software Support, INMOS Limited,
1000 Aztec West, Almondsbury,
BRISTOL BS12 4SQ. Tel: Ø454 616616

Mr Robert Stallard,
Dept of Computer Studies, Loughborough Univ of Technology,
Ashby Road,
LOUGHBOROUGH, Leics LE11 3TU Tel: Ø5Ø9 263171

Mr Simon Turner,
Plessey Electronic Systems Research Ltd,
Roke Manor, ROMSEY,
Hants SO5 ØZN Tel Ø794 515222 (x2219)

Mr Hugh Webber,
RSRE, St Andrews Road,
GREAT MALVERN,
Worcs WR14 3PS Tel: Ø6845 2733 (x2239)

Dr Peter Welch,
Computing Laboratory, The University,
CANTERBURY, Kent CP2 7NS Tel Ø227 66822 (x629)