

# Next Generation Transputer

February 26, 1988

## 1 Introduction

Concurrent processing has arrived. It has become clear - and is widely accepted - that specialised concurrent processors can be applied in a wide range of application areas. It has also become clear that over the next five years, effective *general purpose* concurrent computers will be developed. Both of these represent major opportunities for Inmos, but require that both product architecture and manufacturing process technology are state-of-the-art. Competitors are unlikely to ignore these opportunities for long!

## 2 Specialised (Embedded) Systems

The major requirements are high performance. Specialised processor configurations - and specialised processors - will continue to be employed. A serious attempt should be made to integrate the DSP products and the transputer products; Fault tolerance is an important area and requires architectural support. High speed input and output must be supported.

## 3 General Purpose Concurrent Computers

The development of general purpose concurrent computers depends on:

1. Increased hardware support for global communication and global memory access
2. Effective user-level concurrent programming languages and parallelising compilers for conventional languages

Both of these developments are taking place at present. The minisupercomputer manufacturers provide global address space using common bus architectures. They provide parallelising FORTRAN compilers. As the number of processors increases, they will use multi-stage interconnection networks in the processor-memory interface. The presence of large local caches (filled from the common memory via the interconnection network) will make these

machines effective up to large numbers (1000s) of processors and will allow the bandwidth of the network to be kept reasonable - allowing it to be implemented at an acceptable cost and reliability.

It would be straightforward for Inmos to produce transputers (and routing devices) which support both global communication and global memory - the evolutionary transputer already goes some, way towards this. Consequently, provided that the appropriate compiler technology is developed for transputer machines, Inmos is well placed to become a standard architecture for volume low-cost general purpose concurrent computers.

## **4 Architectural principles**

Unless Inmos intends to move into other manufacturing technology, it is essential to make extensive use of concurrency in the near future in order to match the performance of competitive sequential processors. It will be necessary to develop on-chip multiprocessors sharing common memory. This, coupled with limited automatic parallelising of ordinary sequential programs will allow Inmos to offer chips of (apparently) sequential performance exceeding that of competitive products.

The use of concurrency with limited automatic parallelisation will allow effective exploitation of increased component density without a need for a corresponding increase in speed. This appears to match the requirements of the technology - and the customer.

## **5 Silicon components**

Inmos will develop a simple component processor which can be replicated to construct on-chip multiprocessors. It will be designed for use with high speed wide-access on-chip RAMS. A floating point unit will be provided for this processor (see below). Scheduling and communication between processors on the same chip will be very fast (sub-microsecond). Data will normally be passed by reference.

### **5.1 Numeric processing**

For high speed regular numeric computations, it must be possible to compete effectively with pipelined vector processors. This applies to traditional vector supercomputer applications and (more important) to DSP applications.

Either

1. Inmos will achieve this performance with an on-chip multiprocessor
2. Inmos will develop a vector floating point processor

In the latter case, it will also be desirable to extend the parallelising compiler technology to deal with vectorisation. At the very least, the normal vector processing operations must be supported.

## **5.2 Symbolic processing**

A new processor will be developed. This will support dynamic storage allocation including garbage collection. Recursive data structures and recursive programs will be supported directly. Software/microcode support for distributed reference-count based garbage collection will be provided (this is the only known technique).

Significantly more support for abstraction will be provided (a more suitable procedure call) and this will be associated with suitable language support for modules etc.

Sufficient processing power (ie at least two processors) will be provided in products to allow data structures to be output or input concurrently with processing.

## **5.3 Image processing**

One (popular) approach to image processing and computer vision is the use of large two-dimensional arrays of simple processors. Inmos will develop a very simple 16 bit transputer capable of implementing large arrays of this type. Its communications architecture will be compatible with other products, allowing effective interfacing to general purpose transputer arrays.

## **5.4 Memory**

Wide access to on-chip memory will be exploited. On-chip memory capacity must be expanded to be competitive with state-of-the-art process technology. Inmos must provide manufacturing process technology for 1 Mbit on-chip RAM by 1990.

## 5.5 Communications

The speed of the communications link must be increased as far as possible. The design aim is to maximise useful data rate at the expense of silicon area. 10 Mbytes/second on each link would be a reasonable target by 1990.

The evolutionary transputer will provide a virtual channel system and message routing system. It will also provide for trapping non-local addresses allowing software support for 'global' memory.

Eventually, given a large on-chip memory, fully automatic support for 'virtual' memory will be provided. This will cause data to move automatically to local memory from remote nodes. In the meantime, instruction level support for software implementations of 'virtual' memory could be provided (primarily good support for data abstraction and high speed hashing - ie integer division).

## 5.6 Interfaces

Standard interfaces will be developed. These will include:

1. A memory interface
2. A fast parallel interface
3. A programmable (pin-wiggling) port

The aim is to use a small number of standard interfaces in a wide range of products.

# 6 Software

## 6.1 Languages

A superset of occam known here as occam3 will be developed. It will allow recursion and recursively defined data types. Processes will be communicable as 'first class' objects. There will be an abstraction mechanism based on modules and directly supported by the processor instruction set. Occam3 will be the system language of the next generation transputer.

Standard languages supported by Inmos will include FORTRAN, C, C++, LISP and ADA. Automatic parallelising of these languages will be much easier on the next generation transputer in view of the reduced overhead

of initiating remote (offloaded) processes - and also the access to non-local data.

## **6.2 Compilers**

Substantial progress has been made automatic parallelising of standard languages - and also of the (more exotic) functional and logic languages. Inmos will develop parallelising versions of FORTRAN, C and LISP - probably in the framework of an ESPRIT project.

## **6.3 Development tools**

# **7 Preparing the market**

The presence of message routing etc. in these products means that a significant amount of market preparation will be needed. The three ways most easily exploited are:

1. Conferences
2. Disclosure to existing customers
3. A book in the Prentice Hall - Inmos series

A collection of papers will be written which will form the basis of the book. They will also be submitted to conferences and used as the basis of presentations to customers.

# **8 Silicon Products**

The aim is to develop a universal silicon architecture from which products can be derived by combination of the components outlined above. Probably the following should be designed:

1. A general purpose transputer with at least two processors (main + communications) and floating point support. Message routing, 8 links, 128Kbyte memory.
2. A fast numeric processor with either vector processing or 4 (?) scalar floating point processors, 128Kbyte memory.

3. A 16 transputer chip, 4Kbytes per processor for 2-dimensional (image) processing
4. A high speed routing chip 32 links + simple processor.
5. An input-output transputer with fast parallel interfaces etc.

## 8.1 Resources

Some increase in resources is needed. The detailed design of the current transputer should take place concurrently with the architecture work needed to derive the outline specification of the next transputer. Only this way can Inmos avoid the unfortunate gap between products which is currently delaying the evolutionary transputer.

Probably 3 compiler writers are needed for work on parallelising compilers. Probably 2 more computer architects are needed and probably more link, processor and memory designers are needed.