

THE INMOS SAGA
A Triumph of National Enterprise?

Chapter 9

Mick McLean and Tom Rowland

©1985
Frances Pinter (Publishers), London
ISBN 0-86187-559-1

The Challenge of the Transputer

Despite all the corporate and political turmoil, it must be remembered that the company had been growing all the time. Put crudely, by 1983 Inmos was something worth fighting over. Apart from the two facilities at Cheyenne Mountain and Newport, there was the British Design Centre. This had been painstakingly built up during 1980 and thereafter emerged as one of the company's most important assets, although it has to be said that many of the Americans regarded it as something of a joke for a long time.

In some ways, setting up a design team in the United Kingdom capable of turning out products at the leading edge of semiconductor technology was even more difficult than setting up an efficient and profitable semiconductor manufacturing facility in the United States. Attracting staff of the right calibre was a very big problem. 'It was really quite impracticable to attract the right people to work in the UK,' recalled Barron; 'bright engineers were reluctant to move to what they saw as a technology backwater and, in any case, we could not have afforded them.' Technologists in the United States felt, with considerable justification, that they were working where the technical action was, and that they would lose their edge if they went to work in Bristol.

Even if they could have been dissuaded from this view, it would have cost the company around £90,000 a year, according to Barron's calculations, for the full expatriate package each would have demanded. This was just for fairly standard engineers; those with substantial track records or managerial experience would have been even more costly. Apart from the sheer expense involved in recruiting from the United States, It would have created

massive personnel problems to have people on American pay scales working alongside those hired in Britain for a relative pittance. But many amongst the British press had expected Inmos to bring back high-flying technologists to Britain. Several British papers ran stories on the ‘reverse brain drain’, and all were disillusioned when nothing happened. This failure to recruit in the United States illustrates two of the firm’s recurring problems at this time: its lack of credibility with the British press and the massive disparity between the amount paid to its employees on either side of the Atlantic. The problem of salary differentials has continued to deteriorate. ‘When we started the firm, the ratio of US salaries to those in the UK was about two to one,’ said Barron; ‘now it is nearer three to one’. These were relatively minor problems for the British management, however, since its toughest challenge was neither improving its public image nor recruitment but rather the consolidation of its existing team.

One result of the troubles of 1980 had been that Barron had been able to spare very little time actually to manage the company. This was especially damaging since, although Barron had hired a set of individuals with considerable imagination and originality, many of the brighter designers had adopted quite contradictory views about what needed to be done, and what form Inmos’s first microcomputer should take. All at Inmos’s technology centre agreed that the initial microcomputer offering should be a total break with the traditions of the past. For too long British computer experts had been forced to use the designs originating from the United States, and, frustrated by their own inability to manufacture a satisfactory alternative, British experts, along with their colleagues in Europe, had become the sharpest and most vociferous critics of existing trends in microprocessor development.

The microprocessor, the heart of a digital computer on a single cheap chip of silicon, had been invented almost by accident by a Silicon Valley firm, Intel, in 1971. In response to a request from a Japanese maker of pocket calculators, Intel had designed a chip, the 4004, which could be programmed to perform a wide variety of calculator functions. In this context the chip’s ‘programmability’ consisted of its ability to read, and act upon, a set of instructions that were stored in a separate chip memory (not like those made by Inmos but more primitive, called a read only memory, or rom). The 4004 and the chips that were developed from it were thus like the largest computers, very flexible in their operation because they were adaptable to the execution of a wide range of tasks by the provision of different stored programs. The problem was, from the point of view of the Bristol experts, that the various families of processor chips which had been evolved from Intel’s pioneering part all shared its original limitations.

The 4004 had been designed to perform simple arithmetic functions in an efficient manner; the kind of instructions it could understand were therefore made up of such simple types as ‘add’, ‘subtract’, ‘store’ and ‘multiply’.

Later chips simply extended the instruction range to include the kind of functions found in the large computers of the 1960s: ‘branching’ instructions to enable the program to choose alternative options according to the values of the variables it was dealing with, and ‘interrupt’ instructions to allow the processor temporarily to stop calculating while a problem generated by another chip in the system was sorted out. Although such chips increased vastly in sophistication during the 1970s, such development was based on the original structure (architecture) of the original devices. More and more instructions were introduced to cope with the need for microprocessors to handle textual (alphanumeric) information as well as numbers, and the most advanced products could cope with the manipulation of pictures in the form of patterns of dots (pixels).

Microprocessors also became faster at processing instructions and dealt with ever larger chunks of information in a given time period. The 4004, for example, could only deal with numbers between 0 and 15, represented by four digital bits (larger numbers could be handled by combining a series of operations). By 1978 the latest processor chips could cope with numbers between 0 and 65536, made up of 16 bits. Since 8 bits are effectively needed to encode a single alphanumeric character, 16-bit chips, like the Intel 8086 and Motorola 68000, were far better than their predecessors at not only doing fast arithmetic but also manipulating patterns of words and pictures. Chips had also incorporated more and more of the extra functions, such as the ability to store data and accept it as input from, for example, a keyboard, or to send it out to a screen. Such more highly integrated chips were known as microcomputers rather than microprocessors.

But the purists at Inmos still thought processor architecture had taken a wrong turning when it started slavishly to copy the features of pocket calculators and early large computers. For Barron, and for the others in the team he had recruited, the full potential of the mos silicon chip would not be fully exploited without devising a more appropriate architecture. The main difficulty was that the young British ‘whizz-kids’ in Bristol had totally failed to agree on what the ideal architecture should look like. While Barron should have been acting as referee, and ultimate arbiter, in the intellectual discussions taking place at Bristol, he had, instead, been doing his best to ensure the company’s future by arguing with the politicians. By the summer of 1980, Inmos’s microcomputer strategy was in a shambles. Barron recalled:

All of our people had different ideas and were failing to shake down together because there was nobody there to give them a strong lead. I was too busy off dealing with the politics to provide enough positive input. The delay over our funding not only held up the setting up of production in the UK but also seriously hampered the development of our microprocessor products.

Two of the main protagonists in Bristol were David May and Robert

Milne. May had been recruited from Warwick where he had been a disciple of Tony Hoare, an academic guru who, when he had worked with Barron before, had chosen a professorial post at Belfast University rather than joining Computer Technology, Barron's first firm. Milne had been hired from Scicon, a large London-based firm specialising in the production of complex computer programs. Before then, he had worked with Chris Strachey, a leading computing expert, at the Programming Research Group of Oxford University. In the absence of Barron, both May and Milne had taken up inflexible positions over their choice of the most desirable architecture for Inmos's 'Transputer' chip.

Milne favoured a design specifically tailored to work with a particular high-level programming language, 'Ada', which had been developed at the request of the United States' Department of Defense. The Department of Defense had been worried about the vast diversity of programming languages used by its large number of equipment contractors. Ada was designed, therefore, to be a general-purpose language in which instructions could be written to do anything from guiding nuclear missiles to sorting out the payroll for the Oakland Navy base. High-level languages had been developed to cut the cost of producing the vast amount of software needed to run the ever-growing number of computer-based applications. Instead of programming a chip, or a large computer, with a set of elementary binary instructions which it could immediately 'understand', a separate program, called a 'compiler', was devised which was used to translate commands written in an approximation to English (or another ordinary natural language) into the patterns of noughts and ones the computer chips could deal with.

Unfortunately, various specialist languages had grown up that were all tailored to the solution of different kinds of problems: Fortran and Algol for mathematical applications, for example, Cobol for commercial systems and Pascal for handling systems which needed an instant response from the control computer (so-called 'real time' systems). Not only were each of these languages fundamentally different from each other; often two versions of the 'same' language produced at different times or by rival manufacturers would not be mutually compatible. The Department of Defense's Ada initiative was thus designed to resolve the increasing problem of computer Babel by producing a single, all-purpose, standard language.

Although Ada faced problems right from the start, since so much time and effort had been put into writing programs in existing languages and training programmers in their use, there were many who felt that the massive influence, and huge purchasing power, of the United States' military would be sufficient to establish Ada as a new world-wide standard. This view was especially prevalent within the many companies that made considerable profits from supplying complex military software. These software houses saw the advent of Ada as a unique opportunity for foreign firms to break into the lucrative American defence market. Milne favoured this approach

for the Transputer: that it should be the first chip in the world specifically tailored to run Ada.

This approach was anathema to Tony Hoare, even though Ada incorporated some of his own ideas, and to his disciples at Inmos. For Hoare, the trend in high-level language development, towards ever more complex features to cope with ever-expanding application areas, was a basic mistake. As languages became increasingly complicated it became more and more difficult to work out whether the translations they produced were correct. This criticism, according to Hoare, was especially relevant to real time systems, and particularly worrying in connection with such systems as these used by the military. Languages like Ada, Hoare argued, were so complex that no one would ever know if a missile, or early warning system, had been correctly programmed, and the consequences of such a failure would be truly disastrous. Hoare was, in computing terms, a fundamentalist, and May and Barron had been raised in the same creed. What they all wanted was a new simplicity in computers, in their structure and in the languages used to program them. In this context simplicity need not be the enemy of performance. Indeed, Barron and May thought a simply-structured chip might be capable of far faster operation than the existing ranges of complex devices. By increasing the elegance of architectures and languages, the full potential of mos chips — their ability to perform lots of elementary operations at speeds approaching that of light — could be fully harnessed.

The Milne/May debate was not to be resolved for some time. Some essential decisions had, however, been taken at a very early stage and they were no less revolutionary in their implications than the choice of instruction format and programming approach. Since 1977 Barron had been arguing that the microcomputer of the future would inevitably incorporate a certain amount of random access memory. Many such devices had already been launched by 1980, but the memory had always been added almost as an afterthought. In Barron's view, processing and memory (used to store instructions and the data that were being manipulated) were so closely associated as to become eventually indistinguishable. The transputer was therefore designed from scratch to combine memory and processing capability. If every microcomputer could have intimate access to its own data, argued Barron, then it would be much easier to design systems that put together a multiplicity of microprocessors to tackle really tricky problems like understanding human speech. Here was another fundamental point of computing philosophy: rather than trying constantly to improve the speed and power of an individual processor, Barron was convinced that the way ahead lay in harnessing the power of lots of small processors together to provide the most cost effective way of building ever more capable computers.

Hence one arrives at the third major revolutionary aspect of the Transputer: the means by which it communicated with other devices. Conventional processors communicate with other chips, and with the input and

output devices of the system of which they are a part, via a 'bus'. This consists of a set of wires, one for each bit that the chip is capable of handling simultaneously, which links all the devices together. If one chip needs to send data to another, it first has to establish whether the bus is free, then transmit the data preceded by a signal representing the 'address' of the intended recipient, and it then has to send another signal to check if the data have been received correctly. Although the bus is simple in physical form and concept, its use in practice is quite complicated even if only a few parts are hooked up to it. If a system consists of a large number of chips, managing the data traffic on the bus gets tricky and special chips, called 'bus drivers', have to be used to regulate the data flow.

For Barron's vision of a powerful computing engine made up of multiplicity of individual processors all working at the same time (concurrently), the bus concept is unworkable since the chips would spend more time managing internal communications than actually processing data. The alternative approach was to give each processor a fixed number of communication circuits, which it could manage for itself, and allow each circuit to be linked directly to one other similar part. This point-to-point communications approach was a great leap into the unknown since buses had been around for long enough for every designer to have become familiar with their use and for a variety of different standards to have emerged and found almost universal acceptance. So, while it had been decided early on to incorporate a large, fast memory on the transputer chip and adopt a radical new method of communications, the detailed structure of the transputer had still not been chosen by as late as the beginning of 1981.

In the end, Barron grew tired of waiting for a consensus to emerge from his team and decided to enforce his own preference. Along with May, Barron had decided that the transputer should be what others now describe as a 'reduced instruction set computer' (risc). Instead of the hundreds of different types of instructions recognised by the most sophisticated conventional processor chips, the Transputer would only use a small number. With its reduced instruction set, on chip memory and direct communications, the transputer was seen as an ideal device to control for example, the next generation of home appliances, and factory automation equipment. The same virtues that suited it for these tasks would also make it easier to link several transputers together to tackle more general-purpose computing applications. This is not to say that the transputer's risc architecture would prevent it from providing a high level of processing performance. On the contrary, the risc theory suggests that, since out of any instruction set only 10 per cent of the instructions are used in 90 per cent of cases, the performance penalty imposed by having to construct the more esoteric instructions out of the simple ones is more than outweighed by the gain in throughput obtained by making sure that the most common ones are executed in the most efficient fashion.

May was given the job of designing the risc transputer in the spring of 1981. But Barron tried once again to involve Milne in the project, even though his architectural idea had been firmly rejected. Milne was thus asked to design a programming language which would complement May's risc hardware. But still the team could not reach agreement. Milne wanted to write a better version of Ada, which just did not fit in with the essential simplicity of the transputer architecture.

After six months' wrangling, Barron again had to put his foot down. The language was to be as simple and elegant as the transputer structure itself. Barron, Hoare and May went off to a hotel for a week-long brainstorming session and returned with the specification for the new language. Milne's final contribution, before resigning and going to work for British Telecom, was to dream up a name for the language. William of Occam was an early English philosopher who had proposed that philosophy should 'not multiply entities without necessity'. Known as 'Occam's razor', this principle had been adopted by later, and greater, British thinkers such as Bertrand Russell. It also seemed to epitomise the Inmos approach to computing, so the name 'occam' won the naming competition organised by Barron outright. It was an unfortunate irony that Barron had offered a case of champagne as the prize, for Milne was a teetotaler and was thus unable to enjoy it.

Occam was just as revolutionary as any other aspect of the transputer. It was intended not just as a programming language but also as a means of describing the structure of a computing system. It bucked the trend in the evolution of computer languages towards ever greater sophistication and complexity. It was not oriented towards specific applications, but allowed programmers to write compilers for higher-level languages that were. And above all it was simple, being defined in terms of a few primitive constructs from which more complex structures could be built. By September 1981, when the occam specification was published, most of the ingredients of the transputer had finally been sorted out, but the job was far from over. The agonisingly difficult and risky business of actually transforming a bunch of radical ideas into a working and saleable piece of silicon required two further, and vital, components: a design system and a manufacturing process.

By the 1980s, the procedures for designing silicon chips had to some extent been automated. The semiconductor industry had been one of the first to exploit its own products in the form of computer systems to aid the chip designer in laying out the complex patterns of the various layers of an integrated circuit. Other systems were also used to simulate the performance of the finished article before it was actually made and to take over the tedious task of preparing the test routines and the actual photographic masks used during manufacture. A healthy industry had grown up supplying such specialised design tools, and at Colorado Springs, for example, the Inmos team had purchased all the necessary equipment from Silicon Valley suppliers. The staff at Bristol had, however, rejected such an option right

from the start. Deciding that existing equipment was not powerful enough for the task of designing the complex random logic circuits needed to make a transputer. Inmos had set up a team of five people in the spring of 1980 to build a set of ideal chip design tools from scratch. By the time the ultimate form of the transputer had been finalised it was still by no means certain whether this home-grown design system would eventually work. It was not until 1983 that the Inmos computer aided design system, which had been codenamed ‘Fat Freddy’ after an American hippie cartoon character, was finally to prove itself.

The second missing ingredient was a manufacturing process. It had originally been intended that the transputer should be designed to use the same process as the 64k dynamic ram. But this approach had been vetoed by Paul Schroeder, who had argued that the risks involved were too high because the operation of the processor might interfere with the very delicate mechanism of the on-chip dynamic ram. In the jargon used by chip designers, this problem, when it does occur, is known as ‘noise’. The decision to abandon dram technology in favour of the theoretically more stable sram route put a major delay into the transputer programme. This was because the 16k static ram process would not support the level of complexity required for the transputer, and so the transputer design was forced to wait for the development of the next generation 64k static ram technology.

The transputer designers reviewed the technology options and came to the conclusion that the best available technology for both the transputer and the next generation memory products was cmos — complementary metal-oxide semiconductors — the coming thing at the time. Its intrinsic, low-power requirements and other advantages made it ideal for making most chips, and it was gradually being realised that cmos would also be needed to make any very complex chip. With so many functions packed into a tiny area, power dissipation was becoming a critical variable. Even with the mainstream nmos technology, which was much less power-hungry than bipolar, and which it had largely displaced for digital applications, it was becoming harder to increase chip density without running into problems with the amount of heat generated. Opinions differed, however, about how much growth potential could be extracted, by design ingenuity, from the inherently much simpler nmos process. Inmos’s memories had all been made in nmos, and the Colorado Springs group saw no need to develop cmos technology. They argued back and forth across the Atlantic on the subject until 1982 when Colorado Springs needed to develop a process for its next generation memory products and finally came to appreciate the advantages of cmos.

Even then the transputer was not out of the woods. Its future was dependent on the United States developing a process and manufacturing the prototypes. The development of the cmos process was slow. In consequence, a queue of new products built up for prototyping in Colorado Springs. This queue was made up of a range of 64k static rams, a range of 64k dynamic

rams and the transputer tagging along behind. Just as in 1982 Colorado Springs had found it difficult to start up the manufacture of the 64k dynamic ram while manufacturing the 16k statics in volume, so the same problems were experienced in 1983 and 1984 as the new generation products came on to the production-line.

As a result the British designers were only to receive three batches of working silicon prototypes of the transputer during 1983 and 1984. It made the target Barron had announced for transputer samples look rather foolish. In the event, midnight on 31 December 1984 went by without a transputer to be seen. Although the public reaction to the transputer had been enthusiastic, even by 1985 it was too early to tell whether the transputer was a potential moneyspinner or just a brilliant concept that was so radical in every aspect to be way ahead of its time.

The reactions of Inmos's American staff to the transputer saga had been somewhat bemused. Most Americans, both within the company and outside it, had been deeply impressed by the chip's intellectual concepts; Barron had addressed a huge audience at a Silicon Valley electronics exhibition in the autumn of 1983 and had been delighted with the warmth of the response. One American chip-maker had offered to make the transputer under licence, an offer Barron had refused until it could be negotiated on the more favourable terms made possible by having actual chips to sell. But the American Inmos employees could not understand the desperate need of those in Bristol to challenge every single convention of microcomputer design. In particular, it was hard for those in Colorado Springs to understand why the British team had decided to build its own design tools. 'For a start-up company trying to get products out very quickly, that did not look like a very good idea,' commented Heightley. The seemingly endless wrangles over the precise structure of the transputer chip were also incomprehensible to the Americans, as was the Bristol team's insistence on the next generation process for their brainchild.

These differing perceptions illustrate the great contrast between Inmos's staff on either side of the Atlantic. Apart from the enormous pay differential, in the United States the team was made up of engineers, 'silicon hackers', as Barron has described them. In England the people were mostly pure scientists, wanting the best, and in the American view forgetting that the best is often the enemy of the good. The Americans' scepticism and concern over the transputer's delays never, however, developed into outright hostility. But conflict was inevitable, for with Cheyenne Mountain having the only facility for producing batches of prototype chips for development purposes. American and British designs were bound to vie for priority.

As we shall see in a later chapter, the management policies adopted in the United States made the problem far worse and eventually began to affect the Cheyenne Mountain facility itself. Briefly, there was hardly any space for any kind of development work there for a period as all resources were

concentrated on turning out current lines in as high volumes as possible.

It was not until the end of 1984, however, that such contention was to reach crisis point. Colorado was by then trying to debug the design of its latest memory, the cmos 64k static, just when the prototype versions of the transputer desperately needed processing. The problem was only to be finally resolved by setting up a prototyping line at Newport, where the 1985 chip slump had created spare capacity. So the transputer was finally to come home for the last stages of its development, but the transfer across the ocean imposed the last in a long series of delays. It is ironic that it was not until 1985 that the advantages of having a close relationship between design and manufacture, which had done so much to precipitate the traumas of 1980, were put into practice for Inmos's flagship British component.

Although it is still too early to say whether the transputer will turn out to be the worldbeater its British inventors conceived it to be, it is important to discuss the factors that will determine its chances. In a decade when high-technology industry has become increasingly demand-driven — listening to the needs of the marketplace rather than pushing its technology on reluctant users — one could argue that the transputer seems to be a throwback to an early era. If one accepts this premise, then the image of white-coated boffins sitting in a darkened room working out what people really need, rather than what they think they want, is hard to avoid in connection with the transputer. The 4004 had also been revolutionary, but at least it had been ordered by a customer. This is, of course, something of a caricature, but it is possible to draw parallels between the transputer and a whole host of British inventions that were scientific triumphs but commercial disasters.

The caricature could be grossly unfair and misleading. The ultimate aim of the whole Inmos project was, we should remember, to create a major semiconductor company. Starting from scratch in the late 1970s, Barron argued that the only possibility of achieving this was to take the high-risk options and attempt to innovate by creating new and exciting products in advance of the established opposition. In practical terms this meant being able to provide both memory and processing on a single chip by the early 1990s. Again, in practice, to make this ambition realisable, Inmos has no option but to turn itself into a major force in both memories and microprocessors. Now, it so happens that two giant United States corporations, Intel and Motorola, have the microprocessor market neatly stitched up between them. That is not to say that others do not make, and successfully sell, these parts in very large numbers, but the market is clearly dominated by the products of the two giants.

So Inmos had to find a way to break into the microprocessor business. The obvious route would have been to become one of those happily selling alongside the big two by producing a lookalike product that had a marginal performance advantage and hence finding itself a small niche. The second obvious alternative was to have become a second source for one of the es-

tablished processor designs. In effect this would have meant making under licence and paying royalties on sales for a particular range.

It was not viable for Inmos to be a second source, as those who succeed here tend to have either specialist marketing or manufacturing skills they can bring to bear. One is, after all, competing with the original manufacturer. The one advantage a second source must have is that provided by a better sales force. Inmos did not have this advantage, nor an established base of its own customers to attack with a licensed technology. The problem with taking the first option and competing with a derivative of an existing architecture and a marginally better product has to do with Inmos's size and structure. It is a risky option, and those firms that succeed tend to be far larger than Inmos and well established in other chip markets. So, if Inmos wanted to succeed as a semiconductor house of international stature, Barron argued that its products would have had to be significantly different to those of the established competition. Such products would need to be significantly better on a variety of different technical criteria if they were to stand any chance against the established might of the big microprocessor houses.

Both Intel and Motorola have 32-bit processors of their own which hit the market at around the same time as the transputer. Inmos's and Barron's fervent hope is that because their part offers so much more than conventional designs and can be used to solve many more of a designer's problems than a conventional 32-bit processor, they will be able to grab a large part of the business created by the arrival of the next generation. The microprocessor trade is far less risky than the memory business, so if the transputer strategy works the company will have a far more secure base. Barron is confident that neither of the parts on offer from Intel or Motorola is nearly as well organised or technically executed as the transputer. Also because of their on-board memory and communications, transputers can be linked together to make computing engines of vastly superior power than anything else available, or likely to be available for sometime.

Another advantage with regard to the transputer is that the processor market has so far remained immune from attack by chip-makers from the Far East, which is more than can be said for memories. One reason for this is that the Japanese have taken a long time to develop a software base of sufficient sophistication to turn out good processor products. Barron also argues that the transputer does very definitely address market needs, but the gross needs of an expanding, technology-using community rather than the fiddly little needs of marginal improvements on existing components. So Inmos maintains that the transputer provides a great deal of power in a very small space. Users are also given access to a complex device that is nevertheless embedded in the simple system provided by occam. Once occam is understood, Inmos claims that it will be far easier to use the transputer to good effect.

There is no doubt that there is a great deal of enthusiasm for transputer chips in the market, but that does not necessarily infer that many people are going to buy them in large quantities. This is a point that Heightley recognised soon after he joined the company. The only real test of how well a product meets users' needs is the quantity bought. So conventional wisdom preaches that when planning a new device it is usually as well to base it on an existing product for which there is a known demand. The Inmos static rams, for example, were not revolutionary, but merely offered significantly better performance than rival but similar chips. Their sales were thus fairly well assured.

The transputer is such a risky venture, such a challenge, precisely because it is so radically different from anything that has gone before. If it succeeds it will be a triumph, but the odds have been against it from the start. It has done well to get this far, and certainly the chances of success will get better from now on. The technology judgement to which Inmos is now fully committed is that this route offered the only chance of breaking the American stranglehold on the microprocessor market. There is no doubt, in any case, that Inmos needs a broader product line than that offered by high-performance memory chips, and that it must develop a transputer, or something like it, if it is to live up to the promise of its business plan. Heightley was philosophical about the device.

If nothing comes out of the transputer project then we will have wasted a considerable investment. But by the standards of the microcomputer industry we have not spent that much on it and if we end up with a success, then we will have acquired it on the cheap.

Only time will tell which of Heightley's alternatives comes true.