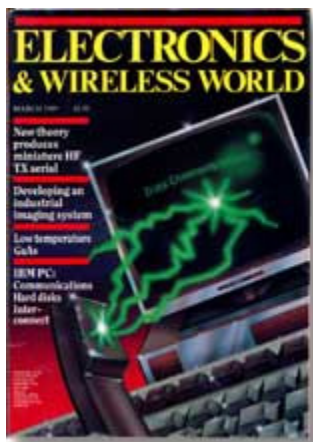


## Air Traffic Control ...



# The Kernel Logic Machine

Electronics & Wireless World, March 1989, p254

**Cost-effective array of a million computers is ideally suited to Europe's air traffic control problem, weather forecasting, and a host of hitherto impossible tasks**

IVOR CATT

Occasionally, a number of technical advances come together to give a quantum leap forward. This occurred recently as a result of three factors - the increased density of components on an integrated circuit, the successful fabrication of fault-tolerant complete integrated circuit wafers at Anamartic Ltd, and a new approach to structuring these wafers called the Kernel Logic Invention. The result is that the latent, explosive power of semiconductor technology can be unleashed - one million computers working together in an array to solve large, complex problems at high speed.

### INTRODUCING KERNEL LOGIC

An improved approach to wafer-scale integration became possible back in 1972 because chips of reasonable yield contained, or would soon contain, as many as 10,000 components. Using an external piece of special test circuitry composed of 100 TTL packages, a single row (spiral) of perfect chips could be 'grown' into an imperfect wafer each time power was switched on to the machine (see panel). Burroughs Corp. (now Unisys) at Cumbernauld built three inch working wafers which demonstrated the feasibility of the spiral approach. The same successful team of engineers later moved to Sinclair Research Ltd (renamed Anamartic), where in 1985 they successfully manufactured the first pre-production working wafers intended for the market. A four-inch wafer full of 16Kbit drams used the spiral algorithm to interconnect the good memory, bypassing the bad, to a total of 0.5Mbyte on the wafer. However, because of the slump in the ram market at the time, this product was never brought to market. In 1989, Anamartic will market a solid-state disc made up of a pack of six-inch wafers containing 1Mbit drams to a total of about 20Mbyte per wafer. Its size could be something like a six-inch cube.

In 1987, 15 years after the spiral algorithm was patented (UK Patent 1377859, described in Wireless World. July 1981, p 57), the number of components in a chip of reasonable yield had risen to one million, an increase of one hundred times beyond the vintage of that invention. The Kernel Logic patent exploits the fact that much more 'fault tolerance' capability can be designed into today's dense chip.

To understand kernel logic, think in terms of the faults in a wafer. One model suggests that tiny faults exist at random points across the wafer, so that if a wafer with 250 faults is cut up into 500 chips, half of them will contain a fault and so be scrapped. Now consider a tiny section at the south-west corner of each chip, which I call the kernel. If this kernel is small enough, its yield will be very large. It is easy to calculate the size of kernel required so that 80%, say, of the wafers manufactured will have a perfect kernel in the corner of every chip on the wafer. The other 20% of manufactured wafers - those with chips containing one or more faulty kernels-are scrapped.

When power is switched on to the wafer, the kernel logic spontaneously puts its chip through a test routine, and decides whether the chip it controls is perfect. If it isn't then the kernel logic cuts off communication with the outside, and the faulty chip disappears from the system.

Chips adjudged by their several kernels to be perfect are allowed to intercommunicate. There is then a simple procedure whereby control circuitry outside the wafer is informed as to which chips are perfect and which chips have been removed from the two-dimensional array. Perfect chips are instructed to link up into an array structured according to the needs of the external control circuitry. (Workers in artificial intelligence would restructure the machine to match the structure of their data).

Communication into and out of the wafer is by means of signal lines at both ends of every column and also of every row of chips. The structure lends itself naturally to expansion into a Cartesian array of interlinked wafers, resulting in an array of 1000 by 1000 processing nodes, each with its own microprocessor and 1Mbit ram, at a cost of the order of one pound per processing node.

### **HISTORY OF WAFER SCALE INTEGRATION**

The first attempt to achieve WSI was at Texas Instruments in the USA in the 1960s. A wafer was made with an array of ordinary, identical chips with conventional bonding pads. These chips were then probed in the usual way, and a record of which were good and which were faulty was fed into a large computer. The computer designed a unique final layer of metallization which would interconnect the good chips on that particular wafer and avoid the bad. The major problem with this approach, and the reason why it failed, was that it was necessary to assume that this last layer of metallization would have 100% yield.

The other famous debacle in WSI was at Trilogy. Amdahl, the father of the IBM 360 series of computers, left IBM and succeeded in taking a share of their massive market with his company Amdahl Corp. He then ventured out to beat IBM's fastest computers for speed by cramming an IBM look-alike machine into five wafers, where signal lines and therefore signal delays would be less. Amdahl raised \$250 million on Wall Street in the biggest start-up in history. His wafers used a conventional approach to fault tolerance. A wafer was very complex, and had over one thousand wires bonded to it. The failure of his WSI company in the early 1980s was the second major blow to the credibility of WSI. It is doubtful if the assertion in the Butcher article (see bibliography) that Trilogy made working wafers is true.

Other companies have approached the use of wafers in ways which would lead to their supplying only a niche market. Wafers have been used as a substitute for the printed circuit board, with flipchips bonded onto them. Laser mending of faults has also been tried, but such expensive doctoring of wafers falls outside the mainstream of attempts to exploit the wafer for its potential low cost and high reliability. The Butcher article discusses other WSI projects at length.

### **A DIGITAL ANALOGUE OF REALITY**

The first signs of the new concept appeared in my own writing 20 years ago ([New Scientist, 6 March 1969](#)), later developed in "Computer Worship" (Pitman, 1973, page 128) in which I discuss 'situation analysis' and 'situation manipulation'. A clearer, more developed outline was published in this journal in my Wireless World January 1984 article 'Advance into the past', (see [The Nub of Computation, page 59](#)). (The way in which an array processor composed of kernel logic nodes would tackle problems is more clearly stated in 1984 because at that point the appropriate hardware possibility existed, whereas it did not a decade earlier.) More recently, in the television series "The Mind Machine" on BBC 2 in September last year, the concept is clearly stated, usefully validating the approach.

Parallel work in cognitive science has been done by Kenneth Craik and Phil Johnson-Laird, see bibliography.

The idea that I have nurtured is that future events should be predicted by speeding up the system clock and projecting a 'data cube' into the future. We do not have predictive algorithms. Rather, in the case of airline collision avoidance, for instance, we lift the current data state in our data cube into a second array, running at a faster clock rate. Two aircraft projected into the future (each occupying a larger and larger volume of space into the future to cover all possibilities) then collide, and the collision of the two over-size aircraft is reported back to the current data cube, pointing to a potential hazard in the near future. This forward projection is soon erased, to be replaced by a more recent valid current data cube, which in its turn will be accelerated into the future in search of possible hazards. This approach probably has a different conceptual base from the more conventional approach of calculating all kinds of possible hazards, and it seems to be more comprehensive and easier to effect. (This second data cube could conveniently reside in higher pages in the same 1Mbit ram as the original data cube.)

### **KERNEL LOGIC ARRAY PROCESSOR HARDWARE**

To configure good chips (processors) in a wafer, the external controller can send in an instruction with a physical chip address. The address has two fields, an easting and a northing. This class of instruction has its address decremented each time it passes through a chip so that the address becomes 00 00 when it reaches its destination. A chip that is seven chips in and 13 up has a physical address 1307.

The interrogated chip then sends a reply, that it is good or faulty, rippling outwards, so that one or more replies are received by the external controller via a path of good chips. The controller then studies the pattern of good and bad chips and instructs most of the good ones on how to link together to make a perfect two-dimensional array.

The architectural constraints of this fault tolerance lead to the extremely powerful array processor machine described here. The standard kernel logic array processor contains a two-dimensional array of 1000 by 1000 processing nodes. Since each individual wafer contains an array of perhaps only 30 by 30 processing nodes, we have to use 1000 wafers in order to give the one million processing nodes in the standard machine. It is therefore necessary to interconnect the rows and columns of an array of 30 by 30 wafers to give one million nodes interconnected in a two-dimensional array.

Four wires are stitch bonded down each column of chips (=nodes) on each wafer. These wires give lower resistance and faster links than is possible with the standard aluminium metallization on a chip. This means that a wafer will contain a set of about 100 vertical wires stitch bonded from top to bottom of the wafer. Each wire is connected to a pad on each chip that it passes over. These wires are then extended across to the two adjacent wafers, the wafer above and the wafer below. Each group of four wires comprises a ground line, a power line, a clock line and a data line. The transmission line represented by the pair of wires, ground and clock, is capable of delivering a 100MHz clock rate. Also, serial data can be clocked into each node at a 100Mbit/s rate. Such data includes 'global' instructions, broadcast to every processing node in parallel.

In practice, the number of wires will probably be reduced to three, and 'OV' will be delivered instead through the wafer substrate. Various other deviations are possible in practice. For instance, to improve fault tolerance, the columns of stitch-bonded wires will probably be at an angle of 45° to the rows and columns of chips (nodes). Another possible variation will be for one set of four stitch-bonded wires to serve two columns of chips (processing nodes) rather than one, but discussion of such deviations here obscures the grand design.

Each chip (=node) will have the ability to communicate 100 Mbit serial data locally to its four neighbouring chips to the north, east, south and west. This will be via conventional aluminium surface metallization. In the case of chips on the border of a wafer however, local east-west inter-chip data lines will be bonding wires connecting the data lines from the right-hand edge of edge chips to the left-hand edge of chips in the next wafer to the right. Similarly, local north-south between-wafer inter-chip data lines will be bonding wires connecting the data lines from the bottom chips of one wafer to the top chips of the next wafer below. In addition to these, the columns of global stitch-bonded wires down a wafer will be extended between wafers, right down through the column of 30 wafers. So a single global wire will have 1000 stitch bonds, and traverse the full height of the 1000-wafer machine. That is, it will traverse 30 wafers.

Each node comprises a processor, something like a serial 6502, and one megabit of ram. It also contains four serial output ports and four serial input ports, enabling local data transfer with adjacent nodes to the north, east, south and west. Each local inter-chip link can support data transfer at a serial hit-rate of 100Mbit/s. (The result looks much like a two-dimensional array of transputers interconnected through their serial ports.) The normal operating mode will be for all processing nodes to simultaneously carry out a series of instructions (a program) globally broadcast to all nodes down the vertical stitch-bonded wires. However, the global array controller will sometimes hand control to an individual processing node, whereupon a processor will implement a subroutine stored in its own ram.

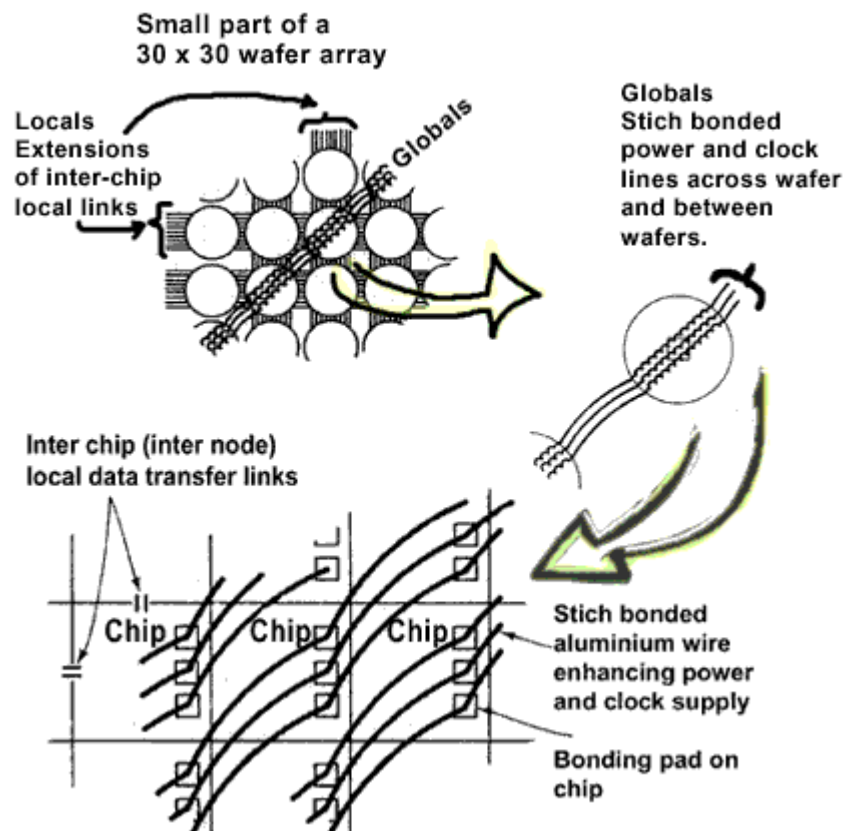
The instruction set will include typical classes of microprocessor instructions, with some additions, as follows. First, there will be configuration instructions, which deal with the configuration of a perfect array of processing nodes by bypassing the faulty nodes. There will be local intercommunication instructions, when each node will transfer data to its neighbour to the east, and so on. In many cases, a flag in a node will determine whether that node will carry out a particular global instruction. There will be a new class of conditional (jump or branch) instructions, when a processing node decides whether it will become autonomous for a short time, obeying a subroutine in its own 1Mbit ram instead of obeying instructions coming down the global stitch-bonded lines.

Practical considerations will have a strong influence on the choice of ram and processor. Since the development time for a state-of-the-art ram is four years, it is necessary, to benefit from the latest increases of ram bit density, to base the kernel logic design on the leading ram manufacturers' process, whether it be 1 Mbit, 4 Mbit, or whatever, even though the ideal memory size at a processing node is somewhat less, perhaps only 100 Kbit. We then aim to take advantage of developments in microprocessor hardware and software and try to get the ram manufacturer to agree to mix a modified state-of-the-art processor into the ram wafer.

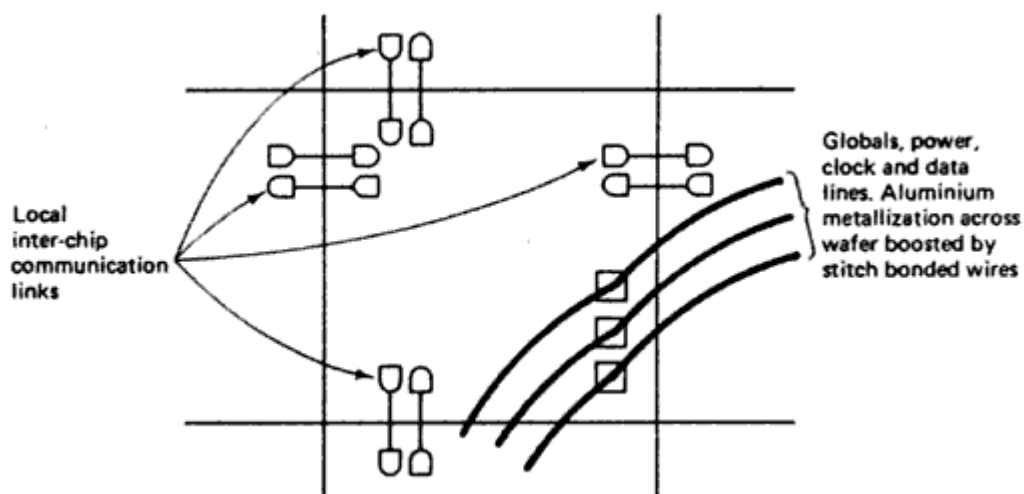
### **STITCH-BONDED CLOCK AND POWER WIRES**

Conventional chips use narrow lines of aluminium metallization on their surface to deliver power and clocks to every part of the circuit.

Anamartic retained this approach in their successful wafer-scale engineering using my spiral approach. However, the resistance of such interconnections, already a minor embarrassment in a large, high power chip, became crippling in the case of a wafer, with its longer distances and greater total power (i.e. current). However, the problem is not severe if, like Anamartic's, the wafer merely houses dynamic ram. At any one time in an Anamartic wafer, only one ram on the wafer is being read and only two more are being refreshed. The rest of the wafer consumes little power. Our situation is different, because we have processing nodes active at the same time throughout the wafer. Limitation on power delivered would mean limitation in the speed of those processors, which is unacceptable. Processing nodes must all be capable of operating at maximum speed all of the time.



Fortunately, stitch bonding technology is ideal for the purpose. At a cost which is only a fraction of the cost of the processed wafer, parallel columns of aluminium wires can be stitched across the wafer, reducing the effective resistance of the aluminium track beneath. The yield on such stitch bonding is very high, and faults, on the rare occasions when they do occur are to a harmless open circuit to the bonding pad (the aluminium beneath covering for the break) rather than to a short. These wires can be either 0.12 or 0.25 mm in diameter, giving the kind of low resistance needed both for power lines and for high-speed clock lines. Further, the characteristic impedance of the transmission line made up of the pair of lines (clock and OV) that delivers the clock is reasonable and convenient to drive.



### CAN YOU PROGRAMME IT?

The kernel logic machine comprises a two-dimensional array of 1000 by 1000 processors, each with its local 1 Mbit ram. The processor will be something like a 6502 microprocessor.

In normal operation, program instructions will be broadcast in parallel from an outside controller to all one million processing nodes, which will obey the instructions in parallel, but operate on different, local data. (This is SIMI) - single instruction, multiple data.) The instruction set will include the groups of instructions contained in a 6502 or Z80, with some additional groups.

One small group of instructions will control the configuration of the perfect 1000 by 1000 array from a larger, imperfect array. This (re) configuration will take place every time the machine is switched on, and gives it a fault-tolerant, self-repair capability.

Another small group of instructions will cause local inter-node communication of data in parallel. For instance, one instruction would cause every node to exchange a particular word of data with the node immediately to the north. This local, ripple-through, intercommunication will be fast, but it will take 20 cycles for a word to traverse 20 processing nodes. (It will be used for the zoom facility mentioned elsewhere.) A 20-bit delay is of course less significant when working serially.

It is possible for the external controller to relinquish control of one group of nodes, or even of all processing nodes, so that each node can carry out a subroutine stored in its own 1Mbit ram. (At any time, the central controller can regain control of all processing nodes.) Generally, when this occurs, the external controller would divide up the one million nodes into no more than four or five groups, and each group will act in concert. The notion of a million processing nodes all implementing different programmes at the same time is unthinkable, not because of technical limitations, but because of the impossibility of assembling enough humans (programmers) for enough time to dream up all the different activities for so many computers. Of necessity, groups of processors will act in concert, obeying the same series of programming code, though not necessarily applying it to the same data. When the first kernel logic machine has been delivered and become operational, a significant fraction of all the processors in operation in the world will reside in that one kernel logic machine. It follows that they must operate in groups, and not as individuals.

On initial memory load from the external controller, each 1 Mbit memory is loaded with a number of flags. These can be employed later by the global program to define which sectors should, for the next period of time, run under global control, and which under their own local routines. The "flag" in each memory might be merely the address or 'grid reference' for that processor.

Recapture of control by global instructions could be effected by the equivalent of the Z80 DMA, or less preferably by interrupt. Using DMA, local control is relinquished when the marker flag) in local memory is found, tailing for a return to global control.

Programming the kernel logic machine is straightforward because its structure mirrors the structure of the problems to be solved by the machine-weather forecasting, air traffic control, and so forth.

## **APPLICATIONS OF THE KERNEL LOGIC MACHINE**

For the last 20 years I have suggested that something on the lines of the Kernel Logic Machine is ideally suited to a large range of important applications. At last the technology has arrived and made it possible to construct the machine we always wanted. It will lead to enormous cost savings and speed improvements in many applications covered by the general descriptors finite and linear element analysis, finite difference methods, and computational fluid dynamics (CFD). In "Supercomputers and the need for speed", New Scientist, 12 Nov 88, page 50, Dr Edwin Galea, research fellow at Thames Polytechnic, says

"The flow of air, water, burning gases, the Earth's atmosphere, ocean currents and molten metals provide scope for the partnership of computational fluid

dynamics and supercomputers."

"Only supercomputers can provide the speed and memory required to perform the detailed calculations for the complex geometries and flows encountered in the design of aeroplanes, automobiles and ships."

. . . manufacturers are already approaching the limits of the capabilities of single processors, . . . ."

"Only parallel processing - the concurrent use of more than one processor to carry out a single job - offers the prospect of meeting these requirements."

Galea talks in terms of a partnership of a supercomputer with CFD software. The software causes the single-processor (von Neumann) computer to behave like an array processor, but at a heavy cost in loss of speed.

As Galea says, the physical processes involved in flow behaviour occur on a very tiny scale, so CFD divides the flow region into thousands of small computational cells and solves the governing equations in each cell. Generally, applications involve perhaps one million cells. A conventional, single-processor computer is caused by software to compute the next change in each cell one at a time, so that its speed is reduced by a factor of one million - hence the need to start off with a very fast computer. Even then, this massive drop in speed is unacceptable, and the application demands parallel processing, when duplicate hardware is devoted to each cell. The kernel logic machine provides this multiplicity of hardware.

Galea's [1988] article estimates the total sales of supercomputers so far to be \$1000 million, and says the market is growing. Most supercomputer applications, and the applications which are expensive in computer run time, are CFD. The kernel logic machine will cause an acceleration in the growth of the supercomputer market, because applications which were too slow and expensive to run on a Cray machine or on the small-scale array of a dap or perhaps 100 transputers, will be successfully attempted on a million processor kernel logic machine. This is a very attractive market; the development of computer graphics for a space adventure movie; a task taking one hour on a kernel logic machine which previously absorbed the run time of a \$5 million Cray machine for months. Another lucrative application is whole-world modelling in real time for the purpose of weather forecasting. This is only practicable on a kernel logic machine.

Applications for the kernel logic machine include airborne early warning systems, air traffic control Europe, in which one machine in London is linked to a second machine in Milan and a third in Barcelona, etc., TV image enhancement, TV compression for satellite transmission, aerodynamic design of motor cars, aircraft and spacecraft, study of airflow through gas turbine engines, weather simulation and forecasting, prospecting for oil and gas by analysing rock structures.

### **AIRBORNE EARLY WARNING AND AIR TRAFFIC CONTROL**

In modern warfare, enemy aircraft attack by approaching very low and at high speed, so that they appear over the horizon only a short time before they reach their target. The defensive response to this is to have an aircraft flying high up so that it can look over the horizon with its radar, and give early warning of attack. The radar continually scans a cone of space stretching in front of it, starting at top left and ending at bottom right. In each complete scan, it transmits a series of pulses, one in each direction ahead of it. A single scan creates one picture "frame", but the reflections from "targets", or enemy aircraft, are weak. By repeated scanning, it builds up a picture of what is in the space. This picture is developed by a process of repeated addition of frames known as "burn-through". This process relies on the fact that the noise is random and averages out, whereas the target recurs in successive frames, and grows out of the noise.



The scanning of the space is similar to the scanning of a TV camera, except that at every point in the raster there is a further, depth scan in the third dimension. If a pulse from the transmitter is reflected from a more distant target, the reflection arrives back later, and thus its distance can be determined. A Nimrod or AWACS radar aircraft groans under the weight and volume of the digital signal processing hardware needed, plus the massive power supplies needed to generate the DC power to drive the hardware, plus the generators needed to generate the electric power, plus the fuel needed to supply the generators, plus the cooling equipment needed to cool the hardware.

The conventional approach is for the aircraft's digital signal processing to look for over-large signals being received by the radar dish among the random noise. These larger signals might be reflections of the aircraft's own output bouncing back off the target. However, they might just be noise. The procedure is to sum up repeating larger signals from one region of space, and at some point make the decision that this must represent a target. This target is then tracked through the region of space being monitored. The practical problem is that each target which has been identified and is being tracked consumes more time in the central von Neumann computer, and the total system overloads and fails if more than a handful of targets are detected. We have to ask the enemy to limit the number of aircraft they use in their initial surprise attack.

By contrast, the kernel logic machine commits one processor in its array to one element in the raster of space. Within that processing node, the first page in its 1 Mbit memory is committed to the cube of space nearest to the aircraft. Further pages in memory are committed to further cubes of space, all of them in the same direction from the radar aircraft, but at different distances. This way, space is divided into one thousand million data cubes in a 1000 by 1000 by 1000 array, although in fact the array only contains one million processing nodes. The third dimension is accommodated by stacking up through pages in ram. (The disadvantage is that there is only one set of inter-node communication links, not one set per page of ram, so there is a resulting drop in local inter-node communication data rate proportional to the number of segments ("pages") used in a ram.) Possible targets need not be thresholded into definite or downgraded to random noise in the kernel logic machine, because such a powerful machine will not be overloaded if the number of targets tracked exceeds 100—the point at which today's early warning tracking systems overload.

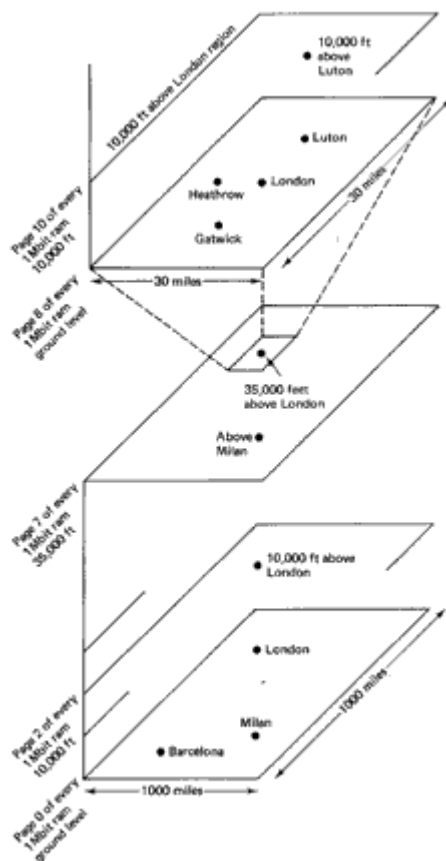


**Potential targets need not be thresholded in a kernel logic machine because it will not be overloaded when the number of targets tracked reaches 100 – the overload point for today's early warning systems.**

Parallel processing in an array makes implementation of the tracking software much more straightforward and fast. Each detected target is a sort of amoeba which moves through the array, carrying its amplitude, velocity and probability with it, to be reinforced from that region of space; or alternatively to diminish down towards zero each time the radar scanner picks up no reflection. Uncertainty over the latest direction and velocity of an amoeba-like possible target results in the amoeba growing into a larger probability volume. However, at the same time, failure of the target (signal) to rise above noise during the last scan (last frame) leads to a reduction of its probability weighting at all points within its amoeba.



Air traffic control Europe would use essentially the same machine, with minor enhancements. Europe will be divided into 1000 by 1000 squares, each of one mile square. However, since this is inadequate for the London airspace, an enlarged model of 30 miles square around London will be housed in the upper reaches of 1 Mbit rams of the array processor. This model will use the full 1000 by 1000 array, and so provide a high precision array of 30 by 30 nodes for each square mile. In an ordered manner similar to the action of the zoom lens in a camera, the local London micro-model and the Europe macro-model will update each other once per second. During this update, the new data will ripple through the array in parallel in an ordered manner.



For Air Traffic Control Europe the Kernel Array Processor commits one processor to the airspace above each one square mile of earth, one page of RAM in that processor per 10,000 feet of height. Higher pages still are committed to an enlarged data cube around a major airport.

The reporting of position and speed by a commercial aircraft will result in the collapse down to point size (a single processing node) of a tracking aircraft which, because of increasing uncertainty resulting from lack of recent position reporting or recent definite radar detection, had developed into a large amoeba.



Aircraft collision avoidance will be achieved by causing the current data cube contained in the kernel logic machine, that is the most recent record of location and velocity of all aircraft, to be transferred to an identical machine (in the higher pages of the 1 Mbit rams) which will be accelerated into the future by (in effect) increasing clock rate. Potential hazards between a pair of aircraft will then be flagged up because of actual collision between two of the growing (future tense) amoebae in this accelerated machine, one representing each aircraft that is at risk.

### **TV IMAGE COMPRESSION**

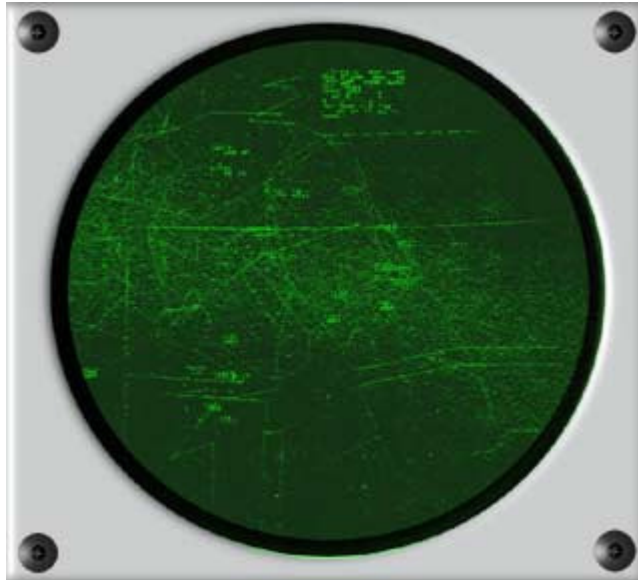
The cost of transmission of TV signals by satellite can be high. We may be able to justify investment at source and at destination in order to reduce the data flow needed to send one TV channel. If we use the standard kernel logic machine, each TV frame is loaded into the 1000 by 1000 processor array in parallel down 1000 columns. Since a TV frame has far less than 1000 by 1000 pixels, we would need only one quarter of our standard machine, costing well below \$1 million. Also, since the power of the machine is still far greater than is needed for the purpose, we will probably make each processing node time share between four or eight pixels, thus reducing the cost of the machine from \$3 million for the standard array to \$200,000 or so. There are 1000 input channels in parallel, each channel having a serial input rate of 100Mbit/s. This gives a total input data rate of 100,000Mb/s; well above the bit rate of a sequence of rasters of TV pixels. The compressed result is outputted down the columns, exiting from the array at the bottom. The compression will involve comparison of the new frame with previous frames, and the most recent 20 frames will be stored in the array. It is possible that the compressed output will travel in parallel down the columns of processors, and then finally exit to the right along the bottom (extra) row of processing nodes, which will have a bit rate capability of 100Mbit/s.

### **TV IMAGE ENHANCEMENT**

If, as seems likely, a reasonable performance TV data compression machine will only cost \$200,000 or so by reducing the number of processing nodes and making the survivors time share between four or eight pixels, then the same machine will be attractive for TV image enhancement. We can envisage all sorts of modifications to the video tape being programmed in via such a machine. We could correct for errors in shooting, and also programme in the background to a scene being shot in much more sophisticated ways, developing forward from the blue background.

### **ANALYSIS OF MEDICAL SCAN IMAGES**

X-ray and ultrasound scanning machines are expensive, and so sophisticated processing of the resulting images may be justified. Further, it is likely that if we add more image processing power using the kernel logic array, we will be able to tolerate lower quality in the scanning hardware, and therefore lower price.



### **AERODYNAMIC DESIGN**

A recent article by Dr E. Galea (see bibliography) discusses the pressing need for array processors in aerodynamic design and the ideal machine is clearly the standard kernel logic array processor with one million processing nodes. Galea shows that wind tunnel testing is unsatisfactory for car design because the ground beneath the car `moves', introducing major errors in the results. This is one of many reasons why supercomputers are gaining favour in such applications.

### **WEATHER SIMULATION AND FORECASTING**

The kernel logic array Processor will commit one processing node to each square mile of area. This is a good example of finite element analysis, where pressure, temperature, etc in one square will affect adjacent squares, and the array processor will have the power to let these effects ripple through the array. Weather forecasting will radically improve as a result of the greater (and also more appropriate, because distributed,) processing power.

A network of kernel logic array processors will make possible, and highly profitable, the real-time monitoring of weather throughout the globe giving highly accurate forecasting through the absence of the edge problem.

Ivor Catt's Kernel Consultants, PO Box 99, St. Albans, is currently seeking X5 million financial backing to build the prototype kernel logic machine.

---

## **Bibliography and References**

### **[Advance into the past](#)**

by I. Catt, Wireless World, Jan 1984, p.59

### **Brighter prospects for wafer-scale integration**

by R. Dettmer, Electronics & Power, April 1986, p.283-8

**Catt Spiral patents:**

UK 1377859, filed 3 Aug 1972, US 3913072, filed 3 Aug 1972, Germany 2339 089, Japan 1188600

**Catt Spiral**

picture Electronics & Wireless World, June 1988, p.592

**[Dinosaur among the data?](#)**

I. Catt, New Scientist, 6 Mar 1969, p.501/2.

**Kernel Logic international application**

PCT/GB88/0057 filed 15 July 1988

**Mental Models**, by P. Johnson-Laird, CUP.

**[Sinclair and the Sunrise Technology](#)**

by I. Adamson and R. Kennedy, Penguin, 1986, p.50-55.

**Supercomputers and the need for speed,**

E. Galea, New Scientist, 12 Nov. 1988, p.50.

**The Decline of Uncle Clive**

by I. Adamson and R. Kennedy, New Scientist, 12 June 1986, p. 33-6.

**The Nature of Explanation,**

by Kenneth Craik, CUP, 1943.

**Wafer scale integration: a fault-tolerant procedure,**

by R. C. Aubusson and I. Catt, IEEE Journal of Solid State Circuits, vol. SC-13, June 1978

**Wafer scale integration,**

by I. Catt, Wireless World, July 1981, p.37/8

[Jump to Top of Page](#)

## WHAT IS CATT'S SPIRAL?

There is only one proven method for generating a perfect array of chips out of an undiced wafer that contains faulty chips among the perfect ones. My approach is to develop a one-dimensional array (spiral) of good chips, adding further chips on to the far end, but all testing being under control of external test circuitry at the beginning, near end of the array. Each prospective additional chip is put through its paces by instructions travelling down the developing array through the chips already passed as good and already included in the array. If the next chip is adjudged faulty, it is disconnected and another chip adjacent to the penultimate one is tested out instead.

In my approach, the distinction between faults in manufacture and faults developing in service is blurred. On switch off, the array connections are destroyed-all links having been volatile - and the array is reconstructed from scratch each time the machine is powered up.

The chip does not test itself. The problem that a mad chip might demonstrate its madness by reporting that it is sane is evaded by having the main testing hardware outside the wafer. But all the same, the fact that powerful test-dedicated circuitry and also chip interconnection logic will consume only a tiny portion of today's chip's real estate is exploited.

I steal up on wafer-scale integration in a somewhat crabwise fashion. If (as is clear) we should start off with all chips, good and bad, cheaply interconnected during chip manufacture, and then open and close these connections by volatile information as a cheap way to exclude faulty hardware, it becomes inevitable that the major unit will be of maximum size - i.e. a complete wafer.

If I ask a mad man (mad chip) whether he is mad, then surely his answer is useless? The flaw in that remark is that I could ask not the whole chip, but only a small portion of that chip. Now today, it is possible for a portion of the chip to reply to such a question, yet that portion to be so tiny that the possibility of its being faulty can be, for practical purposes, ruled out

There are three weaknesses in the spiral approach. It is a one-dimensional array so access is limited to one entry point This is particularly limiting if the array contains many processors, each one needing continual input of raw data and also needing to deliver the results of its data processing.

The second and third weaknesses result from the high resistance of the aluminium lines across the surface of the wafer. This limits the amount of current and therefore power that can be delivered to the wafer. And secondly it limits the clock speed to 30MHz. Both of these are more damaging for an array processor than for an Anamartic wafer which is quietly storing data in ram. All three weaknesses are overcome in the kernel logic machine.

[Jump to Top of Page](#)

---

## Diagram headers

(see illustrations in above article)

In a kernel logic parallel processing array for air traffic control over Europe new data would update the array in a ripple-through manner every second. Aircraft collision avoidance will...

March 1989 ELECTRONICS & WIRELESS WORLD

... be achieved by transferring current data to an identical machine in the higher pages of the 1Mbit rams which will in effect be accelerated into the future by increasing the clock rate.

Connections between adjacent processing nodes have to be extended between wafers, as shown. In practice wafers may need to be arranged as an hexagonal or triangular array rather than a rectangular array.

A 'chip' or processor node is linked to the outside world in three ways: softwareselectable links to adjacent good chips, conventional metallized power and clock lines not shown, and stitch-bonded 0.13 mm wires to enhance power and clock by reducing resistance and increasing speed.

Potential targets need not be thresholded in a kernel logic machine because it will not be overloaded when the number of targets tracked reaches 100 - the overload point for today's early warning systems.

For air traffic control Europe Kernel Array Processor commits one processor to the airspace above each one square mile of earth, one page of ram per 10,000 feet of height. Higher pages are committed to an enlarged data cube.

[Jump to Top of Page](#)

[homepage](#)

