DR. DAVID WALLACE
University of Edinburgh

This paper covers some of our interests and experience at Edinburgh using the Distributed Array Processor (DAP). This is a machine which was built by International Computers Limited (ICL), a U.K. company. It is a very interesting device because it is truly an intrinsic parallel processor, consisting of a 64 by 64 hardwired array of elementary processors. The design of the machine was published ten years ago and the device itself has been available for about three years. Each of the processing elements is extremely primitive. However, because there are 4,096 of them, it is a rather powerful machine which approaches the performance of CRAY-1 for many of the problems of physics that we are interested in. The DAP is also very inexpensive. I think that the reason for this is twofold. First, these modular-structure machines are cheaper to design and to build so they can be produced more cheaply. Second, the major mistake that ICL made when they built the machine was to tie it into ICL mainframes in the hope that this would sell more mainframes. What happened was they didn't sell many DAP's. As a result Edinburgh got a good bargain; ICL was selling the DAP's for about a quarter of a million pounds in the end. I understand that six DAP's have now been built. I regard it as a successful first-generation machine on which we have already been able to do a lot of interesting physics and the group at Edinburgh is enthusiastically committed to this type of machine for our future computing requirements.

Four topics are covered briefly here: 1) what the machine is; 2) how we got it (involving the national funding bodies and links with industry in the UK); 3) a little bit about how it is used and, 4) a few remarks about prospects for this kind of architecture.

Figure 1 is a schematic description of the machine. It is a 64 x 64 array of processing elements, PE's, each connected to its nearest neighbor on the square, with periodic boundary conditions if required. Each PE is very simple; arithmetic operations are done by sequential single-bit manipulations. Switches control the transfer of data between neighbors on the array. Each PE has 4K of RAM so that in total there are 2M bytes of central memory. There is a master control unit broadcasting through the machine which controls the whole system so this is a single instruction but multiple data (SIMD) device; all the processing elements are doing the same things at the same time but on different data. ICL were really rather secretive about what the master control unit looks like, but for the user everything is very explicit and straightforward, as I shall indicate later.

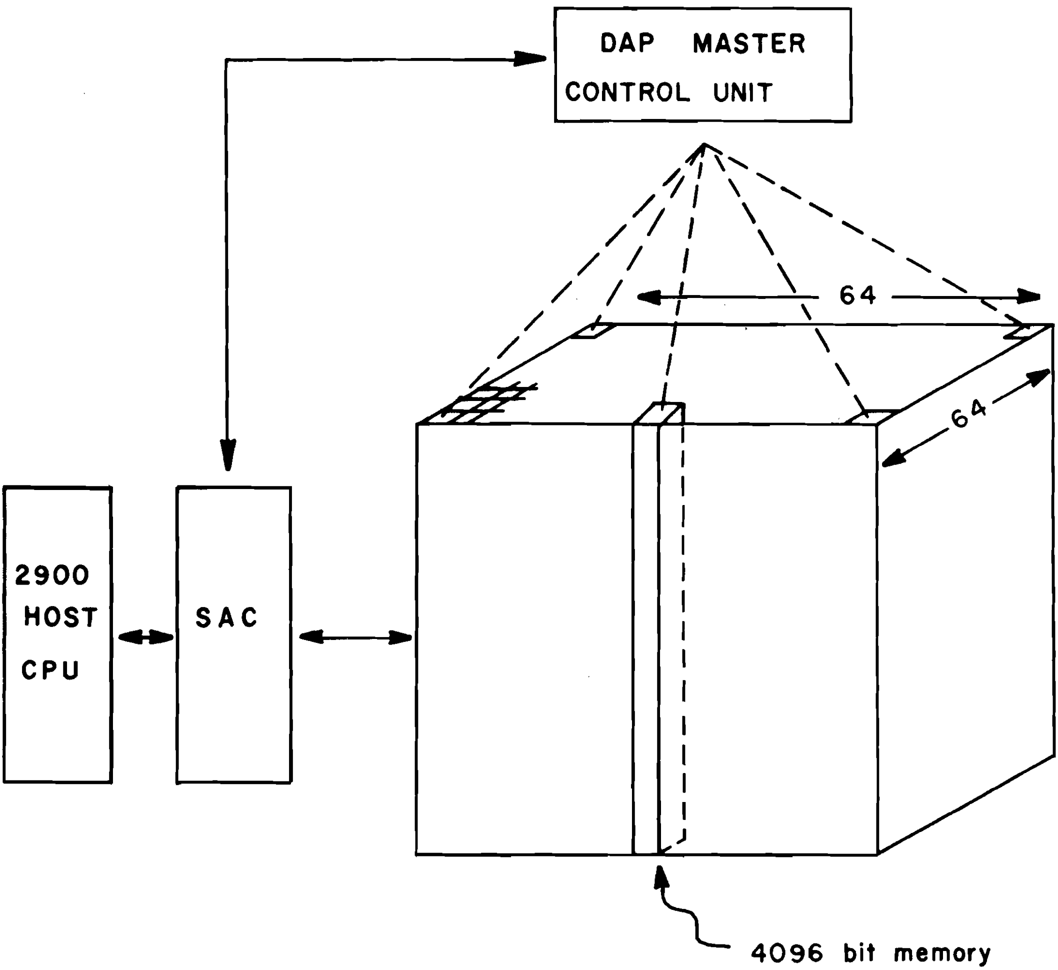

Fig. 1 DAP. Schematice architecture.

To understand how we got the machine it is necessary to understand the funding within the UK. By the standards of funding within the US, the UK has actually followed quite a sensible policy in recent years and the scale of the funding for a UK effort is not bad. The Science and Engineering Research Council (SERC) had a policy that it would fund central facilities which are what they describe as state-of-the-art computers, whatever that actually means. Three or four years ago this was easy to decide and they bought time on a CRAY which was installed near Manchester. This meant there was a general CRAY facility available to users in the scientific university community in Britain - not very much of it, but it was generally available. The SERC also set up a DAP unit at Queen Mary College in London. This arrangement was quite interesting in that the head of the DAP support unit also had an appointment with ICL so there was a clear link there on installing a machine at an early stage in a University in the expectation that fruitful developments would be made. Of course, this didn't quite meet Ken Wilson's criteria that prototypes should go free to universities because SERC had to pay for it. More recently the CRAY machine has been re-installed in the University of London Computing Centre for general southern region users and there will be a CYBER available in Manchester shortly. On top of that we have been able to acquire our DAP. We needed to find 270,000 pounds sterling or roughly $400,000 for it. After failing first in our efforts to set up a national Scottish facility we spent Christmas and New Years preparing an application to SERC in three blocks of 90,000 pounds sterling for work at Edinburgh in astronomy, solid-state physics, and elementary particle physics. This application went to different subcommittees of SERC who decided they would or wouldn't fund it. It's a very long story. The strength of our regional computing center links with ICL at this time should be gauged by the fact that they allowed us to ship the machine in and essentially set it in concrete in the machine room before SERC had decided to support it, in fact, when two of the three subcommittees had decided not to support it. That demonstrates ICL's commitments to get a machine to us. The relationship continues and we hope that they will build on the experience gained on the DAP.

Next, a little bit about the software for the machine. The way ICL set up the software for this new kind of machine on the first attempt is impressive and fun to use. It is a development of Fortran which they call DAP Fortran. Let me mention three features: (a) There is a lot of choice in specifying variables and constants. First, it is possible to have real and integer variables of various lengths (e.g., 1,2,...8 byte integer variables). Logical variables are particularly powerful and simple to handle. Both of these features one might of course expect in a machine with bit serial arithmetic. Second, in addition to the usual scalar etc. variables of standard Fortran one can also declare vectors of length 64 and 64 x 64 arrays whose elements are distributed over the 4,096 PE's. If A, B and C are declared as such arrays then in an equation such as A=B+C, the operation of adding B to C and putting it in A is done simultaneously on the different data in all of the 4096 PE's. The same holds for operations with the standard mathematical functions, e.g., A=SIN(B). (b) One of the most powerful facilities that ICL has built in is the ability to switch off some of the processing elements and decide not to calculate at those elements. That is done by the use of logical masks which are simply defined as logical, as shown in Fig. 2. The DAP has built-in logical functions which may look somewhat unusual but they are precisely the kind of functions that are needed for scientific computation, for example, alternating rows by one ALTR(1), as shown in Figure 2. More complicated masks can be built up with simple lines of programming, for example the chequerboard defined as alternating rows logically equal to alternating columns (Fig.2). This is the kind of mask that is needed if an algorithm says that one must perform calculations only on every other processing element, for example, if all odd sites must remain passive. These are typical requirements in the kind of calculations that we do. The typical FORTRAN statement that one then uses is A(L MASK)=B and this just puts B into A everywhere that L mask is true. That is very simple software for the user. (c) A final point worth mentioning is the shift operation which transfers information between the various PE's. For example a=B+SHWC(C,3) simply takes C and puts it into a processing element three units to the left, adds it to B and puts the result into A. This is done in parallel throughout the machine. Similarly there are shifts east, north and south with cyclic or planar boundary conditions: SHWC, SHEC, SHNC, SHSC, SHWP, SHEP, SHNP, SHSP. This is simple to implement and it is again precisely the kind of software that one has to have for the kind of calculations that we do.

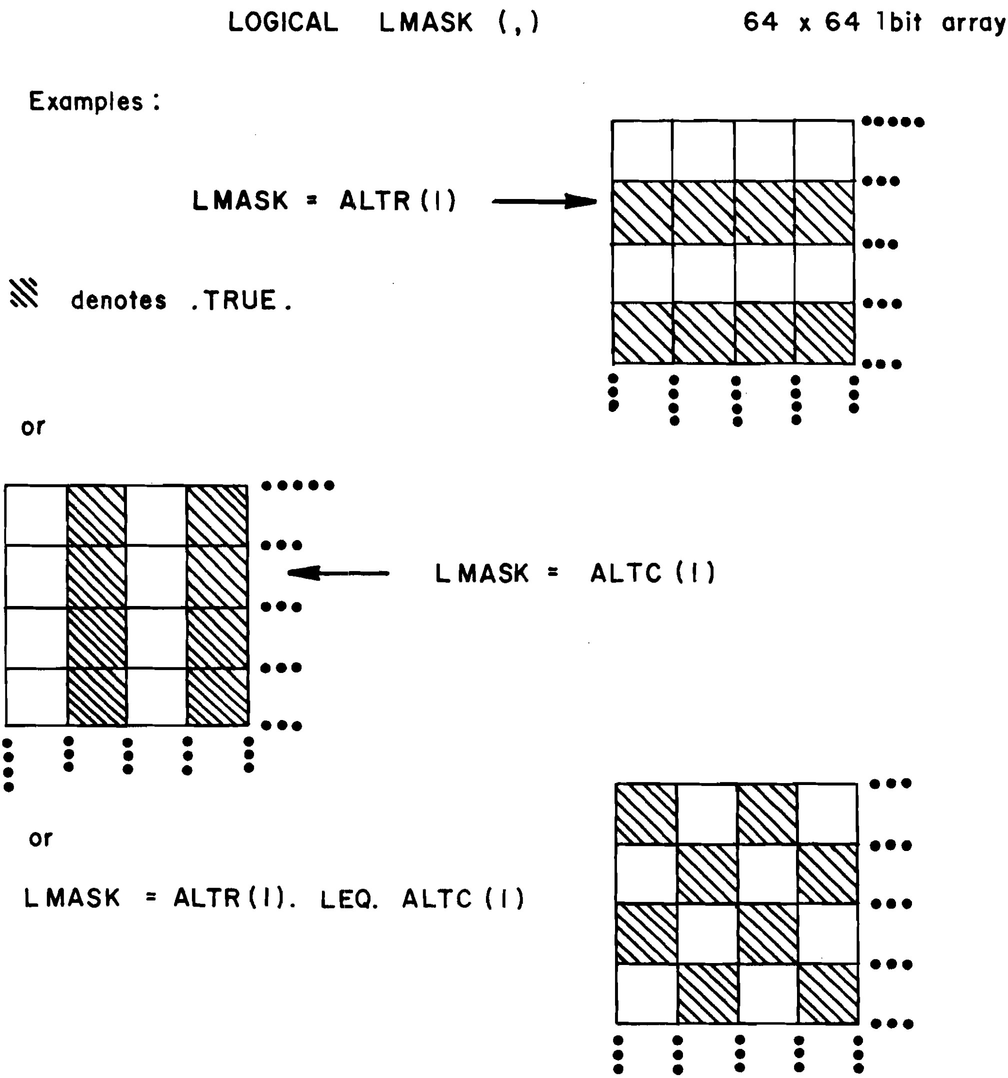

Fig. 2 Logical masks.

We use the machine for essentially the same kinds of calculations that Ken Wilson or Norman Christ would be interested in and which they will mention. These are problems where one wants to simulate a physical system. It is necessary to discretize that physical system (i.e., to approximate it by a lattice of points) in order to put it onto a computer and then one just associates so many of the lattice points of the physical system with a processing element. The updating algorithm is then begun and there is parallel updating of the simulation for all the processing elements in the computer.

Finally, what are the prospects for this kind of machine? The potential for future development is certainly very high. For example Goodyear is now building the "massively parallel processor" (MPP) for NASA and there are other developments along these lines in other companies. It seems certain that the next generation of machine will be twenty to thirty times faster and still be bit serial processing. Our general philosophy about the DAP is that it is a very good design for an engine, it is ideal for the kind of calculation that we do, and it has given us links with companies which will certainly increase.

The following is a selection of references covering general information and some specific applications developed at Edinburgh. The original DAP reference is: S.F. Reddaway, in Proc. 1st Annual Symposium on Computer Architecture (IEEE/ACM), Florida (Dec. 1973), pp. 61-65.

For further information on the DAP and its software see, for example, R.W. Hockney and C.R. Jesshope, Parallel Computers (Adam Hilger Ltd. Bristol, 1982); G.S. Pawley and G.W. Thomas, J. Comp. Phys. 47 , 165 (1982). 165.

Reviews of the Edinburgh group's work and further references are given in:

K.C. Bowler, in Proceedings of the Three Day In-depth Review on the Impact of Specialized Processors in Elementary Particle Physics, Padova, March 23-25, 1983 (University of Padua).

K.C. Bower and G.S. Pawley, to appear in Proceedings of IEEE, January 1984.

G.S. Pawley, in Proceedings of the Conference on Monte Carlo Methods and Future Computer Architecture, Brookhaven May 1983.

D.J. Wallace in Proceedings of Les Houches Workshop, March 1983, to appear in Phys. Reports.