Editorial – The Maturing of VLSI – and its Consequences for the Computer Industry

Editor's Note: The present special issue is aimed at identifying topics which will be especially significant in the near future. Although all of the topics chosen have an 'information' flavour, not all the developments to come will involve information theory. A view from the hardware side is presented in this guest editorial, supported by the first two papers in the issue.

The computer industry is at present undergoing a major upheaval. This situation has come about as a result of the maturing of Very Large Scale Integration (VLSI) to the point at which it is possible to fabricate single-chip processors that have a power equal to that of the processors in the fastest currently available minicomputers. Parallel developments of equal importance have led to high-density memory chips. The next generation of single-chip processors will break into what is now the mainframe range.

In the early years of semiconductor process technology, progress was slow and uncertain. VLSI was talked about and there was a general understanding that it would eventually arrive, but no one had any idea when that would be. The first success was achieved by Intel around 1971 but, as they were able to put no more than 1000–2000 transistors on a chip, little interest was aroused in the computer industry. A more significant breakthrough occurred about three years later, when it became possible to put enough transistors on a single chip to make a processor of simple but practical design. The first to come was the Intel 8080, and it was followed by the Motorola 6800. Neither of these processors was powerful enough for use in a minicomputer, and for that reason the established computer industry almost ignored them. They did, however, give rise to the low-cost personal computer; this was a strikingly new departure, and was destined to give rise to a new branch of the industry.

A further breakthrough came in the mid-1980s, with the coming of much more powerful single-chip processors. These, along with the new high-density memory chips, made possible personal computers that were sufficiently powerful to interest users who had found the earlier models entirely inadequate for their needs. Development of these new more powerful personal computers was taken up by people who saw their importance in the context of the older industry, rather than by engineers from the personal computer side of the industry. This accounts for the fact that they were, from the beginning, described as work stations rather than personal computers.

By this time the semi-conductor industry was getting into its stride, and progress became much more predictable. The power of single-chip processors steadily increased and, as I said above, has now reached the top of the old minicomputer range and is pushing up into the mainframe range. This progress will continue as transistor density on the chip is increased and chips become larger. The gain in speed will come partly from the increase in intrinsic speed that results from the transistors being made smaller, and partly because it will be possible to accommodate more high-speed memory on the chip itself. It is safe to predict that, within a few years, single-chip processors will come to match in speed the multi-chip processors to be found in present-day computers of the highest performance.

Now that the single-chip processors going into work stations are as powerful as those that used to be found in minicomputers, the minicomputer is becoming nothing more than an overgrown work station. It uses the same processor, but has more memory, more disc storage, more peripherals, and more communication equipment. The same applies to servers of various kinds, especially file servers.

The overgrown work station will continue to increase in power until it has superseded not only minicomputers as we know them today, but large mainframes as well. However, this is still to happen. Very large mainframes and supercomputers continue to exist and are in a healthy state. Their power depends on the fact that they combine very high speed with very great I/O bandwidth and massive bulk storage capability. There is room for them to develop in these directions.

The maturing of VLSI is having a profound effect on the computer industry. Computing equipment is now much smaller and it consumes much less power. The effect may be summed up by saying that, whereas in the past the products of the industry have weighed hundredweights or even tons, in the future they will weigh pounds. Several years ago, the established companies began to realize that their existing production facilities were on too large a scale for their future requirements. They initiated, and are now completing, major and painful operations designed to cut them down to size.

To add to the computer industry's problems of adjustment, other major changes are going on. These have come about in the context of work station development, and it is possible to see them as a natural consequence of that development. On the other hand, they may be seen as developments that were inevitable, if not overdue, and only needed some appropriate trigger to cause them to happen.

The first change concerns operating systems. For more than twenty years it has been clear that operating systems would eventually follow programming languages and become processor-independent. The first major processor-independent operating system to have emerged is UNIX, and it has emerged in the context of work stations. The current situation is that the scientific market demands that all work stations and servers based on them must have UNIX.

From the earliest beginnings of the computer industry to the present time, companies engaged in the manufacture of computers have had their own proprietary processor instruction sets. On the basis of these they established and held together their individual customer bases. This is still the case on the traditional side – as distinct from the personal computer side – of the industry. It is emphatically not the case in the new market based on work stations that is now emerging. Whatever the processor inside the work station, it is UNIX that the user sees. The question he asks about any new software is whether it will run in a UNIX environment, not whether it will run on a particular processor.

The development of work stations leads naturally to the open-system approach to computer installations. According to this approach, a computer user installs a local area network and then shops around for file servers, hard copy servers, bulk data servers, and work stations to connect to it. He builds up a system to suit his needs and to take advantage of the most attractive offerings of the industry. Whether or not this becomes the standard model for the computer system of the future, there can be no doubt that the harmonious working together of equipment from different vendors will become a permanent requirement in the industry. For example, all work stations or personal computers connected to the network, whether they run UNIX or some other operating system, must be capable of accessing the same file server. Companies will sell components as well as complete systems and the components will form a major part of their business. This is already beginning to happen, and is having the consequence of making the industry much more competitive than it was in the past.

Users as well as computer manufacturers will have problems of adaptation to the changing state of affairs, especially if they have been accustomed to rely on their chosen vendor for system support. Falling margins due to increased competition will make it hard for vendors to maintain the same standards of user support as formerly. It will be tempting for users to shop around for equipment, but problems are bound to arise from time to time when nominally compatible items are added to a system. Not all users, in the first instance at any rate; will be prepared for such problems. Hardware vendors and independent companies alike are seeing here a business opportunity in the provision of support services in system management.

The technical directions are now clear, and the industry is in the process of adapting itself as best it can to the biggest upheaval that has occurred for many years. It is unfortunate that this upheaval should come at a time when there has been a general recession in industry as a whole. I make no predictions about exactly how things will work out. There may well be some surprises, but I remain optimistic for the future health of the computer industry and the major companies in it.

MAURICE V. WILKES

Book Review

CHRISTOPHER JOHN HOGGER, *Essentials of Logic Programming*, Oxford University Press. £15.00, ISBN 0-19-853832-4

Christopher Hogger wrote his first book on logic programming some ten years ago. This new text could be seen as an update of the earlier book but really it is not so. It is a new introduction to logic programming, which is not only up to date now but has the benefit of the author's further ten years of teaching the subject and the lessons which have been learned from that.

The book is organised as a set of 60 themes, and the treatment of the themes, compared to chapters in a conventional book, is short and swift. Within each theme only a small amount of material is treated, but each theme concludes with a set of comprehensive exercises, which are supplied with full answers. This frequent reinforcement is a substantial aid to understanding and remembering.

This book is about logic programming and not, for example, about logic-programming languages such as Prolog and Lisp. It is theoretically rather than practically based and includes *inter alia* Herbrand interpretations and an introduction to First-order Logic, the problem of Resolution and the problem of Theorem provers. It begins with a background and overview, continues with First-order and Clausal-form logic and problem solving. It goes on to consider the Herbrand and Domain resolution and concludes with the problem of program verification.

This is very much a tutorial text, but it is aimed at the higher levels of computer science courses. A mathematical background or some background in Logic is essential. For students on M.Sc. courses in Expert Systems or a B.Sc. course in Artificial Intelligence it is ideal, on the other hand the highly theoretical approach makes it quite unsuitable for vocational-orientated courses such as the BTEC HND.

The treatment is straightforward, the book is not littered with many references and quotations which would make it difficult to read, and the short themes make for easy learning. Its 300 pages contain a wealth of information and at £15 it is particularly good value for students.

JOLANTA JAGIEŁŁO
London

NIGEL WOODHEAD, *Hypertext & Hypermedia: Theory and Applications*, Addison-Wesley, London, 1990. 231 pp. £19.95, ISBN 0-201-5444-3.

This book is a discursive overview of hypertext and hypermedia. It begins with an introduction to the subject together with its history. It then goes on to illustrate, by example, the potential that the technology offers. It also suggests a methodology for building applications, looking at owner, author and user issues. In addition, a listing of sources and products for various systems is given. It must be stressed that the book provides an overview, and more detail on the research mentioned in the book must be sought elsewhere. An excellent set of references is given to enable the reader to do this. Nigel Woodhead is a technical writer, with experience in hypermedia applications. This means that the book is written in an objective way and none of the products or ideas are given any special treatment.

The book is structured, like many others in this area, to be read in a nonlinear way. Due to the wide range of people it is aimed at, it is not anticipated that everyone will read the whole book. Navigation is via contents pages at both section and chapter level. Links in the text to other paragraphs are also provided; these are used sparingly, but help to give the reader a flavour of their potential. Page numbers attached to these links would be a welcome addition; this would encourage a greater number of readers to experiment with non-linear reading.

The book is divided into an introduction and three sections. The introduction provides a history of the development of hypermedia technologies and also introduces some terminology for those new to the area. Section one, 'The potential of hypermedia', relates the formal characteristics of hypermedia to those in other areas. The areas covered are databases, object-oriented approaches, text-processing approaches, artificial intelligence and electronic publishing. Section two, 'Owners, authors and users', discusses the issues relating to these three parties. The section on authoring issues is particularly comprehensive, covering tools, styles and methodology. Section three is a listing of the packages available for the IBM PC, Apple Macintosh and workstations Accompanying each is a brief description and review, and also supplier details. There are also appendices which give details of available shareware and shellware.

The author suggests that the book is of use to software developers, technical authors and electronic publishers, as well as teachers and students. It is indeed useful for these people but it may not be of significant depth for software development professionals and they may find many redundant sections, which might make it not a worthwhile investment. Otherwise, the design of the book, both the presentation and the flexible layout, justifies the cover price.

The book is very comprehensive and easy to read. This makes it especially good for students of many disciplines, including business studies, computer science and information technology, although it is more suitable to postgraduate and research students since there are no worked examples or exercises. As an overview the book works well and it is a good initial step into this area. The numerous references allow readers to follow up the areas of interest easily.

CHRISTOPHER HANKINS
London