The Impact of the Microprocessor

Anthony C. Davies

The Impact of the Microprocessor

Anthony C Davies
Emeritus Professor, King's College London, Visiting Professor, Kingston University, Kingston-upon-Thames, Surrey, UK
tonydavies@ieee.org

Abstract: A description and explanation based mainly on the author's personal experiences of the changes in the curriculum for electrical engineering undergraduates and in the required expertise of practising electronics engineers which occurred from the mid-1960s. The changes began with the introduction of digital system design methods, and increased with the subsequent introduction of microprocessors as widely-used programmable components, for which software design expertise was an essential part of their utilisation.

Keywords: Microprocessor, electrical engineering students, curriculum

1 The Higher-Education Background in UK

In the 1960s and early 1970s teachers in UK universities had considerable freedom to interpret the syllabuses of courses which they taught. Around 1965, I was teaching a final year undergraduate course for electrical engineers which contained no material on digital electronics. However the syllabus contained the phrase "and/or gates", no doubt intended to occupy only a few minutes of one lecture. I decided to interpret that as an excuse to include a substantial amount of material on digital logic including Boolean algebra, combination logic design, synchronous logic design, which developed into around half of the course content. By the mid-1970s, I was involved in teaching essentially the same material to the first year undergraduates, where it had been transferred against the opposition of the older academic traditionalists. Much the same happened at about the same time in many other UK universities.

Boolean algebra had been included earlier but typically taught by mathematicians who may have known about the original application to logic of a different kind but usually had no idea of the application to switching circuits and electronics. Such teaching did not have a favourable impact on most engineering undergraduates.

In parallel with this it had become common for engineering undergraduates to learn programming in languages such as FORTRAN, used to solve engineering problems, submitting their work by punched-cards or paper tape and receiving the results hours (or sometimes, days, later). By about 1970, they were typically also using electromechanical teletypes for multi-user access to a central computer providing fast return of results. Later, a few had access to and use of a mini-computer such as the PDP-8 for measurement, instrumentation or control applications and this could be in a real-time context.

Errors in programming resulted in wrong answers or crashed programs leading simply to modifications and re-submissions. No concepts such as software engineering or software design were involved. It was considered a sufficient education to have a FORTRAN language handbook and see a few examples [1].

Electronic engineering undergraduates were unlikely to be taught about computer architectures or assembly language programming except in a very superficial way.

It was against this background that the microprocessor as an electronic component appeared in the form of the 4-bit Intel 4004, announced in 1971 and the 8-bit Intel 8008 the following year. General availability had arisen by 1973, and the impact was really being felt in industry by 1975, with the availability of 8-bit microprocessors such as the Intel 8080 and Motorola 6800.

2 The Initial Context and Impact of the First Microprocessors

The microprocessor created a shock amongst local electronic component distributors. They found out they had to sell a new kind of component which needed not just the support of a single page datasheet but, rather, huge manuals fall of unfamiliar material plus additional supporting supplies such as a deck of punched cards comprising a cross assembler to run on a mini-computer or mainframe computer, truly a "New World" for which their sales force was totally unprepared and untrained.

Figure 1: 126 page User manual for Intel 8008 [2]

A further complexity for both the distributors and potential users of these components in new engineering designs was that each major semiconductor device manufacturer wanted to produce its own unique microprocessor range (see Appendix and Table I). Each microprocessor had a different architecture, a different assembly language and different package pin-outs and needed different 'support-chips' to make a working system. Deciding which microprocessor was the best to choose was a major hurdle for practising engineers as was deciding what was the fundamental expertise needed to use these devices and what was salesman's hype.

At the time there were many companies making integrated circuits, but most were not computer manufacturers. Microprocessors involved making a quite new class of product and so moving into uncharted territory for them. There was no move towards standardisation, each manufacturer hoping to become a market leader. Some 'second-sourcing' agreements were made (for example between Fairchild and Mostek for the F8 and later Zilog and Mostek for the Z80), since this was seen as a sales-advantage with some classes of customer.

3 The Education and Training Responses

All this created a market for short courses of a few days duration run by entrepreneurial engineers who could hire some small rooms in a hotel to present introductory courses focusing on a particular microprocessor type and including hands-on work with inexpensive kits. Typically such courses included simple machine-code and assembly language programming tasks which could be immediately executed on the kits and the participants carried out such assignments as controlling a seven-segment numeric display, responding to the pressing of a push-button, generating simple sequences and waveforms and perhaps even starting and stopping a simple miniature electrical motor.

My assessment of such courses at the time was that they were rather too expensive for many electronics engineers in industry and too detailed in content for the senior managers who could afford the fees but perhaps would have felt that assembly language programming of the kits was beneath their dignity. I therefore had an idea to start running very low cost but useful courses for practising electronics engineers, and accordingly, with the support of my university and IEEE and the aid of two engineers from a nearby industrial research laboratory[1] who had been working intensively with microprocessors, our first short course of weekly evening lectures was run for a fee of only £10 in October 1975 (Figure 2a).

By any standards that was extremely cheap and affordable, our aim was not to make money but to do something useful without the risk of actually losing money. This aim was comfortably achieved and led me to promoting and teaching a sequence of frequent short courses on various aspects of microprocessors over the following years (such as the one illustrated by Figure 2b). Other universities and polytechnics in the UK also began running such courses.

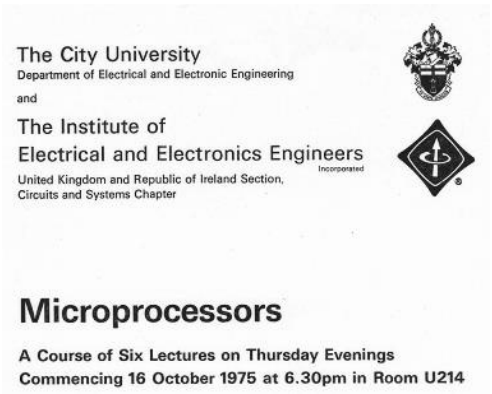

Figure 2a: First microprocessor short course at The City University

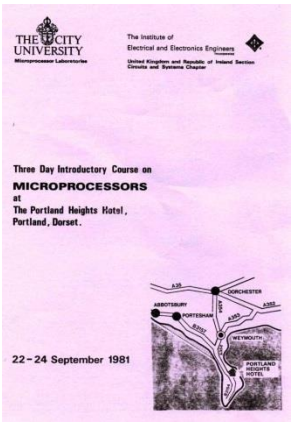

Figure 2b: Short course in a hotel near a military research establishment

[1] David Wright and Daphne Shipperlee, Standard Telecommunications Laboratories, Harlow, Essex

Figure 3a: Typical single-board 'evaluation kit'

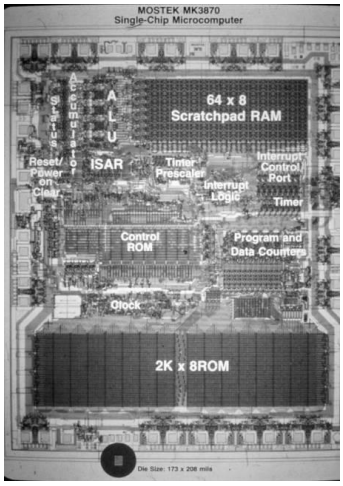

Figure 3b: Single-chip microcomputer: a new kind of component for the electronic-system designer

4 New Outlooks and the Educational Curriculum

It was clear by this time that a new electronic design paradigm had arisen. There was the prospect of using programmable electronics rather than hardwired logic (which came to be called random logic) in many kinds of electronic products. An obvious consequence was a need for major retraining of engineers in industry, a new responsibility for Learned Societies in the engineering domain, and a need to do something about modernisation of the typical electronic engineering curricula of universities and polytechnics. However understanding what to do and how it should be done represented a challenge, with indecision and many obstacles for senior people in academia and industry and the Professional Institutions.

At the management level (staff and volunteers) of the IEE (Institution of Electrical Engineers) the realisation that something had to be done was accompanied by a general belief that the scope of the Institution did not permit inclusion of either computer architecture or computer programming and a realisation that many of the people knowledgeable about microprocessors were not members of the Institution and maybe not even qualified to become members. From the perspective of the traditionalists in IEE, the British Computer Society was made up of what they felt were mainly hobbyists and amateurs whom they considered to be irreconcilably distinct from the professional engineers of the IEE.

Nevertheless, seeing an urgent need to become involved with microprocessors, they set up a somewhat independent Microprocessor Application Group, with funding and publicity which enabled IEE to have committees, arrange meetings, and generally support this topic, without really either understanding it or admitting it into the scope of IEE. Growth in this subject area enabled them after a few years to incorporate it

within the scope of IEE, although arguments continued about whether computer architecture and software engineering belonged in the Institution. For a while there was an uninformed view among some senior members that software engineering meant using computers to solve engineering design problems and they could not or would not recognise that it meant designing software in an engineering manner for any kinds of applications, including embedded software in real time systems incorporating microprocessors.

I recall being at an IEE Council Meeting at which one of the older members suggested that IEE should have nothing to do with Computer Architecture because that would lead to jealousy from the Institution of Civil Engineers.

At about this time, The City University in London started a bachelors degree course called 'Computer Engineering' [3]. Figure 4 shows the cover of a brochure for this course. The intention was that this would be for the 'modern' type of future electronic engineer, in recognition of the importance of computers in all domains, including the use of microprocessors in electronic projects. Getting academic approval and professional institution accreditation proved difficult, and instead of bringing in hoped-for collaboration between the electrical engineering and computer science departments it led to incomprehension and something more akin to warfare.

Nevertheless it provided a route by which much more digital electronics and microprocessor work was incorporated into the undergraduate engineering program.

The typical student experience included similar laboratory experiments to those used in the commercial short courses, e.g. designing and executing simple small assembly language programs on various microprocessor kits, to carry out simple real-time tasks. For the first time electronic engineering undergraduates were introduced to the mechanisms of subroutines and interrupts rather than just using high-level languages to solve problems by sending programs to a computer centre and waiting for the results. It became feasible to 'slide' this laboratory work gradually into that for those on the traditional electronic and electrical engineering degree courses without the opposition of the traditionalists.

The first microprocessor-evaluation kits often had only about one-millionth of the amount of random access memory (RAM) to store programs and data that is today provided in a typical basic laptop computer for home-uses running the Windows operating system.

Initially we made much use of a simple 'kit' for the F8, shown in Figure 3(a), which had a small operating system with a few subroutines for input and output, a teletype interface, a loader for machine code programmes and (almost) 1024 8-bit words of user programmable memory. Programs written in assembly language were translated to machine code by a cross-assembler (written in FORTRAN!) running on a main-frame computer (ICL 1905), with output on a paper-tape which was loaded into the 'kit' using a teletype. Soon after, with the availability of microcomputer systems with more capable operating systems and having storage on 7 inch floppy discs, it became possible to assemble and test the programming tasks in a self-contained portable unit, no longer requiring access to a main-frame computer. Fluency with hexadecimal arithmetic was one of the skills students had to acquire.

It should not be assumed that we 'educationalists' involved in these changes were wise and far-sighted prophets, we were just somewhat less hidebound by tradition than some senior academics and professional engineers. We were aware of Moore's

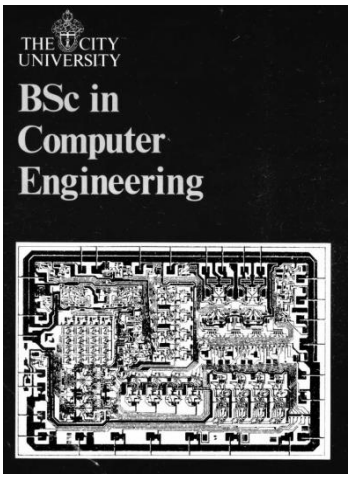

Figure 4: Computer Engineering Course brochure (shows metallization pattern of Am2909 Microprogram Sequencer)

Law [4], and observed the exponential rise in the complexity of digital integrated circuit chips, but (at least in my case) concluded that this could not go on for long, because the Law indicated that soon there would be a million transistors per chip, from which I concluded (and told others) that designing a microprocessor with so many transistors would be beyond human capability. Of course, that proved completely incorrect, and now single-chip processors with transistor counts of one billion are commonplace (and the fabrication cost per transistor is apparently less than the cost of growing one grain of rice). The first microprocessor chips comprised a few thousand transistors

I also recall a public meeting at which an experienced engineer insisted that there would never be 16-bit microprocessors using silicon technology, because it could be proved from fundamental physics and the laws of thermodynamics that too much heat would be generated. At the time the first rudimentary 16-bit microprocessors were already being developed, and now silicon 32-bit and even 64-bit processors are perfectly feasible and in use.

5 The UK Government Response to Microprocessors

While all this was going on there were members of the UK government who recognised that the introduction of the microprocessor represented a fundamental change for the engineering industry and their products. They perceived that if the UK did not take the correct steps to participate successfully in this change the future prosperity of the UK might be severely damaged.

In addition to commercial applications related to national prosperity this was also the time of the Cold War and so the relationship to military systems and defence could not be forgotten.

It was within this context that in 1979, funding of £15 million was provided to the National Computer Centre for a microprocessor application project (MAP), which invited bids for government financial support for training courses and schemes aimed at updating British industry in the use of microprocessors in manufacturing and in end-products. This source of funding attracted the attention of the universities and polytechnics. In the universities it was often seen as a way of creating additional income by running short courses for industry for high fees and so generating comfortable surpluses. In the polytechnics their funding schemes often meant that any surpluses created by their activities, including short courses, reverted to the local authorities that controlled them and brought no benefit to the polytechnics. This meant the polytechnics often ran short courses charging fees which just covered the

direct costs and so spoiled a lucrative market for the universities and also for the small entrepreneurs previously mentioned who were running courses in hotels and making a comfortable income for themselves.

One of the outcomes was that many expensively produced university microprocessor courses attracted few customers and did not create the hoped-for financial surpluses. It could be said that the market was flooded with far more courses than was needed. Nevertheless some did continue successfully and I had personal experience of teaching such short courses in Washington DC and in Berlin as well as in or alongside various industrial companies and government establishments around the UK.

Soon afterwards notice was being taken of what was called the Fifth Generation Computer Project underway in Japan, the purpose of which included research and development to solve some of the difficult problems for which computers had so far been unsuccessful (for example speech recognition, speech understanding, speech synthesis, image processing and recognition and various forms of so-called artificial intelligence). The microprocessor was denoted the 'fourth generation' in computer hardware, and so the Japanese programme was intended to be the next major step.

Shortly after this, the UK Government launched the Alvey Programme (1983-1987), a very well-funded scheme with the aim of moving forward UK research in the computer and information engineering areas with the specific idea of enhancing the position of the UK in associated scientific development and economic prosperity. It was to a large extent based on the assumption that competing with the Japanese Fifth Generation Computer project was necessary. The programme provided a convenient and welcome boost to the funding of related research in UK universities, of particular help to Computer Science, although in retrospect, not much seems to be recalled about the positive outcomes.

There was a sensible realisation that teenage children needed to get some experience of microprocessors, and the many simple and fairly expensive microcomputers becoming available was leading to amateur and hobby groups being formed. One significant consequence was the BBC Micro, based on a design from the Acorn Company (the Acorn Atom) which was chosen from several competing alternatives, and this became widely used to teach programming in schools, clubs and in many university electrical engineering departments. It used a non-standard but somewhat superior form of the BASIC language, with the inclusion of capabilities to support graphics programming and real-time interfacing. The cost was low enough for widespread adoption, and the BBC Micro was powerful and versatile enough to provide an educational foundation for beginners and to be used in simple real-time control engineering applications and experimental work.

Whereas in the early days of semiconductor technology and the move towards integrated circuits, very many of the major electrical engineering companies became involved in fabrication, making their own products, as the complexity of digital integrated circuits increased and the feature sizes of the transistors decreased dramatically (as predicted by Moore [4]), the cost of an up-to-date semiconductor plant became unaffordable for more and more companies. This is now well-known.

In the hope that the UK could remain active and competitive in this field, the INMOS company was formed with government financial support, against a background that most of the British companies which had semiconductor

manufacturing capabilities were closing them down or not keeping up with the general advances and instead limiting themselves to specialized niche markets.

Ultimately, INMOS had its government support withdrawn and its activities effectively terminated. However, it was responsible for one very significant product, the INMOS transputer, a microprocessor with a very different architecture specifically intended to support a multi-processor design paradigm. This approach was supported enthusiastically by many UK Computer Science Departments as a basis for both their teaching and research, and brought them more into contact with industry and contract-supported research. However, the final demise of INMOS and the transputer limited the long-term impact of this, and it seems to have done little in most universities to bridge the gap between Electrical Engineering and Computer Science Departments.

6 Some more Technical Details about Microprocessors

Initially, a minimum working system needed several integrated circuits but soon single-chip microcomputers (microcontrollers) appeared, with everything needed in a single 28 or 40 pin dual-in-line package, either mask-programmable for quantity-production (as in the Intel 8048) or with user-programmable memory which could be erased with ultra-violet light and re-programmed for design and experiment and for small-number production runs (as in the Intel 8748).

A significant constraint was the number of pins available when using the industry standard dual in-line packages. The Intel 8008 was in a 18 pin package (Figure 5a) and subsequently packages with 40 pins were generally used for the early microprocessors (Figure 5b). Such large packages were already at the limit of conventional manufacturing and assembly processes for electronic circuit boards. Texas Instruments developed the 9900, a 16-bit processor in a 64-pin dual in-line package, based on their 990 minicomputer. The package was difficult to handle and represented an upper limit for the dual in-line format. Later, totally new kinds of package had to be developed with far more pins (such as the Pin Grid Array) and some with special cooling arrangements. The Intel 80386 was in a 132 pin PGA. Figure 5c shows the Intel 80486 in a 168-pin package with the corresponding socket shown in Figure 5d. Many other types of package were developed to cater for increased miniaturisation and more pin numbers.

The first microprocessors used PMOS technology but this was soon replaced by the faster NMOS and later by CMOS which allowed much lower power consumption. Bipolar technology offered greater speed but could only achieve much lower circuit complexity per integrated circuit and it was used mainly for military applications until CMOS technology “caught up”.

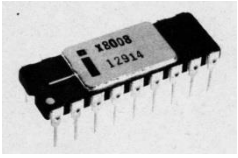

Figure 5a: Intel 8008 in 18-pin dual-in-line package

Figure 5b: 40-pin ceramic dual-in-line package

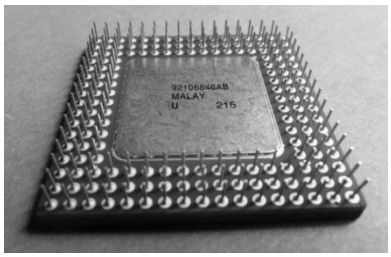

Figure 5c: *Intel 80486 in 168-pin PGA (underside)*

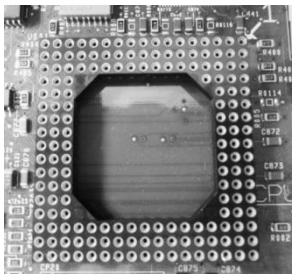

Figure 5d: *Socket for PGA*

7 Languages to Program Microprocessor-Based Systems

Despite the limited performance of the early microprocessors, they enabled the development of home computers and simple desktop computers which began to be used for general office applications such as document preparation and financial calculations. The word processor and the spreadsheet became universally familiar in all office environments.

These small computers provided the means of supporting the programming of microprocessor-based systems. Initially that involved assembly language programming but as the computers became more powerful the use of high-level languages with microprocessors began. This was also a necessary development to enable these small computers to be used by various kinds of beginners (hobbyists, teenagers in schools, and those with no electronics or computer background who were finding interests and applications in such things). Nevertheless, it was still possible to hear 'experts' claiming that high-level languages were inappropriate for use with microprocessors!

In the University teaching environment, there was typically a split between those who preferred the BASIC language and those who preferred languages in the Pascal and Modula-2 style. Most often it was the engineering departments using assembly language and BASIC, while computer science departments understood the conceptual advantages of Pascal and were sometimes rather uninterested in or ignorant about hardware and real-time applications of interest in engineering. The typically limited comprehension of software engineering principles and good software design was an often unrecognised handicap in the engineering departments. Indeed it was often a handicap among practising engineers in industry.

8 Desk-top Computing

The microprocessor made possible the development of the home computer and the office desk-top microcomputer. Initially with very limited capabilities, they soon improved to make their use normal for document preparation and financial management¸ as well as providing a basis for the support of teaching of all subjects

and the tools needed to design many kinds of microprocessor-based electronic systems. The plentiful diversity of initial products[2] in the market simplified with the release and dominance of the IBM PC and Apple, becoming standards with which other designs failed to compete. The open nature of the IBM PC design stimulated the production of IBM-compatible computers from many sources, often with better performance or lower prices, and this stimulated a huge range of hardware additions and software.

The IBM PC design ensured dominance of the Intel microprocessor range and laid the foundations for the success of Microsoft, while Apple provided an alternative built around Motorola products and supporting somewhat a different conceptual basis, with more emphasis on graphics and the human interface in a closed system which discouraged a market in competing but compatible products. The Apple design may have laid the foundation for their ipad and iphone, and similar products from other manufacturers.

9 Conclusions

The 'birth' of the microprocessor heralded a fundamental change in the educational framework, the practice of engineering design and the products of the manufacturing industries. It is now inconceivable that widely-used products from automobiles to telephones to military weapons systems (including aircraft) would be designed without incorporating microprocessors and microcontrollers.

This has fundamentally altered the education for and practice of the engineering profession.

One could say that it was a 'difficult' birth, but one whose consequences changed lives throughout the world, and laid the foundation for the internet and mobile phones and much that is now taken for granted as essential aspects of modern life.

Appendix

Table 1 gives an indication of the somewhat bewildering assortment of microprocessor designs which arose in the early years, with no claim that it is complete. It illustrates the situation that confronted electronics engineers, most of whom had little or no computer experience but who had to choose and learn to understand and use such novel components in products which they had to design. Only with hindsight can it be seen that just a few semiconductor manufacturers (e.g. Intel, AMD, Motorola and Texas Instruments) would survive in this market as the complexity and cost of making state-of-the-art integrated circuits increased. From

[2] A table in Personal Computer World magazine in 1980 [5] lists 83 microcomputer systems on sale in UK, of which only a few are generally remembered today (for example, Acorn Atom, Cromemco, HP85, North Star Horizon, Commodore PET, Sinclair ZX80, Tandy TRS 80)

some ultimately-unsuccessful designs (e.g. the MOS Technology 6502), the ARM computer design developed, and became the central processor used in almost all mobile phones. It is the architecture which supports Blackberry® and the Android operating systems and is used in Kindle e-books.

Table 1: Diversity in early microprocessor designs

Manufacturer	Name	Others (if applicable)	Comments
Intel	4004	8008→8080→8085→8086	Developed into the 80286, 80386, 80486 range leading to current designs
Motorola	6800	6000→68000 and 6809	Developed to 68020 and current designs. 6809 came too late for success
Fairchild	F8		May have been based on LP8000. F8 was second-sourced by Mostek.
Zilog	Z80	Z8, Z8000	Z80 second sourced by Mostek, development via Intel 8080. Z8 is a single-chip microcomputer.
General Instruments (GI)	LP8000	CP1600	CP1600 was an early 16-bit microprocessor.
Rockwell	PPS-4	Later, R6500 was produced	R6500 was developed from MOS Technology's 6502.
Signetics	2650		Popular for a while with hobbyists.
Texas Instruments (TI)	TMS1000	9900 16-bit microprocessor produced later, but failed to find a significant market.	TMS1000 4-bit microcomputer used in many low-cost embedded-computer applications. Later TI prominence was with signal processing products such as TMS 320 series.
MOS Technology	6501	→ 6502	Used in BBC Micro.
National Semiconductor	SC/MP	Later, 16032 → 32016 → 30032. NSC800 (a CMOS version of Z80)	Pronounced "scamp". Independently and later, 32000 series was a very early 32-bit processor.
RCA	1802 (COSMAC) and 6511		COSMAC was an early CMOS design, offering very low power consumption.
Intersil	IM6100		12-bit CMOS design, with architecture similar to PDP-8 minicomputer.
Ferranti	F100-L		16-bit microprocessor, military sponsor based on FM1600 B minicomputer. Promoted as first European microprocessor, but not commercially successful.
Intel	8048	→8048→8051	Single-chip microcomputer
Mostek	MK3870		Single-chip microcomputer, based on F8.

A more detailed comparison of the wide range of microprocessors and microcomputers was made by Depledge [6].

References

[1] D. D. McCracken 'Guide to Fortran Programming' , Wiley, 1961
[2] Intel Corporation, MCS-8 Micro Computer Set, Users Manual, 1973 (Revision 3)
[3] The City University, Bachelors Degree Course in Computer Engineering, Syllabuses, May 1974 (London)
[4] G.E. Moore 'Cramming more components onto integrated circuits', Electronics, Vol 38, 19th April 1965
[5] 'In Store', Personal Computer World, pp115-118, Vol 3, No 10, October 1980, Sportscene Publishers, London.
[6] P.G. Depledge 'A review of available microprocessors' pp44-53 in "The challenge of microprocessors" Ed. M. Hartley and A. Buckley, Manchester University Press, 1979