

tical and experimental methods. But to teach and to learn the experimental method is difficult. It may be more difficult than the theoretical, that is well formalized, complete etc. In consequence, care must be taken that the experimental method is not underweighted in physics education. We think we have paid attention to this in our programme. Leading the student step by step from the very beginning, with simple experiments in the first year to real research in the last year, we believe efficiently overcomes the difficulties. We think, laboratory work is necessary in every phase of education. That is why we have a sequence of laboratory courses, growing with time in level and aims.

It is very important also, that our students know how physics acts in modern industry, how a new technology arises as a teamwork of scientists and engineers from several different fields. It is necessary to prepare the student for this teamwork. In some lectures, summed up above as applied physics, we try to give the stu-

dent some idea of physics in modern industry. We have here about 96 hours measuring techniques (3rd. until 5th. semester), 64 hours electronics (3rd. semester), 32 hours solid state electronics (8th. semester) and 32 hours technology for physicists (8th. semester) with the double aim, to give the students new knowledge for their practical experimental work, but also to show them how physics is used in various fields of technology. Especially in the lectures on technology for physicists, it is shown, using one suitable example (e.g. integrated circuits), how physics is used in the development of a new product, in the different production steps, in production control, and in the application of the product. It is shown how physics, chemistry, mathematics, material science, and several fields of engineering have to act together, to arrive at a stable and efficient production of a modern product, and it is also shown that physics and economics in the production process are not independent of one another.

Very important in this connection is

also the "Industrie-praktikum" at the end of the third year. The student is working for three months in an industrial plant in the place of a physicist. He has to solve a (smaller) problem, as it would be solved by a physicist in the industry. He has to cooperate with other scientists, engineers and technicians, as is typical of teamwork in industry. We think that this work is very important for establishing the right understanding of the part physics plays in industry and for developing an additional motivation for the last two years, especially for the diploma work.

References

1. CASIMIR, H.B.G., *Physics Education* 11 (1976) 11.
2. LECHNER H., *Wissenschaftliche Konferenz der Forschungsgemeinschaft der Methodikbereiche Biologie, Chemie und Physik.* (Humboldt-Universität. Berlin am 17. und 18. März 1976) To be published
3. *Statistisches Jahrbuch der DDR 1975* (Staatsverlag der DDR, Berlin) 1975, p. 344.

Microprocessor Applications in Physics

T. F. Buckley, E.R.S. Candlin, D.J. Candlin and A.Q. Jones, Edinburgh Univ.
(Computer Science Department) (Physics Department)

Why are so many conferences and articles in journals devoted to microprocessors these days? Do they represent some new breakthrough in the computer field? Are they of any use in experimental physics? What is a microprocessor anyway?

We should like to suggest answers to some of these questions, and to show that microprocessors do indeed have a significant role to play, both in the control of equipment and as components in very fast scientific computers. We are currently building a fast "number-cruncher" ourselves, and we present this as an example, to illustrate some of the points which have to be considered when designing a microprocessor system.

How microprocessors work

Microprocessors are very large scale integrated circuits which, on a single chip, have an arithmetic and logic unit, a set of registers, and a control section, linked internally by networks of greater or lesser complexity (Fig. 1). Each microprocessor can perform any of a set of basic operations (e.g. transfer, add) taking

its operands from internal registers or from its input. Microprocessor designs differ widely in speed, word length, operation set and price, but all execute a series of microinstructions in a microroutine in response to a user instruction. We can distinguish two general classes of microprocessor. The first carries a set of microinstructions in a Read-Only control memory, integral with the chip. The manufacturer fixes the available set of (user) instructions when he designs the control microprogram. The second type requires an external microprogram memory. This is provided by the user who can, therefore, define his own set of instructions in terms of microroutines. The programmability of this second type makes them particularly powerful and flexible. Those interested in learning about microprocessors may find Refs. 1-6 helpful.

Useful tool for physicists

We can distinguish two rather different ways in which physicists employ computers. On the one hand, there is large-scale scientific data processing carried out by, for exam-

ple, high energy physicists, meteorologists and crystallographers. These people usually make use of large main frame computers. They are generally competent high level language programmers, but are probably not interested in the lower-level hardware of the machine they are using nor in the details of its operating system. On the other hand, there are physicists who require real-time control of experimental apparatus. For this purpose they use minicomputers, normally dedicated to the specific task.

If we now look at the current commercial use of microprocessors, we can again see two rather different types of application. In the first case, there is the "petrol pump" application, i.e. using the microprocessor as a dedicated controller for a piece of apparatus. In the second case, microprocessors may form the basis of a more complicated piece of computing machinery, for example, the central processing unit of a minicomputer.

These uses are clearly parallel to the requirements of the two broad categories of physics use, and physicists may sometimes be able to buy what

they want off the shelf. However, it is not economic for manufacturers to produce short runs of very specialized equipment, and it will sometimes happen that a physicist will not be able to buy a ready made system that exactly fits his requirements. In this case, he has the choice of either managing as well as he can with what is available, or of building his own system. This has hitherto been possible only for very simple control systems, but the advent of large-scale integration has changed the picture. Microprocessors have now become sufficiently versatile, fast and cheap to be used as general purpose electronic components in the building of complex special purpose processors. Other LSI chips, in particular, memory chips, have also fallen in price. At the same time, their functions and operating characteristics have become more standardized. This has opened the way for the non-professional to build up, from components, a processor that carries out the functions that he wants for his particular application, and to provide himself with a purpose-built system that would not be economic to produce commercially.

If a team of physicists does decide to build its own system, what conditions are necessary for success? In the first place, a certain amount of expertise in logic design and familiarity with hardware is required and secondly, a competence in low-level programming. It is ideal if the team itself can provide these capabilities, but it will probably be more usual for the physicists to hand over these aspects of the work to specialists. In this case, it is essential that both sides should speak the same language. There is a real danger that the physicists will specify their requirements so loosely that the implementer cannot see which functions are essential, and which of lesser importance. Furthermore, the physicists will not be able to participate in design decisions which may have important effects on the final product.

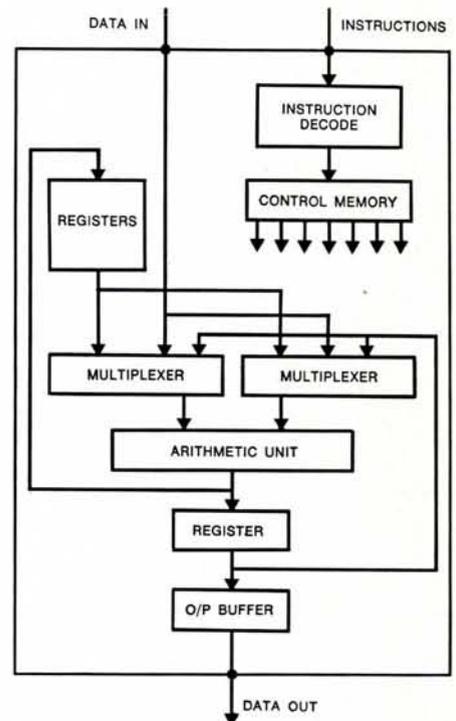
Given a suitable design team, there must obviously be technical facilities for hardware testing and debugging. It is possible to do this on the microprocessor itself by buying a "prototype kit" consisting of microprocessor, read/write control memory, teletype and interface, but this is not the most convenient way of operating; it is much more satisfactory to link the microprocessor to a minicomputer which holds the programs needed for development. Most physics departments will probably have a minicomputer with spare capacity for deve-

Fig. 1. Diagram of a typical (but hypothetical) microprocessor. Not shown are the individual control lines originating in the control memory.

lopment work. Software will be required, so that the system can be programmed. In some cases, software can be bought from the manufacturer of the microprocessor. In other cases, this software will not be available, or will be unsuitable, and will have to be developed by the applications group itself. There is always a tendency to underestimate the amount of effort that has to be put into system software development; in practice, it will probably require about the same amount of effort as the hardware.

We now consider some of the factors that might influence a physicist in deciding whether he should go ahead and build his own system, or whether he should make use of a commercially available minicomputer. If the physicist is looking for a dedicated equipment controller, it is clear that minicomputers are capable of doing the job. The only criticism might be that they are unnecessarily expensive and powerful. As far as the hardware alone is concerned, this is true, although there are some remarkably cheap, basic minicomputers on the market. For example, some manufacturers, e.g. DEC and Computer Automation, offer minicomputers consisting of a processor, a small memory of up to 4k words and simple I/O facilities on one or two printed circuit boards. When we come to consider software, however, the true costs are rather difficult to determine, as they depend on circumstances. The software for a minicomputer might be available, if the laboratory already possesses a machine from the same manufacturer, and in this case there is a powerful incentive to stick to what is already familiar. On the other hand, some microprocessor manufacturers have realized that software is a problem, and are beginning to produce systems that emulate popular machines like the PDP8 and PDP11, for which extensive software is available. The decision whether to go for the simpler, though apparently more expensive, solution of buying a minicomputer, or of building one's own microprocessor system, may depend in the end on the relative availability in a given laboratory of money for capital expenditure, or of manpower for development work.

The same considerations apply if the application involves substantial numerical computation. Many mini-



computers are now very powerful and have excellent processing capabilities (see for example, the review article by J. Wooton⁷). On the other hand, there are a certain number of applications that can still only be handled by main frame computers, which of course are too expensive for the exclusive use of a small group of physicists. We feel that it is in this field that do-it-yourself microprocessor systems may provide a relatively cheap solution to the problem.

EMMA - A Case History

EMMA (Edinburgh-Multi-Microprocessor-Assembly) is a microprocessor-based system being developed in Edinburgh by a small team from the Computer Science and Physics Departments. The objective is to provide very high speed number-crunching at low cost. EMMA runs under the control of a host minicomputer, which handles all input-output and peripherals. From the host's point of view, EMMA behaves like another peripheral; a block of data for processing is sent to EMMA, and when processing is complete, a block of data is sent back again. EMMA is thus confined to a data-processing function; it has no operating-system overheads. In this way, we have been able to keep to a fairly simple design, which still has the performance one might expect from a large main frame computer.

The impetus for the development of EMMA came from the very large scale data processing requirements of modern high-energy physics experiments. In multi-particle spectrometers, up to 10^8 events can be accumulated

in a year, and a large fraction of these events have to be passed through programs for spark-reconstruction, track recognition, momentum determination and kinematical fitting. It is not easy to reduce the processing time per event much below 100 ms on an IBM 360/195, which means a total requirement of 1000 hours CPU time, or more, per year.

We decided instead to build a system of parallel processors, based on microprocessor chips, with the versatility of a general purpose scientific instruction set. The following points all seemed important to us :

1. The system should be simple to construct and maintain. We are only a small team, and heavily engaged with teaching duties.
2. The design should be modular, so that we should have something useful with the minimum configuration, but could extend it, as money became available.
3. We should build a machine that would be generally useful in a physics department, but at the same time we should take into account some of the special characteristics of high-energy physics data processing.
4. From the user's point of view, programming should be similar to programming a standard scientific computer.

We shall start with a description of the EMMA memory design, as this is perhaps the most unusual part of the system. Separate stores are used for program, microprogram and data. All these stores are similar and consist of Schottky TTL circuits (cycle time : 60 ns) mounted on boards to give 1k x 16 bits of memory. These boards contain all the necessary buffering and may be used interchangeably for all storage purposes. The instructions and integer data are 16 bits wide, and microinstructions and floating-point data 32 bits wide. As we are using read/write store for all memories, the program, and more unusually, the microprogram, can be easily changed.

The application program, resident in the program store, drives two independent processors each with its own private data store, and with access to a communal constant data store. This design is a direct result of the type of processing required in high energy physics. In our particular application, each processor works on an event, characterized by initial measurements of spark positions. As these events are statistically independent, there is no interaction between processors. Each processor accepts data from the host minicomputer, processes its event, and passes results back to the

Section	Boards	Approx Cost
Integer processor	3 large, 1 small	£ 500
Control, I/O, memory buffering	2 large, 3 small	£ 400
Memory (16 bit word)	1 small	£ 250
Cabinet, racks, power supplies	—	£ 1000

host when processing is complete. The constant store is accessible to both processors. It contains application dependent constants such as chamber constants and is used in read-only mode.

Each processor is constructed around four 4-bit slice Schottky microprocessor chips (AM2901), with a repertoire of 488 functions and a worst-case cycle time of 100 ns. Since some additional logic is needed in the execution part of the processor, the overall cycle time is 150 ns, rather more than twice the overall memory access time. This allows two processors to access the same program and microprogram stores independently, 180° out of phase. The result is an overall doubling of computing speed, without a commensurate increase in cost. The program is expected to be large, and the expense of second program and microprogram stores is saved, as is separate host I/O control. Integer arithmetic can be carried out by the processor itself, but for the more complicated floating-point operations, a special unit is shared between processors. We may add other shared functional units (for example mathematical functions) later. There is never any scheduling problem with sharing resources in this way, as the processors are out of phase. For each processor, instruction fetch, microinstruction fetch and execution cycles are concurrent, forming a three-stage pipeline, and minimizing the total time per instruction.

In general, each instruction in the program store initiates the execution of a sequence of several microinstructions, so that the processor does not access the program store again until the end of the microroutine. We could obtain a further increase in processing speed by adding a second (lower priority) processor for each of the primary processors, which could access the program store during the spare cycles. Most of the cost of EMMA lies in the program store, so that we should expect to obtain a 60-70 % increase in processing power for a 30 % increase in cost.

EMMA is constructed on two sizes of wire-wrap boards conforming to DEC standards. The smaller has space for 32 16-pin and five 40-pin I.C. sockets, and a 72-pin edge connector. The larger board is equivalent to two

small boards. We plan to use printed circuit boards for the standard memory. The table gives the size and approximate cost of different sections.

An integer machine with two processors and 40k word of store will thus easily fit into five 19 in. racks, with space for enhancements, and cost about £ 12 000. We have designed such a machine down to chip layout, and aim to complete it by the autumn. The design figure of 150 ns for the processor cycle is based on worst case chip times, with some allowance for propagation. If this is achieved, some typical instruction times are :

ADD register to register	150 ns
ADD register to store	300 ns
ADD store to store	600 ns
Branch on condition	300 ns
Subroutine jump	750 ns

These times should be halved to get an effective figure for the two processor system, which thus has a speed comparable with that of a powerful main frame computer.

While building the integer machine, we are also developing the hardware multiplier and the floating point unit. The multiplier is on a CERN-designed board, costs about £ 300 and allows a 16 x 16 bit multiplication in 750 ns, most of which is in operand fetch and store.

Conclusion

We feel from our experience that microprocessors can find many applications in physics and, in particular, can be used to produce relatively low cost special purpose processors. However, anyone embarking on such a project should realise that it can involve considerable effort to produce the hardware and software systems.

References

1. KENNY, R., Microprogramming simplifies control system design *Computer Design*, Feb. 1975
2. NARTINEZ, R., A look at trends in microprocessing/microcomputer software systems *Computer Design*, June 1975
3. KHANNA, V. and DALY, T., Making the most of your micro *Digital Design*, July 1975
4. LANE, A., Microprocessor system design *Digital Design*, August 1975
5. LOGAN, J. and KREAGER, P., Using a microprocessor: A real-life application *Computer Design*, Sept. 1975
6. *Computer*, October 1975 : This issue is devoted to microprocessor design and application.
7. WOOTON, J., Small machines to solve large problems *Computer Weekly*, 8th Jan. 1976