

XOP: A SECOND GENERATION FAST PROCESSOR FOR
ON-LINE USE IN HIGH ENERGY PHYSICS EXPERIMENTS

T. Lingjaerde, CERN, Geneva, Switzerland

ABSTRACT

Processors for trigger calculations and data compression in high energy physics are characterized by a high data input capability combined with fast execution of relatively simple routines. In order to achieve the required performance it is advantageous to replace the classical computer instruction-set by microcoded instructions, the various fields of which control the internal subunits in parallel.

The fast processor called ESOP is based on such a principle: the different operations are handled step by step by dedicated optimized modules under control of a central instruction unit. Thus, the arithmetic operations, address calculations, conditional checking, loop counts and next instruction evaluation all overlap in time.

Based upon the experience from ESOP the architecture of a new processor "XOP" is beginning to take shape which will be faster and easier to use. In this context the most important innovations are: easy handling of operands in the arithmetic unit by means of three data buses and large data files, a powerful data addressing unit for easy handling of vectors, as well as single operands, and a very flexible logic for conditional branching. Input/Output will be made transparent through the introduction of internal fast processors which will be used in conjunction with powerful firmware as a software debugging aid.

1. Introduction

Fast processors have already been used for data compression and trigger calculations in high energy physics for some time.

Many different approaches have been used in order to achieve particular aims: high speed, high flexibility, compatibility with existing systems or easy programming. The number of publications is already quite high; an exhaustive list is found in ref. 1.

For a given choice of hardware parameters, high throughput of a processor system can be achieved by means of parallel execution of parts of the task. If a problem can be split into a number of separate tasks, they can be executed in autonomous but identical processors. In the cases where this cannot be done, concurrency of operations has to be achieved on a low level: a set of elementary operations, contained in one micro-instruction, is handled simultaneously in different subunits tailored for speed and ease of operation. The price to be paid for the resulting high efficiency is more effort from the programmer: in order to exploit the potential power of a microprogrammed machine the user has to know its architecture well. The task can be compared with that of playing the organ: certain things have to be set up beforehand and during the execution both hands and feet must work together in perfect harmony. But the result can be most rewarding!

In the following chapters a new microprogrammable processor called XOP (Expandable Online Processor) will be described which is based upon the principle outlined above.

2. The Origin of XOP: the ESOP microprogrammed processor

ESOP processors²⁾ are used in three experiments at CERN^{3,4,5)} and are being installed in an additional two.

The prototype of ESOP was developed in 1973 to do histogramming in real time when following tracks with the ERASME bubble chamber picture measuring machine. The host computer was a PDP-11/20 and the processor was interfaced directly to the UNIBUS.

When ESOP was chosen to be used with the NA11⁴⁾ and R807⁵⁾ experiments, the original machine underwent a whole series of modifications and additions: larger memories, control through CAMAC and special interfaces.

Last year it was decided to do no more additions or modifications to ESOP as the limits of the original framework had been reached. Instead it was decided to start the design of a new machine, using the latest technology combined with the experience from the manufacturing, testing, installation, running and maintenance of the ESOPs.

3. The Main Elements of XOP

Three main ingredients are essential for fast real time processing:

- i) The ability to collect data from external sources at a high rate, whenever possible combined with some simple sorting or preprocessing.
- ii) A powerful arithmetic logic unit.
- iii) An addressing scheme which allows handling of arrays as well as single variables in a simple, flexible and efficient manner.

Fig. 1 shows a block diagram of the new fast processor, XOP, which is being designed at present. The three units which are meant to fulfill the requirements above can be recognized. A short description of each will follow later.

In order to run the program, an instruction unit is needed, the task of which is to evaluate the next instruction address, taking into consideration possible branch conditions, and to save return points when requested.

In common with most processors of this kind, XOP is not a stand-alone device. Programs are loaded from a host computer via a CAMAC interface through which the results are also read back. In order to keep the interface and the device handling software simple, these tasks are executed in an Auxiliary Processor which uses a program stored in PROM. In particular it is able to recognize CAMAC commands and initiate the relevant internal routines to handle them.

4. How Concurrency of Operations is achieved in XOP

The basic principle of the ESOP architecture has been maintained: each instruction is composed of a set of subfields which contain parameters and operation codes for the execution in the various subunits in that cycle. As usual in fast processors, the instructions are contained in a dedicated memory rather than in common with the data.

But the XOP processor goes further than that: data addresses and return instruction addresses are stored in dedicated files. Auxiliary control functions such as loop-counters, are handled independently of the arithmetic unit and an indexing scheme has been tailored to make operations on arrays straightforward. Combined with the highly flexible conditional multibranch system, "DO-loops" comprising arithmetic operations and "IF" statements can be executed in a single cycle. When used for trigger applications it is essential that data from external sources, such as wire chamber digitizers, can be transferred fast. In order to achieve the necessary high data input rate, the data memory has been divided into blocks of 4K x 16 bits, each with its own independent DMA channel. A variety of hardware interface protocols can be handled by the local processor which will run at a cycle time less than 100 nanoseconds. The most important of these will be the one connecting XOP to the FASTBUS system. In addition, the data can be checked as it is coming in and addresses of key words stored in a local address file for later use by the CPU.

5. Comments on the XOP Block Diagram

A detailed explanation of the XOP hardware is beyond the framework of this paper. Therefore, only a summary of the various units found in Fig. 1 is given here. It should be emphasized at this point, however, that the specifications have to be regarded as preliminary.

5.1 The instruction unit

The most severe problem encountered with ESOP was how to meet requirements for additions or modifications which needed to be controlled by bits in the (fixed) 48 bit instruction word. The resulting overlap of instruction fields made programming cumbersome and slowed down the execution speed. In order to solve this problem once and for all, XOP will use a totally new scheme for the instructions: rather than centralizing the instruction memory (IM) in one separate unit from which an instruction bus is distributed, each unit will have its own "horizontal" fraction of the IM which contains those bits which are used inside it. The instruction ADDRESS is distributed to all units: this bus is only 12 bits wide and can be easily handled. The result is a system which can be expanded without having to do any modifications to the remaining hardware at all. A further advantage is that the IM memory chips can be located close to the circuits where the content is used, thus cutting down on long, wide signal paths.

The logic system for conditional multi-branching used in ESOP has turned out to be very effective. XOP will use an extended version of this in order to accommodate expansions.

5.2 The data memory address control

Two files have been foreseen: one to hold a "base" value and the other an index. The two can be added together to form the effective address. Further the index can be incremented to operate on arrays. Addresses can be saved in consecutive locations of the "base" file.

5.3 The local controllers

There is one local controller per memory module. These serve two purposes:

- i) to act as the interface between an external source such as an RMH wire chamber digitizer or a FASTBUS system.
- ii) Whenever desirable and possible, to do pre-processing on the data as they are read in.

5.4 The arithmetic logic unit

A data file has been foreseen to hold frequently used variables or constants. Its content can either be handled as a single value, where the address is contained in the instruction itself (immediate), or as a block of sequential data. In the latter case the file address is incremented each time a new word is fetched. Variables which change less frequently can be accessed directly from data memory. The layout of the various data buses makes it possible to execute arithmetic operations according to the scheme:

SOURCE1 (OPERATION) SOURCE2--→ DEST.

ALU-conditions are connected to the instruction control to allow for multiple branching.

5.5. Loop counters

The initial values as well as the running counts are kept in a file. The content of a counter can be incremented and checked for overflow in a dedicated arithmetic unit.

5.6 The STACK

Instruction addresses can be saved on, and retrieved from, the STACK file.

5.7 Breakpoints and debugging aid

Rather than working on the instruction addresses, the breakpoint logic will use three dedicated bits in the instruction. Thus there will be no limit on the number of breakpoints and the combination of the bits will specify the action to be taken, such as to halt. The Auxiliary Processor will handle the display of the processor status upon request from the breakpoint logic.

6. Hardware Considerations

As can be deduced from the block diagram and the descriptions above, most of the Integrated circuits will in fact be memory chips. Around 200 4K x 1 bit chips will be needed for the Instruction Memory alone, a large fraction of which will hold immediate addresses and parameters. In addition there is the Data Memory and the different files. Static MOS memories have been regarded as the most suitable for these applications. As

a consequence all the control circuitry must be TTL compatible. It is, however, feasible that some "isolated" units will be done in fast ECL.

The boards will be at least twice the size of a CAMAC unit and will use 96 pin backplane connectors.

The first units will be made in "multi-wire".

7. The Time Schedule

At present only detailed block diagrams have been made. It should be kept in mind, however, that a substantial part of the logic will be adapted from ESOP. Before the final layout is decided upon, a simulation study will be undertaken using the "ISPS" program.^{6,7,8)} The prototype of XOP is planned to be ready in 1982. Based upon previous experience it will then take another year to integrate the processor into an experiment and develop the necessary software and documentation in order to ensure trouble-free operation.

Acknowledgements

The ideas presented in this paper reflect the collaboration with A. Fucci, D. Jacobs and C. Ljuslin, all CERN Data Handling Division.

Valuable comments on the ESOP Architecture and a number of suggestions for improvements were obtained from D. Giddings and G. Lutjens of the NA11 experiment and B. Heck, J. Hooper and S. Nielsen from the R807 experiment.

References

- 1) C. Verkerk, CERN Data Handling Division, DD/81/03, February 1981.
- 2) T. Lingjærde, CERN Data Handling Division, DD/75/17, May 1975.
- 3) D. Jacobs, CERN Data Handling Division, DD/80/22 October 1980.
- 4) C. Damerell et al., Proc. of the Europhysics Conference on Computing in High Energy Physics, Bologna, Italy September 9-12, 1980.
- 5) B. Heck et al., Proc. of the Europhysics Conference on Computing in High Energy Physics, Bologna, Italy September 9-12, 1980.
- 6) M. Barbacci et al., "The ISPS Computer Description Language", Technical Report, Dept. of Computer Science, Carnegie-Mellon University, 1977.
- 7) M. Barbacci et al., "An ISPS Simulator", Technical Report, Dept. of Computer Science, Carnegie-Mellon University, 1978.
- 8) M. Barbacci, "Instruction Set Processor Specifications (ISPS): The Notation and its Applications", IEEE-CS Transactions on Computers, Vol. C-30, No. 1, January 1981.

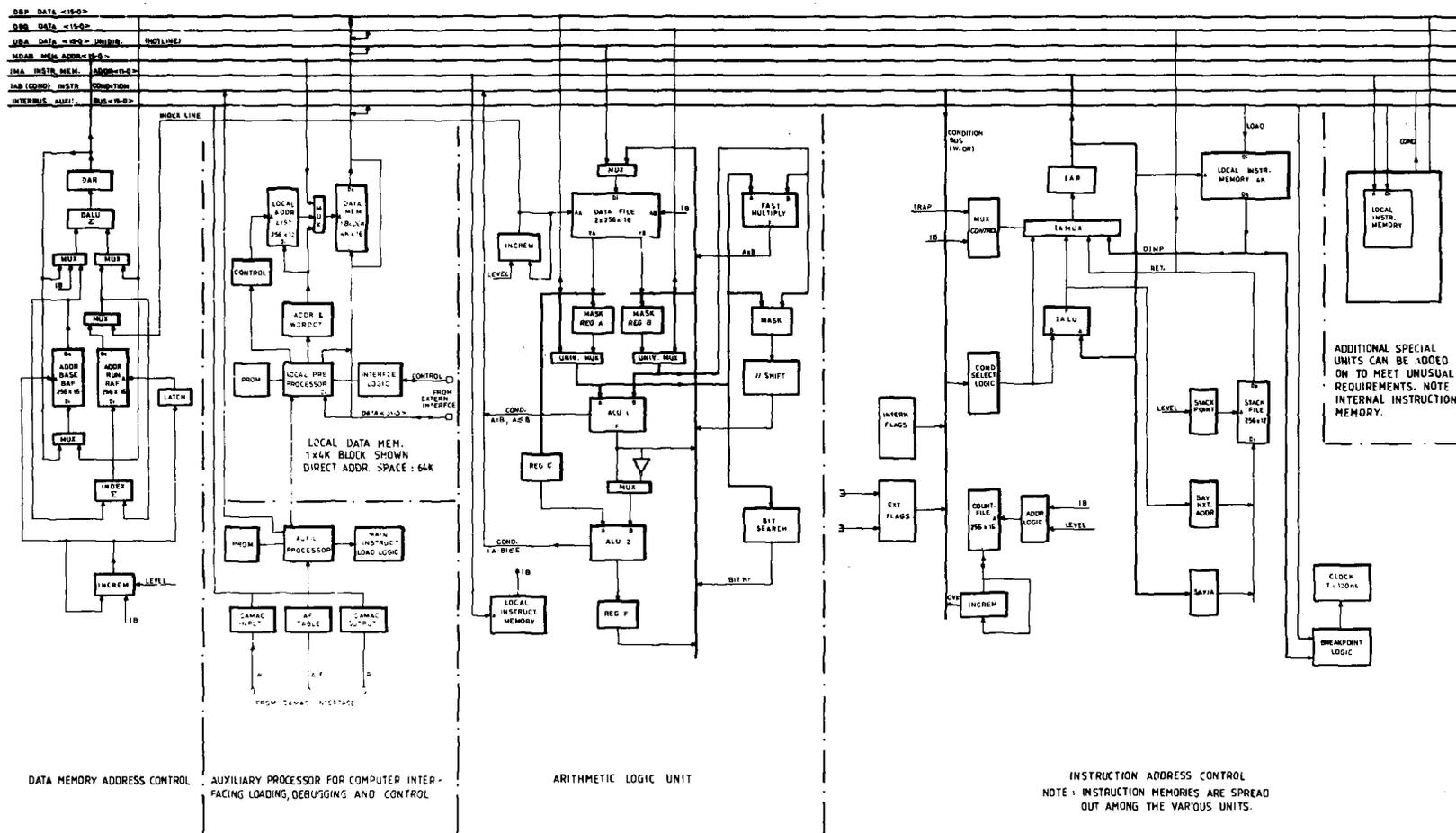


Fig. 1: XOP Block Diagram