CERN-Data Handling Division
DD/85/15
July 1985

XOP, a fast versatile processor, as a building block for parallel processing in high energy physics experiments

P. Bähler, N. Bosco, T. Lingjaerde, C. Ljuslin,
A. Van Praag and P. Werner

(Presented at the Computing in High Energy Physics conference, Amsterdam
June 25-28, 1985)

XOP, a fast versatile processor, as a building block for parallel processing in high energy physics experiments.

P. Baehler, N. Bosco, T. Lingjaerde, Ch. Ljuslin
A. Van Praag and P. Werner

CERN
1211 Geneva 23, Switzerland

Abstract

The XOP processor has been designed for trigger calculation and data compression in high energy physics experiments. Therefore, emphasis has been placed upon fast execution and high input/output rate. The fast execution is achieved by a wide instruction word holding operations which are executed concurrently. Thus, the arithmetic operations, data address calculations, data accessing, condition checking, loop count checking and next instruction evaluation all overlap in time. In conventional micro-processors these operations are performed sequentially. In addition, the instruction set comprises not only the classical computer instructions, but also specialized instructions suitable for trigger calculations, such as bit search, population count, loose compare and vector instructions.

In order to achieve a high input/output rate, each XOP ECLine interface board is equipped with an input and an output port which fulfil the LeCroy ECLine specifications. The autonomous input port allows a data rate of 40 Mbytes/sec, while the program controlled output port allows 20 Mbytes/sec. For Fastbus based systems a dual Fastbus master interface is under design which allows to build up a Fastbus multi-processor system. This design is being done in collaboration with LAPP Annecy for the CERN Lep L3 experiment. Their scheme comprises 4-5 XOP processors, each of them with a master interface on a data input segment and a master interface on a data output segment.

The present paper describes the structure of the XOP processor, the interface capabilities and the software development and debugging tools.

1. The XOP System Configuration.

The XOP system configuration as shown in Fig.1. comprises a VME system which is able to control up to 8 XOP processors. The VME System with a terminal connected to it, is used for software development and debugging. During debugging, the serial link to the host computer allows running of the Cross Software (Assembler, Linker, Pusher) installed on the host computer and the downline loading of the XOP programs, while, for production, the parallel link over a CAMAC module allows the control and the downline loading of XOP programs by the data acquisition control programs. A separate control line from the CAMAC module allows the resetting of the VME System by the host computer for initialisation.

Each XOP processor consists of the following modules: the VME-XOP interface, the instruction control unit, the main arithmetic logic unit, the data address calculation unit and at least one data memory, plus the experiment specific interface modules for accessing the trigger data. For interfacing XOP to a data acquisition system there is at present a module available consisting of a buffer memory with an input and an output port which fulfil the LeCroy ECLine specifications. Currently, there is a test system installed which reads out a LeCroy FERA system with its maximum data transfer rate of 20 Mbytes/sec (the XOP interface allows 40 Mbytes/sec) and which sends the data to a CAMAC ECLine buffer memory with a rate of 20 Mbytes/sec.

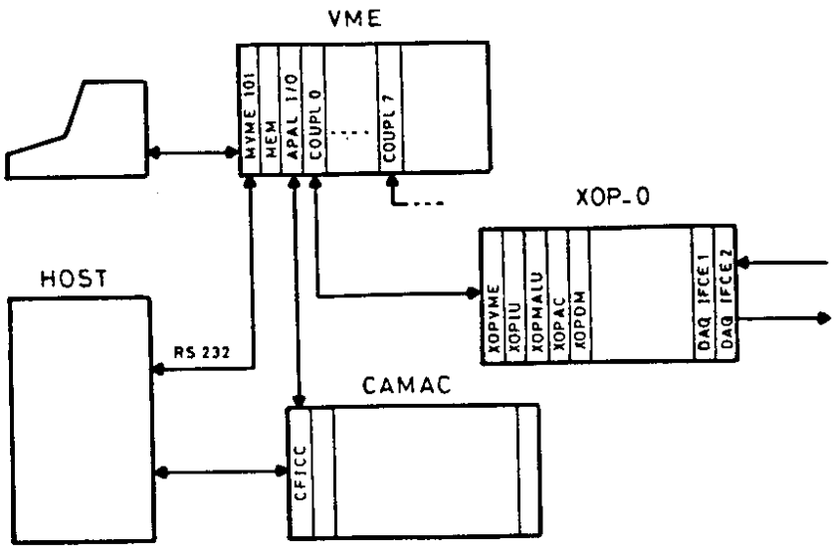

Figure 1: XOP System configuration.

For Fastbus systems a dual Fastbus master is under design. The first prototype is expected to be ready by the end of this year. All Fastbus operations will be implemented, including the pipelined block transfer. The internal structure of XOP and the design of the Fastbus master interface will be capable of coping with a data transfer rate of 40 Mbytes/sec.

2. The internal structure of the XOP processor.

XOP is an extremely fast processor optimized for 16 bit integer calculations. This is achieved by using a wide horizontal instruction word allowing concurrent operation of the different execution units. The interconnection of these units is shown in Fig.2. The backplane of the XOP processor consists of 5 busses (three data busses, one data address and one instruction address bus) and some control lines for signals like the clocks, condition bits, the hold signal and the block signal.

In XOP the instruction memory (IM) and the data memory are separated. The data address bus allows to address a range of 16 Mwords of 16 bits, whereas the instruction address bus allows a range of 64 Kwords of 256 bits. Currently the size of the instruction word is only 160 bits. The dual Fastbus master will expand it up to 192 bits. In XOP the IM is distributed. The instructions are not stored centrally in one instruction memory unit from which an instruction bus is distributed to the different execution units. Instead, each unit has its own fraction of the IM which contains the necessary information to perform its task. This scheme not only reduces the bus to a 16 bit address bus which is distributed to all the units, it also makes it easy to incorporate new execution units which are working in parallel with the already existing ones.

The XOP processor is programmable on two levels, the micro and the nano level. The micro level generates instructions for the different execution units every 100 nsec. If a unit is not ready within the 100 nsec, the micro level cycle is expanded by a multiple of 50 nsec until all the execution units are ready for new instructions. This feature is used for instance by the Fastbus master interface, in order to hold the XOP processor until the requested data from Fastbus is available. The main arithmetic logic unit is using this mechanism to hold the processor during the execution of special instructions which are slower than the common instructions, i.e. the bit search, the loose compare, the divide and of

course the block and vector instructions. All other execution units are always ready after 100 nsec. In fact, the data address calculation unit is capable of generating a new data address every 50 nsec and the data memories are able to store or retrieve data within 50 nsec. Therefore, for these units the micro level allows to specify two instructions per micro cycle. This allows writing the result of the current instruction to the memory and accessing an operand from the memory for the next micro cycle in one cycle. Block and vector instructions take full advantage of this high execution speed. Together with the pipeline facilities it is possible to perform for example an arithmetic operation on a vector element from the data memory and a vector element from the data register file and to store the result in a vector in the register file every 50 nsec.

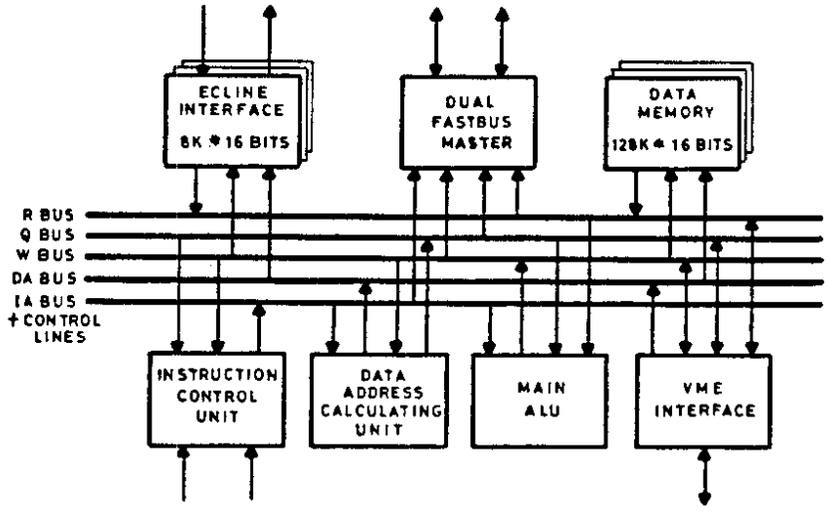

Figure 2: XOP processor, its modules and busses.

The main arithmetic logic unit (MALU) of the XOP processor is a powerful programmable processor by itself, with a cycle time of 50 nsec. The code written for this machine is called the nano code. Each MALU instruction available on the micro level is pointing to a nano routine. The routines for the common instructions are only two nano instructions long. This explains the 100 nsec cycle time of the micro level. Programming on the nano level is only necessary if new special MALU instructions have to be implemented in order to speed up a program.

3. Software development and debugging tools.

For software development Cross software is available on IBM and as well on VAX computers. The Macro Cross Assembler for the XOP processor is based on the Cern Macro Cross Assembler for the M68000 processor. It uses, with some exceptions, the same pseudo operation codes and produces also an object file in CUFOM format (Cern Universal Format for Object Modules). In using the CUFOM format it is possible to make use of the already developed Cross Software, such as the Linking−Editor and the Pusher. This generates a file in the Motorola S−Format which can be downline loaded into the target machine.

The EPROM resident monitor on the VME system is an extension of the CERN supported monitor MoniCa. The monitor incorporates additional commands for the XOP processors, which are

similar to those for the M68000 processor. Only to mention some of them, there are commands for investigating and modifying memory locations, data and address register files, commands for starting and stopping programs, for single stepping and for setting breakpoints. In addition, there is a command handler implemented running in the background of the monitor which accepts messages from a host computer over a CAMAC link.

4. Status and Performance.

At present, three XOP processors are installed and running, two further ones are under production. One of the three processors is used as a test bench for testing XOP processor modules, whereas the other two processors have been delivered to the Lep L3 experiment. There, they are used for software development, for testing the first level trigger and for the development and debugging of the dual Fastbus master. In the end, they will be used together with 2−3 other XOP processors to provide the second level trigger of the Lep L3 experiment.

The Cern collider experiment UA2 has been investigating fast processors for their second level trigger. For bench−marking they used a cluster finding routine with real event data from their previous run. The tests showed that this routine was executed about 10 times faster on the XOP processor than on the 3081E emulator.

In the domain of trigger calculations and data compression where fast integer arithmetic is sufficient, there is hardly a better candidate than the XOP processor − which, afterall, has been especially designed for this particular field of computation.