ISTITUTO NAZIONALE DI FISICA NUCLEARE

Sezione di Trieste

INFN/AE-90/10

17 Settembre 1990

P. Battaiotto, A. Colavita, F. Fratnik and L. Lanceri

A FASTBUS MODULE FOR TRIGGER APPLICATIONS BASED ON A DIGITAL SIGNAL PROCESSOR AND ON PROGRAMMABLE GATE ARRAYS

Servizio Documentazione
dei Laboratori Nazionali di Frascati

A FASTBUS MODULE FOR TRIGGER APPLICATIONS BASED ON A DIGITAL SIGNAL PROCESSOR AND ON PROGRAMMABLE GATE ARRAYS

P. Battaiotto
INFN-ICTP Microprocessor Laboratory, Trieste,Italy
Universidad Nacional de La Plata, Argentina

A. Colavita
INFN-ICTP Microprocessor Laboratory, Trieste,Italy

F. Fratnik
INFN-ICTP Microprocessor Laboratory, Trieste,Italy
INFN, Sezione di Trieste, Trieste, Italy

L. Lanceri
INFN-ICTP Microprocessor Laboratory, Trieste,Italy
Istituto di Fisica, Università di Udine, Udine, Italy

ABSTRACT

The new generation of DSP microprocessors based on RISC and Harvard-like architectures can conveniently take the place of specially built processors in fast trigger circuits for High Energy Physics experiments. Presently available Programmable Gate Arrays are well matched to them in speed and contribute to simplify the design of trigger circuits.

Using these components, we designed and constructed a Fastbus module. We describe an application for the total energy trigger of DELPHI, performing the read-out of digitized calorimeter trigger data and some simple computations in less than 3 ms.

(Submitted to Nuclear Instruments and Methods)

1. INTRODUCTION

Until recently, general purpose microprocessors could not be used as basic computing units in fast trigger circuits for High Energy Physics (HEP) experiments, whenever the computing time per event was limited to not more than a few microseconds. In time-critical trigger applications, even with increased clock frequencies, Complex Instruction Set Computers (CISC) were not fast enough.

This led people in the HEP community to design and construct their own programmable processors, based on more sophisticated architectures. Including interfaces to detector electronics, buffer memories and other special units, a typical processor [1] filled one CAMAC crate.

The cost of such processors and the effort required in their design, programming, and implementation in specific environments, were still rather heavy. A better performance can now be achieved tailoring the trigger processors around presently available Reduced Instruction Set (RISC) microprocessors, like the Motorola DSP56001 [2]. The architecture of these DSPs allows operations on data, addresses and instructions to be carried out in parallel and pipelined; a fast multiplication is available, offering an important improvement in speed and reliability. General documentation and software tools satisfy the more demanding standards of commercial products.

We chose the DSP56001 as a basis for the design of a Fastbus module, intended for trigger applications in the DELPHI Forward Electromagnetic Calorimeter (FEMC) [3]. The interfacing to Fastbus and to the detector front-end electronics was made more compact and reliable by an extensive use of XILINX XC3020 Programmable Gate Arrays [4]. All the circuits could fit in less than one half Fastbus board, at a cost roughly an order of magnitude lower than that of an equivalent processor of the previous generation.

Two such modules are presently installed in the trigger system [5] of the DELPHI FEMC. In this paper we briefly describe the hardware and software features of our module, and give some results on its performance.

2. HARDWARE

The design of the module was kept as simple as possible. A block diagram is shown in Fig. 1.

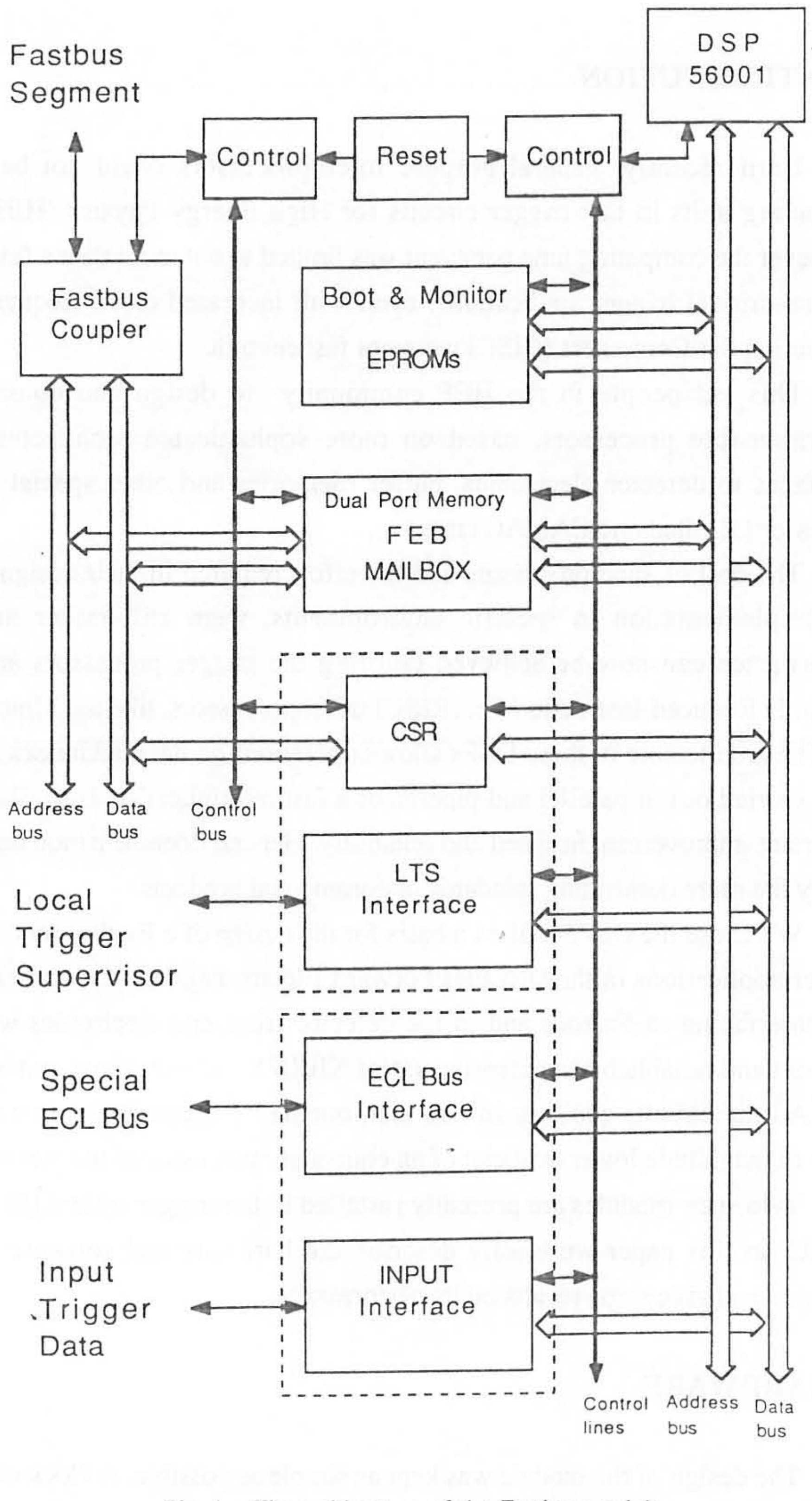

Fig. 1 - The architecture of the Fastbus module.

The DSP performs fast trigger computations, executing programs normally loaded from EPROMs.

The input trigger data are provided through an "INPUT Interface" which in our specific application reads calorimeter trigger data digitizers (FDDP, Fast Data Digitizers and Processors [6]).

Local Trigger Supervisor (LTS), through the "LTS Interface".

A special "ECL Bus Interface", following the ECLine protocol [7], makes the DSP results available to other trigger processors.

The Data Acquisition System (DAS) of our experiment requires Fastbus access to the module. This is obtained via a standard Fastbus Slave Coupler plus Control and Status Registers (CSR) and a Dual Port Memory acting both as a Front End Buffer (FEB) and as a "Mailbox" for communications between DSP and the DAS. This structure also allows downloading of DSP programs cross-assembled or compiled in an host computer.

The dashed boxes in Fig. 1 indicate the circuits that are integrated in Programmable Gate Arrays.

2.1 The DSP processor

The DSP56001 [2] is a RISC microprocessor with a Harvard type pipeline architecture [8], sketched in Fig. 2.

Data and address manipulations are performed by three units:

a) a Data ALU, with 24 bit input registers and 56 bit accumulator registers, performing logic and arithmetic operations, including single-cycle multiplication, between 56 bit operands;

b) an Address Generation Unit, offering several addressing modes for program and data memories;

c) a Program Controller, prefetching and decoding instructions, executing loops and controlling interrupts.

Three different memory address spaces are defined: two for the data memory, called X and Y, and one for the program memory, called P:

a) X and Y data memories include internal RAMs (two, 256 x 24 bits each) and ROMs (two, 256 x 24 bits each), and can be expanded externally to 64K x 24 bits each. We mapped the 31 registers of our interfaces to the external X space , and the Dual Port Memory (1024 x 16 bits) of the Fastbus Interface to the external Y space.

b) the P program memory includes a 512 x 24 bits internal RAM, and a smaller "bootstrap" ROM. An external expansion up to 64K x 24 bits is possible.

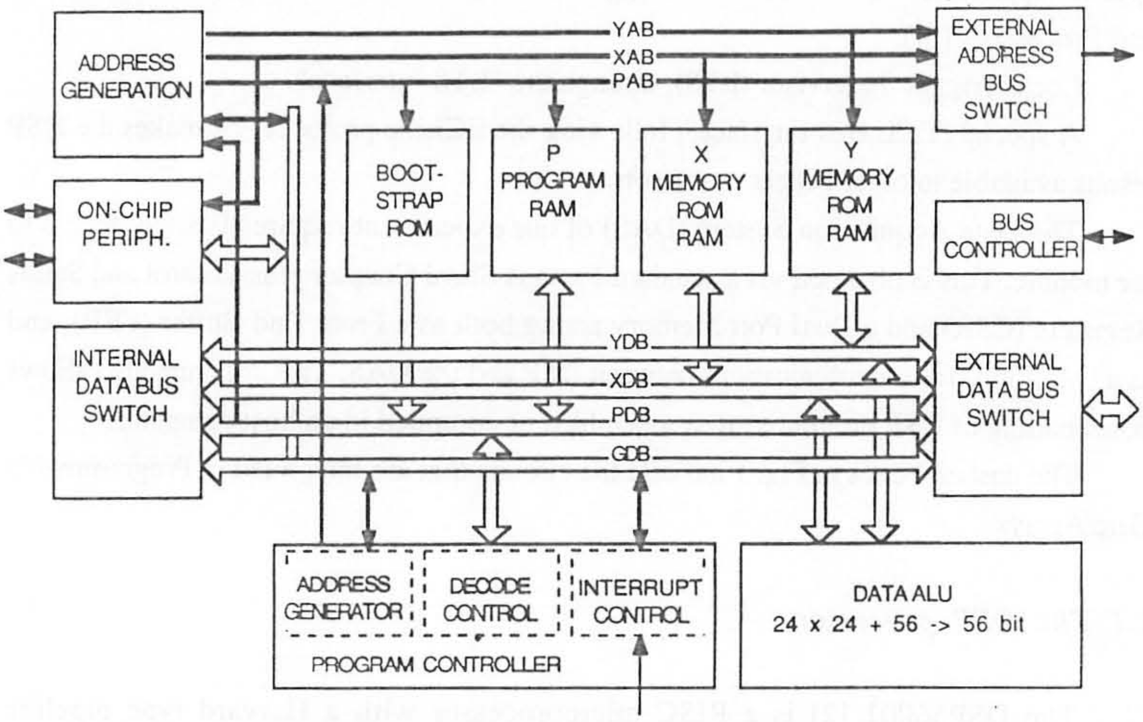

Fig. 2 - A simplified block diagram of the DSP56001.

The separate internal buses for X, Y, P data and addresses are multiplexed to one external bus for addresses and one for data. To fully exploit the potential speed of the microprocessor, external memory accesses must be minimized. We therefore bootstrap our trigger program from an EPROM into the internal P memory; we use part of the external space for an EPROM containing a debugging monitor program, whose execution is not time-critical.

2.2 The interfaces

The registers and logic circuits of the interfaces shown in Fig. 1 were implemented in three XILINX XC3020 Programmable Gate Arrays (PGA) [4]. These VLSI ICs contain 64 logic blocks and 64 input/output blocks. Each logic block includes some programmable combinatorial logic and two flip-flops. The routing of signals between blocks is also programmable. The programming is done using CAD tools and the final programs are stored in EPROMs, from which the PGAs are initialized.

The "INPUT Interface" and of the "ECL Bus Interface" fitted in two equally configured PGAs. The block diagram of one of them is shown in Fig. 3. Each PGA contains an 8 bit slice of all the registers of the two interfaces. Four input registers can retrieve data from four external data sources in parallel, and are separately addressable by the DSP. An output register for the special ECL Bus is accompanied by its handshake control circuitry.

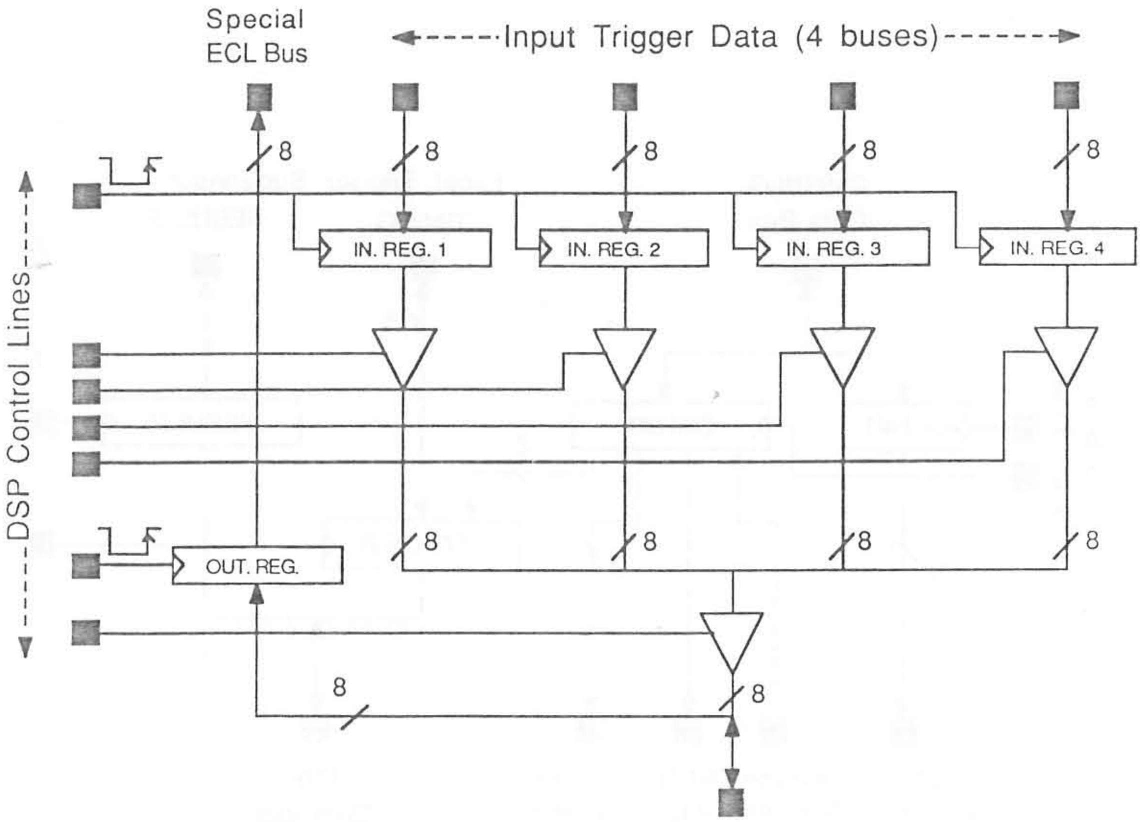

Fig. 3 - The Programmable Gate Array implementation of an 8-bit slice of the "INPUT nterface" and of the "ECL Bus Interface".

A block diagram of the third PGA is shown in Fig. 4; it includes:

a) for the "LTS Interface": a "Status Register", by which the module can be synchronized with the experiment, and a "Register of Results", used to communicate its results, compacted in a few bits, for fast trigger decisions;

b) for the Fastbus access: the "Control and Status Registers" CSR0 and CSR1. The first register is required to access the module as a Slave and perform standard actions (resetting the module, running/halting the processor, requesting its attention). Some of its non-standard bits are used to set options in the trigger program (real or internally simulated data for the trigger algorithm, different types of detector segmentation, etc.), and to select different ways of handling the Front End Buffers (FEB). The second register can be used to select one of four FEB areas in which the DSP is expected to write its results of the next processed event.

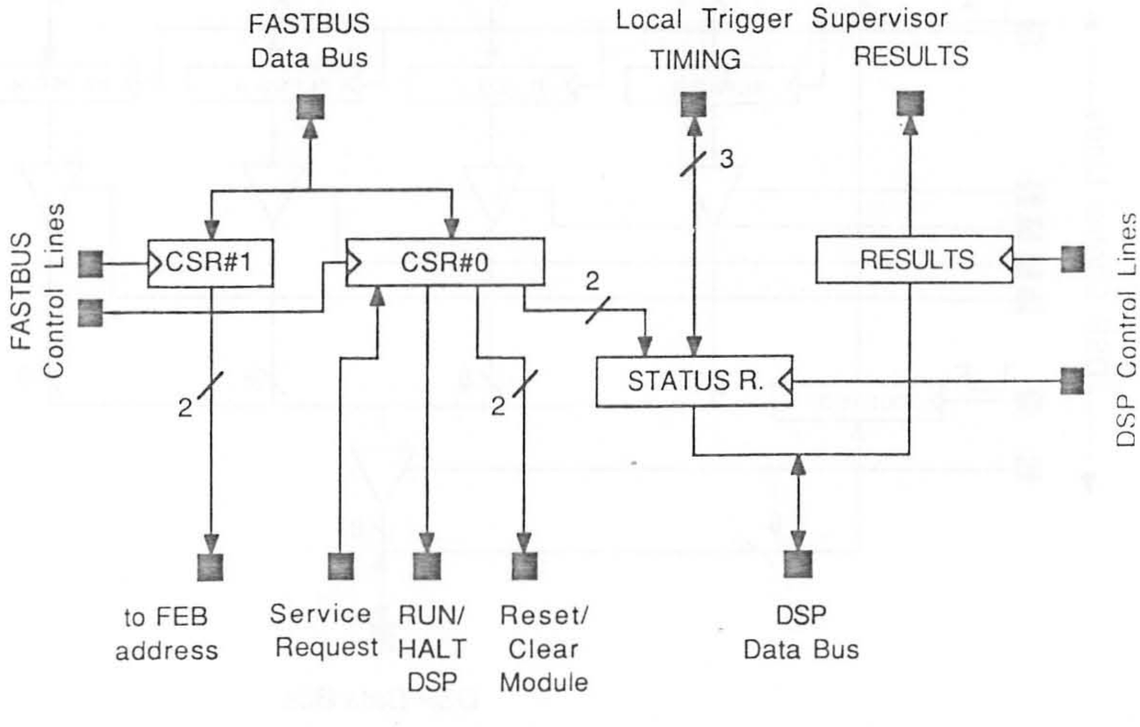

Fig. 4 - The Programmable Gate Array implementation of the "LTS Interface" and of the Fastbus CSR registers.

3. SOFTWARE

We used the DSP software developement tools, a C-language cross-compiler, a cross-assembler and a simulator from Motorola, to develop trigger algorithms and a debugging monitor. A control program for initialization and tests was also written.

3.1 A simple trigger algorithm

In the DELPHI experiment, the first application of our module is a "total energy" trigger for the end-cap electromagnetic calorimeter. At each new valid beam-crossing, the trigger program prepares some constants and then waits for the first level trigger. Digitized trigger data, corresponding to the energies deposited in 24 sectors of the detector, are then read-out, added and compared with pre-defined thresholds. Results are made available: to the Local Trigger Supervisor in the Register of Results, to other trigger processors on the special ECL bus, and to the read-out system in the Front End Buffer.

The trigger algorithms, cross-assembled and tested by the simulator program in an external computer, are recorded on EPROMs and bootstrapped into the DSP internal memory from the "Boot EPROM" at power-on or reset.

3.2 Debugging monitor and control software

A debugging monitor has been written and installed on EPROMs mapped to the external P-memory space of the DSP. It handles some basic tests of the memories and registers of the module, under the control of a terminal, connected to the module, .

On the host computers interfaced to the Fastbus system of the experiment, a stand-alone control program can access the module via its Fastbus Slave Interface. It allows an operator to interactively initialize and extensively test the module. The DAS packages of the experiment use a subset of the available functions for initialization and read-out.

These functions are mainly implemented by the CSR registers described in section 2.2.

The control program can also write and read the Dual Port Memory (DPM) in Fastbus Data Space. As briefly mentioned in Section 2, one half of the DPM is dedicated to the FEB. The other half is used for communications between the module and the control program or DAS. A simple "Mailbox" protocol has been set up: from both ports, messages can be written in the DPM, generating an interrupt in the other port. This protocol can also be used to download DSP programs, cross-compiled in an host computer.

4. PERFORMANCE

The module is clocked at 20 MHz, corresponding to a DSP half-cycle of 50 ns. By fully exploiting the microprocessor architecture, and by carefully matching the timing of the interfaces to the DSP cycles, we were able to pipeline several actions in one cycle: read-out of a set of input trigger data into the "INPUT Interface" registers, transfer of one data word into the DSP internal registers, addition of the previous data in the DSP accumulator. For example, the simple algorithm described in section 3.1 is completed in less than three microseconds, including the recognition of the "Start" signal, the read-out and addition of 24 data words, the comparison with two thresholds and the output of the results to the Local Trigger Supervisor.

5. CONCLUSION

We have designed and built a Fastbus module, based on a commercially available RISC microprocessor, the Motorola DSP56001 and on XILINX Programmable Gate Arrays. The very large scale integration of these components allowed a reduction of about one order of magnitude in both size and cost, with respect to specially designed processors of the previous generation. At the same time, we obtained an important gain in speed, in simplicity and in reliability.

The module is presently used in the trigger system of the end-cap electromagnetic calorimeter of DELPHI. The flexibility of the module makes other trigger applications rather straightforward. Extensions of the system including the addition of more memory and other processing units on the same Fastbus board are being considered.

Acknowledgements

We gratefully acknowledge the support and advice of our colleagues of the DELPHI-Trieste group. The technical skill of Giorgio Maselli from INFN Trieste has been invaluable for the success of the project.

References

[1] T.Lingjaerde, Internal note CERN/DD/75/17.
G.Lutjens, Report CERN 81-07 (CERN, Geneva, 1981), p. 236.

[2] Motorola Inc., DSP56001 Advance Information, ADI1290 Rev.1 (1988).

[3] P. Checchia et al., Nucl.Instrum. Methods A248 (1986) 317.

[4] XILINX Inc., XC3020 Logic Cell^{TM} Array Family (1988).

[5] D. Bulfone et al., DELPHI Internal note 86-100/DAS-42.

[6] F. Bertolino et al., IEEE Trans. Nucl. Sci., NS-36(5) (1989) 1469.

[7] L.B. Levit, G.J. Blanar and M.L. Vincelli, IEEE Trans. Nucl. Sci., NS-33 (1) (1986) 925.

[8] H.S. Stone, High performance computer architectures (Addison-Wesley, Reading, Mass., 1987).