PARALLEL PROCESSING ARCHITECTURE FOR HIGH PERFORMANCE DIGITAL SIGNAL PROCESSING

D. Crosetto, R.W.Dobinson and B. Martin

CERN, Geneva, Switzerland

Abstract

Currently available Digital Signal Processors (DSP) offer extremely high computational performance, yet no convenient mechanism is available to solve the interconnection problem.

Transputers are designed as building blocks for large processor arrays. Both hardware and software support the distribution of processes across configurations of multiple processors in a scalable way. Array of processors can be reconfigured by means of commercially available crossbar switches.

We describe a module that integrates the computing power of the DSP with the communication facilities of the Transputer in an industry standard package.

1. Introduction

Currently available Digital Signal Processors (DSPs) offer extremely high computational performance for a wide variety of signal processing needs. It is often necessary to process a large number of signals in parallel and to correlate their behavior. The use of an array of DSPs for this purpose is very attractive, but no standard and convenient mechanism is available to solve the interconnection problem.

Transputers (from INMOS), on the other hand, are designed as building blocks for large processor arrays. Both hardware and software (Occam), support the distribution of processes across configurations of multiple processors in a scalable way. Arrays of processors can be reconfigured by means of commercially available crossbar switches. However, Transputers lag behind DSPs in processing power. For example, a complex 1024 point fast Fourier transform of a complex array can run an order of magnitude faster on a DSP than on a Transputer.

To facilitate the parallel processing of many correlated signals we propose to consider the integration of the two technologies; DSPs for acquisition and initial processing of digital signals and Transputers for communication and overall signal correlation.

2. Digital Signal Processors

Digital signal processing is concerned with the real-time treatment of digitized analog signals which are discrete in both amplitude and time. Digital signal processors [1] are special purpose microprocessors designed to be particularly efficient for this sort of computations. The distinguishing features of DSPs compared to conventional microprocessors are separate program and data memories (Harvard architecture) and their emphasis on multiply-accumulate operations. DSPs are stripped down processors optimized for simple, repetitive calculations with very high data flow.

Traditionally, DSPs have been considered difficult to program efficiently without assembly language coding and fine tuning. However, this situation is now changing as manufacturers offer design aids such as libraries and software simulators.

Among the wide variety of DSPs on the market, we have chosen to work with the AT&T DSP32C [2] which is a state of the art 32-bit DSP including on-chip memory, floating point hardware and three separate DMA channels for high speed input/output. Its peak performance is quoted by the manufacture as 25 Mflops.

3. Transputers

The INMOS transputers [3] are powerful microcomputers of which the current generation is containing on a single chip the following features: an integer processor, a floating point processor, 4 Kbytes of fast memory, four 20 Mbps communication links, two timers and a real time kernel. Transputers were designed to be connected together and work cooperatively. The Occam language, designed by David May and based on idea of Prof. Hoare [4], has been developed to facilitate the parallel programming of Transputer arrays.

The high level of integration of the transputer permits a complete system of processor, memory and I/O to be realized in less than four square inches of printed circuit area. A standard packaging, the dual-in-line Transputer Module (TRAM)[5] permits such a system to be plugged directly into a socket on a motherboard. The socket carries power, clocks, links and resets between the TRAM and the motherboard. Typical motherboard has 8 to 10 slots. In the 8 slot motherboard, the footprint of the socket is defined such that eight adjacent sockets may be occupied by eight size 1 TRAMS, four size 2 TRAMS , two size 4 TRAMS or one size 8 TRAM. In this way a standard motherboard may be configured according to the processing needs of the application.

4. Fast Digital Parallel Processing module

In order to test an idea of mixing Transputers and DSP's we have decided to construct a test vehicle, the Fast Digital Processing module (FDPP), to measure the performances and test its programmability in a High Energy Physics Data acquisition environment.

4.1. Hardware

The initial packaging of the FDPP is as a 4 unit TRAM board. This allows us to add DSPs to existing TRAM based Transputer systems with relative ease. The FDPP is equivalent to the widely used INMOS B404 TRAM board plus a DSP and an input port for digitized data.

The module contains two processors, a 25MHz T800 Transputer and a 50MHz AT&T DSP32C. The Transputer is equipped with 2 Mbytes of external DRAM in addition to the 4 Kbytes of fast on-chip RAM (40 ns cycle time).

The Transputer shares access with the DSP to two memory banks which can be switched by the Transputer between the two processors in 100ns.

Each memory bank is built of 128 Kbytes of 35 ns SRAM. One of the Transputer links is used as a single byte "mailbox" to and from the DSP.

The DSP has access to the two switchable memory banks (60 ns cycletime) as well as its on-chip memories (20 ns cycle time). In addition there is a 16-bit wide 25 Mhz FIFO input port for digitized signals.

Data is stored in the FIFO by an externally generated write signal, the FIFO can also be reset externally. A FIFO full signal is generated by the FDPP and the DSP can send a reset to the external data acquisition equipment. Data can be transferred between the FIFO and the DSP memories by its on-chip DMA controller at a rate of one 16-bit word every 160 nsec.

In general the data path of the FDPP allows two mechanisms for passing data between the Transputer and DSP:

- short messages are transferred using Transputer links, providing synchronization automatically.

- large blocks of data are communicated using the switchable memory banks.

The Transputer always acts as the master processor. It can reset the DSP under program control. The DSP I/O port is used by the Transputer for loading programs into the DSP and monitoring/controlling its activity. The input FIFO port can be switched from the DSP to the Transputer, thus the FDPP can be operated with or without the DSP. This is a useful feature for both the debugging phase and for later performance analysis.

Fig 1 shows how the basic FDPP building block can be used to synthesize an array of communicating signal processing nodes.

The DSP has a double role. Firstly, it acts as an intelligent and powerful front-end for acquiring data. Secondly, it acts as a performance booster to the Transputer (a likely one order of magnitude improvement for the digital signal processing applications under consideration).

The DSP is not to be used as a co-processor to implement specific instructions, but rather to off-load certain computationally intensive algorithms like matrix manipulation, peakfinding, track finding, cluster finding, pattern recognition, filter and FFTs, from the Transputer which is seen as the master.

4.2. Software Development Systems

We expect it to be straightforward to incorporate the FDPP Transputer into existing software environments, it will simply appears as any other processor in an array of interconnected TRAM boards. Immediate use can be made of the INMOS software development systems, in particular the Transputer Development System (TDS) [6] and Toolset [7]. The Transputer can be programmed in the Occam language developed by INMOS for parallel processing applications and also in well established high level languages such as Fortran, C and Pascal. Recently UNIX-like operating systems for multiple Transputers such as HELIOS have become available [8].

The DSP will be programmed in either assembly language or in C. A library of standard routines is provided for many commonly used functions. Table 1 lists a small subset of these routines together with their execution times on the AT&T DSP32C. Cross software for the DSP could also be developed on a host computer running UNIX or MSDOS operating system.

A simulator for the DSP is also provided on the IBM PC which we have chosen as a host.

For the hardware debugging, we will use an in circuit emulator available for the IBM PC.

5. Status of project

A prototype FDPP has been constructed and is being tested. Fig 2 shows the board layout, components have been mounted on both sides of the printed circuit board.

6. Conclusions

Arrays of DSPs have already been used in high energy physics experiments [9]. However, the interconnections were made in an ad hoc way and all programming done in assembly language with the absence of an operating system. The described project is an attempt to use a well defined interconnection system and benefit from the availability of high level languages and where appropriate the features of an operating system. This should permit a much higher productivity in developing the specific parts of the applications.

Table 1. Partial list of the AT&T DSP32 Library routines.

NAME	DESCRIPTION	TIME
	MATRIX ROUTINES	
_mat2x2	Multiply two 2x2 matrices.	1.44 usec
_mat3x3	Multiply two 3x3 matrices.	3.2 usec
_mat5x5	Multiply two 5x5 matrices.	11.04 usec
_matinv	General purpose square matrix inversion routine	70.00 usec
	FILTER ROUTINES	
_iir2	Calculate the output of a 2-section 4-multiply per section infinite impulse response filter.	1.62 usec
_toneamp	Returns an estimate of the square amplitude of a pair of digital oscillators from the state variables of the tone-pair	4.72 usec
	ADAPTIVE FILTER ROUTINES	
_LMS	Implements the real, least-mean-square (LMS) algorithm.	2.08+0.8 N usec
	FFT ROUTINES	
-ffta	Calculates the fast Fourier transform (FFT) of a complex array. The size of the array must be a power of 2 and the maximum array size is 4096	(N=64) 172 usec
_fftc1K	Calculates a complex 1024-point fast Fourier transform (FFT). Separate arrays are used to store the real and imaginary parts of data.	3.22 msec
	GRAPHICS/IMAGIN ROUTINES	
_grey	Converts an array of color pixels represented by 16bits per pixel (5 bits per color) to an array of pixels with a 5-bit grey scale.	0.72+8.1N usec

References

[1] David A. Mindell "Dealing with a Digital World". Byte magazine, August 1989, pp. 246-256.

[2] WE DSP32C Digital Signal Processor Data Sheet.

[3] INMOS "The Transputer Data Book" Nov. 1988.

[4] INMOS OCCAM2. Reference Manual, Prentice Hall 1988.

[5] Paul Walker, INMOS Technical Note 29, Dual Inline Transputer Modules (TRAMS) INMOS, 1987.

[6] INMOS Transputer Development System (TDS), Prentice Hall 1988.

[7] INMOS, OCCAM2 TOOLSET manual, 1989.

[8] PERIHELION SOFTWARE "The Helios Operating System", Prentice Hall 1989.

[9] D. Crosetto, et all. IEEE Transaction on Nuclear Science. "Parallel arrays of digital signal processor as central decision elements for upper level triggers in High Energy Physics experiments." February, 1988.

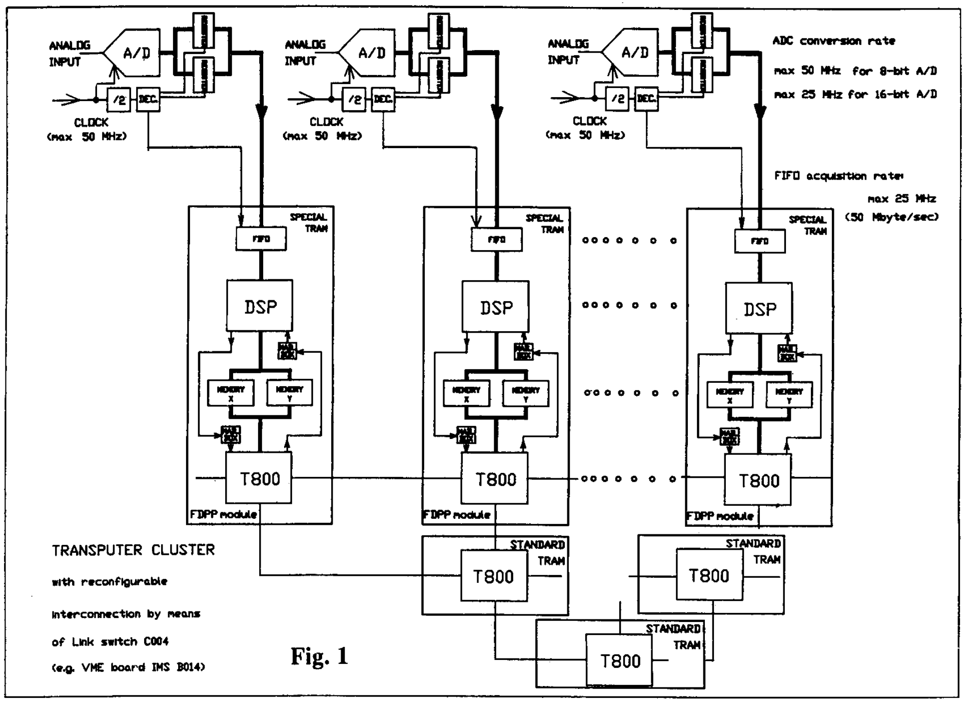

Fig. 1

Fast Digital Parallel Processing module (FDPP)

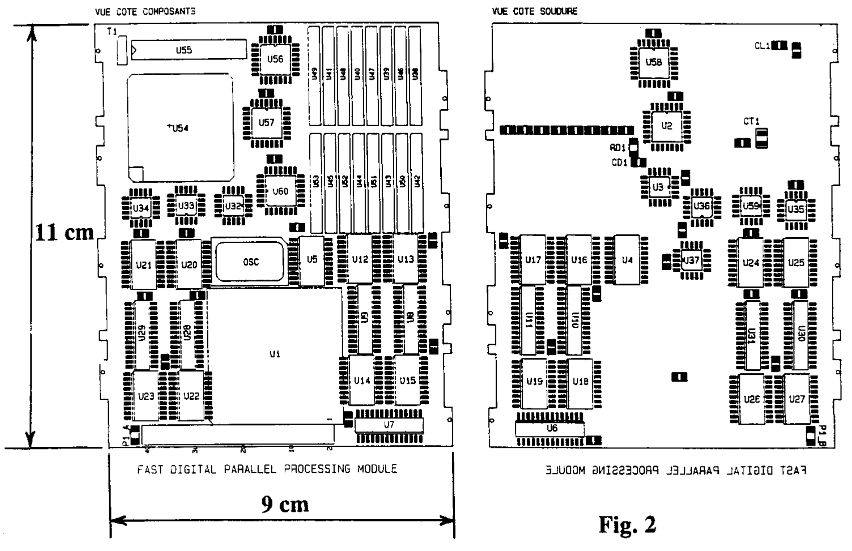

Fig. 2