The following full text is a publisher's version.

For additional information about this publication click this link. http://hdl.handle.net/2066/124371

Please be advised that this information was generated on 2018-11-23 and may be subject to change.
A two-Transputer VME module for data acquisition and on-line event selection in ZEUS


NIKHEF-H, PO Box 4188, NL-1009 DB Amsterdam, The Netherlands

Received 14 January 1993

For the ZEUS experiment at the HERA e−p collider in Hamburg a versatile and fast VME-based processor module has been developed at NIKHEF-H. The single-slot wide VME module contains two INMOS T800 Transputers, each with private memory, and a triple-port memory (TPM), accessible to both Transputers. Both Transputers have access to the VME bus, while an external master can also access the TPM directly. The VME accesses proceed typically with 10 Mbyte/s. Three application-specific integrated circuits (ASICs) handle the internal and external Transputer interfacing. The module combines the VME environment with the parallel processing and link-oriented environment of the Transputer. Over 200 modules are used in the data acquisition and event selection systems of ZEUS.

1. Introduction

ZEUS is one of the two detectors at the HERA electron–proton collider at DESY in Hamburg. In a large detector like ZEUS with 250 000 electronic channels a parallel approach for readout and processing of the data is necessary. Therefore, many processors fetch the data and perform trigger calculations concurrently. For these tasks a processor module has been developed at NIKHEF [1]. In the module, two processors (INMOS Transputers) have access to front-end modules via the VME bus. Several subdetectors of ZEUS use this module for reading out the data, for trigger processing and for communication with other systems [2–4]. Furthermore, the module is used for event building [5] and for the global second-level trigger [6]. The latter applications required that memory of the module had to be made accessible over the VME bus.

The choice for the Transputer was based on:

i) the DMA-driven data transport over four serial links, providing a simple means of communication between Transputers in a network,

ii) the multitasking kernel with task switching times of just a few μs,

iii) support in the occam language for parallel processing and for interprocess communication and synchronization, eliminating the need for an operating system.

The parallelism in the ZEUS on-line system is geometrical: Transputers fetch and process each apart from the subdetector data. But pipeline parallelism is also applied: data from subsequent events is simultaneously processed at different processing levels. Data transport is carried out in parallel with data processing. Statistical fluctuations are derandomized by extensive buffering. Each communication or processing step takes less time than the average time between events.

2. Architecture of the 2TP-VME module

The layout of the module is sketched in fig. 1. The 2TP-VME module features:

i) two T800 (or T425) Transputers running at 20 MHz,

ii) 4 or 16 Mbyte of private memory (dynamic RAM) per Transputer,

iii) 128 or 512 kbyte of triple-port memory (TPM) (static RAM), accessible to both Transputers and from the VME bus; lockable for exclusive access by one Transputer and allowing for read–modify–write access from the VME bus,

iv) VME interface with:

• access to the VME bus by both Transputers,

• memory-mapped VME cycle generation,

• implementation of most VME cycle types, including read–modify–write, block transfer indivisible cycles and interrupt acknowledge cycles,

• pipelined VME write cycles.
A two-Transputer VME module

Fig. 1. Architecture of the 2TP-VME module.

- A pipelined VME read cycle mode called "postfetch mode";
- Generation of VME address modifiers through programmable address modifier registers (eight per Transputer, corresponding to different parts of the memory map);
- Generation of the four most significant address bits through a register (one per Transputer) for access to the full 32-bit address range;
- Routing of VME interrupt requests to the event input of one of the Transputers and of VME bus errors to the event input of the Transputer causing a bus error;
- VME arbiter functions (no need for an external arbiter if the 2TP-VME is the only master in the VME crate);
- Connection between the two Transputers for mutual interrupt capability;
- External connections:
  - RS-422 type balanced drivers/receivers for external connections of all eight Transputer links and additional logical signals via the VME P2 connector;
  - An input for a logical external signal (edge sensitive) for each Transputer, routed to its event input;
- 12 diagnostic front-panel LEDs to display VME accesses and error conditions.

The module (fig. 2) occupies a single VME slot. Two piggyback boards (fig. 3), each with a Transputer, private memory and interface to the local bus (implemented in ASIC “Z-LOC”), are plugged onto a motherboard (fig. 4) with the TPM, the VME bus interface (partially implemented in ASIC “Z-VME”) and drivers/receivers for the links.

A link adapter board provides single Transputer link connections. The board is placed at the rear of the VME crate on the P2 connector. With the RS-422 interface up to 50 m at 20 Mbit/s and up to 100 m at 10 Mbit/s can be bridged. If optical buffering is used, link connections over more than 100 m are possible [5]. The board also provides connections for control (reset, analyse, error) signals, enabling either Transputer to be controlled individually. Alternatively, jumpers on the board can be set such that control signals for a network of 2TP-VME modules are daisy-chained.

3. VME Interface

3.1. VME Bus Cycles

Access to on-board and off-board devices is memory-mapped. The address range dedicated to VME access is subdivided into segments to accommodate all types of VME cycles. There are four main 1-Gbyte segments used (two partially) for VME access. Two provide a 1-Gbyte window on the VME address space for cycles with 32-bit addresses. These windows can be positioned with register 2 (see section 3.3). The other two segments are subdivided into smaller 16-Mbyte ones for 24-bit and 16-bit address cycles. Each segment is associated with one of the eight programmable address modifier registers, the content of which is enabled onto the VME AM-lines when a VME access is performed in the corresponding Transputer memory segment.

The Transputer does not feature 8-bit read and 16-bit read/write cycles. Solutions have been found to provide these nevertheless. In dedicated segments normal 32-bit read and write cycles are mapped onto
Fig. 2. The 2TP-VME module with the piggyback boards and motherboard shown separately.

Fig. 3. Block diagram of the piggyback board.
Fig. 4. Block diagram of the motherboard.
two VME D16 cycles: first a cycle for the high order word and then a cycle for the low order word of the Transputer long word. In other segments 32-bit read and write cycles are mapped onto a single D16 cycle. Whether the low or high order word is read or written is determined by the choice of the address segment. VME D8 read cycles are provided in several segments by a special mode with the byte to be read selected by the A29 and A28 address bits.

Write cycles to VME bus slaves are performed in a pipelined fashion, meaning that the VME interface independently performs the cycle after the address and data have been received from the Transputer. The Transputer can continue processing locally, while the VME interface completes the VME cycle. Read cycles in VME are not performed in such a pipelined fashion unless the so-called “postfetch mode” is used.

### 3.2. Postfetch mode read cycles

A special pipelined read cycle type, called post-fetch mode (PFM) read cycle, has been implemented to speed up reading datablocks from VME. In post-fetch mode the Transputer presents an address to the VME bus, but does not wait for the read cycle to finish; instead it returns with the data present in its local bus databuffers. This data is the result of the previous PFM read cycle that was completed while the Transputer was busy generating the next address and/or transferring the data from its local bus databuffers to local memory or to the TPM. A special end-postfetch-mode instruction reads the last data word. This pipelining of address generation and cycle completion speeds up data block transfer significantly (see section 5).

### 3.3. Control and status registers

Each Transputer on the 2TP-VME module has access to four registers with the following functions:

**Register 0:** to determine the source of a Transputer event input. The flagged event types are: VME interrupt, VME bus error, external error signal, external event signal, external event signal from the other Transputer of the module, dynamic memory refresh timeout.

**Register 1:** to set and inspect event flags (as specified for register 0) under software control for test purposes.

**Register 2:** to set the upper four VME address lines (A28, A29, A30, A31) for 32-bit address cycles and to control read–modify–write cycles and block transfer cycles.

**Register 3:** to generate external outputs: a signal on the external output or an event signal for the other Transputer of the module.

### 3.4. Interrupt and event handling

The event pin of the Transputer is used to implement an interrupt facility. VME interrupts are routed with jumpers to the event input of one of the two on-board Transputers. A read cycle from a dedicated memory segment provides the Transputer with the interrupt vector and generates a VME interrupt acknowledge cycle.

The source of an event can be determined from register 0. After taking appropriate action the event flag has to be reset.

### 3.5. Byte order

The most/least significant byte order definition of a 16-bit or 32-bit dataword in memory according to the Transputer is opposite to the order as defined by VME. The 2TP-VME module automatically performs byte swapping by inverting the A0 and A1 address bits. As a result the ordering of bytes in 32-bit numbers to be read/written by the module from/to VME memory is as defined by the VME specification. The dataword is to be found at the expected address.

Bytes in 16-bit datadwords to be read/written by the 2TP-VME module from/to VME memory are correctly ordered, but the dataword has a VME address with the A1 address bit inverted. An individual byte read/written by the module from/to VME memory is placed at a VME address with both A1 and A0 address bits inverted. Consequently, when the 2TP-VME module writes an array of bytes representing a character string to VME memory it is reversely ordered with respect to the VME specification. For accesses by a 2TP-VME module this effect is not noticeable, but for accesses by another type of processor board it usually is.

### 4. Data transfer rates

In table 1 the cycle times are given for access by the Transputers to memory, either local or on the VME bus. A transfer of data has to be set up first; for a Transputer block move this takes about 100 ns. In table 2 measured data transfer times are presented for large datablock sizes. Typically, with the modes selected in the ZEUS systems transfers proceed with about 10 Mbyte/s.

### 5. ASIC development

An important requirement for the module has been the limitation of its width to a single VME slot. From prototype modules it became clear that even with two
268

Table 1
Memory access times. One of the Transputers has optimized access to the TPM. Values for either Transputer are specified if different. Memory definition: PrivMem is the private memory of the Transputer. TPM is the 2TP-VMEs triple-port memory, VMEMem is the TPM of another 2TP-VME module accessed via the VME bus.

<table>
<thead>
<tr>
<th>Memory access</th>
<th>Cycle time [ns]</th>
</tr>
</thead>
<tbody>
<tr>
<td>From/to PrivMem</td>
<td>200</td>
</tr>
<tr>
<td>From/to TPM</td>
<td>150/200</td>
</tr>
<tr>
<td>To VMEMem</td>
<td>200</td>
</tr>
<tr>
<td>From VMEMem</td>
<td>440</td>
</tr>
<tr>
<td>From VMEMem*</td>
<td>250</td>
</tr>
</tbody>
</table>

*Transfer in postfetch mode.

piggyback boards the number of components would just not fit. Therefore, surface-mount technology was applied and moreover two types of ASICs were also developed. The ASIC on the piggyback board handles the address decoding and contains the control and status registers. The ASIC on the motherboard handles most of the VME interfacing (with several state machines) and controls the local bus. The CMOS chips made with 2 μm laser technology [7] contain 5600 and 10000 transistors, respectively. With this technology no expensive masks are required and therefore a second batch in case of errors in the design is relatively cheap.

6. Concluding remarks

For the ZEUS experiment the 2TP-VME module turned out to be applicable in various areas of the on-line system. Working with Transputers is found to be straightforward. The Transputer can be used as a building block when constructing large processor systems. Programmers unexperienced in real-time and parallel processing can quickly become productive because of the conceptually simple but effective support for this type of application in the Occam and parallel-C programming languages as used in ZEUS. These factors have led to a de facto standardization of Transputers in both readout and second-level trigger systems of this experiment. Moreover, the versatility of the 2TP-VME module has led to its use in other experiments as well.

Table 2
Measured read/write rates for 4096-word blocks of 32-bit data words using the Transputer block move instructions. They are specified for either Transputer if different.

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
<th>Transfer rate [Mbyte/s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>PrivMem</td>
<td>PrivMem</td>
<td>9.8</td>
</tr>
<tr>
<td>PrivMem</td>
<td>TPM</td>
<td>11.2/9.8</td>
</tr>
<tr>
<td>PrivMem</td>
<td>VMEMem</td>
<td>9.8</td>
</tr>
<tr>
<td>TPM</td>
<td>PrivMem</td>
<td>11.2/9.8</td>
</tr>
<tr>
<td>TPM</td>
<td>TPM</td>
<td>13.0/9.8</td>
</tr>
<tr>
<td>TPM</td>
<td>VMEMem</td>
<td>9.9</td>
</tr>
<tr>
<td>VMEMem</td>
<td>PrivMem</td>
<td>6.2</td>
</tr>
<tr>
<td>VMEMem</td>
<td>PrivMem*</td>
<td>9.8</td>
</tr>
<tr>
<td>VMEMem</td>
<td>TPM</td>
<td>6.8/6.2</td>
</tr>
<tr>
<td>VMEMem</td>
<td>TPM*</td>
<td>10.0/9.8</td>
</tr>
<tr>
<td>VMEMem</td>
<td>VMEMem</td>
<td>4.9</td>
</tr>
</tbody>
</table>

*Transfer in postfetch mode.

Acknowledgments

The work described is part of the research program of the “Stichting voor Fundamenteel Onderzoek der Materie” (FOM).

References