An HS-Link Network Interface Board for Parallel Computing

A. Cruz[1], *J. Pech*[1,2,3], *A. Tarancón*[1], *C. L. Ullod*[1], *C. Ungil*[1]

[1] Departamento de Física Teórica, Facultad de Ciencias, Universidad de Zaragoza, 50009 Zaragoza, Spain
[2] Institute of Physics, Academy of Sciences, 180 40 Prague, Czech Republic
[3] CERN, 1211 Geneva 23, Switzerland

Abstract

An interface board capable to connect the PCI bus to two serial links at 1 Gbit/s each is presented. That PCI-HSLink board has been developed at CERN as a component of a testbed for the distribution and analysis of high energy data from new colliders. By using those boards and appropriate cross-link switches in our cluster of 16 Pentium Pro dual nodes (RTNN), a high performance, low cost full parallel machine is achieved.

Keywords: HS-Link, parallel computing

1 Introduction

The HS-Link technology is outlined in sec. 2, as well as the available devices. The PCI-HSLink board developed at CERN is presented in sec. 3, describing its use as traffic generator and network interface card. Finally, the parallelisation of the RTNN cluster composed of 16 PCs using those boards is discussed in sec. 4.

2 IEEE 1355 HS-Link

2.1 HS-LINK Technology

IEEE 1355 HS-Link [1] is a high speed serial link technology. Present day implementations work in the range between 700 MBauds and 1 GBaud. Such bidirectional point-to-point connections (HS-Link or simply *link* in what follows) may be used to connect chips on a printed-circuit, printed-circuits over a back-plane or racks by means of coaxial cable or optical fibre.

Standard IEEE 1355 defines four protocol layers: bit, character, exchange and packet. Characters are successive groups of bits representing data or control information, and are encoded by a 8B/12B DC balanced encoding scheme, where 8 data bits are encoded into 12 code bits. That coding scheme allows to use 256 different values as data and 126 control characters. Eight of them are reserved for the low level protocol of the *link* and are transparent to the higher protocol layer, the packet layer, preventing the emitter from overfilling the reception buffer.

The exchange layer controls the transmission of characters to ensure that the *link* is working properly, which involves such functions as control of flow through the *link* and the startup mechanisms. Packets consist of a destination header, which is used to route the packet through a switching fabric, followed by the actual payload data, and an end-of-packet character. There are no restrictions on the packet size.

2.2 HS-Link devices

2.2.1 The Bullit HS-Link interface chip

The Bullit chip [2] provides a parallel interface to an HS-Link. It consists of a transmitter/receiver pair, input and output FIFO buffers and a low level protocol engine. The FIFOs are 80 character deep and are accesed through a parallel interface. The protocol engine implements the low-level link protocol and performs functions such as 8B/12B encoding/decoding, flow control or link startup and shutdown.

2.2.2 The RCube 8-way HS-Link router

The RCube [3] is an 8×8 router for the IEEE1355 HS-Link networks, based on an 8×8 non-blocking crossbar switch and 8 bidirectional 1GBaud serial links. This results in total cross-sectional data bandwidth of 640 Mbyte/s. The RCube uses "Wormhole Routing", which allows packets of unlimited length to be routed. It also provides very low latency in lightly loaded networks, each RCube has a latency of 150ns. The device also provides adaptive routing which enables efficient load balancing in multistage networks.

2.2.3 HS-Link network construction

Several boards based on the HS-Link devices have been developed at CERN in order to build and study the behaviour of different network topologies under Large Hadron Collider exploitation conditions [4]. One of them is the 4×8-way switch module, which consists of 4 RCube switches with all their links brought to the front panel for connecting to other switch modules or network interface cards using coaxial cables. A T8 microcontroller is used to configure and monitor the RCube switches, and IEEE 135 DS-Links or RS232 connectors are used to control the module. Two connections are provided, which allows a daisy-chain or a star topology to be used for the control of several such modules. More complex HS-Link networks can be constructed by connecting several of those boards, the network topology depending on the cable configuration.

3 The PCI HS-Link interface

In order to access those networks, a board has been developed which acts as an interface between a PCI bus and HS-Links (see fig. 1). It can be used as a traffic generator for a network testbed, or it can allow a processor to access a pair of links through a PCI bus. This second functionality can be used either for network control purposes or to transfer data from processor to processor at high speed and low latency.

The AMCC S5933 is an interface to the PCI bus and the Altera FLEX 10K30 handles the multiplexing of that interface between two HS-Link channels. Each one of those channels is provided with one Bullit and the supporting logic for reading and writing the emission and reception FIFOs. Data can be transferred directly to or from the PCI bus meeting the requirements of inter-processor communication. Each one of the two channels has been also provided with two memory banks, one for emission and one for reception.

The electronic logic linking memory, Bullits and PCI interface is implemented with programmable logic, the functionality of the board depending on the logic actually programmed. Those devices are programmable on board: the Altera 10K30 can be programmed accessing it through the AMCC, while the other six MAX 7128 components are linked by a JTAG compliant programming chain which can be controlled either from the FLEX 10K30 itself or from a specific connector.

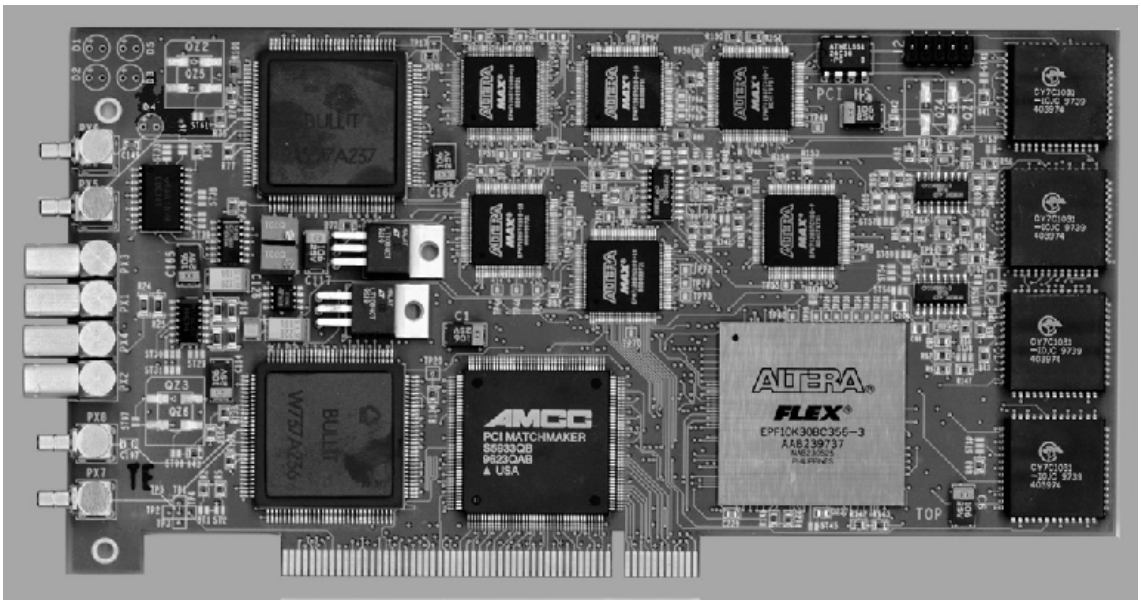

Figure 1: The Interface Board PCIHS.

3.1 PCIHS as traffic generator

When used as traffic generators, the boards are required to send packets according to a certain traffic pattern and to register packets coming from the network, without at any time minding the packet contents. Packet descriptors, storing the destination address, the packet length and the launching time, are stored in the transmission memories of the different HS-Link channels. The programmed logic reads the descriptors, interpret them and launch the packets. Similar descriptors are created when packets are received and stored in the reception memory, which is downloaded periodically to a control processor through the PCI bus. By examining the emission and reception descriptors, the behaviour of different network topologies can be studied.

3.2 PCIHS as network interface card

The PCI bus of a PC works at 33 MHz and its word length is 32 bits. Working in *burst* mode a transfer velocity of 132 Mbyte/s is obtained. The PCI bus manager on Intel based PCs is prepared to work in *burst* mode in peripheric writing, but data transfer from PCI peripheric to computer is always managed by the PC word by word. Yet, the AMCC component chosen as interface in the PCIHS board can also act as master of the transfers in both directions and is capable to act in *burst* mode in both cases. Support for direct memory access (DMA) has been added to the logic defined in the PCIHS board in order to use this feature.

Two Bullits working at 66 MHz allow a bidirectional flow of 132 Mbyte/s. The PCI bus transfer speed is 132 Mbyte/s, but is onedirectional. The presence of memory on the PCIHS board allows the saturation of the bidirectional 132 Mbyte/s channel formed using both *links* jointly. Data flow from the PC is sent directly through the *links*. The memories on the board are then used to store the data arriving through the links while the PCI bus is busy, in order to download it later when the bus becomes available.

Such *store-and-forward* procedure is not the best solution for a high performance communication network, one of the reasons being the latency introduced by the data copying. Yet, the effect is more or less serious depending on the kind of communication to be carried out. Paralleli-

sation of lattice Monte-Carlo simulations often leads to equal amounts of data to be transferred in both directions. We can then proceed as follows: the nodes send each other simultaneously the information to be exchanged, that information is stored temporally in the memory of the receiving boards and transferred later to the PC memory. In this case there is no performance loss. Of course, if improvement of communications is desired with protocols requiring the transmission of control packets between nodes, the latency effect must be considered and the packet size must be adjusted to optimise performance.

3.3 Software support for the PCI HS-Link interface

An UNIX-like controller (*driver*) for the operating system LINUX has been developed to control PCIHS boards when used as traffic generators, and extended to manage the communication tasks. It is a *character* controller providing access to the inner registers, autodetection, support for multiple boards and independent access to each *link*. The driver supports most of the standard UNIX system calls for device control tasks. The `read()`, `write()`, `ioctl()`, `mmap()` and `select()` functions, timers, task queues and interrupt handlers have been implemented.

4 Parallelisation of RTNN

The simplest parallelisation of the 16 nodes of RTNN can be achieved by means of a single 4×8-way switch. We can form two separate parallel computers composed of 8 RTNN nodes, using two RCubes to connect each of those clusters. Parallel routing in the RCubes provides full connectivity inside a group. Using three 4×8-way switches full connectivity of the whole cluster for usual network topologies can be achieved.

Low latency and broad bandwidth can be achieved allowing the application to access directly the network interface. Moving most of the processing of the protocol to be used to the user level, it can be specialised and better integrated in the application, avoiding context switching and improving the computational performance.

Preliminary tests of communication between RTTN nodes have yielded a transmission rate of 79 Mbytes/s, limited by the chipset of RTNN. The replacement of the nodes with newer machines will allow us to achieve the peak transmission.

5 Acknowledgements

We wish to acknowledge the financial support of CICYT (Comisión Interministerial de Ciencia Y Tecnología, Spain), under project AEN97-1708.

References

1 IEEE Std. 1355, *Standard for Heterogeneous Inter-Connect (HIC), Low Cost Low Latency Scalable Serial Interconnect for Parallel System Construction*, IEEE Inc.,USA (1995).

2 *The Bullit Data Sheet*, version 2.0, Bull Serial Link Technology (1995).

3 *The RCube Specification*, version 1.7, Laboratoire MASI, Université de Pierre et Marie Curie, Paris, France (1997).

4 C.R. Anderson et. al. *IEEE 1355 HS-Links: Present Sattus and Future Prospects* Architectures, Languages and Patterns, IOS Press, 1998 `http://www.cern.ch/HSI/dshs/publications/wotug21/hslink/html/hslinkpaper.html`