SmartHSL: an Evaluation Board for the IEEE 1355 Technology

J.J. Lecler, A. Fenyo, A. Greiner, F. Potter,

UPMC / LIP6, 4 pl. Jussieu 75005 Paris, France.

Abstract. This paper presents an evaluation board for the RCube router and the HSL high speed interconnect technology, developed in the framework of the OMI-HIC project. The RCube router is an 8*8 dynamic crossbar, fully compliant with the IEEE 1355 standard for high speed links (HSL). It implements a wormhole routing strategy. The router latency is 150 ns. Each port has a 1Gbit/s throughput in both direction, for a total bandwidth of 640 Mbytes/s. The RCube router has been designed at Pierre et Marie Curie University, and is now commercialized by the BULL company.
The SmartHSL evaluation board is a PCI board containing an RCube router and a Motorola 56301 microprocessor for the PCI interface. One port is used to connect the router to the host processor, and seven ports can be used to build an HSL network. The physical links between the SmartHSL boards are bidirectionnal shielded wires.
The SmartHSL board can be used with any CPU board having a PCI interface. The on-board 56301 processor can access the PCI bus in both master and slave mode, in order to put or get data to/from the HSL network.

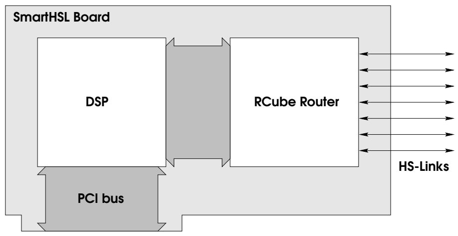

Figure 1: SmartHSL Board Diagram

1 Introduction to the IEEE 1355 Technology

The IEEE 1355 standard defines two substandards named DS-Links and HS-Links, which support 100 Mbit/s and 1Gbit/s respectively. SmartHSL uses the IEEE 1355/HSL technology.

1.1 The Physical Link

The physical media is a point-to-point, full duplex, high speed serial link. It’s a single wire containing two coaxial cables, one per direction, as shown on the figure 2. The throughput of the link is 1 Gbit/s in each direction.

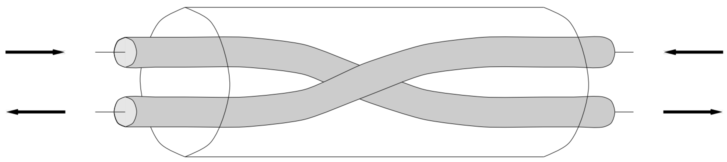

Figure 2: IEEE 1355/HSL Physical Link

The communication is asynchronous, emitter sends data at its own speed and the receiver recovers both emitter's clock and data. This technology avoids skew problems. The bytes are encoded as shown on the figure 3.

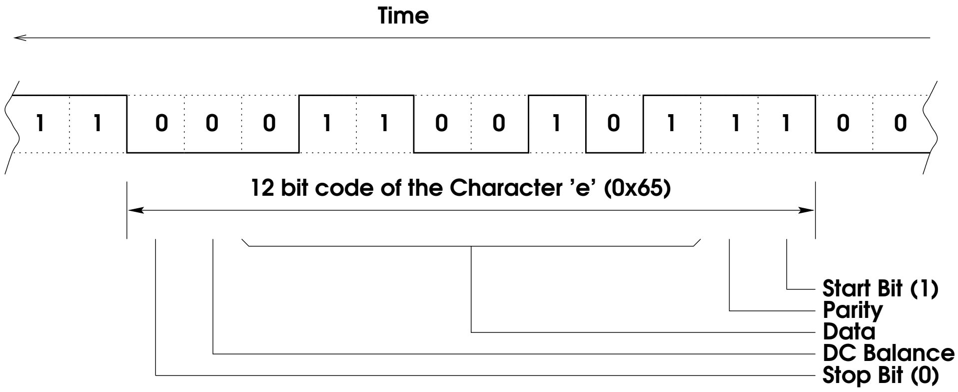

Figure 3: Codage des symboles 12 bits sur HSL

Since each byte requires 12 bits on the link, the throughput on the link is about 80 MB/s. $(1\ Gbit/s\ /\ 12bit/byte = 80\ MB/s)$

The maximum length on coaxial cables is about 5 meters. HSL can also be used on optic fiber (in which case the maximum link length is about 70 meters), but this requires external transceivers, and is not implemented on SmartHSL.

1.2 The Serialization/Deserialization Operator

The serialization/deserialization and clock recovery macro-cell has been developed by BULL S.A. This operator uses a standard .5 micron technology, and can thereafter be integrated at low cost. Furthermore, the macro-cell does not require any external component, the HS-Link is directly connected to the circuit's pin.

On the figure 4 is presented the complete integration of a HS-Link, from the silicon to the physical link.

1.3 The RCube Router

The hardware architecture department of the LIP6 laboratory has developed a VLSI packet router between 8 HS-Links. This router called RCube [2] is a 208 pin, 3.3 V integrated circuit.

The IEEE 1355 standard defines the packet format as shown on the figure 5. A 1 or 2 byte long header specifies the destination of the packet in the network, and the following bytes constitute the pay-load. The packet length is unlimited and is closed by a special EP (End of Packet) character.

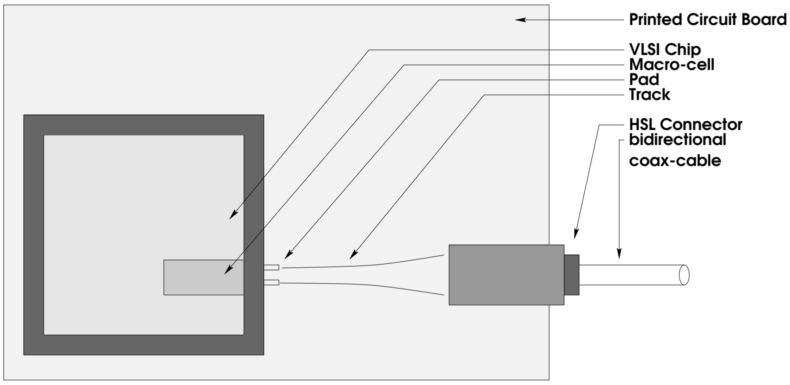

Figure 4: Macro/Circuit/Board/Connecxion Integration

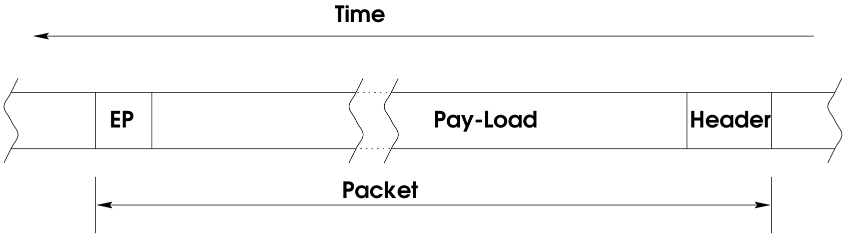

Figure 5: IEEE 1355 Packet Format

An interval routing technique is used, which authorizes a compact representation of the routing tables.

In order to reduce latency, the router uses a wormhole strategy. The header of a packet is reemitted before the tail is received. The inherent latency of the router (without contention) is 150 ns. The flow control uses a credit strategy: router emits to a neighbour, only if it is granted that the receiver can accept 32 bytes of data. This authorization is carried by special FCC (Flow Control Characters) multiplexed in the opposite flow. This technique introduces a 3 % penalty on the bandwidth and prevents any data loss.

If a packet requests a busy output port, it can be redirected to an other output. This capacity is called adaptivity. If all the potential out ports are busy, the packet is blocked waiting.

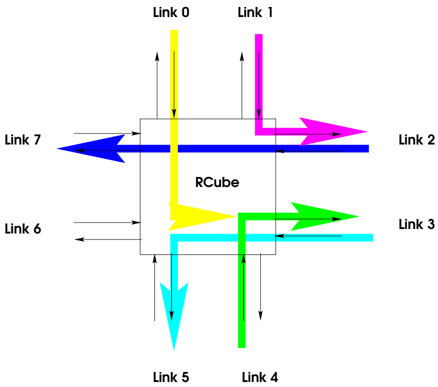

Figure 6: RCube router diagram

2 SmartHSL Features

2.1 Overview

The SmartHSL board offers an elegant solution to enter the world of HSL high speed networks. Since many different communication protocols can be used over such networks, we did not want to freeze a given one. On the other hand, the packet construction and analysis is a complex task, that can not be efficiently done by hosts processors.

So, SmartHSL provides both a network access via the RCube router, and a generic access to the PCI bus of the host computer handled by a dedicated DSP. This DSP (a Motorola's 56301) is the core of the board design. It can be freely programmed by the user.

Here are listed the main primitives available to the protocol designer.

2.2 PCI Access

The PCI Bus (32 bits, 33 MHz) offers both bandwidth and system independence. The DSP56301 [4] provides an integrated PCI controller. Since this port supports full speed data transfers, the bandwidth on the PCI bus is limited by the Host CPU Bridge quality, and/or by the firmware running on the DSP.

As a rule, any master transaction can be initiated by the DSP, using dedicated registers specified in the DSP56301 data sheet. I/O or memory spaces can be targeted, by any transaction type (Read, Read Line, Read Multiple, Write, Write Line, Write and Invalidate). Though, transaction exceptions (Disconnect, Abort, Time Out) must be handled and recovered by software

As a slave, SmartHSL is mapped as a 64 KB window in the PCI memory space. All read or write PCI accesses in this window are handled by the DSP as pop or push in two 32 bit wide FIFOs. Those FIFOs are mapped as pseudo registers in the DSP address space.

2.3 Network Access

From the point of view of the DSP, the access to the HSL network is ensured by a pseudo register mapped in the DSP memory space. The datapath between the DSP and the router is 16 bit wide. Reading this register pops one 16 bit word from the incoming FIFO, writing it pushes one 16 bit word to the outgoing FIFO.

Datapath consists of two 512 word long FIFO (one for each direction). The status of these FIFO can be consulted via another register. This authorizes a flow control on a base of 1, 64, 256 words.

The DSP can be interrupted on the following events: reception register contains a data character, reception register contains a control character (e.g. End of Packet), transmission register is empty.

2.4 System Control and Status

A DSP memory mapped register controls the router reset signal, In order to control the initialization by software. Each link can be reset separately.

Another memory mapped register is used to load the routing tables into the router. RCube use a 258 byte routing table, loaded by 258 successive writes in this register.

The calibration state of the external HS-Links can be observed, to determine whether the link is connected or not. This capability can be used to verify that the effective network topology is identical to the expected one. An interrupt can be triggered on the DSP in case of calibration loss on a link.

The software can also configure whether link 0 is directed to the DSP or is just a standard serial link. This way SmartHSL can be used as an input/output node with 7 outgoing links or an independent 8x8 crossbar.

2.5 System Environment

The DSP56301 uses a 66 MHz clock. It's a RISC processor that can execute one ALU operation and two moves each cycle. 6 DMA engines are also available for parallel data transfers.

The DSP also provides 4 Kword program and 2x2 Kword data on-chip memory. This capacity can be extented by a 64, 128, 256 Kword optional external memory.

SmartHSL also provides a 512 KB flash ROM, which can be used for stand alone boot.

2.6 System Boot and Debug

A switch on the SmartHSL board determines whether the DSP should boot from its flash ROM or from the PCI bus. Utilities are provided under FreeBSD to upload a bootstrap code, and to flash a new bootstrap in the flash.

The DSP provides an on-chip RS-232 interface, which is exported to a miniDIN connector. A text terminal can thus be connected as a console of the subsystem, for debug and monitoring purposes.

Two others low speed serial links are also provided on the card, through RJ11 connectors. These links can be daisy-chained and used for configuration and remote systems (such as external routers) control.

Finally, the DSP has got a JTAG port accessible via an on board connector. This port can be used to connect an external probe or emulator to debug DSP firmware. It supports the JTAG and OnCE (On Chip Emulator) protocols.

3 SmartHSL Architecture and Expected Performances

The global SmartHSL architecture is presented figure 7. On this figure, the data-path from the host to the network is shown with dark gray.

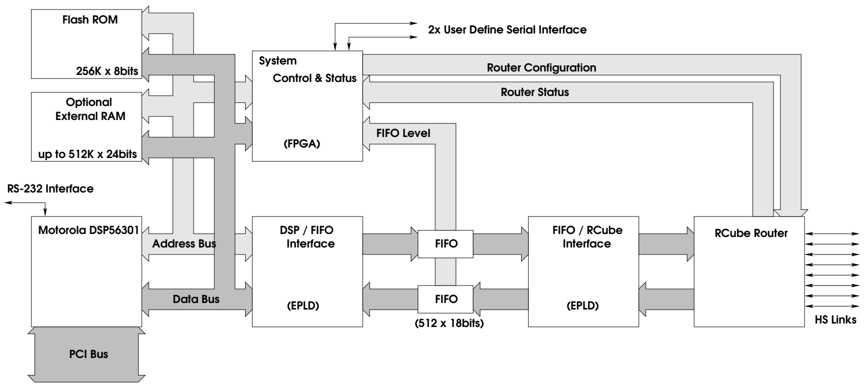

Figure 7: SmartHSL Architecture

3.1 Data Pathes

The PCI Bus peak bandwidth is 132 MB/s. The observed maximum bandwidth is actually between 80 MB/s and 100 MB/s, depending on data transaction type and PCI bridge quality.

The system bus of the DSP is 24 bit wide. Only 16 bits are used to and from the network FIFOs. Accesses to these FIFOs are made with 1 wait-state at 66 MHz. Assuming no other external accesses are made (to external RAM, ROM, or Control chips), by the internal DSP DMA engines, the efficient bandwidth on this bus is 66 MB/s, half-duplex.

The router uses a 80 MHz clock, and is able to import and export one byte per cycle. Time multiplex and demultiplex between the FIFO (16 bits) and the router (8 bits) is ensured by the FIFO/RCube PLD Interface. Thus, the bandwidth on this datapath is 80 MB/s, full-duplex. The same PLD is also responsible of flow control credit management.

In this design, the bottle-neck can be either the DSP System bus (66 MB/s, half duplex) or the DSP computing power, depending of firmware code optimization.

3.2 Latency vs. Bandwidth Trade-Off

Many different communications protocols can be implemented in firmware on the SmartHSL board. Among other things, firmware code writer should be quite careful about HSL flow control management. The simple, byte based, flow control (If the reception FIFO is not empty, then pop a data, if the emission FIFO is not full, then push a data) is very inefficient from the point of view of the throughput. It requires conditional branch which implies DSP pipeline stall states.

Since the HSL datapath FIFO status can be observed from the DSP, it is easy to use a larger granulatity flow control (If the reception FIFO level is above 64, then the DSP can pop 64 data words without any other test). This solution is evidently better from the bandwidth use point of view, but it generates latency.

The expected bandwidth for 64 word bursts is about 50 MB/s per processor.

4 Conclusion and Industrial Opportunities

The SmartHSL board has been successfully tested on Pentium based computers, under unix FreeBSD.

Citius Technologies [1] is a start-up company founded in 1996 by researchers from the Université Pierre et Marie Curie (UPMC). Citius has obtained from UPMC a worldwide license for the RCube router, SmartHSL Board, and other chips and boards using the IEEE 1355/HSL technology.

The main business objective of Citius is to promote this technology, in order to sell fabrication licenses of those chips to one or several silicon foundries. This requires developing the potential market for those components.

The SmartHSL kit has been designed, to let anyone interested in high-speed networks technologies evaluate the IEEE 1355 standard and the RCube router. Its genericity allows its use in many applications, including parallel computers, network switching, digital video systems, ...

---

[1] email: citius@lip6.fr

References

[1] IEEE-1355.
IEEE 1355 Standard for Heterogeneous Interconnect (HIC) Low Cost Low Latency Scalable Serial Interconnect for Parallel System Construction
IEEE Standards Department, Aug 1994

[2] RCube Specification.
http://asim.lip6.fr/mpc/rcube/

[3] PCI Special Group of Interest
The PCI Local Bus Specifications. Rev 2.1
PCI-SIG, July 1995.

[4] DSP56301 Reference Manual.
http://www.mot.com/SPS/DSP/products/DSP56301.html