

# S-LINK Transmission Measurements

Maris Abolins, Beatriz González Piñeiro, Peter F. Peterson

10-Oct-1998

## Abstract

This note describes performance measurements of data transfer via S-LINK between two CES RIO2 8062 processors. Transmission rates as function of packet size and number of packets were measured. A logic analyzer was used to test consistency of measurements.

## 1.0 General Description

Times of transmission of data between a source and a destination connected via S-LINK<sup>1</sup> are measured for packets of data with different size. The results of this measurement are used to estimate the transmission rate as well as the overhead needed by the software and hardware to perform the transmission.

## 1.1 Hardware

Source and destination programs were installed in two RIO2 8062 Power PC 604 based processors<sup>2</sup> running at 200 MHz. The RIO2's used LYNX OS 2.4 as an operating system with their files NFS mounted from a PC running Linux 5.1.

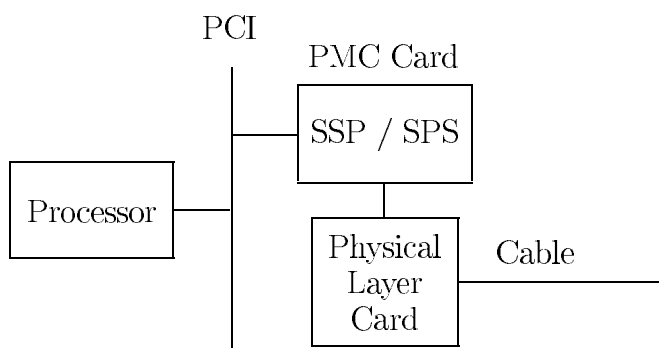


FIGURE 1. : Schematic of source and destination processors

Figure 1 on page 1 is a schematic representation of both the source and destination cards. On the source card the processor communicates via PCI bus with a Simple PMC to S-LINK interface (SPS) mounted in its PMC slot. A Link Source Card (LSC) provides the physical and electrical connection to an 80 pin cable carrying the signals to the destination. At the receiving end a Link Destination Card (LDC) receives the signals and trans-

mits them to the processor via an SSP (Simple S-LINK to PMC) card mounted in its PMC slot.<sup>3</sup>

## 1.2 Software

Data transfer is controlled and effected by means of standard S-LINK PMC library functions.<sup>4</sup> The source and destination programs used for transmission measurements were suitably modified for our needs with efforts made to remove all unnecessary software steps not needed for the data transmission. Time measurements used the *gettimeofday()* function included with the Lynx operating system. The structure of the program is indicated in Figure 2; the input parameters were the number of packets and the size of the packet.

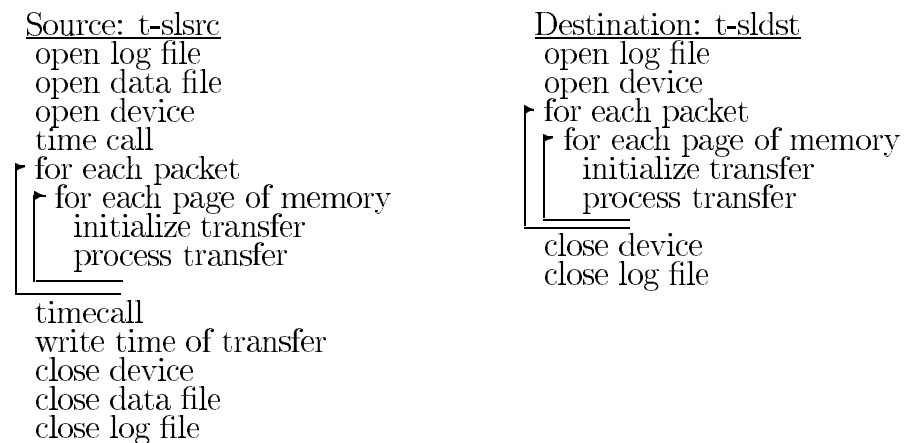


FIGURE 2. :Software Diagram for source and destination programs

## 2.0 Timing measurements using software

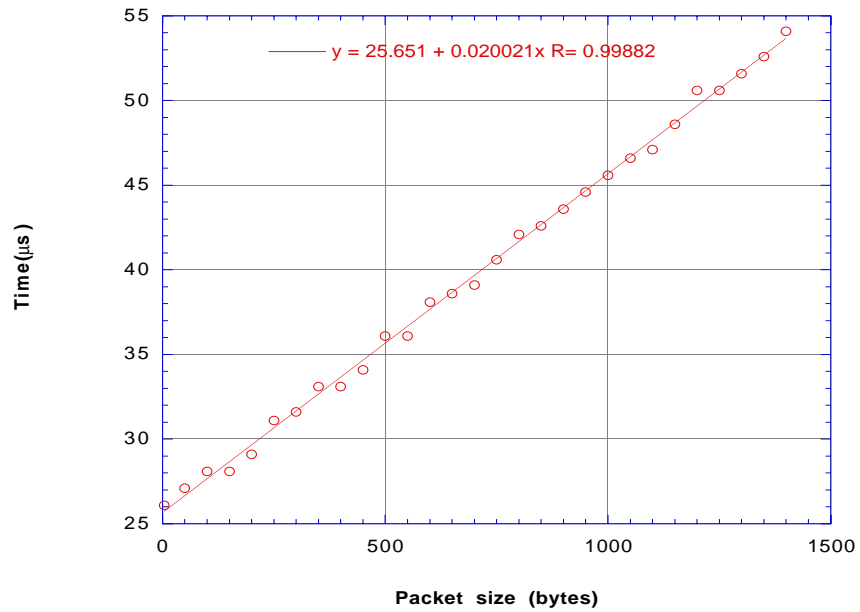
### 2.1 General

Referring to the source program diagram in Figure 2 we note that the timing calls bracket a loop over the number of packets as well as the initialization and transfer instructions. The overhead for two calls to *gettimeofday()* is 3.9  $\mu$ s and has been subtracted from all measurements. Likewise, when measuring single packet transfer times, the extra overhead for the loop over packets is subtracted as well.

### 2.2 Single packet measurements

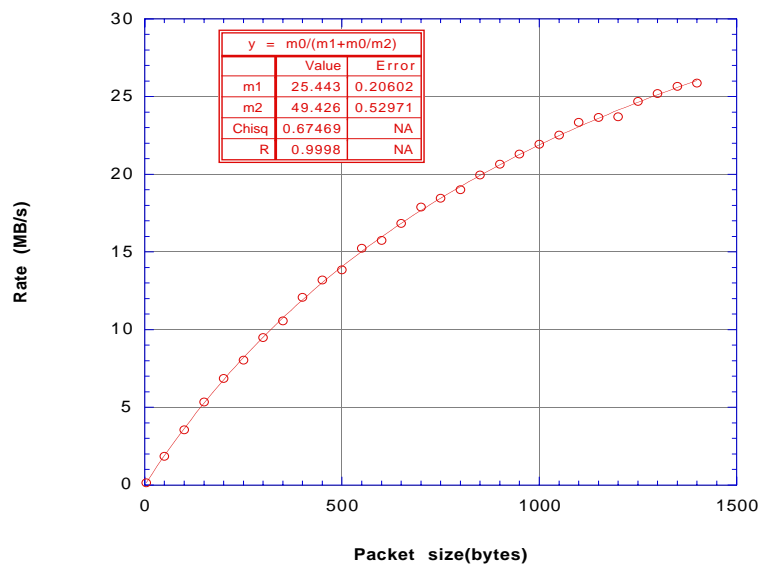
In Figure 3 is shown the transfer time for a single packet plotted as a function of the packet size. The parameters of the linear fit are given in Figure 4 on page 3 in the inset and indicate an overhead of 25.6  $\mu$ s with a slope corresponding to a transfer rate of  $\sim$ 50 MB/s. This

rate would be what could be achieved in the absence of overhead associated with the setup or correspondingly if very large packets are sent.



**FIGURE 3. : Transfer time for a single packet of data**

The transfer rate, shown in Figure 4, results in improving performance as the packet size is increased due to the presence of the unavoidable overhead.



**FIGURE 4. : Transfer rate for a single packet as a function of its size**

### 2.3 Transfer time measurements for many packets

A series of measurements were performed where the number of packets sent was varied along with the number of bytes of data in the packet. Representative data for 1, 10 and 100 packets are shown in Figure 5.

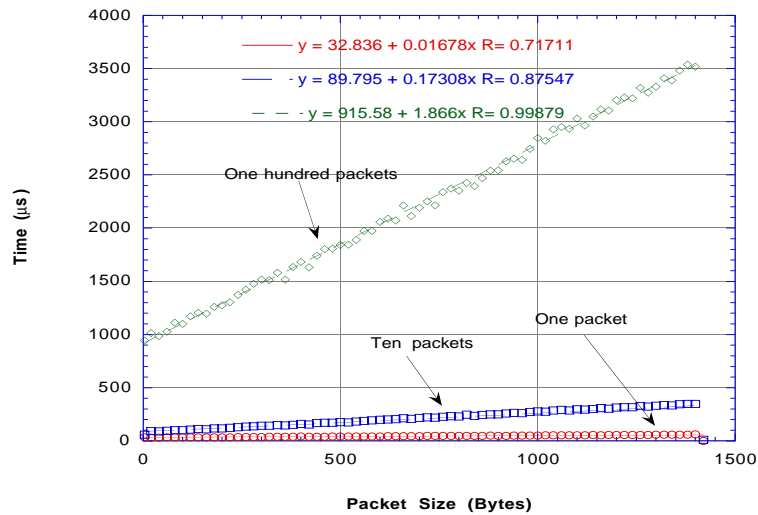


FIGURE 5. : Transfer times for 1, 10 and 100 packets of data

We can summarize the performance of the multiple packet transfer in Table 1, where we have used the linear fit parameters from Figure 5, subtracting the overheads for the timing calls. We note that transfer rates are approximately constant and that the overheads approach 7 - 8  $\mu$ s, close to the numbers for overhead per packet obtained in a previous measurement.<sup>5</sup>

TABLE 1. : Multipacket Timing performance

Number of Packets	Overhead/packet ( $\mu$ s)	Rate Limit (MB/s)
1	25	50.34
10	7.2	51.43
100	8.8	53.79

### 3.0 Measurements with a logic analyzer

As a check of our software measurements we used a logic analyzer to repeat some of our measurements. Four probes of the analyzer were attached to the four lowest order bits of the 32 bit word transferred by S-LINK. We triggered the analyzer on the start control word which has the hexadecimal representation 11111110 and then measured transfer times using the analyzer. The four low order bits of the data as they appear on the logic analyzer are shown in Figure 6. Three packets of data are shown bracketed by start and end control words. Note also the slightly longer length of the first packet relative to the others. The

single packet transfer time measured on the logic analyzer is shown in Figure 7. We note from the parameters of the fit that the overhead is  $\sim 8 \mu\text{s}$  indicating that software is contributing another  $16 \mu\text{s}$  to each transfer. We also note that the rate limit for large packets is  $50.5 \text{ MB/s}$ , consistent with our previous measurements.

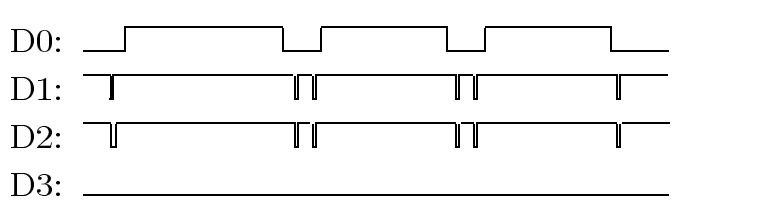


FIGURE 6. : S-LINK data as viewed from logic analyzer

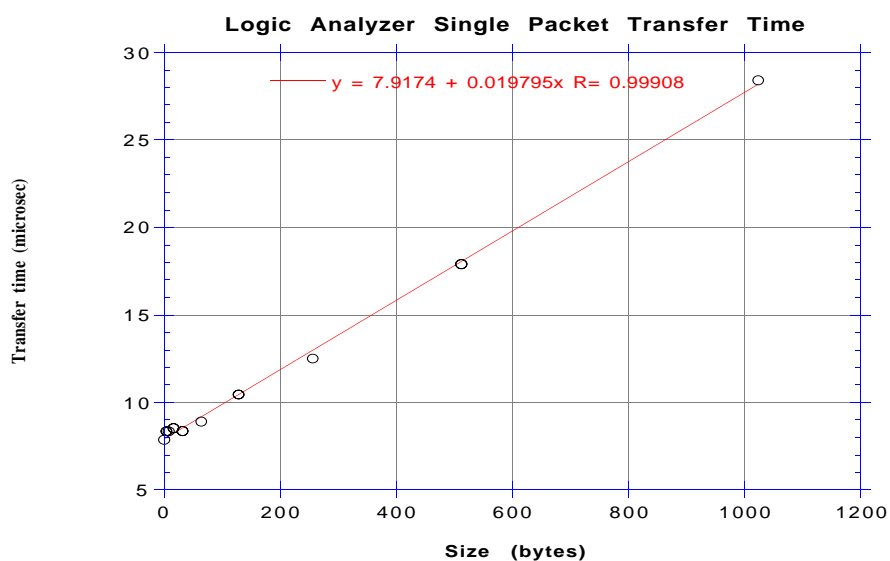


FIGURE 7. :Logic Analyzer measurement of single packet transfer time

## 4.0 Acknowledgments

We thank Jim Schlereth for his help in setting up the project and Giuseppe Mornacchi for his assistance with S-LINK software.

## 5.0 Bibliography

1. S-LINK is a CERN specification for an easy-to-use FIFO-like data-link which can be used to connect front-end to read-out at any stage in a dataflow environment. See [www.cern.ch/HSI/s-link/](http://www.cern.ch/HSI/s-link/)

2. VME mounted Power PC604 processors produced by CES Electronics Systems SA, Geneva, Switzerland.
3. The hardware was designed by John Dawson of Argonne National Accelerator Laboratory.
4. "The S-LINK~PMC library," F. Pennerath, Atlas-DAQ Note 27.  
<http://atddoc.cern.ch/Atlas/Notes/027/Note027-1.html>
5. "S-LINK Performance Measurements in the Environment of ATLAS DAQ/EF Prototype-1", M. Niculescu, ATLAS-DAQ Note 70.  
<http://atddoc.cern.ch/Atlas/Notes/070/Note070-1.html>