ZEUS-Note 95-81

# The PME FDDI-1 board [1] rate tests results

*Authors: Alexander Kalagov and Vladimir Rybnikov*

*1st July 1995*

ZEUS CDAQ/MSU Group

The FDDI node controller (DAS or SAS in a single VMEbus slot) PME FDDI-1 from Radstone has been tested to find out the maximum possible reliable data transmission rate via FDDI between the board and an INDY workstation.
The sustained reliable UDP/IP transfer rate was measured between the board and the INDY workstation (tlt106).

## 1 Test Setup

The PME FDDI-1 node controller is equipped with 8 Mbytes of multiported memory (DRAM), 128 Kbytes FDDI buffer (SRAM), 1 Mbyte processor local SRAM, on board 25 MHz SPARClite RISC processor and a FDDI controller based on the ADM Supernet-2 FDDI chipset. The PME FDDI-1 is also provided with FiberWare (the software interface between host-based higher level communication protocols, TCP/IP, and the physical layer of the FDDI controller and network.

The operating system VxWorks was running on the board, in SRAM.

An INDY SGI workstation was connected to the PME FDDI-1 via a concentrator. Data in the FDDI-1 DRAM was the source of information that should be delivered to the workstation via FDDI network using high level protocol.

## 2 Measurement stages

The measurements were carried out in four stages allowing identification of bottlenecks for obtaining maximum rate via FDDI ( max. bandwidth - 100 Mbps). All tests have been done with VxWorks executable image loaded into static memory. System memory buffer pool and application code were loaded into dynamic memory.

### 2.1 Memory-to-memory transfer rates

Memory to memory transfer tests revealed the following results:
- ( DRAM - to - DRAM), rate = 4.44 Mbyte/s;
- ( SRAM - to - DRAM), rate = 6.15 Mbyte/s.

4000 byte blocks were copied in the tests, thus simulating transferring FDDI frames. The results show that frequent usage of memory-to-memory transfer whilst preparing to transmit data will reduce the FDDI performance.

### 2.2 Low level fddi1 driver routine tests

VxWorks 5.1 BSP for PME FDDI-1 offers an on-board network interface driver with routines that can be used in applications for sending and receiving FDDI frames.

[1] Radstone Technology PLC

The rate for sending data, using UDP frames, was measured for two cases:
- data are in DRAM, rate = 2.11 Mbyte/s;
- data are in SRAM , rate = 2.42 Mbyte/s.

The rate of calls to the low level driver routines, without real data sending, was also measured:
- 4000 bytes frame, rate = 909 calls per second;
- 1000 bytes frame, rate = 2857 calls per second;
- 100 bytes frame, rate = 8000 calls per second;
- 10 bytes frame, rate = 9132 calls per second.

The results of the call rate tests revealed that the driver has a large overhead. This can be explained by the driver's implementation. Memory-to-memory copying is one of the possible bottlenecks of the driver.

### 2.3 Low level routines for UDP frame sending

Routines have been written for sending UDP frames from data in DRAM. They use neither VxWorks network services nor the fddi driver. The routines make use of direct access to the ERICs (DMA controllers on PME FDDI-1) and of FORMAC+ (Supernet-2 chip) the TAG mode for assembling a full frame "on the flight" before sending.

The rates have been measured with and without data checksumming.

- UDP frame sending without checksumming:

Table 1.

| Total data size (Mbyte) | UDP datagramm size (Bytes) | Rate (Mbyte/sec) |
|---|---|---|
| 420 | 4200 | 11.05 |
| 420 | 3000 | 10.24 |
| 420 | 2000 | 9.33 |
| 420 | 1000 | 7.78 |
| 420 | 500 | 4.94 |

- UDP frame sending with hardware checksumming (by ERIC):

Table 2.

| Total data size (Mbyte) | UDP datagramm size (Bytes) | Rate (Mbyte/sec) |
|---|---|---|
| 420 | 4200 | 9.33 |
| 420 | 3000 | 8.75 |
| 420 | 2000 | 7.78 |
| 420 | 1000 | 5.25 |
| 420 | 500 | 3.11 |

Using the UDP frame receiver on the INDY it was found that the number of lost frames, with 1000 bytes/frame, is less than 0.5%.

### 2.4 High level programs with reliable UDP

One scheme of reliable UDP has been implemented for testing reliable data (i.e. a ZEUS event) delivery. This uses additional header information and the sender waits only for confirmation of the entire event being received. If the confirmation is not received then the entire event is retransmitted. For receiving confirmations the sender uses an interrupt routine called by VxWorks on receiving a frame from FDDI.

The receiver on the SGI used standard socket library functions for communications with the PME FDDI-1. Its task was to get all the frames for one event and to store the whole event contiguously, without headers, in memory. In case of success it sent the confirmation, acknowledgement message, to the board.

The results of the rate tests are given in the table 3.

Table 3.

| Event size (Bytes) | Total data size (Mbyte) | UDP datagramm size (Bytes) | Rate (Mbyte/sec) |
|---|---|---|---|
| 102400 | 1000 | 4227 | 6.52 |
| 102400 | 1000 | 4027 | 6.56 |
| 102400 | 1000 | 3027 | 6.44 |
| 102400 | 1000 | 2027 | 5.89 |
| 102400 | 1000 | 1527 | 5.10 |
| 50000 | 500 | 4027 | 5.81 |
| 30000 | 300 | 4227 | 4.92 |
| 30000 | 300 | 4027 | 5.08 |
| 10240 | 100 | 4027 | 3.01 |

The UDP datagramm consists of the data plus 27 bytes application level header. The rate calculations take into account the user data size only.

Measurements were performed on the 28-th of June by A.Kalagov and V.Rybnikov. the sender ran on PME FDDI-1 ”rst002”, under VxWorks 5.2 loaded in SRAM, and the receiver ran on SGI ”tlt106” under IRIX 5.2.

## 3 Conclusion

The rate tests of the FDDI-1 board show that it is possible to obtain a peak throughput of about 6 Mbyte/sec via FDDI network by using the software written. This rate can be achieved under the following conditions:

- The main task of the software is to send data over FDDI.
- The software uses all the resources of the processor while sending data (no other task can run simultaneously).
- The software uses interrupt routines for receiving confirmation to provide reliable UDP.

Taking into account the conditions above it can be seen that the software designed for rate testing (especially 2.4) suits the task of the event transmission from a VME-crate to SGI by means of the PME FDDI-1.

We would like to thank S.Polenz for providing access to both hardware and documentation.