

**ISTITUTO NAZIONALE di FISICA NUCLEARE  
LABORATORI NAZIONALI DI FRASCATI**

---

**LNF-94/004 (NT)  
12 Gennaio 1994**

M. Carboni, H. Beker:

**RESULTS OF THE EVALUATION OF THE HP9000/742rt  
AND THE HP-RT OPERATING SYSTEM**

**RESULTS OF THE EVALUATION OF THE HP9000/742rt  
AND THE HP-RT OPERATING SYSTEM**

M. Carboni

INFN – Laboratori Nazionali di Frascati, P.O.Box 13, I-00044 Frascati (Roma) Italy)

H. Beker

Università di Roma I "La Sapienza", Dipartimento di Fisica Sezione INFN, P.le Aldo Moro 2,  
I-00185 Roma (Italy)

## **1 Introduction**

Modern data acquisition (DAQ) systems are based on a network of computers running specialized software to support and control all the aspects of the DAQ stages. In this framework we have evaluated a solution proposed by Hewlett-Packard based around the HP9000/742rt processor operating in the VME environment.

Hewlett Packard has friendly provided us a complete system formed by a Host System and a Target System. In the present note we report on the results of tests related to the hardware configuration and to the Operating System; these results are of interest for the KLOE experiment in view of a possible usage of real time VME systems in the DAQ structure. We plan to perform similar tests on other VME based real time systems (e.g. Motorola, CES, DEC Alpha in VME, etc.)

## 2 System configuration

The hardware consists of HP9000/747i, running HP-UX operating system and a target HP9000/742rt running HP-RT v1.1 target system. The relationship between these two systems is shown in Fig.1.

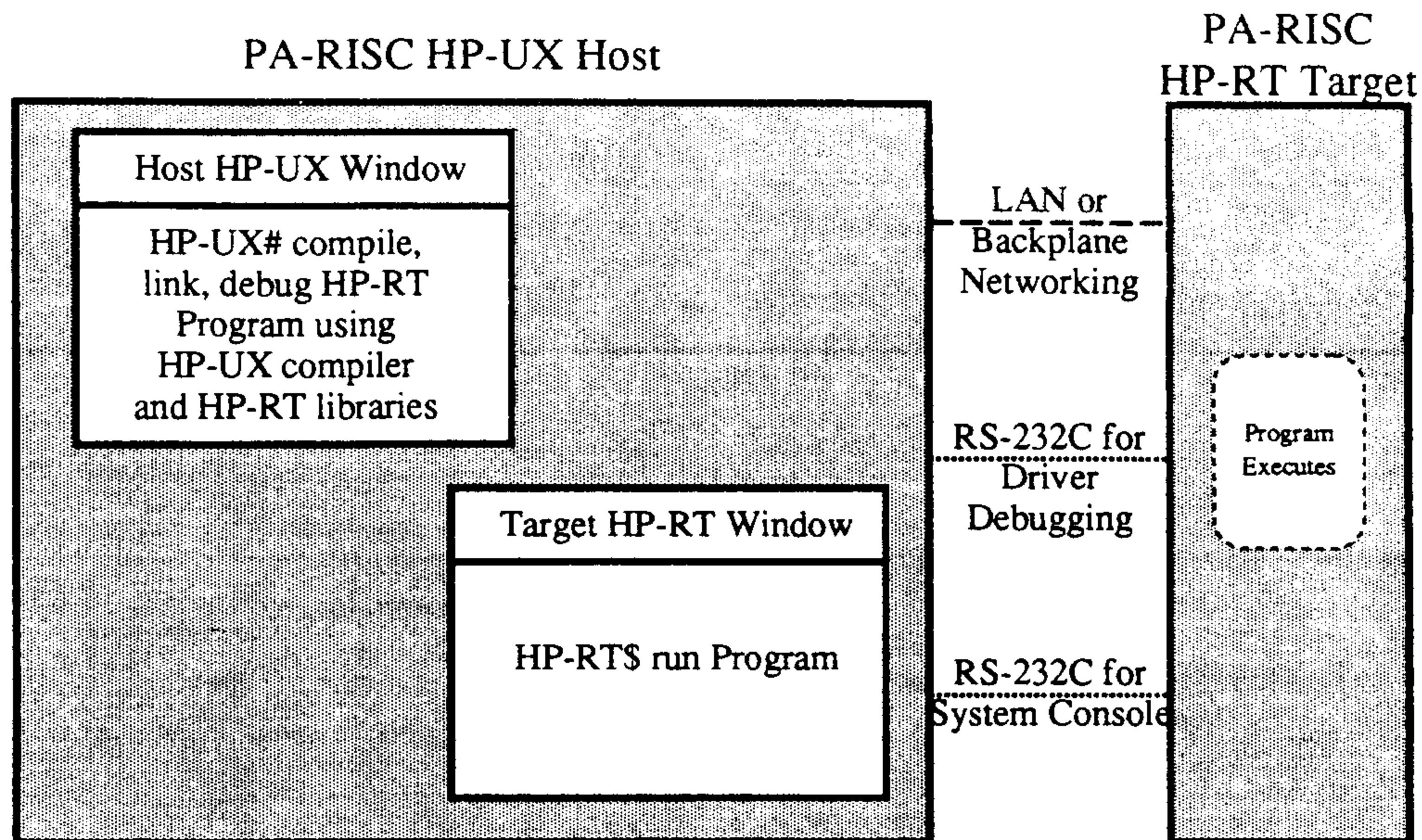


Figure 1: Relationship Between the HP-UX Host System and the HP-RT Target System

- |                      |   |
|----------------------|---|
| <b>Host system</b>   | CPU: PA-RISC 7100, 64kByte cache, clock frequency 50 MHz, 48 MB of RAM, Ethernet, 2 RS232, external SCSI, 2 EISA slots and 6 VME slots.   |
| <b>Target system</b> | the Real Time system is plugged inside the HOST cabinet using 2 VME slots. It offers 2 serial lines, SCSI (not used by us), Ethernet, VME back plane master/slave interface. The processor is identical to the one of the host system. Hence the following benchmarks directly compare the operating system performance. The card we tested was equipped with 16 MB of main memory. |

### 2.1 General remarks on hardware of the HP9000/742rt

An appealing aspect of the board is the processor power as reported in the available benchmarks.<sup>1</sup>

The card offers the standard periphery of most VME processor boards. This, however does not include a VSB interface. On the VME side it offers a full master/slave interface

<sup>1</sup>Overall rating 62 MIPS 13 MFLOPS, 36 SPEC int92, 72 SPEC fp92

| 1 VME cycle per transfer A24/D32              |            |            |
|---|------------|------------|
| CPU register → VME                            | 5.48 MB/s  | write      |
| VME → CPU register                            | 3.81 MB/s  | read       |
| VME → CPU memory                              | 3.78 MB/s  | read       |
| 2 VME cycles per transfer                     |            |            |
| VME → VME                                     | 2.27 MB/s  | read/write |
| for comparison block moves in on-board memory |            |            |
| memory → memory                               | 24.24 MB/s | read/write |
| cache → cache                                 | 38.10 MB/s | read/write |

Table 1: *Data move throughputs*

with a programmable gather/scatter map. Mapping the full VME space leads to a degraded master mode which we have not tested. The results presented here were obtained by mapping only the 16 MByte of the A24 standard address space. Realistic applications can avoid degraded mode. The mapping and AM code handling has to be decided at system creation. Any change in this mapping needed a new kernel generation and hence a reboot.

The fastest way to access VME is obtained by using **global shared** memory (POSIX). This gives access to hardware directly in the user tasks without the need to write drivers.

The software provided with the board includes a **VME library access routines**; this library was not evaluated due to the satisfactory access to the VME provided by the global shared memory.

Only devices producing interrupts on VME need a minimal driver. Drivers have directly mapped access to VME.

For testing the VME master interface we used a CES-FIC 8230 as a slave. Since the on-board memory of this card is DRAM and dual ported, part of the access time measured is due to the slave itself.

## 2.2 Limitations of the HP9000/742rt

The module is not able to perform VME block transfers as a master, D64 cycles and does not have an independent DMA channel. All these missing features are promised for the next version of the board, the 743rt. They are absolutely necessary if we were to use the card as the primary data mover. A VSB master interface is currently not planned by HP.

We had no means to test the slave performance of the card in the absence of a separate fast master. The data sheet of the new board quotes speeds up to 35 Mbyte/s employing

|                                  | HP-RT on the 742rt | HP-UX on 747   |
|----------------------------------|--------------------|----------------|
| Process creation                 | 370 forks/s        | 116 forks/s    |
| Thread creation                  | 1388 threads/s     | not applicable |
| Context switching (of threads)   | 90.000 switches/s  | not applicable |
| Signal intercepts (of processes) | 5000 signals/s     | 2850 signals/s |

Table 2: OS performance evaluation

D64 block move transfers.

### 3 Evaluation of the operating system

HP-RT seems to be a straightforward port of LynxOs to the HP RISC architecture. It has the same POSIX standard real time functions (global shared memory, semaphores, threads etc.)

HP-RT lacks of the compiler and linker and hence needs an independent HP-UX development system. The HP-UX system usually will serve also as the file server even though it is possible to connect a disk directly to the real time system. The connection between the host and the target is performed either through Ethernet or VME (both boot and file server).

Ethernet bandwidth is sufficient in a typical data acquisition system because once the system is booted from Ethernet it can run from a RAM disk (4 MByte) creating no further traffic on the network.

The HP-UX and HP-RT systems are not image compatible and applications running on one system have to be relinked with different runtime libraries to run on the other. Relinking and compilation at present can be performed only on the host system (HP-UX). It seems that if the compiler (preprocessor+compiler+assembler) and linker were available under HP-RT it could be used as a stand alone development system as is the case of other LynxOs systems.

In Tab.2 we quote some real time relevant operating system performance figures which, for the moment, we compare only with HP-UX performance.

### 4 Network performance evaluation

We limited ourselves to test the CPU usage during TCP transfers over Ethernet and memory to memory on the same machine. In the future we plan to measure throughputs on the FDDI backbone, via a FDDI interface by Rockwell made available to us on loan. These numbers can be used as an upper limit for the performance to be expected, taking into account that the physical limit of FDDI is 12.5 MB/s. We used the same bench-marks

presented earlier [1].

We performed the tests both on the HP-UX system and the HP-RT system with identical CPUs so the test compares the high level drivers on these two operating systems. In all cases we chose a block size of 4096 which matches the FDDI frame size and a TCP window size of 56 kByte which is the maximum allowed by HP [2].

| Memory to memory    | HP-RT → HP-RT | HP-UX → HP-UX |
|---------------------|---------------|---------------|
| Sender              | 7.8 MB s      | 36.3 MB s     |
| Receiver            | 17.8 MB s     | 13.3 MB s     |
| Real transfer speed | 3.5 MB s      | 9.8 MB s      |
| CPU usage           | 63 %          | 100 %         |

**Table 3:** *Data transfer on same node, memory to memory, TCP*

In table 3 we notice that on the UX and the RT systems the sharing of CPU time between the sender and the receiver is inverted. Unfortunately the RT system is slower in sending. A possible reason is that the HP-RT system does not use the whole available CPU time in transfers from task to task on the same machine.

In table 4 we present the results obtained across Ethernet; where "Real transfer speed" is the real throughput into the network.

|                     | RT → UX     | UX → RT     |
|---------------------|-------------|-------------|
| Sender              | 5.1 MB s    | 8.3 MB s    |
| Receiver            | 4.8 MB s    | 11.5 MB s   |
| Real transfer speed | 615 kByte/s | 500 kByte/s |
| CPU usage sender    | 12%         | 7.4%        |
| CPU usage receiver  | 13%         | 5.3%        |

**Table 4:** *Data transfer throughput on Ethernet TCP*

Even though above numbers are not quite consistent it is reasonable to assume a maximum writing speed using TCP protocols on FDDI below 6 MByte/s whatever the hardware and low level driver performance is. The receiving speed (of little interest for our purpose) might be slightly higher. These numbers are a bit better than the assured throughput by Rockwell (35 Mbit/s).

It must be noted that the previous figures are obtained without other tasks competing for the CPU and absolutely no VME bus contention. In a demanding environment, as the KLOE experiment, the indication is that, apart from finding better hardware, we would

either have to abandon the TCP protocol or improve the high level driver significantly. We started this work and improved the performance by 30% by expanding the window size to 56 kByte vs the 8 kByte standard.

In tables 5 and 6 we repeat the comparison for the UDP protocol:

| Memory to memory     | HP-RT → HP-RT | HP-UX → HP-UX |
|----------------------|---------------|---------------|
| Sender               | 26.0 MB s     | 12.0 MB s     |
| Receiver             | 11.0 MB s     | 19.0 MB s     |
| Real transfer speed  | 4.8 MB s      | 7.2 MB s      |
| CPU usage            | 61%           | 100%          |
| Protocol Reliability | 98%           | 100%          |

**Table 5:** *Data transfer on same node, memory to memory, UDP*

|                     | RT → UX           | UX → RT     |
|---------------------|-------------------|-------------|
| Sender              | unmeasurably high | 22.0 MB s   |
| Receiver            | 8.0 MB s          | 80.0 MB s   |
| Real transfer speed | 800 kByte/s       | 800 kByte/s |
| CPU usage sender    | seems 0%          | 3.6%        |
| CPU usage receiver  | 10%               | 1%          |

**Table 6:** *Data transfer throughput on Ethernet UDP*

We must note that the UDP protocol is not completely reliable.

## 5 Conclusions

The board in its present form has some serious limitations in its raw VME bandwidth. However, the performance quoted for the next version of the board (743rt) will overcome this aspects.

Another field of concern is the performance of the high level network driver which gives transfer rates at the limit of requirements of the KLOE experiment already for transfers from task to task on the same processor.

The most appealing aspect of the board is the processing speed on the high end of the competition.

The operating system apparently offers all the services we require. With some knowledge of UNIX it is very straightforward to master. Nevertheless, a unified development/target system (i.e. the implementation of the compiler, linker and debugger under HP-RT) could enhance the efficiency in the system programming.

## Acknowledgements

We would like to thank HP Rome, and in particular Ing. I.Penzo for having us provided with the necessary hardware and software. Our thanks go also to Ing. C.Arata, HP Torino, for his valuable help in getting the system going.

## References

- [1] Performances of the UDP and TCP protocols using the FDDI and ETHERNET channels on several RISC machines, M.Carboni et al.,KLOE NOTE 93-61.
- [2] High performance TCP/IP and UDP/IP Networking in DEC OSF/1 for AlphaAXP, C.Chang et. al., Digital Technical Journal Vol. 9 No.1, Winter 93.