8

KEK Preprint 92-134
October 1992
H/D

POSIX Realtime Extension and RISC/UNIX Data Acquisition

Y. YASU, H. FUJII, M. NOMACHI, H. KODAMA, E. INOUE, S. ICHII,
A. MANABE, Y. WATASE and Y. TAKEUCHI

*Contributed to the Computing in High Energy Physics (CHEP92),
Annecy, France, September 21-25, 1992.*

National Laboratory for High Energy Physics, 1992

KEK Reports are available from:

Technical Information & Library
National Laboratory for High Energy Physics
1-1 Oho, Tsukuba-shi
Ibaraki-ken, 305
JAPAN

Phone:	0298-64-1171	
Telex:	3652-534	(Domestic)
	(0)3652-534	(International)
Fax:	0298-64-4604	
Cable:	KEKOHO	

POSIX Realtime Extension and RISC/UNIX Data Acquisition

Yoshiji Yasu, Hirofumi Fujii, Masaharu Nomachi,Hideyo Kodama, Eiji Inoue
Shingo Ichii, Atsushi Manabe and Yoshiyuki Watase
National Laboratory for High Energy Physics

Yasuo Takeuchi
Department of Physics, Tokyo Institute of Technology

We have evaluated DEC OSF/1 Realtime Extension, which is based on POSIX Realtime Extension(P1003.4). The main elements of DAQ, which are device driver for read-out (CAMAC device driver) and buffer manager, have been developed. The real time extension , particularly, the scheduler is important feature on the RISC/UNIX for real time data acquisition. We can show that RISC/UNIX Data Acquisition has wonderful read-out capability and the POSIX Realtime Extension has powerful features.

Introduction

It has been shown that RISC/UNIX system has a possibility to be used for data acquisition system[1,2]. But, standard UNIX operating system doesn't have preemptive scheduling,which is required for real time system. POSIX Realtime Extension may satisfy such requirements of real time data acquisition for high energy physics experiments.

Hardware of data acquisition system at KEK

Figure 1 shows block diagram of the hardware elements in the data acquisition system. We support DECstation with TURBOchannel and SPARCstation with SBus as the data acquisition system. We also adopted VMEbus as the standard Electronics interface. Kinetic VME-CAMAC interface is selected while the CAMAC controller was used for VAX/VMS data acquisition system[3]. The hardware specifications are summarized in table 1. Those hardwares are used for measurements of performance of CAMAC functions.

Basic VME performance on DECstation and SPARCstation

The performance of programmed I/O to a VME module is important because it is frequently used for data acquisition system. In table 1, the specification of VMEbus Adaptors is shown. Performances measured in our setup are shown in table 2.

CAMAC functions

A CAMAC device driver and a CAMAC library are developed for DECstation and SPARCstation[4,5]. CAMAC single read/write functions and CAMAC block transfer functions are supported. LAM handling is also supported. In addition to them, KEK list processing and Kinetic list processing are supported. They reduce overhead of system call by executing the series of CAMAC operations(List Processing) in the device driver. The main CAMAC functions supported in the library are summarized in table 3. The performance of the CAMAC functions are shown in table 4. The performances on MicroVAXII[3] are also shown for their comparison.

DEC OSF/1 Realtime Extension

Processes on UNIX time-sharing system are sensitive to other processes on same system and the maximum expected process time can not be defined. It is not good feature for real time system. One of ideas to avoid it is that the device driver executes most critical part of the data acquisition system. But, all of the critical processing can not be done in the device driver and the implementation of the job to the device driver reduces the flexibility of the data acquisition system. If the operating system becomes preemptive and has real time scheduling policy, one can control the priority of the data acquisition processing. It is better for all of high energy physics experiments. POSIX Realtime Extension is expected to solve such the problems. We could evaluate the real time extension by using DEC OSF/1 Realtime Extension and the result shows that the real time extension is powerful for real time systems.

DEC OSF/1 Realtime Extension supports the functions of P1003.4/D10 except some functions like "real time files". We believe that the essential feature is that the kernel becomes preemptive and has FIFO scheduling policy with real time priority. The programming with asynchronous I/O was frequently done on VAX/VMS data acquisition system. Therefore, we also evaluated the asynchronous I/O function of the real time extension.

Evaluation of the Scheduler

We used a buffer manager called NOVA[6] to evaluate the scheduler. The buffer manager consists of some daemons and clients (collector, recorder, analyzer and so on), which communicate each other by using UNIX MESSAGE QUEUE. The performance measured on three cases of process configuration. Those processes run on ULTRIX V4.2A (with timesharing), DEC OSF/1 V1.0 with fixed priority = 19(all other process are also same priority) and DEC OSF/1 V1.0 with fixed priority = 63(highest real time priority). In one case, only a collector process exists as a client process. The collector process sends a message and the buffer manager receives the message. The message includes data buffer address pointer. Then, the buffer manager sends a message to the collector and these processes are repeated. In another case, a recorder process is added to the previous case. The data buffer information is passed from the collector process to the recorder process. In other case, there are a collector process, a recorder process and a analyzer process. The data buffer information is passed from the collector process to the analyzer process via the recorder process. We measured the round trip time. First, the processes run in a environment with no heavy I/O processes. Next, with 2 heavy I/O processes. The heavy I/O process is simple shell script (% ls -R /). The collector process, the recorder process and the analyzer process have no data processing except NOVA send/receive functions.The results are shown in table 5.

In a environment with no heavy process, the round trip time on ULTRIX V4.2A is faster than that on DEC OSF/1 Realtime Extension. Although the machines are different, we presume that ULTRIX V4.2A is more effective than DEC OSF/1. In a environment with 2 heavy processes, the round trip time on ULTRIX V4.2A (time-sharing) and DEC OSF/1 with normal fixed priority are postponed by the heavy I/O processes while the round trip times on DEC OSF/1 with highest real time priority are almost same time with and without the heavy I/O processes. The result emphasizes that POSIX Realtime Extension is powerful.

Conclusion

A RISC/UNIX Data Acquisition is established. A CAMAC device driver and a CAMAC library are developed. Other tools like buffer manager are also developed. It is shown that POSIX Realtime Extension is very powerful for real time systems. It makes possible to have more flexible priority control on the data acquisition system and then it will satisfy the requirements for high energy physics experiments.

Acknowledgement

The authors would like to thank people of DEC Japan for their helpful supports.

Trademarks

References

[1] M.Nomachi et al., Proceedings of the "Computing in High Energy Physics 1991", editted by Y.Watase and F.Abe , Universal Academy Press, p681 (1991).
[2] Y.Takeuchi, T.Tanimori and Y.Yasu, Development of Data Acquisition System using RISC/UNIX(TM) Workstation, KEK Preprint 92-80 and TIT-HEP-92-05 August 1992
[3] Y.Yasu and M.Nomachi, CAMAC Device Driver on VAX/VMS, KEK Internal 90-2, April 1990
[4] Y.Yasu et al., CAMAC Device Driver for DECstation with TURBOchannel VME adaptor - Is UNIX Real-Time Data Acquisition system possible?-(in Japanese), KEK Internal 91-7, December 1991
[5] Y.Takeuchi, Development of UNIX-VME High Speed Data Acquisition System (in Japanese), Master thesis(unpublished), January 1992
[6] M. Nomachi et al., A buffer manager for SDC Portable Data Acquisition System, to be published(1992)

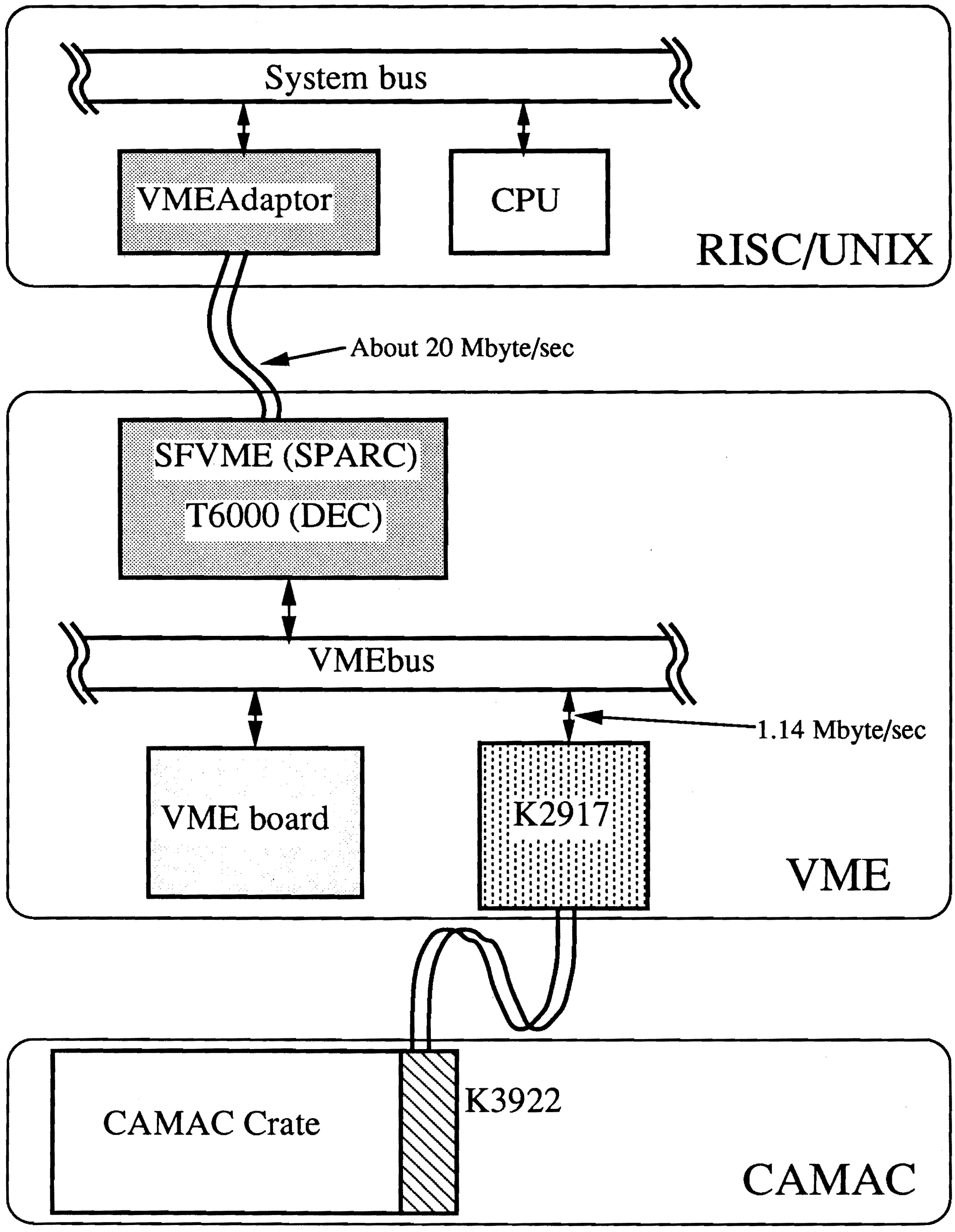

Fig.1 RISC/UNIX-VME-CAMAC SYSTEM

Table 1. Specification of Workstation used for measurement

Computer	SPARC IPC	SPARC IPX	DEC 5000/125	DEC 5000/240
Processor MIPS * Main Memory	SPARC-25MHz 15.8 12Mbyte	SPARC-40MHz 28.5 32Mbyte	R3000-25MHz 26.8 48Mbyte	R3000-40MHz 42.9 48Mbyte
OS I/O bus	SunOS 4.1.1-JLE1.1.1 SBus		ULTRIX V4.2A TURBOchannel	
Computer VME I/F	Solflower SFVME-100		DEC PMABV/T6000	
PIO-write * PIO-read * DMA-write* DMA-read * Max. Cable Length	— — 20MB/S 20MB/S 0.8m (32inch)		6MB/S 3MB/S 21MB/S 17MB/S 3m (9.4ft)	
VME-CAMAC I/F	Kinetic model 2917			
DMA-write* DMA-read *	1.12MB/S 1.14MB/S			

* Catalog value.
The direction of "read" is from VME to host memory.
The direction of "write" is from host memory to VME.

Table 2. Performace of Programmed I/O

	SPARC IPC	SPARC IPX	DECS 5000/125	DECS 5000/240
Read Mbyte/sec	3.2	3.1	1.2	1.5
Write Mbyte/sec	5.4	6.3	3.8	5.0

* VME Memory Board : MVME-224A-1

Table 3. Main CAMAC Functions

FUNCTION TYPE	FUNCTION	CONTENTS
SETUP	CAMOPN CAMCLS	Open CAMAC Device Driver Close CAMAC Device Driver
CAMAC	CGENZ CGENC	Initialize CAMAC Modules Clear CAMAC Modules
CAMAC SINGLE ACTION	CAMAC CAMACW	Execute CAMAC Single Data transfer with 24 bit width Execute CAMAC Single Data transfer with 16 bit width
CAMAC BLOCK ACTION	CDMAL CDMAW	Execute CAMAC Block Data transfer with 24 bit width Execute CAMAC Block Data transfer with 16 bit width
INTERRUPT HANDLING	CELAM CDLAM CWLAM	Enable Interrupt Disable Interrupt Wait for an Interrupt
LIST PROCESSING	CEWAI CKADD CKEXE	Execute KEK List Processing Add Kinetic List Execute Kinetic List Processing

CAMAC LIST COMMAND	GENC	Clear CAMAC Modules
	READ WRITE NDT	Execute CAMAC Single Read Action Execute CAMAC Single Write Action Execute CAMAC Single Action without Data
	IGQ QS AS	Execute Block Data transfer with 16 bit width in IGNORE mode Execute Block Data transfer with 16 bit width in Q-STOP mode Execute Block Data transfer with 16 bit width in Q-SCAN mode
	WAITINT	Wait for an Interrupt
	MOVE MUL SUB BR BEQ	Copy data on Memory of Computer Multiple data on Memory of Computer Subtract data on Memory of Computer Branch Specified Label Branch Conditionally

Table 4. Performance of Main CAMAC Functions

System Call Function	SPARC IPC	SPARC IPX	DECS 5000/125	DECS 5000/240	μVAX II
CAMACW					
read μsec	181	130	139	91	1100
write μsec	181	131	136	73	
CDMAW					
overhead μsec	1210	702	770	392	2000
read Mbyte/sec	0.917	0.939	0.909	0.922	1.2
write Mbyte/sec	0.791	0.813	0.435	0.533	—
CWLAM					
latency μsec	240	153	122	93	1000
CEWAI					
overhead μsec	179	117	138	78	3200

List Processing Command	SPARC IPC	SPARC IPX	DEC 5000/125	DECS 5000/240	μVAX II
READ μsec	27	26	36	29	32
WRITE μsec	29	26	30	24	31
NDT μsec	20	19	23	19	26
IGQ					
overhead μsec	820	464	494	325	400
WAITINT latency μsec	187	150	113	80	160-190
MOVE μsec	3	3	2	2	10

Table 5. Evaluation of Scheduler

Unit = MilliSec.

A) Only Collector

	No Heavy Process	2Heavy Processes
ULTRIX V4.2A (Timesharing) DECStation5000/125	1.1	2.5
DEC OSF/1 V1.0 Fixed Priority=19 DECStation5000/200	2.0	4.1
DEC OSF/1 V1.0 Fixed Priority=63 DECStation5000/200	1.5	1.5

B) Collector & Recorder

	No Heavy Process	2Heavy Processes
ULTRIX V4.2A (Timesharing) DECStation5000/125	2.4	5.0
DEC OSF/1 V1.0 Fixed Priority=19 DECStation5000/200	4.2	8.7
DEC OSF/1 V1.0 Fixed Priority=63 DECStation5000/200	3.2	3.3

C) Collector & Recorder & Analyzer

	No Heavy Process	2Heavy Processes
ULTRIX V4.2A (Timesharing) DECStation5000/125	3.8	7.6
DEC OSF/1 V1.0 Fixed Priority=19 DECStation5000/200	6.7	12.8
DEC OSF/1 V1.0 Fixed Priority=63 DECStation5000/200	4.7	5.0