REAL-TIME CONTROL WITH VMEBUS MULTI-MASTER CONFIGURATION

Toru Ohata, SPring-8, Hyogo 679-5198, Japan

Abstract
A hard real-time performance was obtained by using general-purpose CPU board with generic operating system (OS) under the VMEbus multi-master configuration. Achieved real-time performance is close to the hard real-time. The VMEbus has bus arbitration and can make a configuration of multiple bus masters on a single backplane. We use the master controller to communicate with upper control layer on a distributed control system. And other masters process own real-time applications. Communication between masters is made via backplane. We can easily make a special-purpose controller with real-time properties by software, which substitutes for a custom-made electronics. And recently algorithm reconfigurable devices such as FPGA/PLD may be used. It is the most attractive that seamless development environment is acquired by familiar software on a generic OS. This paper reports the evaluation results of the real-time characteristics under VMEbus multi-master configuration. And we show an example of the real-time application, software-based pattern generator, which controls electromagnet power supply of insertion device at SPring-8.

INTRODUCTION
In SPring-8, we built the original control system named MADOCA (Message And Database Oriented Control Architecture) [1]. The MADOCA is a flexible software framework, which supports many UNIX-based platform. The Solaris had been adopted as one of the standard operating system for the MADOCA [2]. The Solaris has high compatibility of UNIX and POSIX function calls. We have many off the shelf Intel-architecture (IA) products running sufficiently performance on the Solaris. The MADOCA needs the fixed priority control of user processes on the front-end system in order to realize both controls and data acquisitions at a high degree. Most of UNIX OS are lacking of that real-time characteristics, however the Solaris gives powerful real-time extension based on POSIX 1003.1b [3] as the RT scheduling class.

Real-time systems are typically categorized into two classes: hard and soft. In a hard real-time system the time deadlines must be strict. The timing constraints in a soft real-time system are not as stringent. On standard UNIX system, a time for the context switching is nearly 10 milliseconds. The worst-case timing of a response is not guaranteed. The RT class of the Solaris has the real-time characteristics, which are close to the hard real-time.

Recently, there are some proposals about algorithm reconfigurable devices (FPGA/PLD) for real-time control system [4]. A general-purpose CPU operated by OS is inferior in an absolute real-time performance as compared with dedicated FPGA/PLD based system. However, it is important that not only the seamless and flexible development environment but also accustomed one can be introduced. Moreover, high abstraction of hardware is possible only by a generic OS.

It is difficult to keep controls in real-time under any load for generic OS, because deterioration of real-time performance by the control command from upper control layer is unavoidable in the case of a single controller. The VMEbus multi-master configuration is possible to aim the real-time performance of a sub-millisecond order. The VMEbus, IEEE1014, is widely used bus architecture in the control system. It supports several controllers as a bus master on the single backplane. In this paper, the advantage of the multi-master configuration for the real-time application and an example with dual-CPU configuration are discussed.

MEASUREMENTS
To estimate the real-time performance, time jitters have been measured on several conditions with single/multiple master configurations on the VMEbus. In order to investigate the limits of the real-time performance, the same measurement has been done with out-of-date CPU. In addition, a SMP machine has been tested preliminarily for the future choice.

Single/multiple master configuration
The Solaris 8 and Advme8001 VMEbus board computer [5] were used for the measurement. The Advme8001 has a Pentium3 600MHz and 128MB main memory. The cycle of timer interruption of the Solaris was set to 10000Hz. Thus the interval of the timer interruption, TICK, became 100 microseconds. The time jitters were measured by monitoring the output of time-controlled square-wave signal to the serial port. The result of the monitoring with an oscilloscope is shown in Figure 1: A square-wave signals from the Solaris.
The interruption thread has one of the highest priority levels. Even the network interface activity does not significantly distort the interrupt thread. The scheduler does not influence the interruption thread as well. Only the highest interruption of hardware I/O preempts this thread.

Figure 2 shows the results of the time jitters measurement. The upper plots in each graph show profiles without load. As shown in figure 2-a), the distribution of the time jitters is small for no load case. The discrete time structure of TICK intervals is seen. The plot with load shows the wider distribution and about 7% of the events exceed 2milliseconds. In case without load, the result of RT class is similar to that of TS class as shown in figure 2-b). And also we can see the sharp profile of time jitters with load case. It seems deadlines are guaranteed. It may have enough performance for some application. Figure 2-c) shows real-time characteristics of the kernel thread application. The critical section loaded into the kernel space as a driver module. The timer interruption on OS influenced to the kernel thread. About 4% of the events exceed 2milliseconds. The overhead of the context switch between user space and kernel space is taken comparing with the difference between graph a) and c). As shown in figure 2-d), the best real-time performance is obtained by using the interruption thread as compared with others at any conditions. However, the process switch by the scheduler is not expectable, cooperated operations of applications cannot be performed. This means that there is no controllability from network. The multiple master configuration of VMEbus is a solution to solve above problem.

On the slow CPU

To investigate the limits of the real-time performance slower CPU, GMS V155 Pentium 233MHz (MMX), was used for comparison. Figure 3 shows the measurements of time jitters in the interruption thread. They are distributed within range of 500 microseconds and maximum delay is at 1.5 milliseconds. In comparison with Pentium3 600MHz, average deviation becomes 20% large. However the real-time performance is preserved even on the slow CPU. The result shows the real-time performance was not influenced in the difference of CPU clock so much.
The SMP environment

For the sake of future choice, the SMP machine, Dell Power Edge 1300, was tested. It consists of dual Pentium3 500MHz, and 512MB main memory. As shown in figure 4, highly real-time performance was obtained in the interruption thread. The SMP can be considered suitable machine for real-time control. However, some tuning was needed to obtain supreme real-time performance. In usual OS, an interruption thread always preempts a low priority thread. On the contrary, the Solaris can prevent an interruption thread from preemptiong a low priority thread, since a specified process binds tightly to the specified processor set. It can disable interrupts from I/O devices for the processor set. Thus the real-time performance and the controllability near the multiple master configurations were obtained on the SMP machine.

It will be most interesting in a trend of technologies around CPU. Hyper-Threading technology [6] (HT) and multiple core technology [7] on one silicon chip have potential to change the world of the real-time control. Parallel processing is the key feature of both technologies, which are recognized as a SMP machine by OS. These CPU will bring similar result of the SMP.

APPLICATION

By using dual CPU configuration on VMEbus, a Software-based Pattern Generator (SPG) was made for the insertion device ID17 [8] in SPring-8. The SPG changes a polarization of the synchrotron radiation x-ray dynamically by controlling pattern data to magnet power supplies. Target interval of pattern operation is an order of 100 microseconds. A set of patterns is alternating with a few hertz. One CPU processes the real-time pattern operation. It generates pattern voltage values and set them to an analog output board sequentially at the constant interval. The other CPU acts as a server processor for the upper control layer. A SRAM board on the VMEbus is used as a shared memory to communicate between CPUs. An example of pattern signals is shown in figure 5. The pattern contains 8192 points of data with 100 microseconds interval. The SPG supports 32 channels output simultaneously.

CONCLUSIONS

By using general-purpose CPU board with generic operating system, a hard real-time performance was obtained. It is difficult to take hard real-time performance by using single CPU. As one solution, the VMEbus multi-master configuration in the single backplane was proposed. To evaluate real-time performance of the system, time jitters were measured on several environments of single/multiple master configurations. The results show enough real-time performance was acquired on the multi-master configuration with interruption thread application. Not only the Solaris but also a generic OS can use the multi-master configuration. It is good solution to make a real-time control system easily. In addition, the SMP environment also has good real-time performance. New type of technologies around CPU will change the style of real-time control in the future.

REFERENCES

[6] Intel
[7] IBM, Sun Microsystems and Intel