CONTRIBUTION TO EFFICIENCY OF GPIB INSTRUMENTATION

Iveta Ondrášová, Viktor Smieško, Vít Setnička

Department of Measurement, Faculty of Electrical Engineering and Information Technology,
Ilkovičova 3, 812 19 Bratislava, Slovakia
Phone: +421 7 65 42 96 00, e-mail: iveta@pluto.elf.stuba.sk

The article deals with some aspects of evaluation of GPIB (IEEE488) systems performance. Several ways how to increase the system performance are discussed. An example of design of a measurement system used in technical diagnostics is given. Two possible approaches are presented.

1 Introduction

There has been very little research published in the area of software performance testing althoug performance is a problem in many software development projects. WOSP'98 [1] was motivated by a desire to pull together the scattered effort being made, to create a methodology for tackling this problem.

IEEE 488 (GPIB) bus is today a world-known standard, an easy-to-use and time proven interface for automated measurement systems, offering so the widest selection of measurement devices. As in many of these applications the system performance is a crucial parameter, it is necessary to look for possible ways to increase GPIB system performance [6], [7].

2 Possibilities for Increasing GPIB System Performance

The following solutions span both hardware and software options. These can be used either together or independently.

a) Use of Bus-mastering GPIB Controllers for PCI instead of ISA

ISA – based GPIB Controller (e.g. GPIB PCII [9]) has no bus-master capability, i.e. data transfer interrupts the microprocessor activity which causes stopping all processing tasks. PCI peripheral boards (e.g. PCI-GPIB) have their own onboard DMA controller enabling data transfer directly to PC memory. They can take control of the PCI bus and transfer data more efficiently - independently from the microprocessor in the system.

b) *Use of HS488 Handshake Protocol*

The HS488 handshake protocol [9] can accelerate GPIB transfer up to 8 Mbytes/s using standard GPIB cables. HS-488 is implemented in the GPIB controller hardware. Therefore no application software changes are necessary. Both high-speed and standard IEEE 488 instruments can coexist in the same systems. PCI-GPIB controller can use the HS488 protocol, unlike ISA controller GPIB PCII.

c) Use of Multi-board GPIB

Most GPIB systems have a computer equipped with a single GPIB card controller that is designated as the active GPIB controller. One computer may control more than one GPIB bus by using more than one controller card, but only one GPIB controller at a time may be the active one. A multi-board system allows the designer to operate GPIB subsystems simultaneously, and thereby increase the throughput of the whole system.

d) Use of GPIB Instruments with Optimised Measurement and Command Parsing Capabilities and Use of Proper Configuration and Trigger Programming Techniques

Not many instrument vendors offer any specifications of time / performance parameters. Then it is necessary to experiment with different instruments in different modes when configuring instruments and acquiring measurement which would ensure the fastest possible measurement performance. Depending on the application, any of the time components can play more or less significant role.

The particular times may be characterized by a GPIB line analyzer or by using GPIB diagnostic software (e.g. National Instrument′s GPIB Spy software [9]) or by special time-getting functions or tools of the development software (e.g. profiler in Visual C++).

Where measurement times and throughput considerations have a substantial impact on the design, the GPIB instrument performance specification requires careful review and sometimes even technical support from the instrument manufacturer.

An approach to modelling AMS performance is given in [2], [3] and some hints for performance improvement in [4].

e) Use of Data Processing in the PC

In general, processing capabilities of the PC advance more quickly than instrument capabilities. To ensure the most powerful processing rates we should consider off-loading some computation analysis to the PC rather than relying on the instrument capabilities.

f) Use of IVI Drivers for Interchangeability

A new class of drivers called IVI drivers (Interchangeable Virtual Instruments) has been defined by the IVI Foundation, an Industry consortium. The IVI technology is based on involving standard programming interfaces for five classes of instruments (oscilloscopes, dvm, functional generators, switches, power supplies), which results in hardware independent instrument drivers, high flexibility and reusability. IVI drivers achieve better performance by remembering the state of the hardware settings on the instrument and avoid sending out redundant configuration commands.

g) Use of Multithreading in Application Software

Multithreading is a programming method by which a task is divided into different subtasks, or threads that can work independently from other threads in a system. It is possible to increase application performance by simultaneous execution of individual threads in a multiprocessor system, or by allocating appropriate processor time slices to threads in a single-processor system. Multithreading can help to manage system resources more efficiently.

For example, an application might have threads for different responsibilities – user interaction, data acquisition, data processing and data displaying.

h) Use of Native (Binary) Data Format for Data Transfer

Using binary data format for transferring data would avoid the overhead of two conversions. The first conversion in the instrument where binary data is converted into ASCII for a transfer across GPIB (can be very time consuming); the second conversion in the PC – from ASCII to binary for further data processing and analysis.

Some more ideas on GPIB system performance improvement are shown at the end of the following Part.

3 Experiment

a) Task Description

The experiment is oriented to design of automated instrumentation for a particular case of nondestructive diagnostics of cracks, holes, or any other irregularities inside or outside a part under test consisting of soft magnetic conducting material in leakage magnetic field [5].

The physical realisation of testing systems is based on multiple scanning of dc voltage from sensors - Hall type probes, moved along the tested pattern. The data are then processed by vector analysis, as well as archived and interpreted.

b) GPIB Design Considerations (Selecting System configuration, GPIB test instruments, GPIB controller card, choosing GPIB test software)

In one of the detection methods six inputs are handled. In such a case, it is necessary to implement and control triggering multiple GPIB instruments (dc voltmeters). Based on the test requirements and available test equipment, there are two possible solutions which are shown in Figure 1.

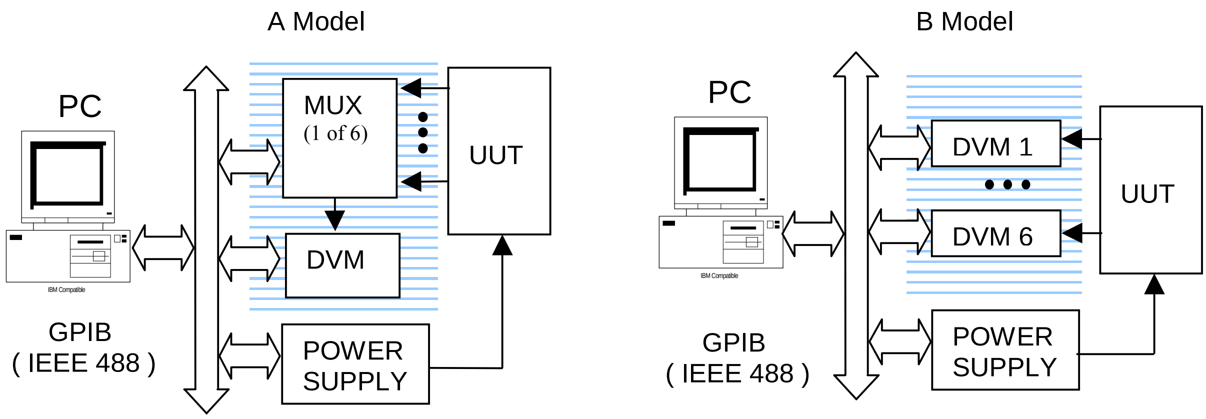

Fig. 1. Two possible models of AMS based on GPIB bus;
A-Model : PC, DVM, MUX (six channels), Unit Under Test, Power Supply
B-Model : PC, six DVM 1 – DVM 6, Unit Under Test, Power Supply

Both of the models are controlled by a PC equipped with one ISA GPIB card (GPIB-PCIIA) based on NI-488.2 device driver [9]. The controller card is compatible with the computer's hardware platform and its OS (Windows 95/98).

A-Model comprises a digital voltmeter DVM (HP34401A [8]) and a multiplexor MUX (HP3495A) with six channels connected to the Unit Under Test UUT. The output of the MUX is programmatically switched to the rear input of the DVM.

B-Model comprises six digital voltmeters DVM 1 – DVM 6 (HP34401A) connected directly to the Unit Under Test.

c) System Performance Testing

Several A and B model variations are measured. The programming structure of each of the models consists of the following steps: initialization, configuration, triggering, data acquisition, data preprocessing or processing, data transfer and data archiving and presentation.

When assessing the performance of the test system, particular times (see [4]) and the total time of the measurement procedure are considered. Table 1 shows results of benchmark test – total times (in seconds) of eight various procedures written in C language in Visual C++.

Model AMS	A Model		B Model	
Instrument	HP3495A – HP34401A		6 x HP34401A	
Range	Auto	10V	Auto	10V
Reading data	62	57	39	38
Reading average	70	67	55	56

Table 1. Times of measurement procedures in seconds for 10 triggers taking 10 samples.

The first comparison deals with different hardware – instruments and two comparisons are based on different instrument configuration and different ways of data processing.

c1) Comparison of A-Model and B-Model with the same type of DVM (HP34401A).
As voltmeters in both models are triggered by an interface command GET, we can talk about serial triggering (A) and parallel triggering (B). In A-Model, the DVM is triggered after closing single channels of the MUX. In B-Model, increase of performance is reached by triggering all six DVMs at the same moment and then the data are read from DVMs sequentially. The times listed in Tab.1 depend first of all on DVM reading rate (according to the DVM configuration, e.g. resolution, integration time, etc.) and also on MUX switching time.

c2) There is no significant difference betwen using autorange (Auto) or fixed range (10V).

c3) Comparison of using transfer of the whole data string and processing in PC (Reading Data) or preprocessing in DVM (Reading Average).
When DVM is configured for mathematical function 'Average', internal microprocessor counts the average value and the result is only transferred through the bus. Otherwise the DVM reads the data string and sends it through GPIB into PC for processing. This comparison sustains the discussion in section e) in 2nd Part, i.e. it is convenient to provide data processing in PC.

d) Results and Future Work

Though the B-Model is considerably better than the A-Model, it is not a very cost-effective solution. However, it serves as a model for testing performance. Besides, it allows simultaneous data monitoring on DVMs displays thus tuning the equipment of UUT to different detection methods.

In practical technical diagnostics application, remote test configuration will be inevitable [9]. Further hints for our GPIB system performance testing and evaluation follow:

- triggering devices: by interface command GET, by SCPI command MEAS?
- testing the possibilities discussed in 2nd Part, in sections a),b),c),d) and f),g),h).
- testing the influence of cable lengths, number of devices powered on the bus.

4 Conclusion

The results proved the former experience in automated instrumentation which suggests that not only hardware performance parameters are a determining factor. Also a program with a better or worse performance can be written. A possibility to analyse performance of a system before carrying out an experiment is of great importance. It requires a systematical approach to measurement system performance based on models of instrumentation and program sequences [1], [2], [3].

5 References

[1] First International Workshop on Software and Performance, ACM 1998, IEEE Transactions on Software Engineering, Vol. 26, NO.11, November 2000, NO.12, December 2000.
[2] Kleinová, I.: Automated Instrumentation Performance Evaluation. Doctoral Theses, STU Bratislava, 1992, 101 pp. (in Slovak).
[3] Kazička, R.: Contribution to the Evaluation of Performance of AT&MS. Doctoral Theses, STU Bratislava, 1995. (in Slovak).
[4] Kazička, R., Ondrášová, I., Smieško, V.: Improving the AT&MS Parameters, J. Electrical Engineering, 46 (1995), No.7, pp 261-264.
[5] Ondrášová, I., Setnička, V., Zubal, R.: Application of Automated Instrumentation in Technical Diagnostics. Proceedings of the Conference Electrical and Power Engineering 2000, Elosys, Trenčín, October 2000, pp. 82-84. (in Slovak).
[6] Patel, A.: Eight Ways to Increase GPIB System Performance, Application Note 133, National Instruments, November 1998.
[7] Porcello, J.: Using GPIB Instruments to Test Communication Systems, Communication Systems Design Magazine, March 1998.
[8] HP 34401A Multimeter, Hewlett Packard, User's Guide, February 1996 USA.
[9] The Measurement and Automation Catalog, National Instruments 2000.