Nuclear Science and Techniques 20 (2009) 307–311

LabVIEW-based auto-timing counts virtual instrument system with ORTEC 974 Counter/Timer

YAN Jie[1, 2,*] LIU Rong[2] LI Cheng[1] JIANG Li[2] LU Xinxin[2] ZHU Tonghua[2] WANG Mei[2] WEN Zhongwei[2] LIN Jufang[2]

[1] *Department of Modern Physics, University of Science and Technology of China, Hefei 230026, China*

[2] *Institute of Nuclear Physics and Chemistry, Chinese Academy of Engineering Physics, Mianyang 621900, China*

Abstract In order to achieve the auto-timing counts measurement of nuclear radiation using ORTEC 974 Counter/Timer, an auto-timing counts virtual instrument system based on the LabVIEW virtual instrument development platform and GPIB instrument control and transmission bus protocol is designed in this paper. By introducing software timing technique, the minimum time base of factory setting improves from 0.1 s to 0.03 s. The timing counts performance and longtime stability are also discussed in detail. The automatic data recording and saving facilitates data analysis and processing. Its real-time display and statistic function is very convenient for monitoring the nuclear radiation.

Key words LabVIEW virtual instrument, Software timing, Timing counts

1 Introduction

Measuring nuclear radiation counts in a certain interval time is a common approach in nuclear research, such as the lifetime measurement of radionuclides and neutron yield monitoring in neutron activation[1]. Special circuits based on hardware timing with resolution up to nanosecond using Field Programmable Gate Array (FPGA) technique[2] and Micro Processing Unit (MPU) with Complex Programmable Logic Device (CPLD) technique[3] are designed for precise timing measurement. On account of the high price and long design periods, however, they are not appropriate for a common application with timing interval requirement of just a few seconds.

In this paper, a software timing method based on LabVIEW (Laboratory Virtual Instrument Engineering Workbench)[4,5] and GPIB (General Purpose Interface Bus) is adopted to design a virtual instrument system of auto-timing counts with ORTEC 974 Counter/Timer[6], which is a 4-channel 100-MHz Counter/Timer under computer control or in manual operation. It can be used as a 3-channel counter with one presettable timer, a 4-channel counter (one counter channel presettable), or as a 3-channel counter (one counter channel presettable) with one timer. Standard computer interfaces built-in to the Model 974 include IEEE-488, RS-232-C and 20-mA current loop. The command format adheres to the NIM/GPIB[7] standard, an easy to use language for programming NIM instruments. The maximum counting rate for negative input signals is 100 MHz; and for positive input signals, 25 MHz. The time base of the Model 974 is of 0.1-s or 1-min increments derived from an internal 1-MHz crystal-controlled oscillator. With the aid of versatile LabVIEW's Timing Functions, the minimum time base is extended from 0.1 s to 0.03 s. The timing performance and longtime stability are tested using random pulse signal from a ^{24}Na γ-ray source and standard frequency pulse signals from an arbitrary function generator.

2 LabVIEW and GPIB

LabVIEW is an integrated data acquisition analysis and presentation package for the graphical programming of scientific and engineering

* Corresponding author. *E-mail address:* yanjie@mail.ustc.edu.cn
Received date: 2009-07-17

applications in areas from simulation, automated test, data acquisition and analysis, to PC-based instrument control. Based on the concept of graphical programming, it provides the best of glossaries, icons and notions known by the users. It includes device driver functions for serial and parallel interfaces, GPIB interface bus and a wide range of mathematical, logic, timing and digital signal generation, processing and analysis functions. GPIB, a digital 8-bit parallel communication interface in data transfer rates of 1 Mbyte/s and higher, using a three-wire handshake, was initially developed at Hewlett Packard and has been an IEEE standard since 1978[8]. The bus supports a System Controller, usually a computer, and up to 14 more instruments. Today, GPIB is the most popular interface between computer and instruments.

3 Auto-timing counts virtual instrument system

The auto-timing counts virtual instrument system (Fig.1) consists of a PC with LabVIEW virtual instrument, a PCI-GPIB card (NI-TNT5004), a GPIB cable and an ORTEC 974 Counter/Timer. The operation commands, data transmission and instrument response between PC and Model 974 are communicated via the PCI-GPIB card. Model 974 is only used to count the pulse signals. Timing and all the operations to the counts, such as real-time display and statistic analysis, data recording and saving, are implemented by the LabVIEW virtual instrument. Fig.2 is the front panel of the LabVIEW virtual instrument, which is featured as follows, compared with the 974.

(1) All the four channel data and related preset parameters are displayed simultaneously, while only one channel for the 974.

(2) During a measurement, control buttons on the 974 are disabled in order to avoid mistakes. The related parameters can be controlled and modified on control panel of the virtual instrument.

(3) Online real-time data statistic analysis, in chart or graph, is provided for an easy monitoring of the measured data.

(4) All the data acquired are recorded and saved automatically.

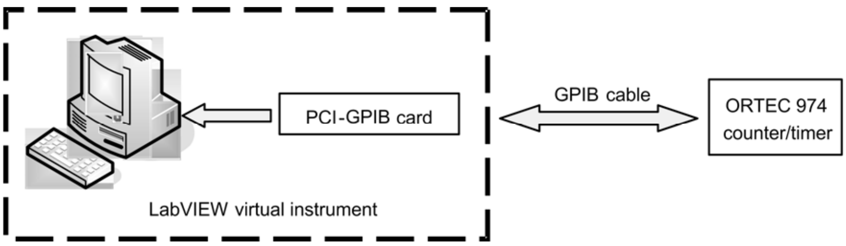

Fig.1 Schematic diagram of the auto-timing counts virtual instrument system.

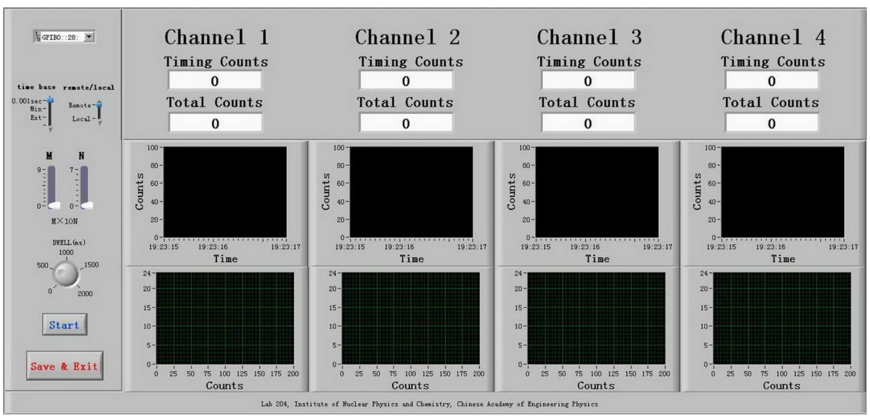

Fig.2 Front panel of the LabVIEW virtual instrument.

4 Process of the software timing

LabVIEW virtual instrument sends operation commands to and reads data from the 974 are implemented by using VISA[4] driver function. Unlike hardware timing of the 974, software timing is the most important core of the virtual instrument. A precise delay time, namely time base, is set between the operations of “start count” and “read count”. After the 974 receives an INIT command and responses correctly, a series of preset commands, such as CLEAR_ALL, SET_COUNT_PRESET, SET_MODE, and so on, are sent to the 974 in sequence. When a correct response returns to the LabVIEW virtual instrument, it sends START command to the 974 and selects a suitable LabVIEW Timing function to accomplish the precise delay time. Subsequently, an SHOW_COUNTS command is send to the 974 to fetch the counts until the Save & Exit button is clicked. By repeating these operations, the auto-timing counts function is achieved. Fig.3 is the flow chart of the software timing of the auto-timing counts virtual instrument system.

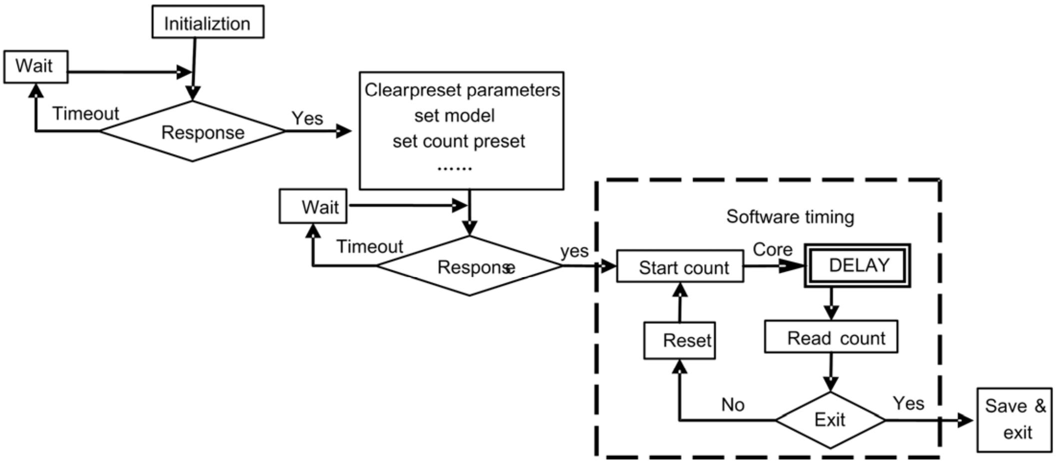

Fig.3 Flow chart of the software timing of the auto-timing counts virtual instrument system.

The LabVIEW Timing functions are defined in millisecond, hence a minimum time base of 1 ms. However, in running a program, the response time of sending commands and receiving response between PC and the 974, plus the real-time data statistic display and preservation, is in the magnitude of millisecond. For this reason, the delay time has a severe limitation. According to the LabVIEW Timing Functions, a delay time longer and shorter than 1 s is treated by adopting “Wait” and “Wait Until Next ms Multiple” Timing Functions, respectively, to avoid the influence of the response time. Since the “Wait” Timing Function causes the virtual instrument to wait a specified number of milliseconds before it continues execution, while the “Wait Until Next ms Multiple” Timing Function causes virtual instrument to wait until the internal clock equals or has passed a multiple of the millisecond multiple input number before continuing execution. The response time is excluded to the former, whereas the opposition to the latter. For the case of delay time longer than 1 s, the errors of the timing counts and the total counts, namely the sum of the timing counts, are less than 0.5%. And the longer the delay time, the smaller the errors (see Section 5). The former Timing Function is suitable for long delay time timing. For a delay time shorter than 1 s, though the response time is included in the latter Timing Function, the last response time has no impact on the next delay time, which is useful for causing loops to execute at specified intervals and synchronizing activities. Probably, it will wait less than the specified number of milliseconds in the first loop iteration, depending on the value of the clock when it starts (that is, how long it takes until the clock is at the next multiple and the virtual instrument proceeds). Consequently, the counts error is mainly due to the uncertainty in the first loop iteration. By contraries, this influence is so small that it can be ignored for the case of delay time shorter than 1 s. The latter Timing Function is suitable for shorter delay time timing. By adopting these two Timing Functions, the timing performance of the 974 is significantly enhanced by extending the minimum time base of factory setting from 0.1 s to 0.03 s.

5 Test of the software timing performance

For a Timing counter, the minimum time base determines the timing precision, and longtime stability is also an important performance criteria. Standard pulse signals with 1-kHz frequency and 4-V amplitude generated by Tektronix AFG310 Arbitrary Function Generator and random pulse signals from the ^{24}Na γ-ray source are used to set up a test system (Fig.4). The sum of the timing counts is compared with the continuance counts from a reference counter ORTEC 994 Counter/Timer[9]. In order to test the timing precision, the measured time is set as 300 s for the 994, whereas ten timing intervals from 0.02 s to 300 s are chosen for the software timing. For the standard pulse signals, there is a discrepancy of ±1 count in the timing counts. The timing counts error only relates to the count rate and plays a major role in the total counts error. Since the amount of the timing pulse signals is proportional to the length of software timing, both the timing counts error and total counts error increase in reduced software timing. Nevertheless, the total counts error between 0.04 s and 10 s keeps invariant for both the standard pulse signals and random pulse signals (Fig.5). When the software timing is shorter than 0.03s, the delay time is comparable to the response time and has no impact on the software timing. Therefore, the minimum base time of the virtual instrument system is about 0.03 s.

Fig.6 shows the result of the longtime stability. The total counts error is only relate to the timing counts error and is nearly a constant no matter how long the measured time is with the timing interval is 1 min. The test results indicate that this auto-timing counts virtual instrument system has a good timing performance and longtime stability.

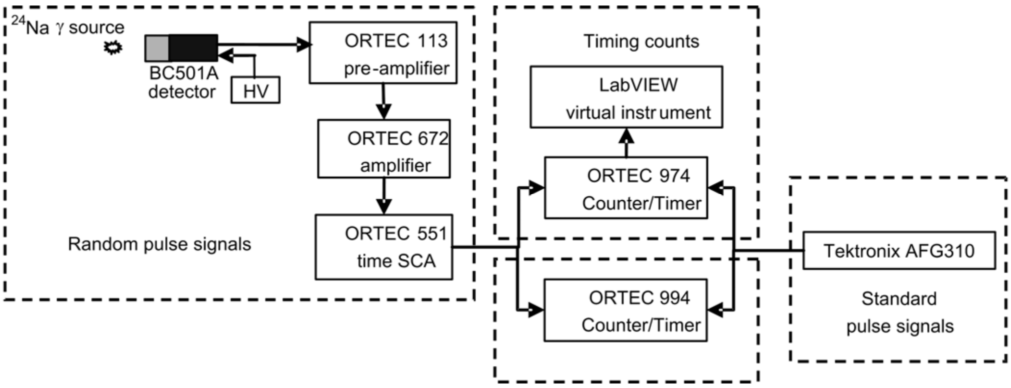

Fig.4 Schematic diagram of the performance test system.

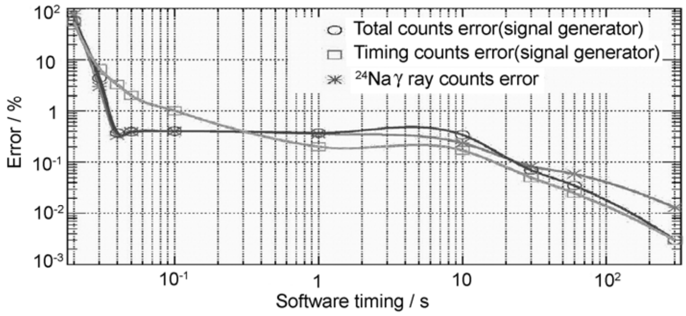

Fig.5 Software timing dependent counts error for the auto-timing counts VI system.

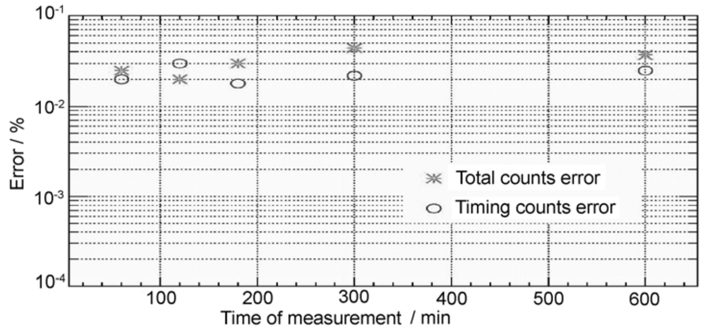

Fig.6 Stability variation with time of measurement.

6 Discussion and conclusion

Based on the LabVIEW programming language and GPIB interface bus, an auto-timing counts virtual instrument system with excellent timing performance and longtime stability is designed. It implements the remote control apparatus and data recording and saving automatically which facilitate the further data analysis and processing. Furthermore, it is convenient for online monitoring of the timing counts attributed to the real-time display and statistic function for all the four channels.

The software timing technology is used to replace the hardware timing in the virtual system for the first time. On the basis of achieving the original functions of the 974, it extends the time base from 0.1 s to 0.03 s. Nevertheless, for the LabVIEW Timing functions are in the unit of millisecond, the software timing technology is restricted to be used in the order of millisecond or above. It is also not the unique method to extend the time base. If the measurement requires a shorter interval time, either ameliorating the electronic circuit of the 974 or adopting the FPGA or CPLD technology is a feasible choice.

Using an arbitrary function generator and a ^{24}Na γ-ray source, the system was tested for the software timing counts performance and longtime stability. This system has been used to monitor the neutron yield and the running status of the D-T neutron generator in the neutronic experiment. With the automatic data recording and saving, workload of the experiment is greatly reduced. The online real-time display and statistic function enriches the monitoring tools for experimenters.

References

1 Knoll G F. Radiation detection and measurement. New York: John Wiley &Sons Press, 1976: 62-74.

2 LIANG Yongfei, BAI Lixin, ZHANG Yiyun, *et al*. Nucl Electron Detect Technol (in Chinese), 2006, **26**(3): 355-357.

3 QUE Yuan, DONG Jianhua, LI Xiaodong, *et al*. J Isot (in Chinese), 2008, **21(**3): 135-139.

4 Austin, Texas: National Instruments Corporation. LabVIEW User Manual. 2008.

5 Travis J, Kring J. LabVIEW for Everyone: Graphical programming made easy and fun, 3th Edition. Pearson Education, Inc. 2007: 354-388.

6 Advanced Measurement Technology, Inc. Model 974 quad counter/timer operating manual, 2002.

7 U.S. NIM committee. Standard NIM digital bus (NIM/GPIB), DOE/ER-0173, August 1983.

8 GPIB manual. [EB/OL]. [2009-07-17]. http://www.ni.com/gpib/.

9 Advanced Measurement Technology, Inc. Model 994 dual counter/timer operation and service Manual, 2002.