INTERNAL PERFORMANCE MONITORING BY EVENT SAMPLING

INTRODUCTION

The subject invention provides an event-driven method and means for sampling hardware generated signals within a data processing system based upon the occurrence of selected processing events and conditions. The event signals are sampled and recorded by means which are distributively interwoven within the CPU structure.

BACKGROUND

Many computer performance monitoring tools have been developed for evaluating the performance of computer systems. They have been conceived with various goals. Some are software, some hardware. Most hardware monitors have been separate from the system they measure, connected to it by manually inserted probes or by a plug interface.

The traditional distinction among monitor types is between counters and recorders. The counter type counts the number of occurrences of each of a set of events, with the counted output normally representing some kind of meaningful information. The recorder type collects data about defined events on recording media. Later analysis of both types is usually needed to make the collected data intelligible. IBM monitors in the early 1960's measured specified states in an IBM 7090 data processing system, such as: total CPU operation time; channel A operation time; channel B operation time; CPU busy with no I/O in process; tape equipment operation time, CPU in wait state; and card equipment operation time.

Software monitors are widely used today but are limited to sampling data stored in memory. They cannot detect hardware states per se. Also, software monitors universally have the drawback of distorting the performance of the system they are measuring, because software monitors compete with the program being measured for use of the resources in the system.

Monitoring functions may also be separated into two other subtypes: those that sense hook instructions put into a program to assist a monitoring operation, and those that sense some characteristic stored by an unmodified running program. For example, a hook may be put into a program routine so that the number of times the hook instruction executed would indicate the number of times the routine was entered, or the number of times routine looped, depending on where the hook was inserted. Both software and hardware monitors have been used to sense and count the occurrences of a hook instruction. Also, monitor functions that have been used to sense non-hook program characteristics, for example, have counted the occurrence of specified operation codes, or plotted the address distribution of accesses to main storage. Non-operable instructions have been inserted as hooks to cause a program interrupt that initiated the recording of an identifying characteristic of the hook instruction. Further, the Monitor Call (MC) instruction in the IBM System/370 architecture was provided for use as a hook instruction insertable into program code.

Monitors and their use have been described in publications, such as a book entitled "Evaluation and Measurement Techniques for Digital Computer Systems", by M. E. Drummond, Jr., published in 1973 by Prentice-Hall Inc., Englewood Cliffs, N.J.

Hardware monitors have been commercially sold, such as the Comten 8028 monitor and the Tesdata monitors. Software monitors have been in public use for many years such as the IBM "Resource Measurement Facility" (RMF) and the Candle "Omegamon" program.

Examples of early patents on data processing system hardware monitors externally connectable to a system are represented by U.S. Pat. No. 3,399,298 to H. M. Taylor entitled "Data Processing Profitability Monitoring Apparatus"; U.S. Pat. No. 3,588,837 to R. D. Rash et al entitled "System Activity Monitor"; and U.S. Pat. No. 4,068,304 to W. F. Beausoleil et al entitled "Storage Hierarchy Performance Monitor" (assigned to the same assignee as this application).

Another externally connected monitor is disclosed and claimed in U.S. Pat. No. 4,435,759 to R. I. Baum et al (assigned to the same assignee as this application). It provides a hardware monitor with a software correlation characteristic. It is externally connected to a uniprocessor or multiprocessor system to collect selected hardware events in that system. It relates the collected hardware events to causative software by simultaneously capturing and recording the address of a potentially causative software instruction at the time the hardware event is being sampled for collection. Collecting is done on every Nth occurrence of a predetermined hardware event, to capture the causative instruction address and one or more other hardware states that can be correlated with the captured instruction address. Hence, the captured instruction addresses relate the simultaneous collected events to the software that potentially caused them. U.S. Pat. No. 4,435,759 also discloses a set of monitors, externally connected to the CPU's in a multiprocessor, with interconnections between the monitors, all monitors also being connected to an external control processor. The control processor issues read commands to the plural external monitors to synchronize their capture and outputting of events in the different CPU's being monitored. The control processor in this way groups the captured events, by receiving and recording each group as the set of events resulting from each read command.

U.S. Pat. No. 4,590,550 (assigned to the same assignee as this application) to J. H. Eilert et al entitled "Internally Distributed Monitoring System" discloses a timer-driven performance monitor, built into and distributed within the system it measures, thereby eliminating some monitoring problems caused by the external location of prior monitors.

SUMMARY OF THE INVENTION

The present invention differs from U.S. Pat. No. 4,590,550 in providing an event-driven monitor which, as discussed below, is internal to a processor (which may be in a UP or MP) to allow a concentrated mode of data capture especially suitable for measuring the performance of very fast LSI hardware in relation to its driving software, including during the developmental debug phases of the hardware or software. Nevertheless, the subject invention is built on the base environment defined in U.S. Pat. No. 4,590,550. The objectives of this invention are to:

1. Provide self-contained monitoring for a system in which LSI technology may prevent the attachment of an external hardware monitor.