

Finally, with reference to FIG. 5, there is depicted a logic flow chart illustrating the collection of monitoring data in accordance with the method of the present invention. As is illustrated, the process beings at block 130 and thereafter proceeds to block 132 which illustrates the storing of data into a buffer within data collection card 22 (see FIG. 4). Next, block 134 depicts a determination of whether or not the present buffer being utilized is full, and, if not, the process returns to block 132 and additional data is stored within that buffer.

In the event the buffer currently being utilized is full, as determined by block 134, then block 136 illustrates the switching to a second buffer in the switched bank memory system of data collection card 22. Next, the data contained within the filled buffer is output to memory within second processor 26, in the manner described above with respect to FIG. 4. Thereafter, block 140 illustrates a determination of whether or not additional data is being received and if so, the process returns to block 132 which depicts the storing of additional data within a buffer. In the event no additional data has been received, then the process terminates, as illustrated in block 142.

Upon reference to the foregoing, those ordinarily skilled in the art will appreciate that the Applicants in the present invention have created a software monitoring and development system which avoids the disadvantages of all known previous software monitoring and development techniques. By loading the software under development with a minimal number of uniquely identifiable elements or "hooks" and outputting a predetermined frame of data in association with each of these hooks, upon the encountering thereof, the processor overhead and output bus space required for known software based monitoring and development systems are not required.

Next, by rapidly transferring these frames of data to a second processor, the problems associated with known hardware based software monitoring and development techniques are avoided. The analysis and tagging of each data frame associated with a particular hook may then be accomplished "off-line" by the second processor without the prospect of processor speed differences between a first generation processor and a second generation processor causing great gaps in the data accumulated from the software application under development.

In this manner, the hybrid system disclosed herein permits the highly detailed trace of a software application during its development for long periods of time without unduly loading the processor being utilized or requiring a highly sophisticated high speed processor for hardware monitoring applications.

While the invention has been particularly shown and described with reference to a preferred embodiment, it will be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope of the invention.

**We claim:**

1. A method for monitoring the performance of a selected software application with a first processor having an output bus, said method comprising the steps of: inserting a limited number of uniquely identifiable elements at selected locations within said selected software application; processing said selected software application within said first processor;

automatically coupling to said output bus a unique identification of each of said limited number of uniquely identifiable elements, in association with a preselected data frame, in response to each encountering of one of said limited number of uniquely identifiable elements during processing of said selected software application;

coupling a second processor having memory means coupled thereto to said output bus;

associating a chronological order time value with said identification and said preselected data frame; and

storing said identification, said preselected data frame, and said chronological order time value within said memory means coupled to said second processor wherein said performance of said selected software application may be monitored.

2. The method for monitoring the performance of a selected software application according to claim 1, further including the step of associating an identifying header frame with said identification, said selected data frame and said chronological order time value prior to storage thereof.

3. The method for monitoring the performance of a selected software application according to claim 1, further including the step of storing within said memory means coupled to said second processor an overrun indication in the event said identification and said preselected data frame are coupled to said output bus at a speed in excess of the speed at which said identification and said preselected data frame may be stored.

4. A data processing system for monitoring the performance of a selected software application within a first processor having an output bus, said data processing system comprising:

means for inserting a limited number of uniquely identifiable elements at selected locations within said selected software application;

means for automatically coupling to said output bus a unique identification of each of said limited number of uniquely identifiable elements in association with a preselected data frame, in response to each encountering of one of said limited number of uniquely identifiable elements during processing of said selected software application within said first processor;

a second processor coupled to said output bus of said first processor, said second processor having memory means coupled thereto;

means for associating a chronological order time value with said identification and said preselected data frame; and

means for coupling said identification, said preselected data frame and said chronological order time value to said second processor for storage within said memory means coupled thereto.

5. A data processing system for monitoring the performance of a selected software application within a first processor having an output bus according to claim 4, further including means for storing an overrun indication within said memory means coupled to said second processor in the event said identification and said preselected data frame are coupled to said memory means at a speed in excess of the speed at which said identification and said preselected data frame may be stored.

6. A data processing system for monitoring the performance of a selected software application within a first processor having an output bus according to claim