A Technique for Recording and Signal Averaging Counts from Several Input Sources
Simultaneously Employing a Small Computer

March 5, 1971

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

H. Bradford Thompson

Professor of Chemistry

The University of Toledo, Toledo, Ohio

PROJECT: Molecular Structure Studies: A Gas Phase Electron Diffraction Unit
Employing Counting Techniques

MASTER

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

COMPUTER INTERFACE FOR RECORDING COUNTS FROM SEVERAL CHANNELS SIMULTANEOUSLY¹

H. Bradford Thompson

Department of Chemistry

The University of Toledo

Toledo, Ohio 43606

In the design of a gas-phase electron diffraction apparatus using electron-counting techniques, a device was required that would accumulate counts on seven channels simultaneously and build a histogram-type record for each channel as the apparatus was moved through each of forty recording positions. In order to minimize errors from drift, time-averaging was to be used, the apparatus making as many as 1000 sweeps through the recording positions during an experiment. The most economical way to achieve this was to use an instrument computer. The Digital Equipment Corporation PDP-8/I was chosen for our installation. While the features of this machine dictate the precise form of our interface, the basic scheme is probably applicable to other computers. Since the computer was assigned a number of tasks besides count-acquisition (display, consistency checks, monitoring, control, etc.), the time required for count-acquisition was to be kept to a minimum. The interface described here performs this task admirably, using the Memory Increment Data Break facility of the DEC 8-series machines. The time fraction actually used by the Data Break for count acquisition was less than .01.

In our experiment, the instrument remains at each position for about 0.1 sec., and raw data occur at an average rate of 100kHz to 500kHz. These incoming pulses are prescaled on each channel by a factor of 64 or 128 before reaching the circuitry shown in figure 1. Prescaling serves two functions: it reduces the data-input rate to a level the interface can handle, and it evens out the time between pulses into each Data Buffer so that the chance of jamming a channel and thus missing a pulse is insignificant.

The sequence of events at a given recording position is as follows. The computer senses the position (in our case, through a signal to a Schmidt-trigger on a DEC AX08 peripheral) and gives a 6xx6 instruction to the interface (figure 1), with address of the first of seven storage locations in the AC. This instruction turns on the Enable flip-flop, sets bits 3-8 of the core addresses to be used into the Address Buffer, clears the data buffer on each input channel, starts the timer module and sets each prescaler to 32 or 64 (i.e., half-full). This setting of the prescalers provides for "rounding" to the nearest half-count in terms of Data Buffer input. Meanwhile, as the computer continues to operate, the scale-of-8 triggered by the BTS-1 pulse during each instruction constantly scans the channels via the octal decoder, and provides bits 9-11 of the data core address. (Figure 1 shows the circuit for one of the three address bits.) When a Data Buffer is set and the scale-of-8 comes to the matching address, a positive level will occur at the output of the R-141 And/Nor module. This permits the next BTS-1 pulse to set the Break-Request flip-flop. This in turn has three results: (1) a break request is sent to the PDP-8/I, (2) the scale-of-8 counter is halted so that the address will not be changed until it is accepted by the computer, and (3) the appropriate Data Buffer is cleared. The Break Request flip-flop is reset by the leading edge of the address Accept signal from the computer, and the required Memory Increment and Direction signals are obtained by a NAND between the B-Break signal and the interface Enable flip-flop. The scale-of-8 resumes scanning, and can find another count as early as the positive-going BTS-1 change that occurs during the B-Break. On the PDP-8/I this can be within three microseconds.

Entry of data ceases when the timer module returns to the off state. At this time, a logical NAND is performed so that a signal is sent to disable the Enable flag only when the last filled Data Buffer is cleared. Thus prescaled counts cannot

enter the buffers after the Timer period, but counts already in the buffers are read by the computer.

By this process, prescaled counts from seven channels have been accumulated in seven consecutive addresses. Upon arrival at the next recording position, the computer must simply repeat the process in the preceding two paragraphs, using a new storage address. The time allowable at a given position is, of course, limited by the capacity of one memory location. In our case, incoming counts at 500 kHz, prescaled by a factor of 128, could be counted for one second before overflowing the 4K capacity of one word of core. The upper limit on the incoming pulse rate is governed by the cycle period of the Scale-of-8. As a worst case, if each Data Buffer required servicing and if each computer instruction were 4.5 μ sec in length, this cycle period could be as great as $8 \times 4.5 + 7 \times 1.5 = 46.5 \mu\text{sec}$. At 500kHz prescaled by 64, we would be within this limit by a factor of more than two. With this much prescaling, this factor is far more than required. With less prescaling, larger safety factors may be advisable, since pulses into the Data Buffer will not be evenly spaced. Potential users of this scheme should consult standard references on counting techniques. An upper limit on prescaling is provided by the loss of precision by roundoff at the end of the read period. In our case probable errors from this source are an order of magnitude smaller than the random count statistical errors expected in the experiment.

By this scheme, one PDP-8 can do the work of several averagers, with time left to monitor and direct the experiment. For example, the data from each sweep can be checked for consistency in several ways before it is added to the accumulated count totals. Clearly, this type of interface is applicable to a range of numbers of channels, count rates, experimental recording sequences, and so on.

Interface to PDP-8/I Computer

Figure 1. Interface Logic.

Interface is shown in DEC negative logic symbols as implemented for our installation. Implementation in later model modules using IC's should be straightforward

APPENDIX A

Description as it appeared in DECUSCOPE, the newsletter of the Digital Equipment Computer Users Society.

In our experiment, the instrument remains at each position for about 0.1 sec., and raw data occur at an average rate of 100kHz to 500kHz. These incoming pulses are prescaled on each channel by a factor of 64 or 128 before reaching the circuitry shown in Figure 1. Prescaling serves two functions: it reduces the data input rate to a level the interface can handle, and it evens out the time between pulses into each Data Buffer so that the chance of jamming a channel and thus missing a pulse is insignificant.

The sequence of events at a given recording position is as follows. The computer (in our case, through a signal to a Schmidt-trigger on a AX08 peripheral) starts a clock (in the AX08) and gives a 6xx6 instruction to the interface (Figure 1), with address of the first of seven storage locations in the AC. This instruction turns on the Enable flip-flop, sets bits 0-8 of the core addresses to be used into the Address Buffer, clears the Data Buffer on each input channel, and sets each prescaler to 32 or 64 (i.e., half-full). This setting of the prescalars provides for "rounding" to the nearest half-count in terms of Data Buffer input. Meanwhile, as the computer continues to operate, the scale-of-8 triggered by the BTS-1 pulse during each instruction constantly scans the channel via the octal decoder, and provides bits 9-11 of the data core address. (Figure 1 shows the circuit for one of the three address bits.) When a Data Buffer is set and the scale-of-8 comes to the matching address, a positive level will occur at the output of the R-141 And/Nor module. This permits the next BTS-1 pulse to set the Break-Request flip-flop. This in turn has three results: (1) a break request is set to the PDP-8/1, (2) the scale-of-8 counter is halted so that the address will not be changed until it is accepted by the computer, and (3) the appropriate Data Buffer is cleared. The Break Request flip-flop is reset by the leading edge of the address Accept signal from the computer, and the required Memory Increment and Direction signals are obtained by a NAND between the B-Break signal and the interface Enable flip-flop. The scale-of-8 resumes scanning, and can find another counter as early as the positive-going BTS-1 change that occurs during the B-Break. Counting at a given

PDP-8/1 INTERFACE FOR RECORDING COUNTS FROM SEVERAL CHANNELS SIMULTANEOUSLY¹

H. Bradford Thompson
Department of Chemistry
The University of Toledo
Toledo, Ohio

In the design of a gas-phase electron diffraction apparatus using electron-counting techniques, a device was required that would accumulate counts on seven channels simultaneously and build a histogram-type record for each channel as the apparatus was moved through each of forty recording positions. In order to minimize errors from drift, time-averaging was to be used, the apparatus making as many as 1000 sweeps through the recording positions during an experiment. Since the computer was assigned a number of tasks besides count-acquisition (display, consistency checks, monitoring, control, etc.), the time required for count-acquisition was to be kept to a minimum. The interface described here performs this task admirably, using the Memory Increment Data Break facility of the 8-series machines. Negative logic (R and W series) is used to match our PDP-8/1; however, the general scheme should be easy to implement in M-series logic.

position is terminated by a 6xx1 signal from the computer.

By this process, prescaled counts from seven channels have been accumulated in seven consecutive addresses. Upon arrival at the next recording position, the computer must simply repeat the process in the preceding paragraph, using a new storage address. The time allowable at a given position is, of course, limited by the capacity of one memory location. In our case, incoming counts at 500kHz, prescaled by a factor of 128, could be counted for one second before overflowing the 4K capacity of one word of core. The upper limit on the incoming pulse rate is governed by the cycle period of the scale-of-8. As a worst case, if each Data Buffer required servicing and if each computer instruction were 4.5μsec in length, this cycle period could be as great as $8 \times 4.5 + 7 \times 1.5 = 46.5\mu\text{sec}$. At 500kHz prescaled by 64, we would be within this limit by a factor of more than two. With this much prescaling, this factor is far more than required. With less prescaling, larger safety factors may be advisable, since pulses into the Data Buffer will not be evenly spaced. Potential users of this scheme should consult standard references on counting techniques. An upper limit on prescaling is provided by the loss of precision by roundoff at the end of the read period. In our case probable errors from this source are an order of magnitude smaller than the random count statistical errors expected in the experiment.

A final source of error to be considered is that counts may be left in the buffers when the count period is terminated. In our case about one count in 4000 is lost this way, and in the course of 100 or more sweeps this error becomes statistically insignificant. A remedy is readily available if needed, however. The prescalers or the inputs to the Data Buffers could be disabled at the termination of the counting period, and the 6xx1 Disable instruction given at least eight machine instructions later. This would require one additional flip-flop, but might not require an additional device selector and code, since the Disable (6xx1) and Clear (6xx2) instructions used here can probably be combined, leaving one instruction free to clear the added gate flip-flop.

By this scheme, one PDP-8 can do the work of several hard-wired averager devices, with capacity left over to monitor and direct the experiment. For example, we intend that the data from each sweep be checked for consistency in several ways before it is added to the accumulated count totals.² Clearly, this type of interface is applicable to a range of numbers of channels, count rates, experimental recording sequences, and so on. We hope to report on other aspects of our own interface as they are developed and tested.

¹This work is supported by a grant from the United States Atomic Energy Commission.

²The aid of Mr. Gregory Fehn in development of software and in statistical analysis of error sources is gratefully acknowledged.