SLAC-PUB-809
November 1970

NEW DIGITIZING AND MEMORY SYSTEM FOR WIRE SPARK CHAMBERS*

by

R. G. Friday, D. W. G. S. Leith, K. D. Mauro, and B. Richter

Stanford Linear Accelerator Center
Stanford University, Stanford, California 94305

ABSTRACT

A new digitizing and memory system for use with magnetostrictive wire spark chambers has been developed. The system is based on the use of fast shift registers as the memory element, and appears to reduce the cost of readout systems for large spark chamber arrays by a factor of about 10. Circuitry is described for use with a large spark chamber system requiring the digitization and storage of 1000 sparks per event. The principle has applications in any situation requiring the digitization and storage of the arrival times of a burst of pulses.

(Submitted to Nuclear Instruments and Methods)

*Work supported by U. S. Atomic Energy Commission.

I. INTRODUCTION

In the past several years, wire spark chambers have come into widespread use in high energy physics as devices for the precise determination of the position of a particle which crosses the gap of a chamber. Many methods of sensing, digitizing, and storing the spark position information have been devised, but the most popular has proved to be the magnetostrictive system, in which position of a spark in space is related to the time of arrival of a strain pulse traveling at sonic speeds in a magnetostrictive wire.

The conventional method for digitizing this information uses a master clock (typically 20 mc) and one scaler (typically 12-bit) for each spark to be digitized in an event. All the scalers start counting the clock on the application of high voltage to the spark chambers, and one scaler is stopped on the arrival of each spark-induced strain pulse at the sensing point on the magnetostrictive wires. Such systems typically cost from $100-200 per spark recorded. As spark chamber systems grow larger, the maximum number of sparks to be recorded per event has also increased, and the cost and complexity of the digitizing system have become significant.

Recent advances in integrated circuit technology have made practical other methods of digitizing and storing spark chamber information. The system described here uses a shift register memory and costs about $10/spark recorded.

II. CONCEPTUAL DESIGN

The system is built around a set of fast, cheap shift registers which have recently become available. It uses a single master counter whose reading is transferred to the memory whenever a spark is detected on one of several magnetostrictive pickups. The basic operation of the system is illustrated in the simplified block diagram of Fig. 1.

The digitizing cycle begins with a signal from the spark chamber trigger system to the controller, which gates the clock signals into the counter. The counter will continue accumulating clock pulses throughout the recording period. The outputs from the magnetostrictive pickups go to both the encoder which indicates which pickup line is active by a binary bit pattern at its output, and to the spark detector. The spark detector output signals the controller to shift the contents of the clock and the encoder into the memory. This read-in cycle is repeated whenever the spark detector is triggered by a signal on one of the chamber pickups.

At the end of a predetermined period equal to the transit time of a magnetostrictive pulse down the longest of the pickup wires, the digitizing cycle is ended and the data stored in the shift register transferred to the computer. The controller first advances the data through the shift register at the clock speed until the first spark read-in is at the output of the shift register. The controller then signals the computer to begin reading in data. The computer first reads the spark counter, which indicates to the computer the total number of sparks which have been recorded and must be read in. The computer then

reads the output of the shift register and signals the controller to advance the next word for read-in.

This cycle continues until all the data have been transferred to the computer at which time the system is reset and is ready for the next event. The total cycle time is limited by the propagation time in the magnetostrictive wire and by the speed of the computer. For typical systems of moderate size the cycle time will be about a millisecond.

III. THE 1000-SPARK SYSTEM

a. General

The block diagram of a system which we have designed to be used with a large spark chamber array is shown in Fig. 2. The operating principles are the same as described above, but the system has become more complex because of the size of the spark chamber array (64 magnetostrictive wire inputs), the maximum number of sparks to be recorded (10 per input = 640), the speed of available shift registers (5 MHz for the most reliable units compared to the 20-MHz speed of the clock), and the particular computer available at the laboratory (IBM-1800). The large number of inputs generates a new problem - that of input line identification. The number of sparks, available component speeds, and particular computer available cause no new problems of principle, but do cause some complication in the circuitry.

b. Input Identification

The 64 input lines could be identified by a 6-bit binary code. However, with a system this large, the probability of having signals

on more than one input arrive simultaneously becomes appreciable. We have chosen to use a 12-bit system for input identification which is not troubled by the coincident arrival of several signals.

The inputs are broken down into 8 groups of 8. The arrival of a signal at input i of group j sets a holding flip-flop whose output drives input i of multiplexer j and input j of OR circuit i in the priority encoder system. If only one input is active the priority encoder presents its bit address i to 3 of the inputs to the shift register memory. At the same time the priority encoder enables the output of the multiplexer whose i^{th} input is active. The multiplexer outputs go to the memory on 8 more lines. The spark detector then loads the contents of the 15-bit scaler, the multiplexer address bits, and the priority encoder bits into the memory. The one-of-eight decoder then resets holding flip-flop i of all 8 groups of chamber inputs.

Signals arriving in time coincidence at more than one input are handled in different ways depending on whether or not they arrive at the same input i of more than one group. If they do, the procedure outlined above is followed with more than one of the multiplexer inputs to the shift-register memory activated.

If signals arrive at different inputs i, the priority encoder reads the largest value of i. The clock and input code of this largest value of i are read into the memory in the usual way. In addition the digital comparator enables the coincidence bit and it, too, is read into memory. (The coincidence bit allows the computer program which analyzes the data to recognize a coincident group and use the counter data corresponding to the first member of the group.)

The spark detector will trigger again if any input to the priority encoder remains active after the largest bit i is reset. This cycle continues until all of the coincident inputs are read into the memory.

c. Memory

The memory is actually composed of 4 parallel banks of 28 × 256-bit 5-MHz shift registers (INTEL-1402). The choice of 5-MHz shift registers was forced on us by the availability of components, and the 4 parallel banks are required to achieve the 20-MHz rate needed to obtain sufficient resolution on spark position in the spark chambers. The 5-MHz shift registers require about 200 nsec to load, and, since the minimum interval between events is 50 nsec, a buffer is required. The 30-bit latches serve this purpose - they are enabled in sequence by the spark detector. The first event will be loaded into latch A by the spark detector output which also advances a 2-bit counter steering the next load command to latch B, etc. Thus, each of the 4 memory banks receives 1/4 of the data and operates at a maximum 5-MHz rate while the entire memory operates at 20 MHz.

d. Operation and Readout

The system is started by a pulse from the spark chamber trigger system which starts the 20-MHz counter and enables the spark detector output. Each event is stored in the memory and advances the spark counter. This process continues until the 20-MHz counter overflows.

The overflow (carry) signal from the 20-MHz counter disables the counter, blocks the path from the spark detector to the 10-bit spark counter, transfers the reading of the spark counter to its associated 10-bit latch, and allows the 20-MHz clock to directly advance the spark counter

and shift registers. The spark counter and shift register continue advancing until the spark counter overflows, which stops the advance pulses and turns off the busy signal. Since the spark counter overflows on the 1024th pulse and the shift-register memory is 1024 words long, the first event stored is now at the output end of the memory regardless of actual number of sparks recorded. The number of sparks recorded is still held in the spark counter latch. The readout process now begins.

Parts of the readout system are specific to our computer and these details will not be described here. The readout multiplexers are controlled by FF2 and FF3 of Fig. 2. They select the 10-bit spark counter latch as the first word to be read by the computer, and drive the large multiplexer at the shift register memory output. FF2 and FF3 are off initially. The first computer read request turns FF2 on, enabling the spark latch select line. The second read request turns FF3 on, disabling the spark latch select line, and routing the first 14 bits of shift register group A to the computer. The third read request turns FF2 off, routing the second 14 bits of group A to the computer. The fourth read request turns FF2 on again, advancing a 2-bit counter causing the memory output multiplexer to read the first 14 bits of memory group B, and shifts the data in memory group A up by one. This process continues until all events have been read into the computer.

IV. CONCLUSION

The system has been tested in the laboratory and works well. Total parts costs for the 1000-spark system are around $2,000, and we estimate total costs as roughly 5 × parts cost. Further advances in the integrated circuit art will undoubtedly result in faster shift registers becoming available, allowing the elimination of the interlaced memories with a consequent simplification of the circuitry.

This digitizing and memory scheme should have applications in any situation requiring the digitization and storage of the arrival times of a burst of pulses.

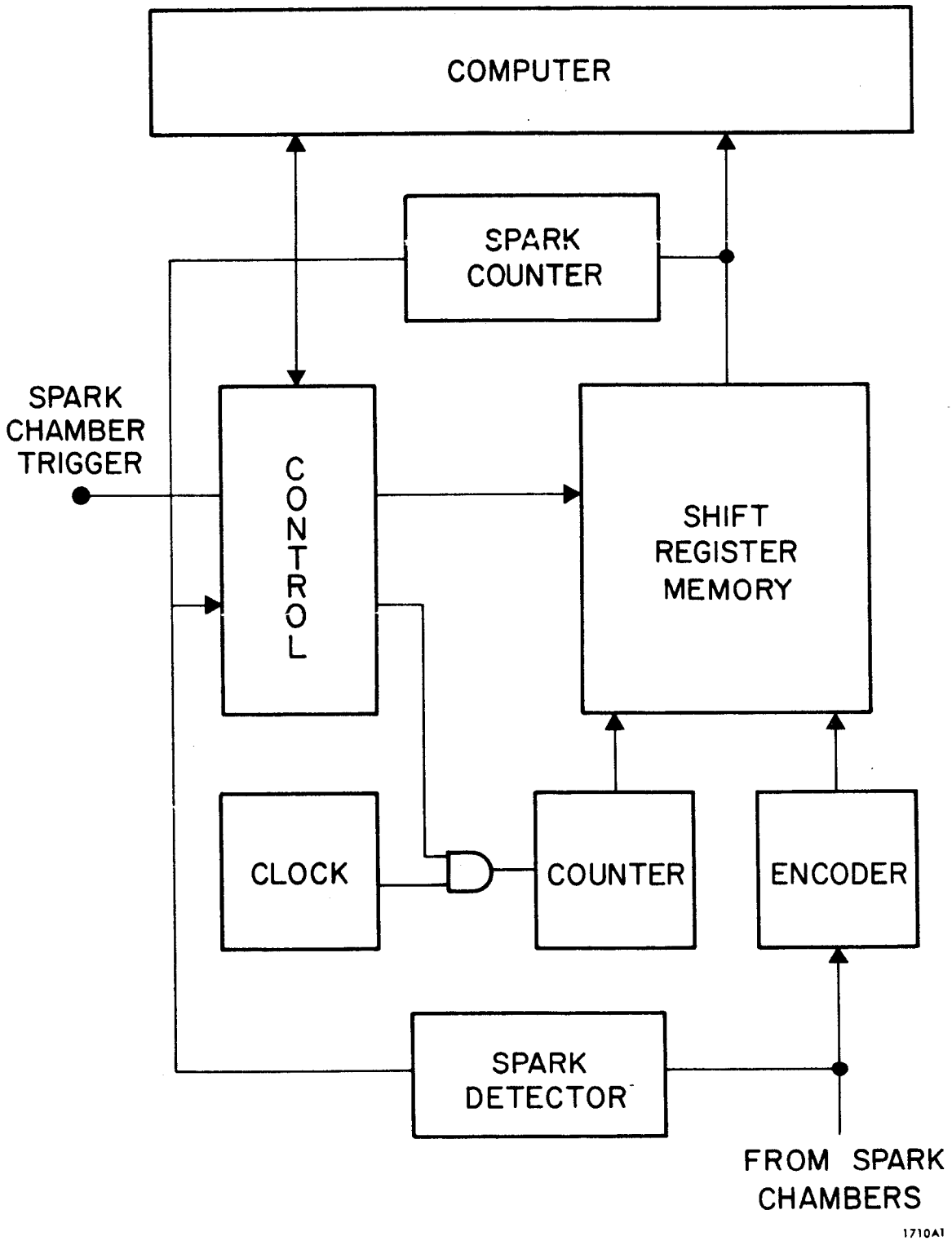

Fig. 1

Simplified block diagram.

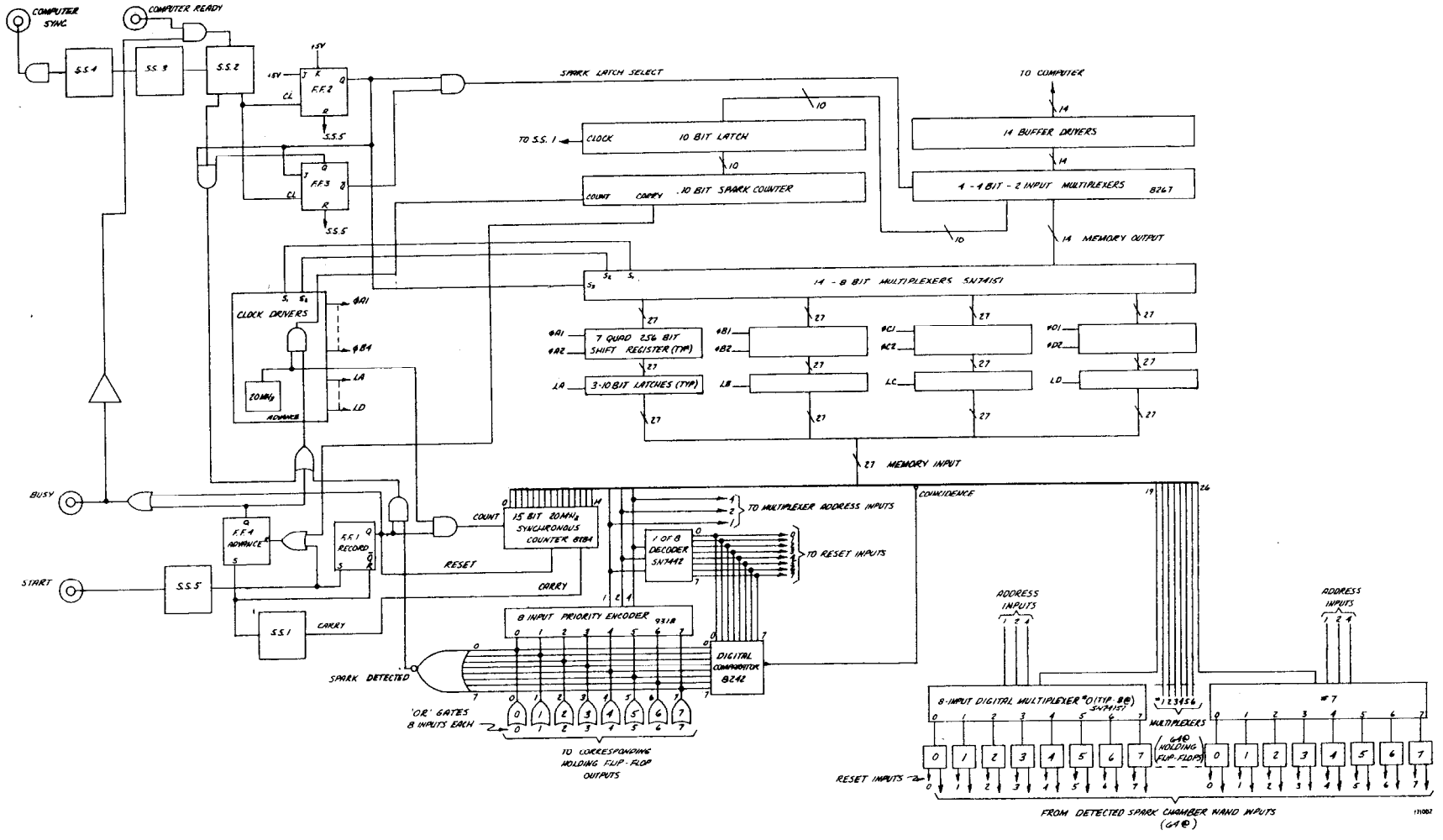

Fig. 2

Block diagram of 1000-spark system.