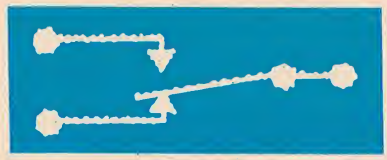


Don't dismiss the relay for data-processing systems

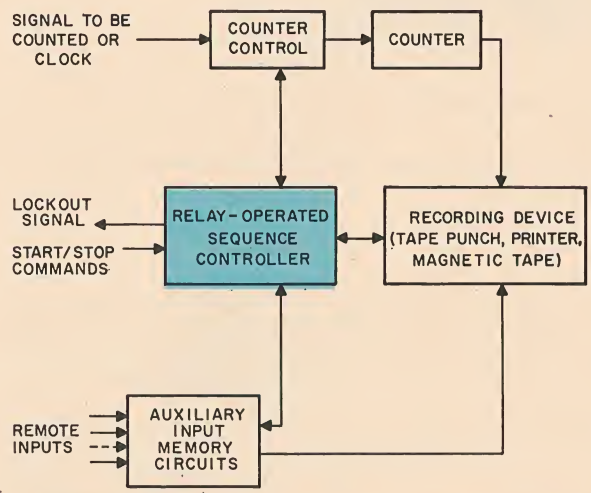
For slow-speed instrumentation, the relay offers distinct advantages over its semiconductor counterpart. It offers simplicity of design, low cost and higher efficiency.

THE TRANSMISSION of electrical signals without distortion or loss of level is essential in data processing. When the rate of data transmission is confined to the low-frequency (audio) range, the relay offers certain advantages. Unlike solid-state devices, it requires neither elaborate auxiliary circuitry nor well-regulated power supplies. In addition it can provide more efficient load carrying and more faithful signal reproduction than its semiconductor counterpart.

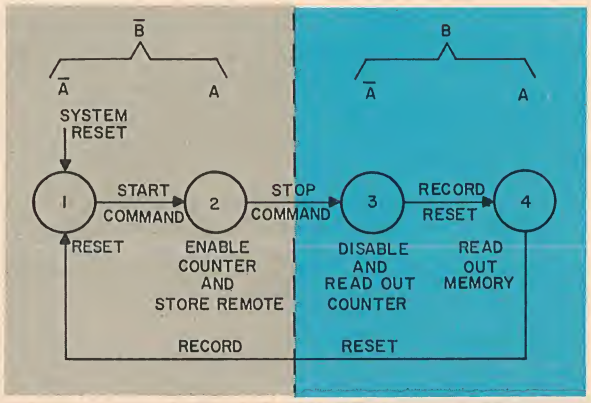
In a variety of instrumentation problems, the end products are printed digital records, punched tapes or magnetic recordings. In data processing for psychological testing particularly, the data change slow enough that relay circuitry is preferred. Many automatic analysis applications, such as heart-rate analysis or industrial batch monitoring systems, fall into this category. An elementary data processor for such applications appears in Fig. 1.

The basic system includes a counter and its associated control circuitry for accumulating data on either time between events or events themselves. A relay-operated sequence controller is used for controlling the entry of data onto the record, entry and reset of the auxiliary memory, and start and stop of the counter. The auxiliary memory element stores such pertinent information as batch number, dates or experiment number. The system is completed with a recording device.

Gordon Silverman
Electronic Engineer
The Rockefeller Institute, New York, N.Y.



1. **Basic data-processing system** uses a relay-operated sequence controller as the key element in the programing sequence.



2. **Phase-state relationships** of a basic data-processing system indicate the sequence of operations.

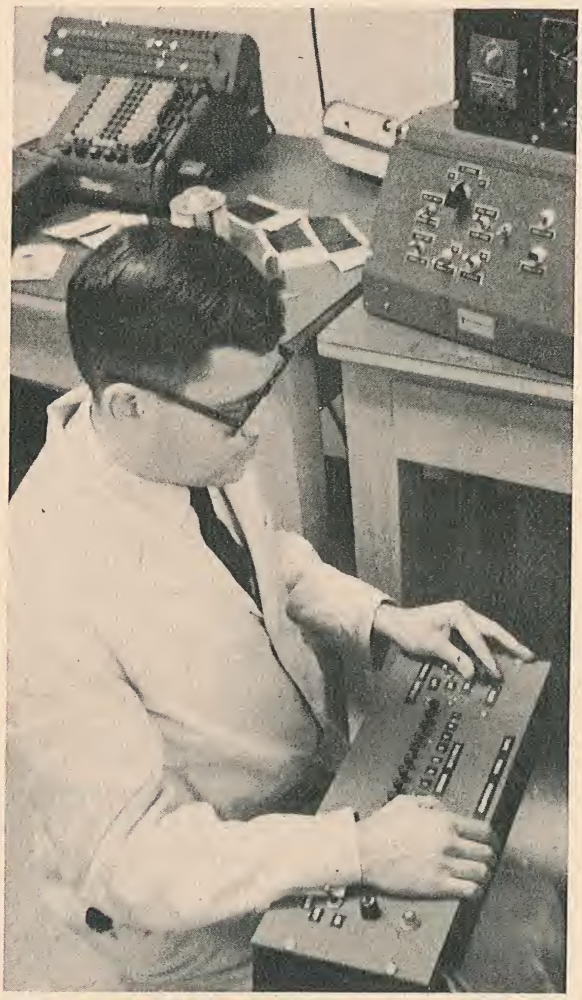
Relay operates sequence controller

The heart of the "wired program" machine is the sequence controller. The design of this device determines the essential characteristics of the data processor. Consider the functional role of the data processing system (as represented in Fig. 2). Here the circles represent the state spaces of the machine, and the arrows represent the sequence of operations. Superimposed on these arrows are the conditions to permit the machine to pass from one state to the next. Since four states are indicated, two bistable circuits are required.

The letters above the state spaces refer to the condition of the two bistable circuits. Thus State 1 (Reset) will be the condition $\overline{A}\overline{B}$ (Not A and not B). State 2 (which readies the counter and samples and stores the remote data) is decoded as $A\overline{B}$ (A and not B). State 3 stops the counter and reads out the number stored in it. Finally the memory is read out during State 4 (AB).

Initially some manual reset is provided, which forces the relay flip-flops into State $\overline{A}\overline{B}$. When the machine receives a command to start from the process it is monitoring (it might be the onset of the first heart beat, the first element in the batch, or the stimulus in the psychological experiment), it sets the first flip-flop. This flip-flop, having been in State $\overline{A}$, is now in State A, and the total machine is in State $A\overline{B}$. The stop command (from the second heart beat, the n^{th} element of the batch, or the proper subject response in the psychological experiment) simultaneously resets A to $\overline{A}$ and sets $\overline{B}$ to B. The machine is now in State $\overline{A}B$, which permits the output to pass to the recorder.

When the recording device has completed its program, it will normally provide a reset signal, which is used to advance the sequencer to its next state. Here the memory bank outputs are recorded. When the recording device has completed its read-out, it again supplies a reset signal. This signal is used to return the sequencer to its reset state (State 1), where it awaits a new start command.



The author, Gordon Silverman, checks the performance of his relay electronic dash-processing system. The relay memory stages use fewer components than their semiconductor equivalents and do not introduce signal distortion.

Relay vs semiconductor criteria

The phase state relationships are:

$$\text{Reset} = \overline{A}\overline{B} \quad (1)$$

$$\text{Enable Counter} = A\overline{B} \quad (2)$$

$$\text{Record Counter} = \overline{A}B \quad (3)$$

$$\text{Record Memory Output} = AB \quad (4)$$

These equations are used for the setting or resetting of a particular flip-flop. Consider the dotted line in Fig. 2. All states to the left of the line are $\overline{B}$ (in addition to either A or $\overline{A}$), and all states to the right are B. Any arrow (or transition) crossing this line from left to right would be a condition for setting the second flip-flop (going to State B). Any arrow crossing this line from right to left would be a condition for resetting this flip-flop to State $\overline{B}$. Thus,

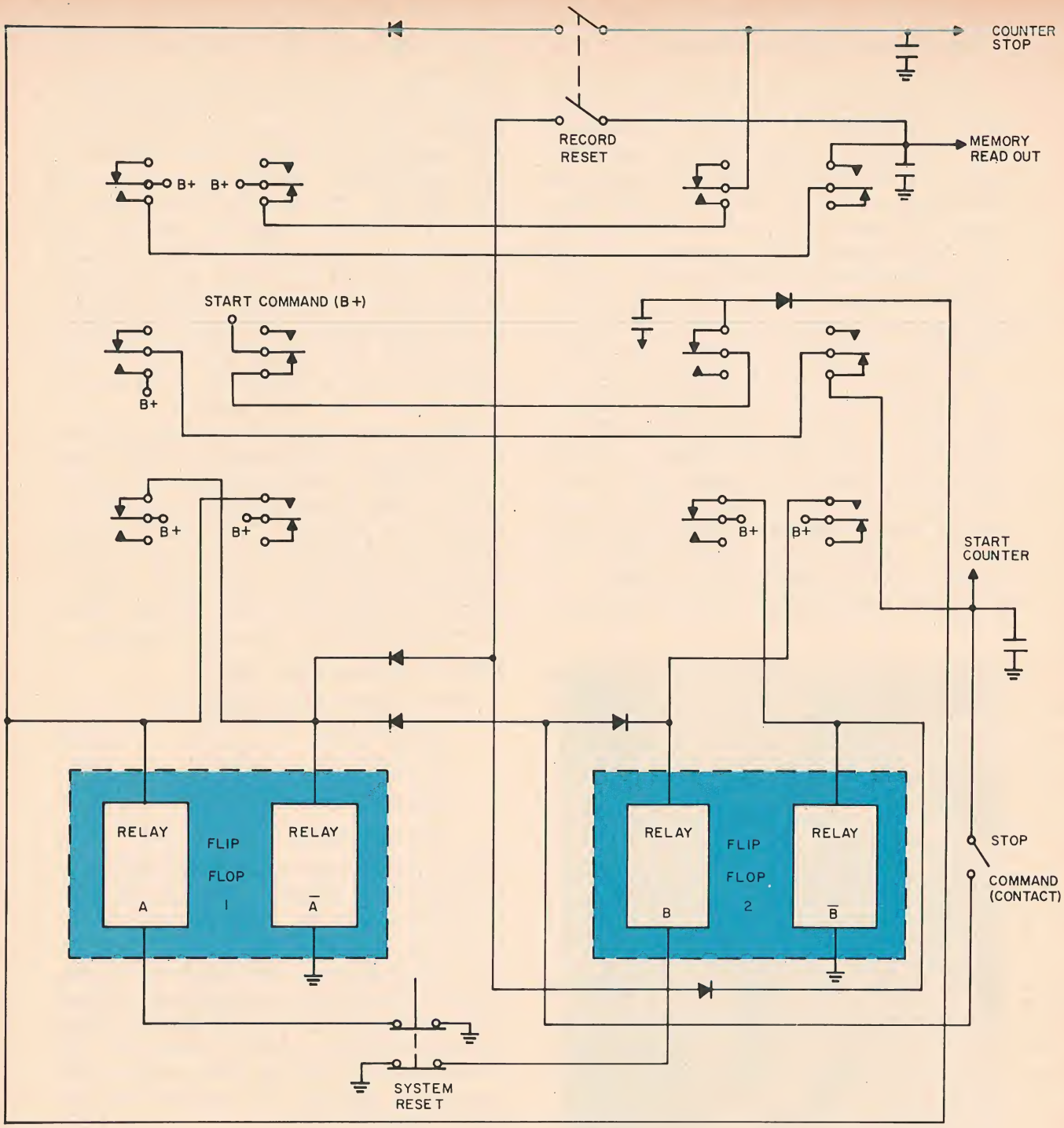
$$B = (\text{State 2}) \text{ and } (\text{Stop Command}) \quad (5)$$

$$\overline{B} = (\text{State 4}) \ (\text{Record Reset}) + (\text{System Reset}) \quad (6)$$

$$A = (\text{Reset}) \ (\text{Start Command}) + (\text{State 3}) \ (\text{Record Reset}) \quad (7)$$

$$\overline{A} = (\text{System Reset}) + (\text{State 2}) \ (\text{Stop Command}) \quad (8)$$

The actual design of the sequencer is thus reduced to a logical arrangement of the phase state diagram and the writing of appropriate equations from it. To contrast the two device approaches (relay and semiconductor) in data-processing systems, a relay-implemented processor and a semiconductorized proces-

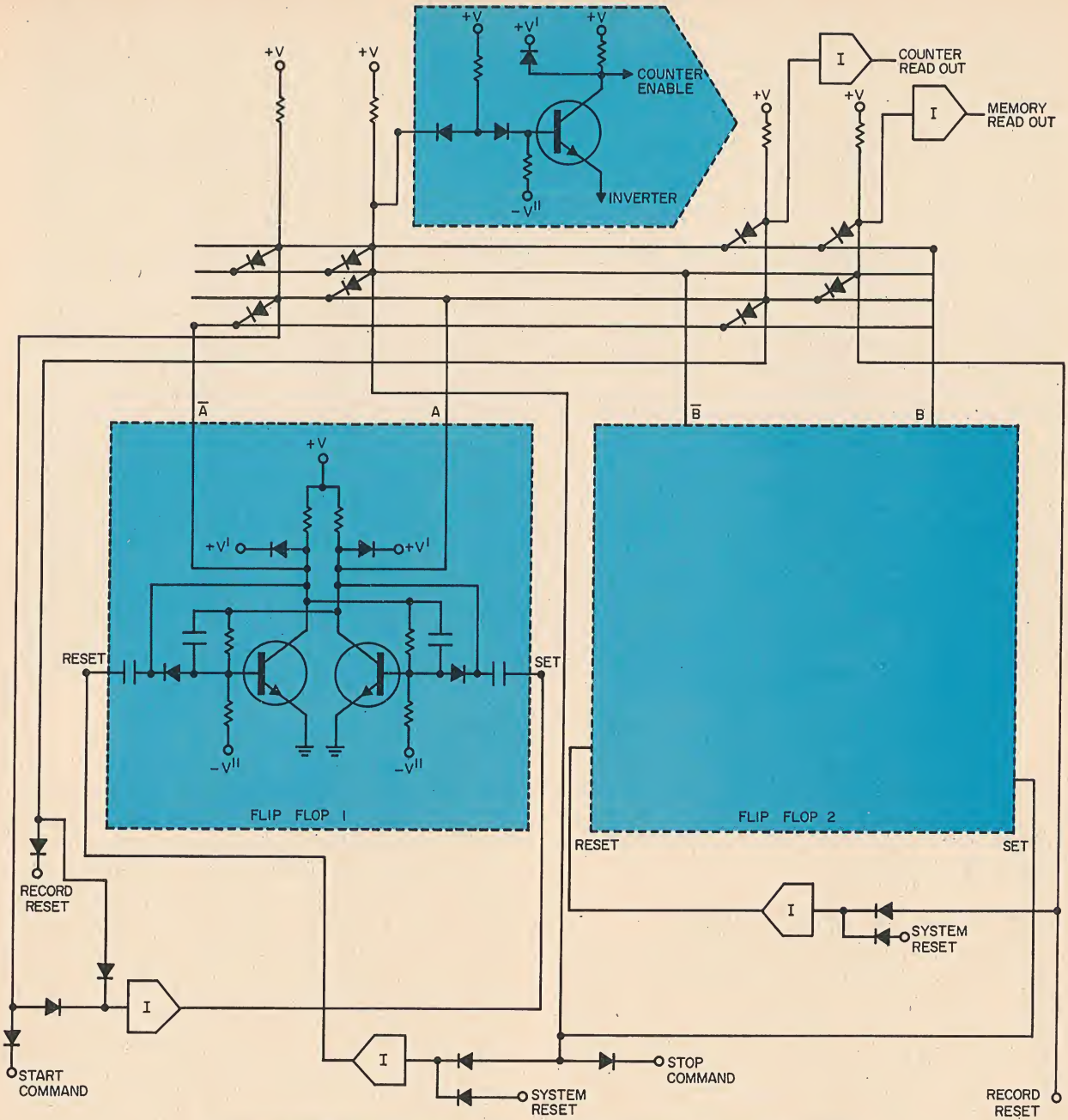


3. **Relay-operated data-processing sequencer** features less complex circuitry than an equivalent semiconductorized system (compare with Fig. 4). The phase-state diagram (Fig. 2) forms the basis for the design of the sequencer, shown in the reset state.

sor are presented (Fig. 3 and Fig. 4, respectively). The relative complexities of these two approaches are readily apparent. Obviously the transistorized approach includes a greater number of components, with added costs of manufacturing.

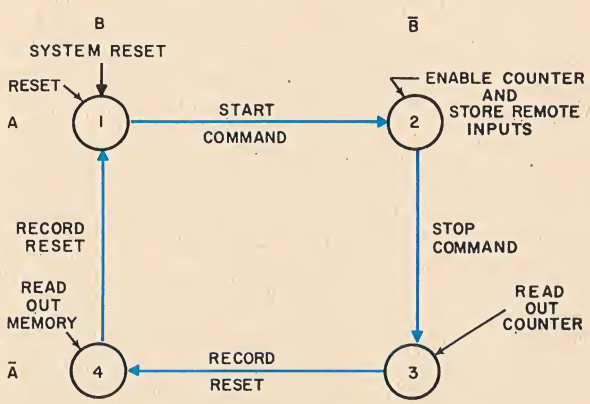
The diagrams reveal critical "race" conditions. Consider, for example, the transition from State 2 to State 3. Notice that flip-flop *1* must go from State A to State $\overline{A}$ and flip-flop *2* from State $\overline{B}$ to State B. If this transition were to occur non-simultaneously, either one of the states AB or $\overline{AB}$ could occur during the transition. If the machine enters either of these states during its transition from State 2 to State 3, it could stop. This is because the only external transition signal present at the time is the stop command. A data block would have thereby been missed. This is one kind of "critical race."

A similar situation exists for the transition from State 4 to State 1. For relay-operated sequencers, multiple relay transitions are to be avoided for this reason. An im-

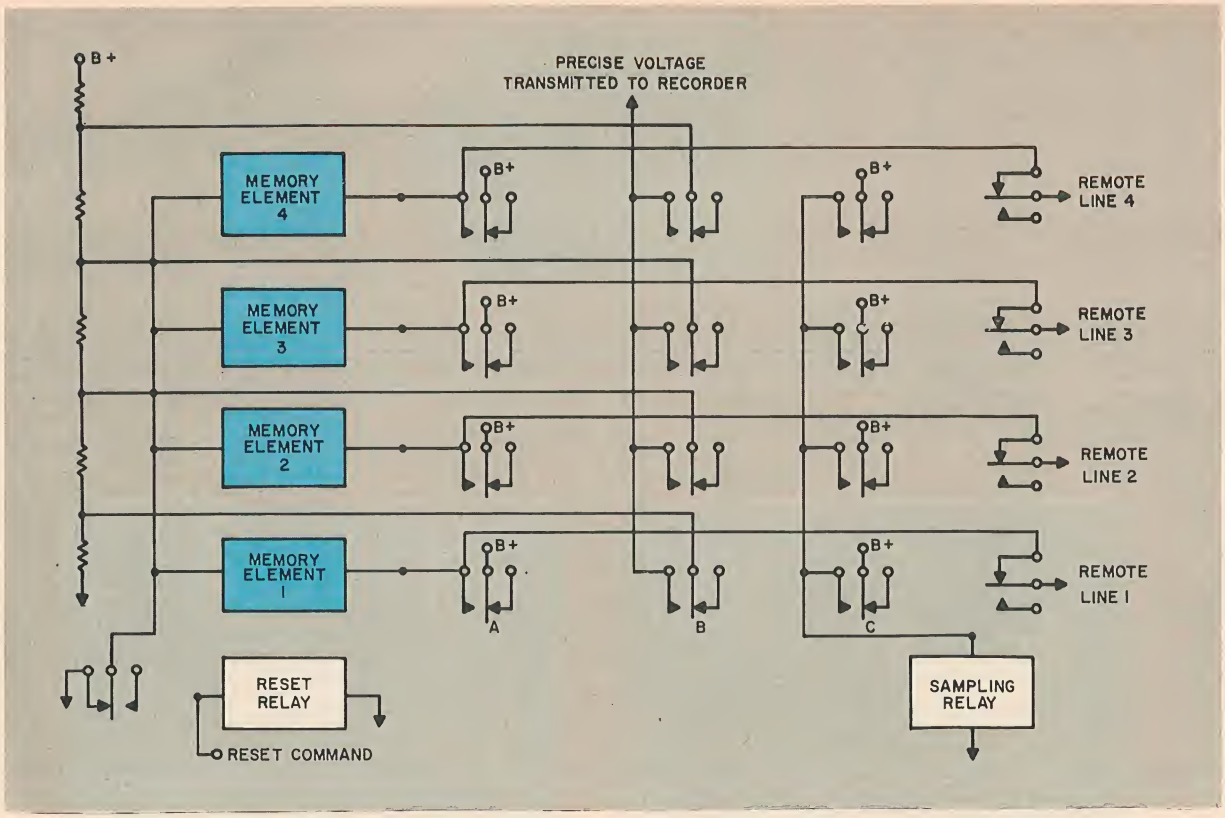


4. **Solid-state sequencer** (the functional equivalent of Fig. 3) is more costly and complex than its relay-based counterpart. This type is more suitable for fast-rate (above audio frequency) data processing.

proved sequencer would employ a "unit-distance" arrangement. In this case only one relay is operated when going from one state to another. A phase-state diagram for this improved sequencer (unit distance) is shown in Fig. 5. Note that in going from any state to the next, only one relay changes state (AB, $A\bar{B}$, $\bar{A}\bar{B}$, $\bar{A}B$, AB, etc). Transition states are occasionally added to achieve a unit-distance phase state diagram. These serve the purpose of assuring that multiple transitions do not occur. For example, a sequencer with two relays (4 binary states) cannot be traversed from a given point back to that point in three steps without a multi-



5. **Modified sequencer phase-state relationships** use a unit-distance separation of functions to insure that multiple transitions do not occur.



6. **Relay memory circuit** features fewer components than semiconductor counterpart and does not distort signal being processed.

ple relay transition. However, the fourth, unused, state could be used as a dummy transition, thus providing a unit-distance design arrangement.

Relay suitable for memory function

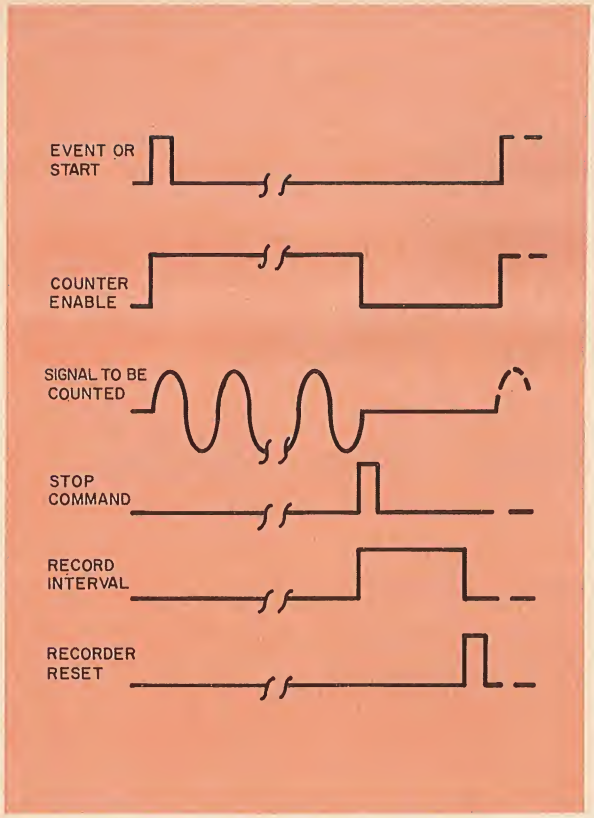
The use of relays in data-processing systems is not limited to the sequencer. It is often required to memorize a particular element of an array (the record number in a batch-recording situation) for subsequent recording. In addition it may be required to transmit to the digital recorder a precise voltage representing this number. These requirements are satisfied by a representative 4-line relay memory (Fig. 6).

If one of the remote lines is energized, a voltage will appear on its corresponding memory element. This relay closes, and consequently locks itself in (via contact set *A*). At the same time the sampling relay is energized (via contact set *C*), disconnecting the remote sources from the memory bank. The selected memory element will then transmit a precise voltage (on the set *B* contacts) to another device (conceivably a digital recorder) for later recording. The memory element remains closed until a reset command is received. The reset relay opens the common ground line, thus unlocking the previously selected memory element. When this system is compared with an equivalent memory bank that uses semiconductors, the latter's increased complexity is evident.

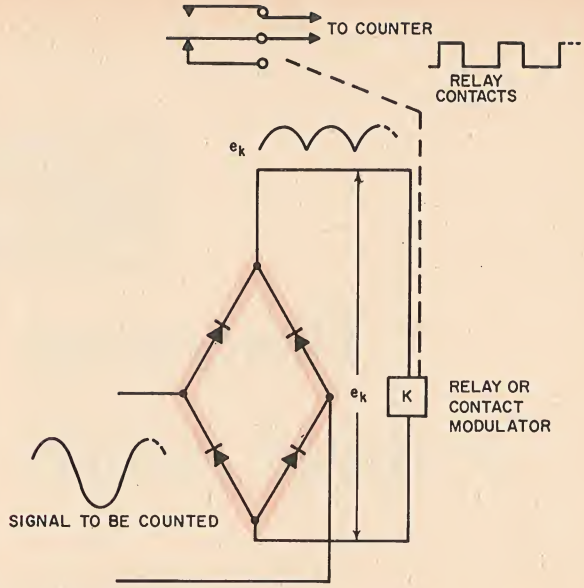
To provide an undistorted voltage to the digital recorder, an electronic transmission gate with matched elements is usually required in the semiconductor case. This is because if the diodes are not matched, their offset voltages (potential drop at the operating current level) will not cancel, causing a voltage other than the true input to appear at the output. In addition, to turn the gate ON and OFF at the appropriate times requires a gate driver. The driver applies the proper bias voltages to the control diodes, so as either to back-bias the signal diodes (thus cutting off the transmission gate) or to back-bias the control diodes (allowing the output voltages to be controlled by the input signal and thereby turning the transmission gate on). These elements significantly add to the cost of such a memory. Additional supply voltages are also required. The equivalent relay system avoids these additional elements.

Limitations of relay EDP

Data-processing systems of the type considered thus far are asynchronous. The commands from the external system are not dependent on any internal data-processor clock. As such, the maximum data rate is



7. **Timing sequence of a relay processor.** Relay transition times limit the data-processing rate.



8. **Diode-bridge and relay** arrangement for improving the resolution times of the counter element in the relay EDP system.

limited by the time necessary to process the data. A timing chart for the processor appears in Fig. 7.

Two limitations must be considered when employing relay data-processing circuits. The first is the error in the timing (or measure of the event) introduced. This error arises because of the finite time required for the counter to be activated after the receipt of a start signal (shown as simultaneous in the illustration). This reduces to the time required to change the state of a relay flip-flop. The time for this is twice the transition time of a single relay (since one relay must be energized and the second de-energized). A representative time is 60 msec for ordinary (long life) elements.

Special-purpose, high-speed (and higher cost) elements can reduce this time to 5 msec, not including reed relays, which will operate faster but have fewer numbers of contacts. In some situations it may be possible to eliminate this error by anticipating the actual start of the event.

If, in addition to the above error, the signal being counted is asynchronous with the control signals (as would be the case if line frequency were being counted), the error introduced by the counter must be included. If the signal is introduced just after the counter has passed the trip level, the counter will run the first cycle and then have to wait a complete cycle before detecting the first count pulse.

This error, called resolution error, can be cut in half by increasing the resolution. A circuit that will increase the counter resolution is shown in Fig. 8. The relay or contact modulator contacts close twice for each cycle. If the counter now misses the first cycle, it has to wait only a half cycle before the next count pulse.

The second limitation of this data processing system rests in its method of resolving sequential stimuli. After the receipt of a stop command, the recording process begins. When the records are in the form of a printed tape (a digital recorder), the time required to print is typically 30 msec. During this interval, it is necessary to hold the contents of the counter, or memory, fixed, thus making it unavailable for any new, incoming event. The fastest data rates here are about 16 events per second (two words being required for each data block).

Two methods can be suggested to increase these data rates. With a punched paper tape record, the recording can be improved. Representative machines can process a word in 8 msec. Thus, to record a 3-digit (word) counter output and a memory word (a total of 4 words) on 8-level tape, approximately 16 msec are required. The second approach entails a register that stores the counter output while it is being recorded. This frees the counter to accept a new event. The effective data rate is thus doubled, at the expense of an additional register. ■ ■