Optimum Control Logic for Successive Approximation Analog-to-Digital Converters

T. O. Anderson
Communications Systems Research Section

Optimum control logic is a popular subject with designers and manufacturers of high-resolution high-speed low-cost modular analog-to-digital converters of the successive approximation type which are found in abundance in today's module market. Also, with miniaturization being a strong consideration, it may not be long before the complete control logic will be available in a single medium-scale integrated circuit chip. The designs described here may be a strong contender for such chips. One novelty of the designs presented here is found in their optimum logic and in their minimum component count, considering presently available components. Another novelty is that they are modular or iterative, i.e., the logic structure is the same for all bits. A high-resolution converter logic is then simply an extension of the logic for a low-resolution converter.

I. Introduction

This article describes another step in DSN analog-to-digital converter development that provides possible circuits for large-scale integrated circuit (LSI) production which would make the analog-to-digital converters (ADCs) more attractive as to cost, size, power, etc. This would make the Deep Space Station data acquisition and preprocessing system designs more attractive. Optimum design is first qualified in terms of minimum logic and component count for implementation. Because of the rapidly advancing medium-scale integrated circuit (MSI) technology, component count is referenced to the present state of development.

The most common design strategies are discussed in general. Two fundamentally different schemes are elaborated on, and for each one of these, several different logic designs are described in detail. The designs presented here are logically simpler and their component count smaller than those most commonly found. Of special interest in these designs is the fact that they are

modular or iterative, i.e., the logic structure is the same for all bits. A high-resolution converter logic is then simply an extension of the logic for a low-resolution converter.

II. Qualification of the Optimum Claim

The claim for these circuits to be optimum, speed of operation being equal or notwithstanding, refers to the total involvement of their implementation. Due to the rapidly decreasing cost of MSI/LSI circuits and due to presently emerging high-speed versions of complex low-power metal-oxide semiconductor (MOS) circuits, the optimum claim also refers to the present state of the art of component technology. The circuits presented here are logically minimum and, in combination with a selective choice of presently available components, should be safely claimed to be optimum.

III. Successive Approximation Logic

The operation in successive approximation logic is well known. It is, in fact, self-evident: the most common operational strategy is as follows: A flip-flop FF1 is set at clock time t_0. It contributes its particular weight to a summing bus, which is one input term to a comparator amplifier. The other is the analog signal to be converted. The output of the comparator amplifier becomes the control bus which determines whether the FF1 shall remain set or whether it shall be reset at t_1. At t_1 the next most significant flip-flop FF2 is set and the same simple procedure is repeated. Any FF is operable for only two clock pulses per conversion cycle which proves to be a useful observation in deriving a design strategy for optimum implementation.

To set a FF and then again reset it if its weight contribution was too large appears at a first consideration to be cumbersome. A design was explored in which a sequencer functioned as a trial-register and a final register was set only if the trial was affirmative. This design, however, required an OR function between the two registers and, therefore, was abandoned as less than optimum, even though its reasoning appeared attractive.

IV. Two Fundamentally Different Designs

One consists of one sequencer and one code register, and the other consists of a single set of FFs connected so as to simultaneously function both as sequencer and code register, referred to in the following as sequencer/code register design.

V. Sequencer and Register Design No. 1

Figure 1 is a logic diagram of one of the most commonly used sequencer and code register designs. This connection is very simple and straightforward.

The sequencer is a shift register which is initially reset for each conversion cycle and shifts as "1" through the register. An output from the sequencer sets a FF in the code register through its DC set input. The output from the FF that is being set is then used as a clock for conditional reset of the previous FF. The control bus is the data input to all code register FFs.

Shown in Fig. 2 are some additional connections, for example, how to conditionally turn off the code register FF for the least-significant bit (LSB). One additional FF in the sequencer is used. This FF is also used in the CONTINUOUS MODE/MANUAL MODE mechanism. In the manual mode its output is used to control the clock enable gate. As the last FF turns on, it inhibits the clock, and the sequencer "hangs up" until manually reset through the manual start signal. In the continuous mode the turn on of the last FF will automatically reset the sequencer and a new conversion cycle will start automatically. The last FF will then also turn itself off. With present components designed for high-speed of operation, this type of connection can easily be made to operate reliably with very little delay or energy storage in the reset line.

The component count for the design discussed above can be reduced as the price for MSI serial in-parallel out (SIPO) shift register packages decreases. Typical shift register packages presently available that would be applicable are 74164, which is an 8-bit SIPO, and 8273, which is a 10-bit SIPO shift register.

VI. Sequencer and Register Design No. 2

In the design discussed above, the conditional turn-off of a FF in the code register is clocked by the secondary effect of the succeeding FF that is turning on. This mechanism then exhibits a slight but unnecessary delay when considering the connection shown in Fig. 3. This is also a simple connection and it works as follows: as the output from the sequencer turns on a code register FF,

the same output from the sequencer turns off the preceeding FF. As shown in the figure, the Q output from the sequencer will DC set a code register FF while the positive transition of the Q output from the same stage of the sequencer provides the clock for the conditional turn-off of the preceeding code register FF.

VII. Sequencer and Register Design No. 3

As mentioned previously, each FF in the code register is set and conditionally reset once per conversion cycle. This reasoning suggests a scheme in which the outputs from the sequencer are two-pulse outputs used as clock terms for code register FFs of the J/K type. The first pulse will set and the second will conditionally reset the FF. For a maximum speed of operation, the second pulse of one output should coincide with the first pulse of the succeeding output. Figure 4 is a logic diagram of such a connection.

The sequencer is simply a shift register which is initially reset to 11000 . . . 00 and shifted right. The shift clock is a common term in the output enable gates. This connection assures overlapping double pulse outputs. Figure 5 shows the truth table for the sequencer outputs.

VIII. Maximum Speed Sequencer

The sequencers described in the previous three designs have been designed as shift registers where a single '1' or '1' have been shifted through a register from a reset condition. The shift occurs on a transition of the shift clock which then must be a pulse train with pulses of a certain duration occuring at a certain rep rate. At higher speeds the clock then looks like a square wave of a certain symmetry. At maximum speed the clock may be an entirely symmetrical square wave.

If one would operate every other stage in such a shift register on the complement of the clock square wave, as shown in Fig. 6, a '1' (or any other preset pattern) would travel through the register at twice the rate it would if all stages operated on the same clock. The code output from any one tap would overlap that of the preceding stage with half a clock period. For distinctive nonoverlapping pulse outputs, the output should be enabled with the clock to that stage as shown in Fig. 6.

A timing chart for this connection is shown in Fig. 7. A single '1' is considered propagating through the register. The general scheme of fast propagation of successive approximation logic described here for a sequencer is also valid for designs where a single set of FFs simultaneously functions both as sequencer and code register.

IX. Sequencer/Register Design No. 1

The sequencer/register expression implies that a single set of FFs is connected so as to simultaneously function both as a sequencer and a code register. The design strategy is as follows:

With all FFs reset their zero condition is serially AND-ed in a series of AND gates from the least significant to the most significant, from right to left as shown in Fig. 8.

A shift "1" connection from the most significant bit to the least significant bit from left to right is easily identifiable.

In Fig. 8 the clock is common to all FFs; however, for maximum speed, every other stage may be operated on the complement of the clock as described above. The turn-off term is also common and so is the control bus, the output from the comparator amplifier. Both the turn-on and turn-off terms for each FF are controlled by the AND-ed zero condition for all the bits of lesser significance.

J/K FFs with 2 term AND functions for both the J and the K inputs such as SN 74105 are used.

The operation is as follows: All FFs are reset. On the next clock pulse the most significant FF will set, since it is the only one that has a true input. The control bus is not true; even if it were, the first FF would still set, since it is a J/K FF. On the next clock pulse the first FF will conditionally reset. The second FF will set because the first FF is set and provides a true input. All other FFs are reset and provide the enable term through the series AND gates. Once the second FF is set, both inputs to the first FF are entirely disconnected for the remainder of the conversion cycle. On the third clock pulse the second FF may turn off but the third FF will turn on and disconnect both the first and the second FFs. Each FF is then enabled or operative for exactly two clock pulses.

Of special interest is the fact that the series propagation of the all-zero condition, through a series of AND-

gates, occurs between conversion cycles at which time the analog circuits must be allowed the settling time for zero to full-scale excursion. During the conversion cycle the gates are then settled and in a "waiting" condition.

Because of pin limitation, a 16-pin dual in-line package (DIP) can accommodate only a single J/F FF with two enable terms for each input. This makes the chip count for this connection somewhat less than optimum. A slight modification will improve this condition.

X. Sequencer/Register Design No. 2

The component count can be reduced if one uses a simpler FF of which there are two per package. This connection then requires external gates. Two J/K FFs are available in one 16-pin DIP; for example, SN7476. Four 2-input gates are contained in one 16-pin DIP; for example, SN7408.

A simple design which will decrease the component count is shown in Fig. 9. The basic strategy of series propagating of the all-zero condition as discussed in the previous case is maintained. The enable term operates on a single gate for each FF. This gate is then the clock gate. In the previous case the J/K inputs were enabled for two clock pulses. In this case the J/K inputs are never disconnected, only the clock gates are enabled for two clock pulses each.

XI. Conclusion

Because of the large number of systems in the DSN using analog-digital-analog conversion, the subject of optimum logic for a successive approximation analog-to-digital conversion is important. This article has given examples of logic which are presently used as well as examples which are not used presently. Two basic types have been discussed, one with a sequencer and a code register as two separately distinguishable items and one where a single set of FFs is connected through auxiliary gates to simultaneously function both as a sequencer and a register. Auxiliary control logic is suggested as well as a connection for speeding up the conversion cycle. Components are suggested but not specified.

The purpose of this article is then to provide guidance for the design of DSN standard analog-to-digital converter modules rather than to provide an absolute or rigid detail design. Also, the article can provide a first step in the design and layout of a single LSI component as a DSN standard analog-to-digital conversion module. Such a module is expected to be more important in future DSIF systems as more and higher frequency operations become digitized.

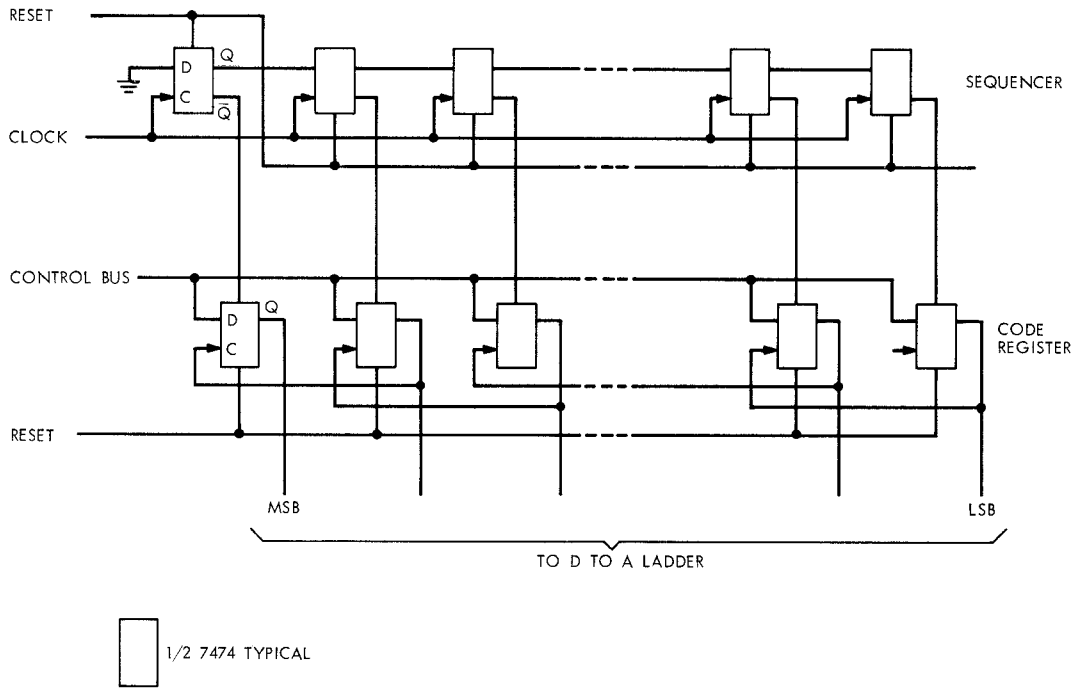

Fig. 1. Sequencer and code register with conditional secondary effect turn-off of code register FFs

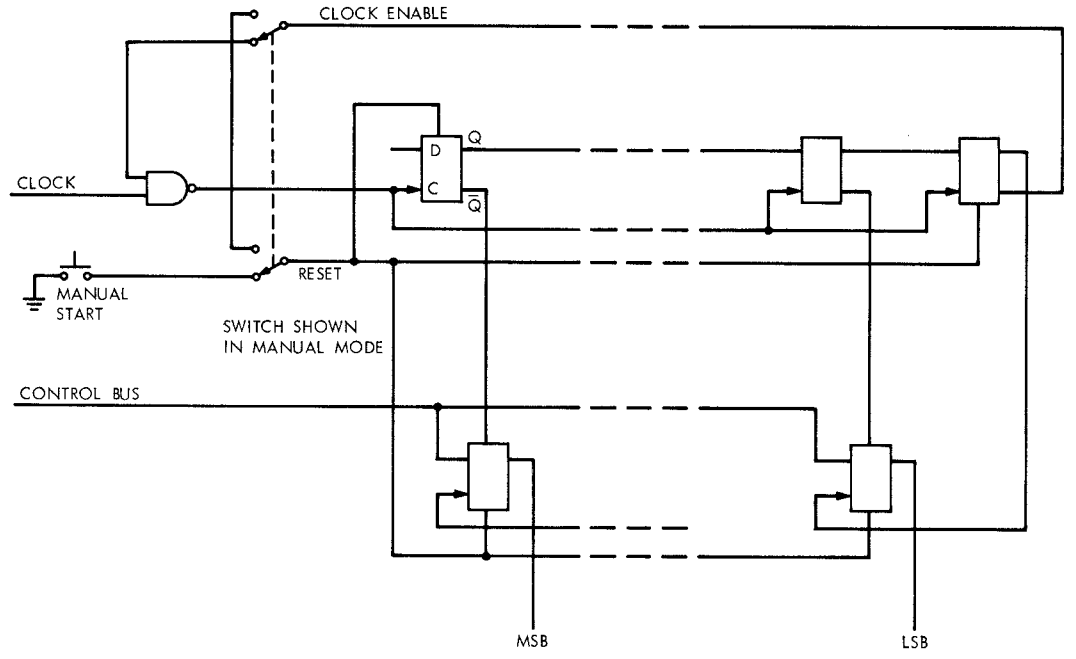

Fig. 2. Continuous or external start conversion control logic

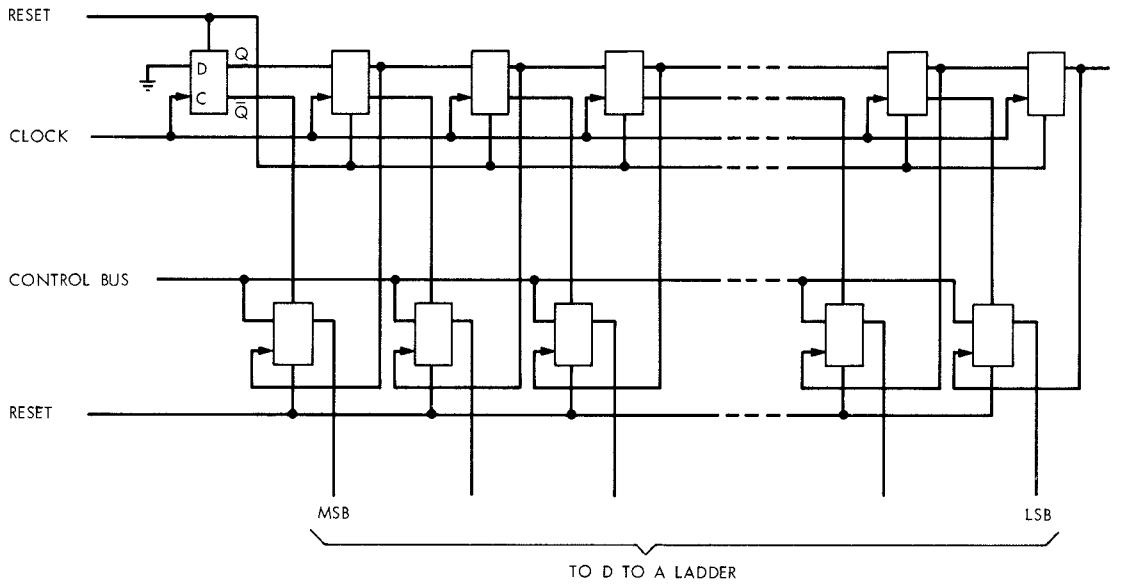

Fig. 3. Sequencer and code register with conditional direct sequencer turn-off of code register FFs

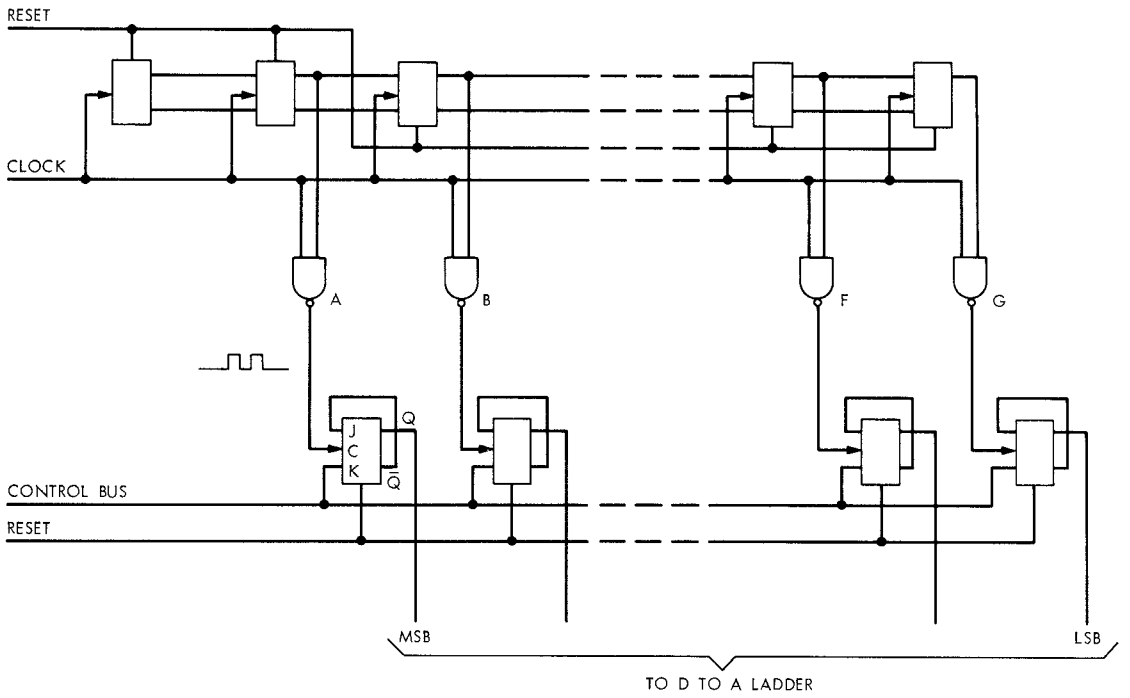

Fig. 4. Overlapping double pulse sequencer and simple J/K code register

A	B	C	D	E	F	G	H
1	1	0	0	0	0	0	0
0	1	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	0	0	1	1	0	0	0
0	0	0	0	1	1	0	0
0	0	0	0	0	1	1	0
0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	1

Fig. 5. Overlapping double pulse sequencer truth table

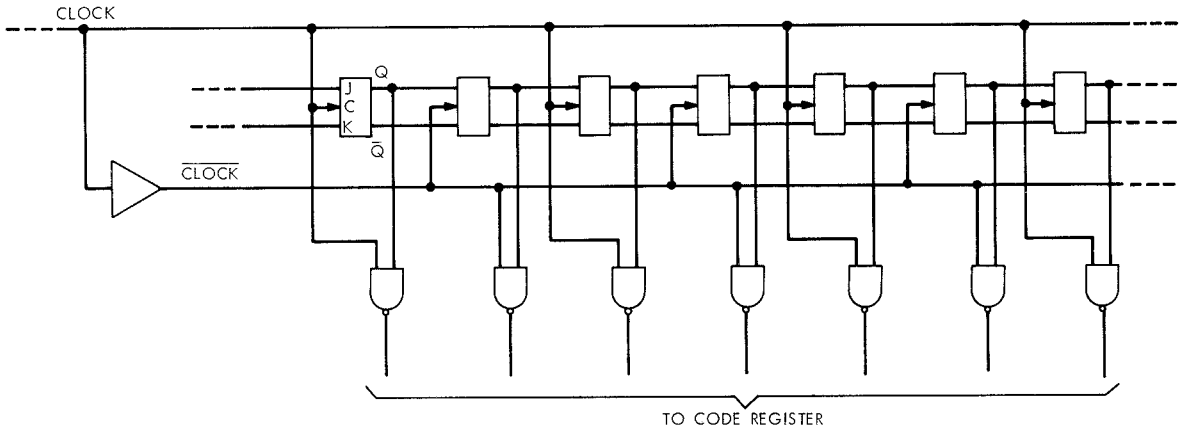

Fig. 6. Maximum speed sequencer connection using alternating true and complement clock phase

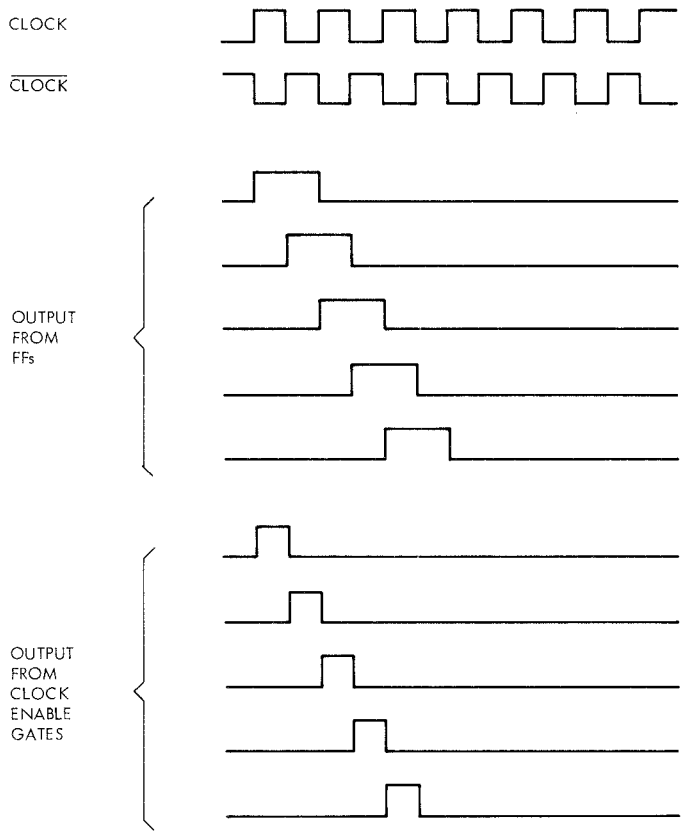

Fig. 7. Timing chart for maximum speed sequencer connection

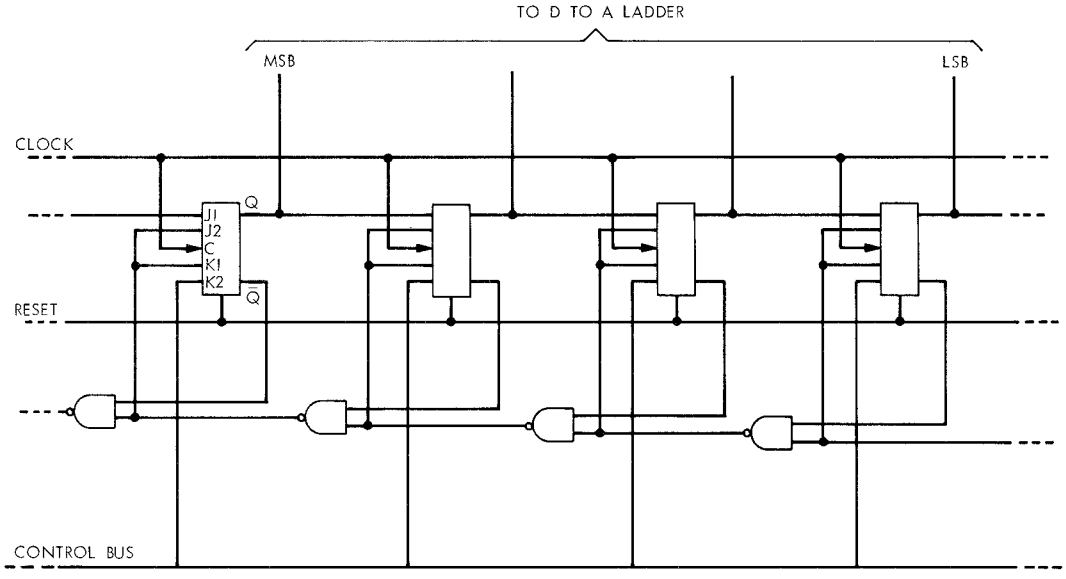

Fig. 8. Sequencer and code register using a single set of J/K FFs plus steering gates

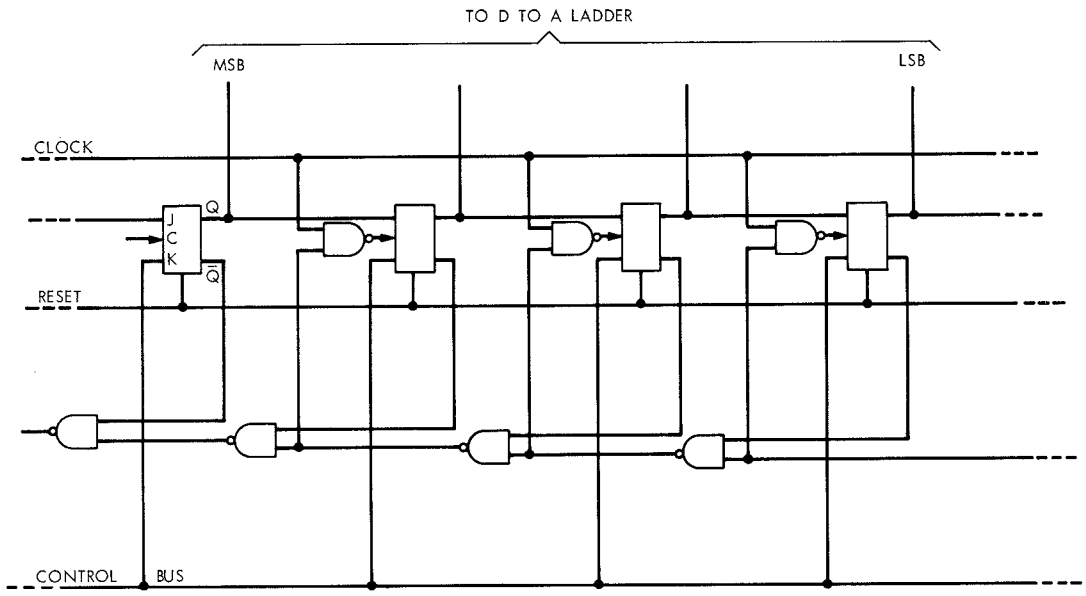

Fig. 9. Sequencer and code register using a single set of simple J/K FFs plus steering gates