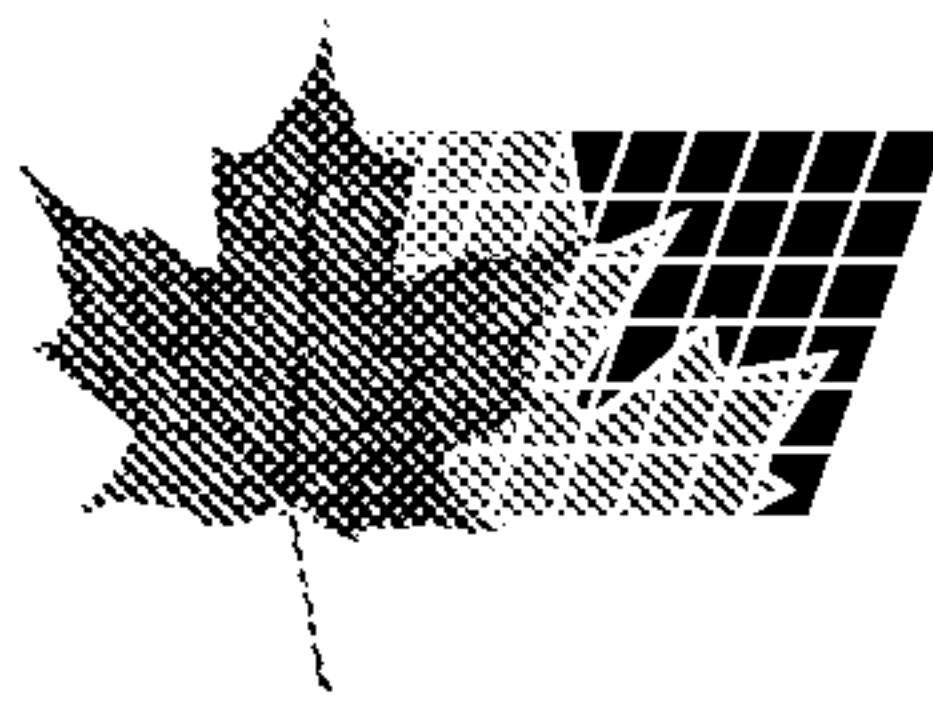




(11) (21) (C) **2,238,294**
(22) 1998/05/20
(43) 1998/11/21
(45) 2000/12/26

(72) SHIMADA, Michio, JP

(73) NEC CORPORATION, JP

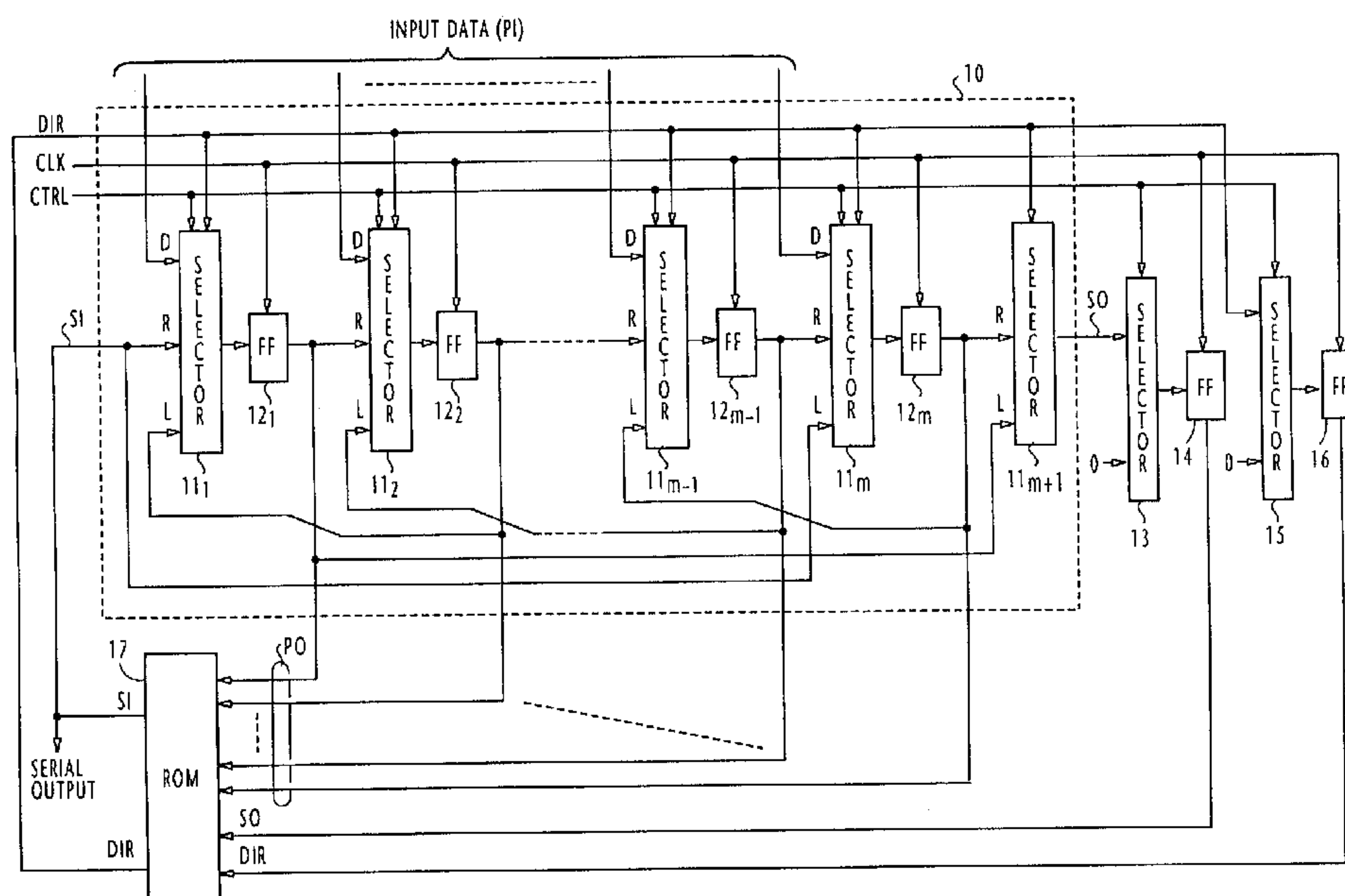
(51) Int.Cl.⁶ G06F 7/58, H04L 9/00

(30) 1997/05/21 (9-146072) JP

(54) **GENERATEUR DE SUITES DE NOMBRES**

**PSEUDOALEATOIRES COMPORTANT UN REGISTRE A
DECALAGE BIDIRECTIONNEL ET UN CIRCUIT DE
REACTION UTILISANT UN GRAPHE EULERIEN**

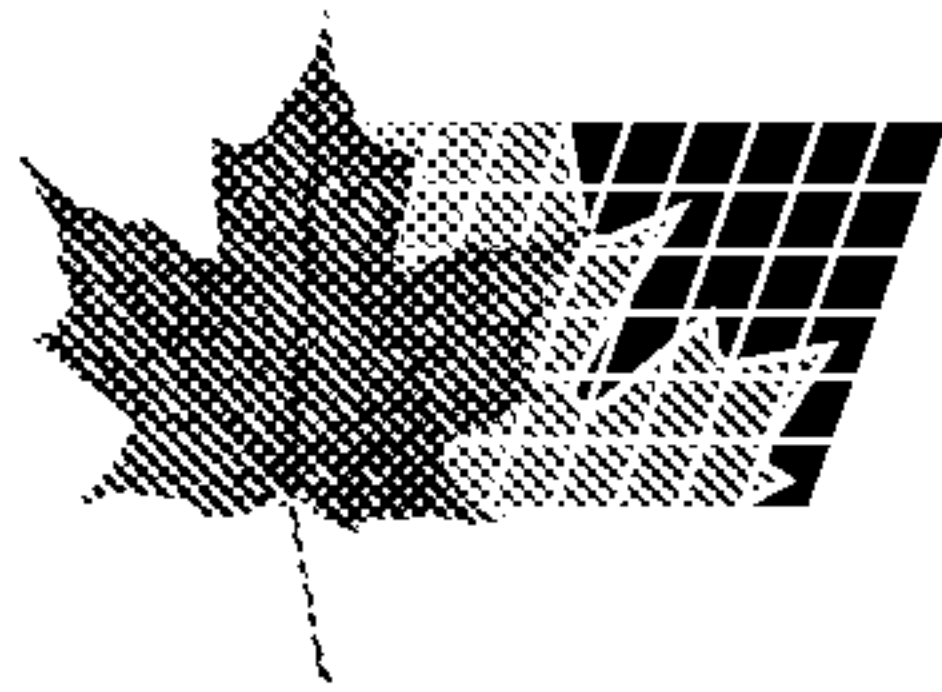
(54) **PN SEQUENCE GENERATOR WITH BIDIRECTIONAL SHIFT
REGISTER AND EULERIAN-GRAPH FEEDBACK CIRCUIT**



(57) L'invention est un générateur de suites de nombres pseudoaléatoires qui comprend un registre à décalage construit de façon à être chargé avec une suite multibit. Ce registre à décalage réagit à la réception de l'i-ème impulsion d'horloge et de l'i-ème bit de contrôle de direction pour décaler la suite multibit dans l'une de deux directions en transmettant l'i-ème bit de sortie et en recevant l'i-ème bit d'entrée. La suite multibit définit successivement l'un des noeuds d'un graphe eulérien connecté par branchements. Un circuit de réaction est connecté au registre à décalage pour convertir les données d'entrée. Ces données d'entrée sont constituées

(57) A pseudorandom number sequence generator comprises a bidirectional shift register arranged to be loaded with a multi-bit sequence. The shift register is responsive to an ith clock pulse and an ith direction control bit for shifting the multi-bit sequence in one of two directions, delivering an ith output bit and receiving an ith input bit. The multi-bit sequence successively defines one of nodes of an Eulerian graph connected by branches. A feedback circuit is connected to the shift register for converting a set of input data to a set of output data. The input data comprises a multi-bit sequence stored in the shift register in response to an





(11) (21) (C) **2,238,294**
(22) 1998/05/20
(43) 1998/11/21
(45) 2000/12/26

d'une suite multibit qui est stockée dans le registre à décalage en réponse à la réception de l'(i + 1)-ème impulsion d'horloge, de l'i-ème bit de sortie et de l'i-ème bit de contrôle de direction et les données de sortie sont constituées de l'(i + 1)-ème bit d'entrée et de l'(i + 1)-ème bit de contrôle de direction. Ces données de sortie sont transmises au registre à décalage de façon que les suites multibit produites par le registre à décalage se déplacent dans le graphe en n'empruntant chaque branchement qu'une seule fois et produisent l'(i + 1)-ème bit d'entrée pour former une suite de nombres pseudoaléatoires.

(i + 1)th clock pulse, the ith output bit and the ith direction control bit and the output data comprises an (i + 1)th input bit and an (i + 1)th direction control bit. The output data is supplied to the shift register so that multi-bit sequences produced by the shift register move around the graph following every branch exactly once, and outputting the (i + 1)th input bit to form a pseudorandom number sequence.



NE-875

- 15 -

ABSTRACT OF THE DISCLOSURE

1 A pseudorandom number sequence generator comprises a
2 bidirectional shift register arranged to be loaded with a multi-bit sequence.
3 The shift register is responsive to an *i*th clock pulse and an *i*th direction
4 control bit for shifting the multi-bit sequence in one of two directions,
5 delivering an *i*th output bit and receiving an *i*th input bit. The multi-bit
6 sequence successively defines one of nodes of an Eulerian graph connected
7 by branches. A feedback circuit is connected to the shift register for
8 converting a set of input data to a set of output data. The input data
9 comprises a multi-bit sequence stored in the shift register in response to an (*i*
10 + 1)th clock pulse, the *i*th output bit and the *i*th direction control bit and
11 the output data comprises an (*i* + 1)th input bit and an (*i* + 1)th direction
12 control bit. The output data is supplied to the shift register so that multi-
13 bit sequences produced by the shift register move around the graph
14 following every branch exactly once, and outputting the (*i* + 1)th input bit
15 to form a pseudorandom number sequence.

NE-875

- 1 -

TITLE OF THE INVENTION

"PN Sequence Generator with Bidirectional Shift Register and
Eulerian-Graph Feedback Circuit"

BACKGROUND OF THE INVENTIONField of the Invention

The present invention relates generally to pseudorandom number sequence generators, and more specifically to a pseudorandom number sequence generator, or nonlinear feedback shift register, suitable for producing stream ciphers.

Description of the Related Art

Stream ciphers are used to scramble important data to protect it from being tapped or tampered with by unauthorized attempts. Nonlinear feedback shift registers are most often used to produce a pseudorandom number (PN) sequence. Details of this type of nonlinear feedback shift register are described in "Analysis and Design of Stream Ciphers", R. A. Rueppel, Springer-Verlag, 1986. The prior art nonlinear feedback shift register is based on a Hamiltonian graph which is concerned with the possibility of travelling around a graph, going along branches (edges) in such a way as to visit every node (vertex) exactly once. Usually, a trial-and-error method is used to discover a Hamiltonian cycle if the shift register has a small number of stages. However, the trial-and-error method is impractical for a shift register if it is required to produce a PN sequence of long period. In addition, if a Hamiltonian cycle were found, the resultant PN sequence would contain a substantially equal number of 0's and 1's which cannot be logically compressed for implementation of simple circuitry.

SUMMARY OF THE INVENTION

It is therefore an object of the present invention to provide a PN sequence generator that can be designed with ease and implemented by simple circuitry so that it produces PN sequences of desired periods.

To obtain this object, the present invention is based on the Eulerian

NE-875

- 2 -

1 graph which is concerned with the possibility of travelling around a graph,
2 going along branches (edges) in such a way as to use every branch exactly
3 once.

4 According to a first aspect of the present invention, there is provided
5 a pseudorandom number sequence generator comprising a bidirectional
6 shift register arranged to be loaded with a multi-bit sequence, the shift
7 register being responsive to an *i*th clock pulse and an *i*th direction control bit
8 for shifting the multi-bit sequence in one of two directions, delivering an *i*th
9 output bit and receiving an *i*th input bit, wherein the multi-bit sequence
10 successively defines one of a plurality of nodes of a graph connected by
11 branches. A feedback circuit is provided, which is connected to the shift
12 register for converting a set of input data comprising a multi-bit sequence
13 stored in the shift register in response to an (*i* + 1)th clock pulse, the *i*th
14 output bit and the *i*th direction control bit to a set of output data
15 comprising an (*i* + 1)th input bit and an (*i* + 1)th direction control bit and
16 supplying the output data to the shift register so that multi-bit sequences
17 move around the graph following every branch exactly once, and outputting
18 the (*i* + 1)th input bit to form a pseudorandom number sequence.

19 According to a second aspect, the present invention provides a
20 method of generating a pseudorandom number sequence, comprising the
21 steps of loading a multi-bit sequence into a bidirectional shift register.
22 Responsive to an *i*th clock pulse, the multi-bit sequence is shifted by one bit
23 position in one of two directions according to an *i*th direction control bit, an
24 *i*th output bit is delivered from the shift register and an *i*th input bit is
25 supplied to the shift register, the *i*th output bit and the *i*th direction control
26 bit are stored. The multi-bit sequence in the shift register successively
27 defines one of a plurality of nodes of a graph connected by branches.
28 Responsive to an (*i* + 1)th clock pulse, a set of input data is formed, which
29 comprises the *i*th output bit, the *i*th direction control bit and a multi-bit
30 sequence stored in the shift register. Responsive to the (*i* + 1)th clock pulse,

NE-875

- 3 -

1 the set of input data is converted to a set of output data comprising an $(i + 1)$
2 th input bit and an $(i + 1)th$ direction control bit, shifting the multi-bit
3 sequence along the shift register by one bit position in one of two directions
4 according to the $(i + 1)th$ direction control bit, and supplying the $(i + 1)th$
5 input bit to the shift register, and holding an $(i + 1)th$ output bit delivered
6 from the shift register and the $(i + 1)th$ direction control bit, so that multi-
7 bit sequences move around the graph following every branch exactly once,
8 and outputting the $(i + 1)th$ input bit to form a pseudorandom number
9 sequence.

10 BRIEF DESCRIPTION OF THE DRAWINGS

11 The present invention will be described in further detail with
12 reference to the accompanying drawings, in which:

13 Fig. 1 is a block diagram of a prior art pseudorandom number
14 sequence generator;

15 Fig. 2 is a state transition graph of a 3-stage shift register of Fig. 1;

16 Fig. 3 is an illustration of contents of the ROM of Fig. 1 according
17 to the graph of Fig. 2;

18 Fig. 4 is a block diagram of a pseudorandom number sequence
19 generator according to one embodiment of the present invention;

20 Fig. 5 is a state transition graph of a 3-stage bidirectional shift
21 register of Fig. 4;

22 Fig. 6 is an illustration of contents of the ROM of Fig. 4 according
23 to the graph of Fig. 5;

24 Fig. 7 is a block diagram of a modified embodiment of this
25 invention;

26 Fig. 8 is an illustration of contents of the ROMs of Fig. 7 according
27 to the graph of Fig. 5; and

28 Fig. 9 is a block diagram of a further modification of this invention.

29 DETAILED DESCRIPTION

30 Before proceeding with the detailed description of the present

NE-875

- 4 -

1 invention, it may be helpful to provide an explanation of a prior art nonlinear
2 feedback shift register with reference to Figs. 1 to 3. The prior art nonlinear
3 feedback shift register has an m-stage shift register 1 and a function generator
4 which is implemented by a read-only memory 2. Shift register 1 includes
5 selectors 3_1 to 3_m and flip-flops 4_1 to 4_m alternately connected in a series
6 circuit so that the output of the selector of each stage is connected to the data
7 input of the flip-flop of the same stage. Each selector has two input data
8 terminals A and B.

9 Individual bits of multi-bit input data (PI) are supplied to the data
10 terminals A of selectors 3_1 to 3_m . The output of the flip-flop of each stage is
11 connected to the B terminal of the selector of the next stage. Shift register 1
12 produces multi-bit parallel output data (PO) from the outputs of the flip-
13 flops 4_1 to 4_m . The ROM 2 is supplied with this output data PO to convert
14 its parallel bit pattern to a single data bit. The output of the ROM represents
15 a pseudo-random number which is supplied to the B terminal of the selector
16 3_1 of the first stage as serial input data (SI) as well as to an external utilization
17 circuit, not shown.

18 Selectors 3_1 to 3_m respond to a logic-0 control signal for selecting
19 the signal at their terminal A and respond to a logic-1 control signal for
20 selecting the signal at their terminal B. Flip-flops 4_1 to 4_m are driven by
21 clock pulses so that they hold the selected data bits in response to a clock
22 pulse and shift them along to the succeeding flip-flops in response to the
23 next clock pulse. The bit sequence stored in the flip-flops 4_1 to 4_m at a
24 particular instant represents an internal state of the shift register of that
25 instant. With the control signal being at logic level 0, the multi-bit parallel
26 input PI is loaded into the flip-flops as an initial internal state of the shift
27 register in response to a clock pulse. When the control signal is at logic level
28 1, the internal state of the shift register changes when the flip-flops are
29 clocked so that their outputs are shifted to the next by one bit and a new
30 input bit is supplied to the shift register.

NE-875

- 5 -

1 The internal state of the shift register 1 varies and the way it changes
2 is a process of selecting one of all possible bit sequences. Fig. 2 is a state
3 transition graph illustrating all possible internal states of a 3-stage shift
4 register of the prior art. The three-digit number in each circle represents
5 one of all possible internal states of the shift register. Solid and broken lines
6 indicates all possible branches that can occur. The arrowheads attached to
7 the solid and broken lines indicate a change from a given state to the next
8 when logic level 0 or 1 is entered to the given state as an input bit SI to the
9 shift register and one-bit shifted to the right. For example, state 000
10 changes to state 100 when a "1" is entered to its leftmost bit position and all
11 three bits are one-bit shifted to the right. If 0 and 1 are entered to state 000
12 and state 111, respectively, the internal state of the shift register does not
13 change at all.

14 According to graph theory, the circles of Fig. 2 correspond to nodes
15 of a graph and all lines correspond to branches. Design of a nonlinear
16 feedback shift register is reduced to the determination of a Hamiltonian
17 cycle, i.e., a cycle that begins with a starting node and passes every node
18 (vertex) only once in its way back to the starting node. In Fig. 2, the solid
19 lines indicate the paths followed by the Hamiltonian cycle.

20 Fig. 3 shows a conversion table implemented by the ROM 2 for
21 mapping one-bit input data (SI) to 3-bit output data (PO) in the case of the
22 3-bit shift register of Fig. 2 whose internal state changes according to an
23 Hamiltonian cycle. A sequence of pseudorandom numbers "11101000" is
24 repeatedly produced by the ROM 2.

25 Although a Hamiltonian cycle can be easily discovered in such a
26 simplified example using a trial-and-error method, this method is
27 impractical if the number of nodes is increased in order to maximize the
28 period of a pseudo-random number sequence. In addition, if a
29 Hamiltonian cycle were found, the resultant PN sequence would contain a
30 substantially equal number of 0's and 1's which cannot be logically

NE-875

- 6 -

1 compressed for implementation of simple circuitry.

2 The present invention is based on the Eulerian graph which is
3 concerned with the possibility of travelling around a graph, going along
4 branches (edges) in such a way as to use every branch exactly once. More
5 precisely, if each node of a graph has two or four branches emanating from
6 it, there exists a path in the graph that passes every branch exactly once. It is
7 well known that discovery of such a path is simple.

8 In Fig. 4, a pseudorandom number sequence generator according to
9 one embodiment of the present invention is illustrated. The PN sequence
10 generator of this invention comprises a bidirectional shift register 10 having
11 "m" stages of selectors 11 and flip-flops 12 and an additional selector
12 11_{m+1} . Each of selectors 11_1 to 11_m has three data input terminals D, R
13 and L. For each selector, the data terminal D is a terminal for receiving an
14 individual bit of an input bit sequence PI except for all 0's and all 1's, the
15 terminal R for receiving an input bit from a left-side stage when the shift
16 register is one-bit shifted to the right, and the terminal L for receiving an
17 input bit from a right-side stage when the shift register is one-bit shifted to
18 the left.

19 The outputs of flip-flops 12_1 to 12_{m-1} are connected to the terminal
20 R of the selectors 11_2 to 11_m , respectively, and the outputs of flip-flops 12_2
21 to 12_m are connected to the terminal L of the selectors 11_1 to 11_{m-1} ,
22 respectively. The output of flip-flop 12_1 is connected to the terminal R of
23 selector 11_2 and to the terminal L of selector 11_{m+1} and the serial data input
24 SI is supplied to the terminal L of selector 11_m as well as to the terminal R
25 of selector 11_1 .

26 Direction control signal (DIR) and mode control signal (CTRL) are
27 supplied to the selectors 11_1 to 11_m . Selector 11_{m+1} responds only to the
28 direction control signal for receiving the output of flip-flop 12_m or flip-flop
29 12_1 to produce a serial output bit SO.

30 When the mode control signal CTRL is "0", the multi-bit input

NE-875

- 7 -

1 sequence PI is loaded into the flip-flops 12_1 to 12_m in response to a clock
2 pulse. When the control signal CTRL is "1" and the direction control signal
3 DIR is "1", the shift register 10 causes its internal state to move one bit
4 position to the right in response to a clock pulse, delivering the rightmost bit
5 as an output bit SO from the flip-flop 12_m to the selector 11_{m+1} and
6 receiving an input bit SI from a function generator, or ROM 17 at the
7 terminal R of selector 11_1 . With CTR L= 1 and DIR = 0, the shift register
8 causes its internal state to move one bit position to the left in response to a
9 clock pulse, delivering the leftmost bit as an output bit SO from the flip-
10 flop 12_1 to the selector 11_{m+1} and receiving an input bit SI from the ROM
11 17 at the terminal L of selector 11_m .

12 A selector 13 is provided to respond to a logic-1 level of the mode
13 control signal CTRL for applying the output of selector 11_{m+1} to a flip-flop
14 14 or applying a "0" bit when the control signal is at logic level 0. With the
15 control signal CTRL being at logic level 1, the flip-flop 14 holds the output
16 of selector 13 as the leftmost or rightmost bit of a previous state of the shift
17 register 10.

18 Further provided is a selector 15 which responds to a logic-1 level of
19 the mode control signal CTRL for applying the direction control signal
20 DIR to a flip-flop 16 or applying to it a "0" bit when CTRL is 0. With the
21 control signal CTRL being at logic-1, the flip-flop 16 holds the output of
22 selector 15 as an indication of the direction in which the internal state of the
23 shift register was shifted during the previous clock period.

24 The outputs of flip-flops 11_1 to 11_m are supplied to a mapping table
25 created in a ROM 17 as parallel output data PO and the outputs of flip-
26 flops 14 and 16 are supplied to the ROM 17 as representative of the data
27 SO and DIR of the previous state. ROM 17 provides mapping of the PO,
28 SO and DIR inputs to output bits SI and DIR. The SI output of ROM 17
29 is coupled to the selectors 11_1 and 11_m and the DIR output is used to
30 control the selectors 11_1 to 11_{m+1} .

NE-875

- 8 -

1 Fig. 5 is a state transition graph of the bidirectional shift register 10
 2 of Fig. 4 when it is implemented as a 3-stage shift register. Since all nodes,
 3 except for two-branch nodes 111 and 000, have four branches, Eulerian
 4 cycles do exist in the graph. One Eulerian cycle is indicated by solid and
 5 broken lines which respectively represent the rightward and leftward
 6 directions of the shift register. By travelling around the graph, starting with
 7 node 000, it can be seen that every branch is used exactly once to find a
 8 return path to the starting node. The arrowheads attached to the solid and
 9 broken lines indicate a change from a node to the next when logic level 0 or
 10 1 is entered to the node as an input bit SI to the shift register and one-bit
 11 shifted in a direction indicated by the characters "R" and "L" attached to
 12 the logic levels "1" and "0". For example, node 100 changes to node 110
 13 when all three bits are one-bit shifted to the right in response to an i th clock
 14 pulse, producing an output $SO = 0$, and an input $SI = 1$ is entered to its
 15 leftmost bit position, and node 110 changes to node 111 when all three bits
 16 are one-bit shifted to the right in response to an $(i + 1)$ th clock pulse and an
 17 input $SI = 1$ is entered to its leftmost bit position.

18 The data SO and DIR of the i th clock period and the data PO of
 19 the $(i + 1)$ th clock period are mapped in the ROM 17 to the data SI and DIR
 20 of the $(i + 1)$ th clock period as illustrated in Fig. 6. The input and output
 21 data associated with the transition from state 100 to state 110 are enclosed
 22 by a broken-line 20 in Fig. 6. This part of Fig. 6 is repeated in Fig. 5 to
 23 illustrate the relations of its contents to the associated nodes and branches.

24 The embodiment of Fig. 4 can be modified as shown in Fig. 7 in
 25 which the ROM 17 of Fig. 4 is replaced with a pair of first and second
 26 function generators, or ROMs 30 and 31 and a pair of modulo-2 adders 32
 27 and 33. Each of the ROMs 30 and 31 receives the multi-bit parallel output
 28 data PO from the bidirectional shift register 10. If the shift register 10 is a
 29 3-stage register which finds an Eulerian path as indicated in Fig. 6, then the
 30 ROMs 30 and 31 can be implemented in such a way as to map the 3-bit

NE-875

- 9 -

1 input parallel data PO to binary outputs as shown in Fig. 8.

2 The outputs of ROMs 30 and 31 are summed with the SO and
3 DIR outputs of flip-flops 14 and 16, respectively, in modulo-2 adders 32
4 and 33 to produce output bits SI and DIR as shown in Fig. 8. Shift register
5 10 receives these outputs of the modulo-2 adders 32 and 33 as new input
6 data. It is seen from Fig. 8 that the modulo-2 summation of the SO and
7 DIR inputs of the i th clock period with the outputs of the ROMs 30, 31
8 produces the same SI and DIR outputs during the $(i+1)$ th clock period as
9 those of Fig. 6.

10 This modified embodiment is advantageous in that the amount of
11 data mapped in memories 30 and 31 can be significantly reduced in
12 comparison with the amount of data in ROM 17.

13 Since the outputs of ROMs 30 and 31 can be respectively
14 represented as $SI(s)$ and $DIR(s)$ for a given internal state "S" of the
15 bidirectional shift register 10, the following relations apply to any Eulerian
16 cycles:

$$17 \quad SI_{(i+1)} = SI_{(i)} \oplus SI(s) \quad (1)$$

$$18 \quad DIR_{(i+1)} = DIR_{(i)} \oplus DIR(s) \quad (2)$$

19 where the symbol $\oplus$ indicates the modulo-2 addition. These relations hold
20 true to the relationships mapped in the ROM 17 of the previous
21 embodiment.

22 A further modification of the present invention is shown in Fig. 9 in
23 which a main function generator, or logic circuit 40, an auxiliary function
24 generator, or ROM 41 and a modulo-2 adder 42 are provided instead of
25 the ROM 30 of Fig. 7. Logic circuit 40 transforms the multi-bit data PO
26 from the shift register 10 into a single bit output according to a predefined,
27 main function. Memory 41 transforms the multi-bit data PO into a single
28 bit output according to a predefined, auxiliary function which serves to
29 correct errors in the main function. The outputs of logic circuit 40 and
30 memory 41 are summed in the modulo-2 adder 42 and the output of the

NE-875

- 10 -

1 adder 42 is supplied to the modulo-2 adder 32.

2 Prior to selecting an Eulerian path, the main function is determined
3 in a brute-force approach so that the logic circuit 40 can be implemented as
4 simple as possible. An Eulerian path is then selected in such a way that the
5 ROMs 41 and 31 produce as many logic-0 outputs as possible. If the
6 outputs of these memories contain many logic-0 outputs, they can be
7 logically compressed to reduce the memory size. In this case, the ROMs 41
8 and 31 can be implemented with simplified logic circuits. Since 3^{2m-2}
9 Eulerian paths are known to exist in a state transition graph of an m-stage
10 directional shift register, the probability of finding an Eulerian path that
11 satisfies the above-mentioned relations is high.

NE-875

- 11 -

What is claimed is:

- 1 1. A pseudorandom number sequence generator comprising:
2 a bidirectional shift register arranged to be loaded with a multi-bit
3 sequence, the shift register being responsive to an *i*th clock pulse and an *i*th
4 direction control bit for shifting the multi-bit sequence in one of two
5 directions, delivering an *i*th output bit and receiving an *i*th input bit,
6 wherein the multi-bit sequence successively defines one of a plurality of
7 nodes of a graph connected by branches; and
8 a feedback circuit connected to the shift register for converting a set
9 of input data comprising a multi-bit sequence stored in the shift register in
10 response to an (*i* + 1)th clock pulse, the *i*th output bit and the *i*th direction
11 control bit to a set of output data comprising an (*i* + 1)th input bit and an (*i*
12 + 1)th direction control bit and supplying the output data to the shift
13 register so that multi-bit sequences move around the graph following every
14 branch exactly once, and outputting the (*i* + 1)th input bit to form a
15 pseudorandom number sequence.
- 1 2. A pseudorandom number sequence generator comprising:
2 a bidirectional shift register arranged to be loaded with a multi-bit
3 sequence, the shift register being responsive to an *i*th clock pulse and an *i*th
4 direction control bit for shifting the multi-bit sequence in one of two
5 directions, delivering an *i*th output bit and receiving an *i*th input bit,
6 wherein the multi-bit sequence successively defines one of a plurality of
7 nodes of a graph connected by branches;
8 a first flip-flop responsive to the *i*th clock pulse for storing the *i*th
9 output bit and responsive to an (*i* + 1)th clock pulse for delivering the stored
10 *i*th output bit;
11 a second flip-flop responsive to the *i*th clock pulse for storing the *i*th
12 direction control bit and responsive to the (*i* + 1)th clock pulse for delivering

NE-875

- 12 -

13 the stored i th direction control bit; and
14 a function generator connected to the shift register and the first and
15 second flip-flops for forming a set of input data comprising a multi-bit
16 sequence stored in the shift register in response to the $(i + 1)$ th clock pulse,
17 the i th output bit and the i th direction control bit delivered from the first
18 and second flip-flops to a set of output data comprising an $(i + 1)$ th input
19 bit and an $(i + 1)$ th direction control bit and supplying the $(i + 1)$ th input bit
20 and the $(i + 1)$ th direction control bit to the shift register so that multi-bit
21 sequences move around the graph following every branch exactly once, and
22 outputting the $(i + 1)$ th input bit to form a pseudorandom number
23 sequence.

1 3. A pseudorandom number sequence generator as claimed in
2 claim 2, wherein said function generator comprises:
3 a first function generator for receiving said multi-bit sequence from
4 said shift register and producing an output bit;
5 a second function generator for receiving said multi-bit sequence
6 from said shift register and producing an output bit;
7 a first modulo-2 adder for summing the output bit of the first
8 function generator with the output bit of the first flip-flop to produce said
9 $(i + 1)$ th input bit; and
10 a second modulo-2 adder for summing the output bit of the second
11 function generator with the direction control bit of the second flip-flop to
12 produce said $(i + 1)$ th direction control bit.

1 4. A pseudorandom number sequence generator as claimed in
2 claim 3, wherein said first function generator comprises:
3 a main function generator for receiving said multi-bit sequence from
4 said shift register and producing an output bit;
5 an auxiliary function generator for receiving said multi-bit sequence

NE-875

- 13 -

6 from said shift register and producing an output bit; and
7 a modulo-2 adder for summing the output bit of the main function
8 generator with the output bit of the auxiliary function generator.

1 5. A pseudorandom number sequence generator as claimed in
2 claim 1, wherein said function generator comprises a memory for mapping
3 all possible bit patterns of said multi-bit sequence and all possible bit
4 patterns of the output bits of the first and second flip-flops to a plurality of
5 pairs of an input bit and a direction control bit.

1 6. A pseudorandom number sequence generator as claimed in
2 claim 3, wherein said first function generator comprises a memory for
3 mapping all possible bit patterns of said multi-bit sequence to a plurality of
4 output bits and wherein said second function generator comprises a
5 memory for mapping all possible bit patterns of said multi-bit sequence to a
6 plurality of output bits.

1 7. A method of generating a pseudorandom number sequence,
2 comprising:
3 a) loading a multi-bit sequence into a bidirectional shift register;
4 b) responsive to an *i*th clock pulse, shifting the multi-bit sequence
5 by one bit position in one of two directions according to an *i*th direction
6 control bit, delivering an *i*th output bit from the shift register and supplying
7 thereto an *i*th input bit, storing the *i*th output bit and the *i*th direction
8 control bit, wherein the multi-bit sequence successively defines one of a
9 plurality of nodes of a graph connected by branches;
10 c) responsive to an (*i* + 1)th clock pulse, forming a set of input
11 data comprising the *i*th output bit, the *i*th direction control bit and a multi-
12 bit sequence stored in the shift register; and
13 d) responsive to the (*i* + 1)th clock pulse, converting the set of

NE-875

- 14 -

14 input data formed by step (c) to a set of output data comprising an $(i + 1)^{th}$
15 input bit and an $(i + 1)^{th}$ direction control bit, shifting the multi-bit
16 sequence along the shift register by one bit position in one of two directions
17 according to the $(i + 1)^{th}$ direction control bit, and supplying the $(i + 1)^{th}$
18 input bit to the shift register, and holding an $(i + 1)^{th}$ output bit delivered
19 from the shift register and the $(i + 1)^{th}$ direction control bit, so that multi-
20 bit sequences move around the graph following every branch exactly once,
21 and outputting the $(i + 1)^{th}$ input bit to form a pseudorandom number
22 - sequence.

Smart & Biggar
Ottawa, Canada
Patent Agents

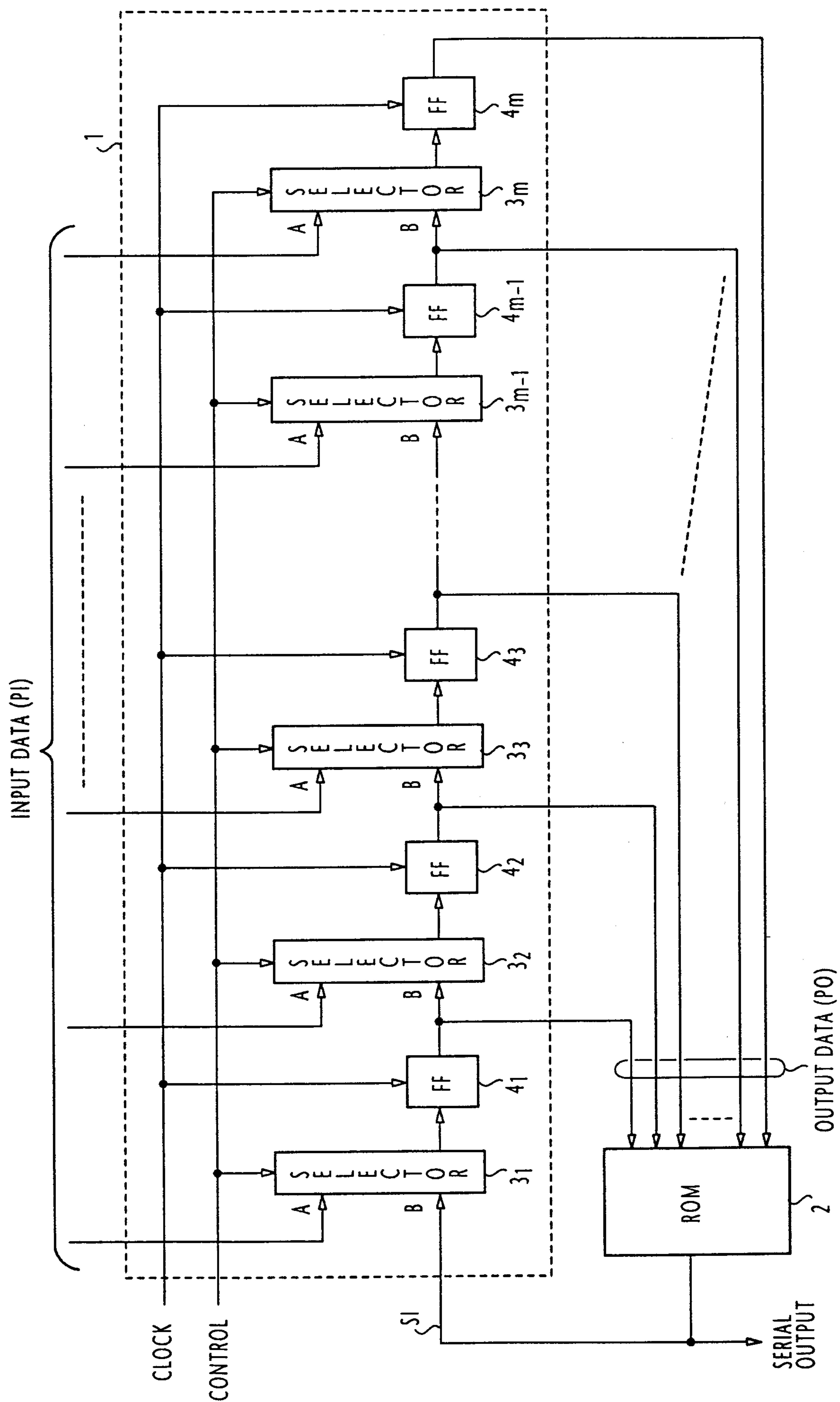
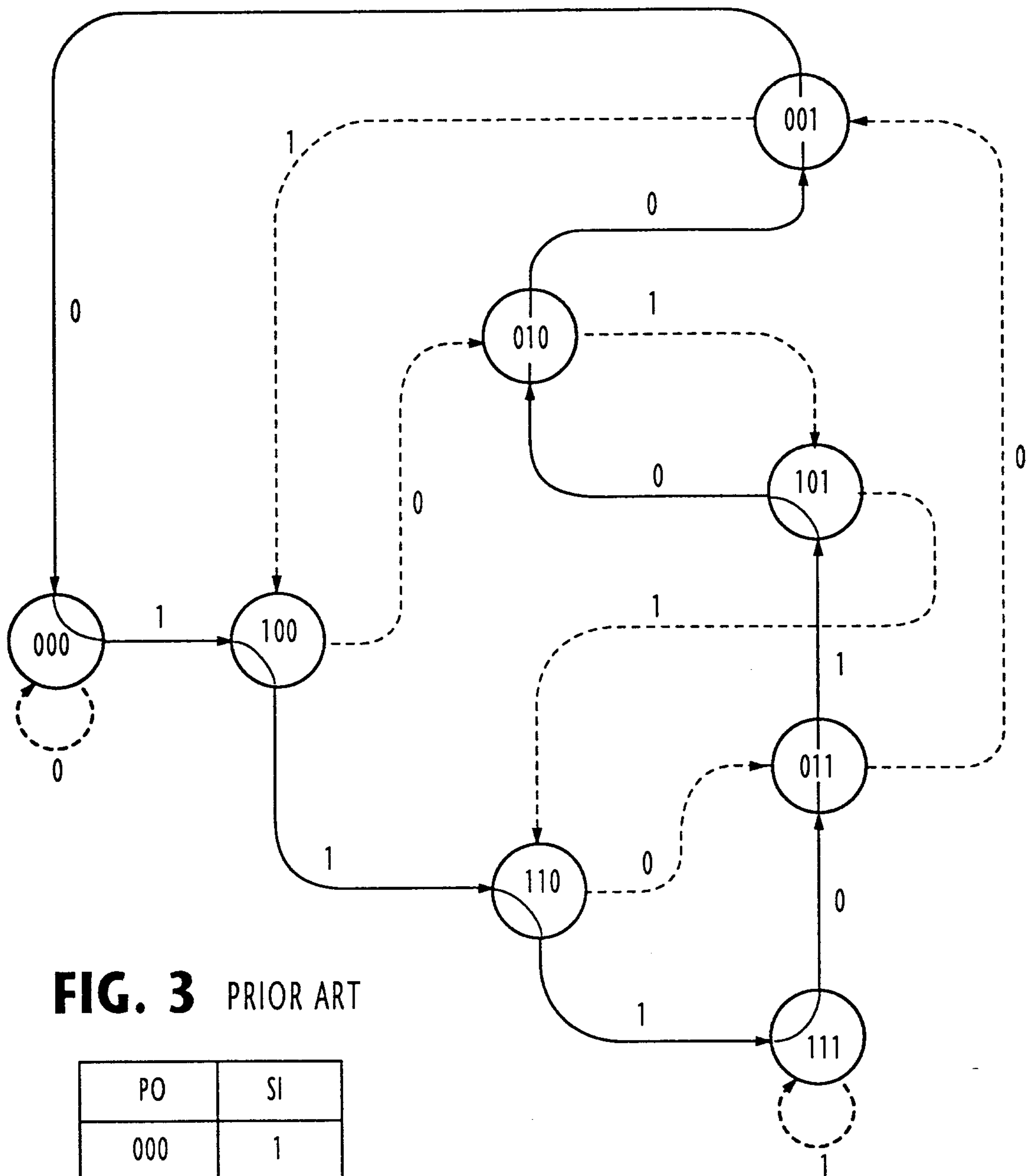
FIG. 1 PRIOR ART

FIG. 2
PRIOR ART**FIG. 3** PRIOR ART

PO	SI
000	1
001	0
010	0
011	1
100	1
101	0
110	1
111	0

FIG. 4

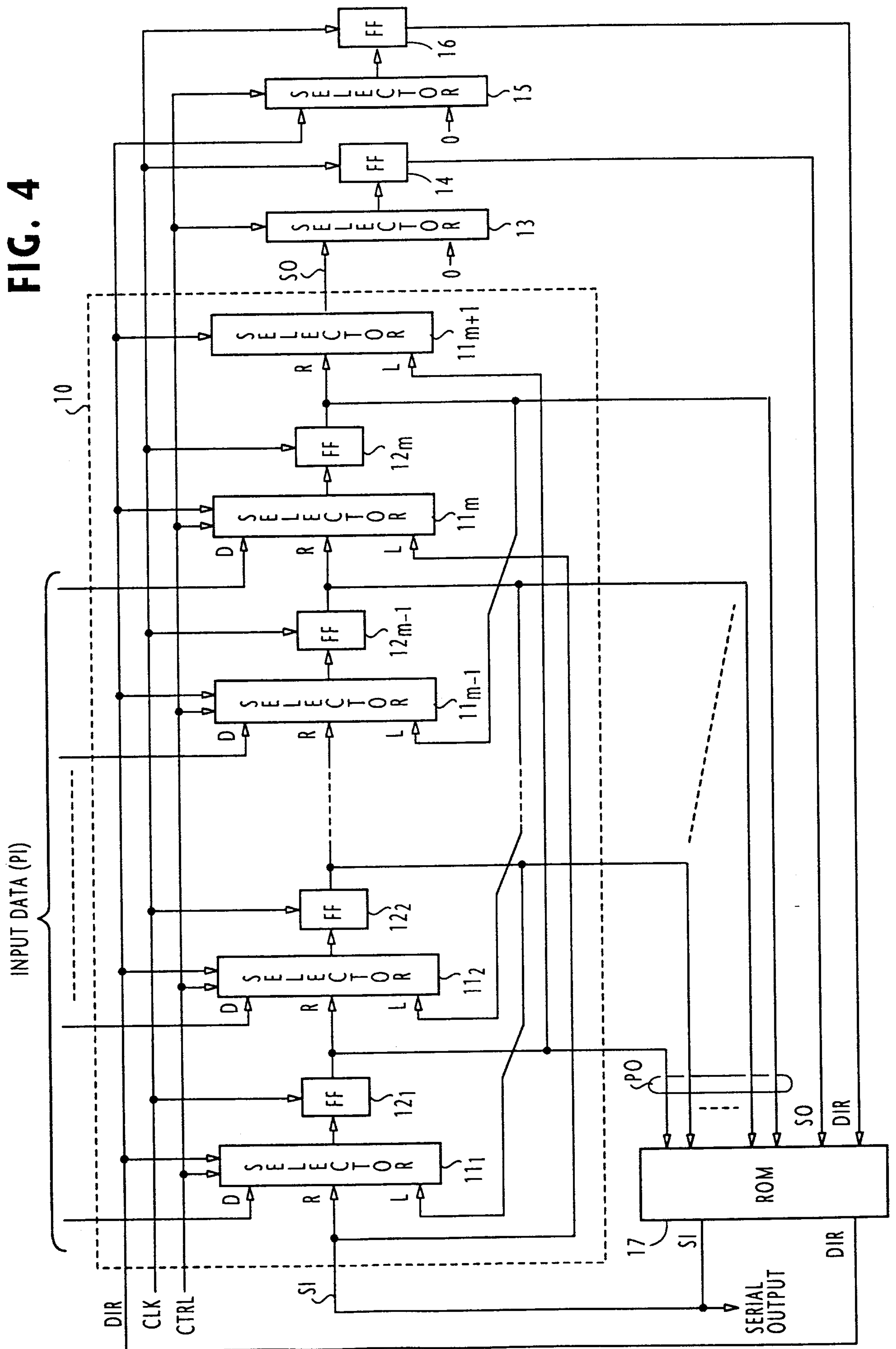


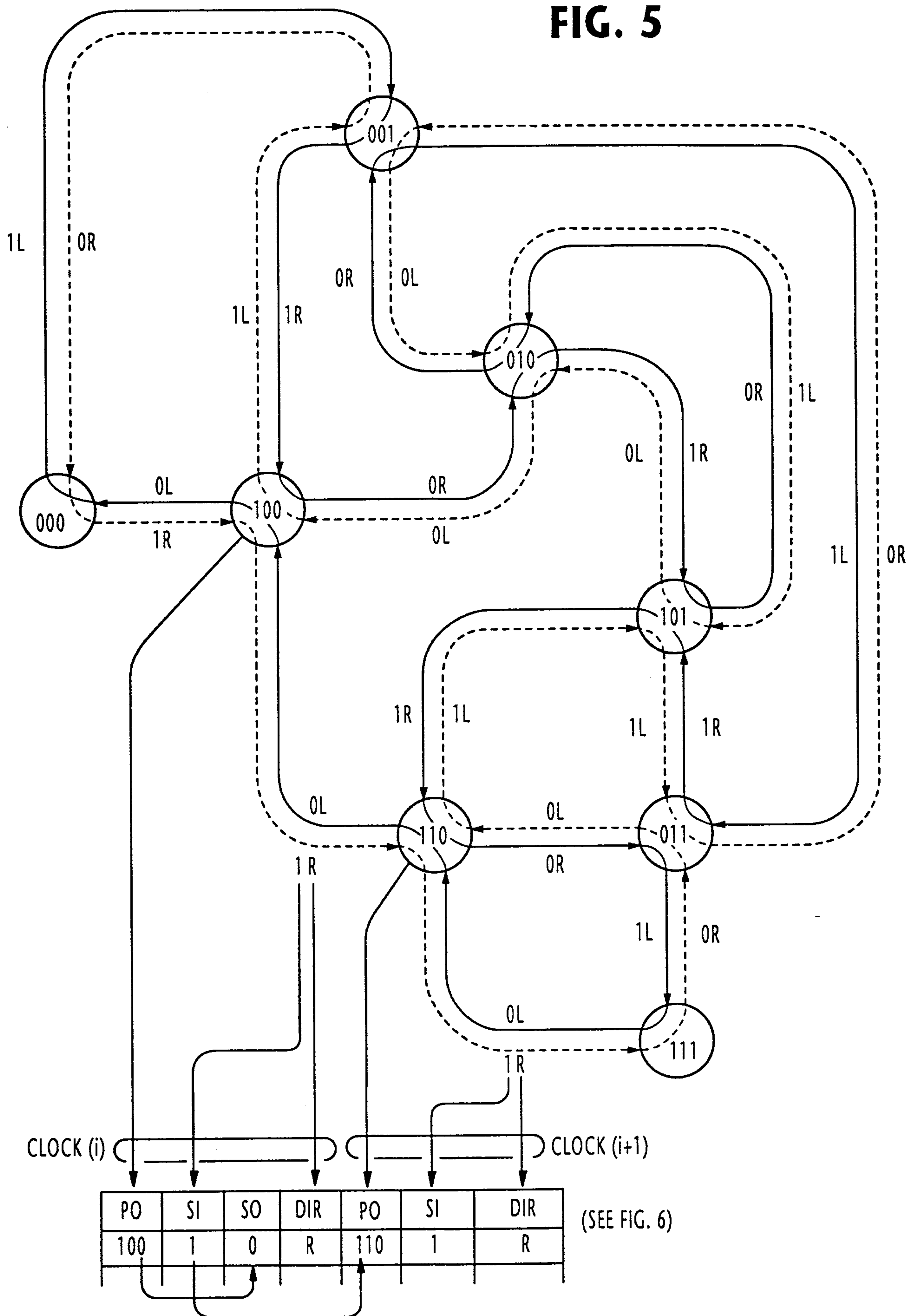
FIG. 5

FIG. 6

LEGEND:
R = "1"
L = "0"

		INPUTS OF ROM 17		OUTPUTS OF ROM 17		
CLOCK (i)		CLOCK (i)		CLOCK (i+1)		
PO	SI	SO	DIR	PO	SI	DIR
001 100	0 0	1 1	R L	000	1 1	R L
011 100 010 000	0 1 0 1	1 1 0 0	R L R L	001	0 0 1 1	L R L R
101 001 101 100	0 0 0 0	1 0 1 0	L L R R	010	0 1 0 1	L L R R
111 101 110 001	0 1 0 1	1 1 0 0	R L R L	011	0 0 1 1	L R L R
000 010 110 001	1 0 0 1	0 0 1 1	R L L R	100	1 1 0 0	R L L R
110 010 011 010	1 1 1 1	1 0 1 0	L L R R	101	1 0 1 0	L L R R
100 011 111 101	1 0 0 1	0 0 1 1	R L L R	110	1 0 0	R L L R
110 011	1 1	0 0	R L	111	0 0	R L

20

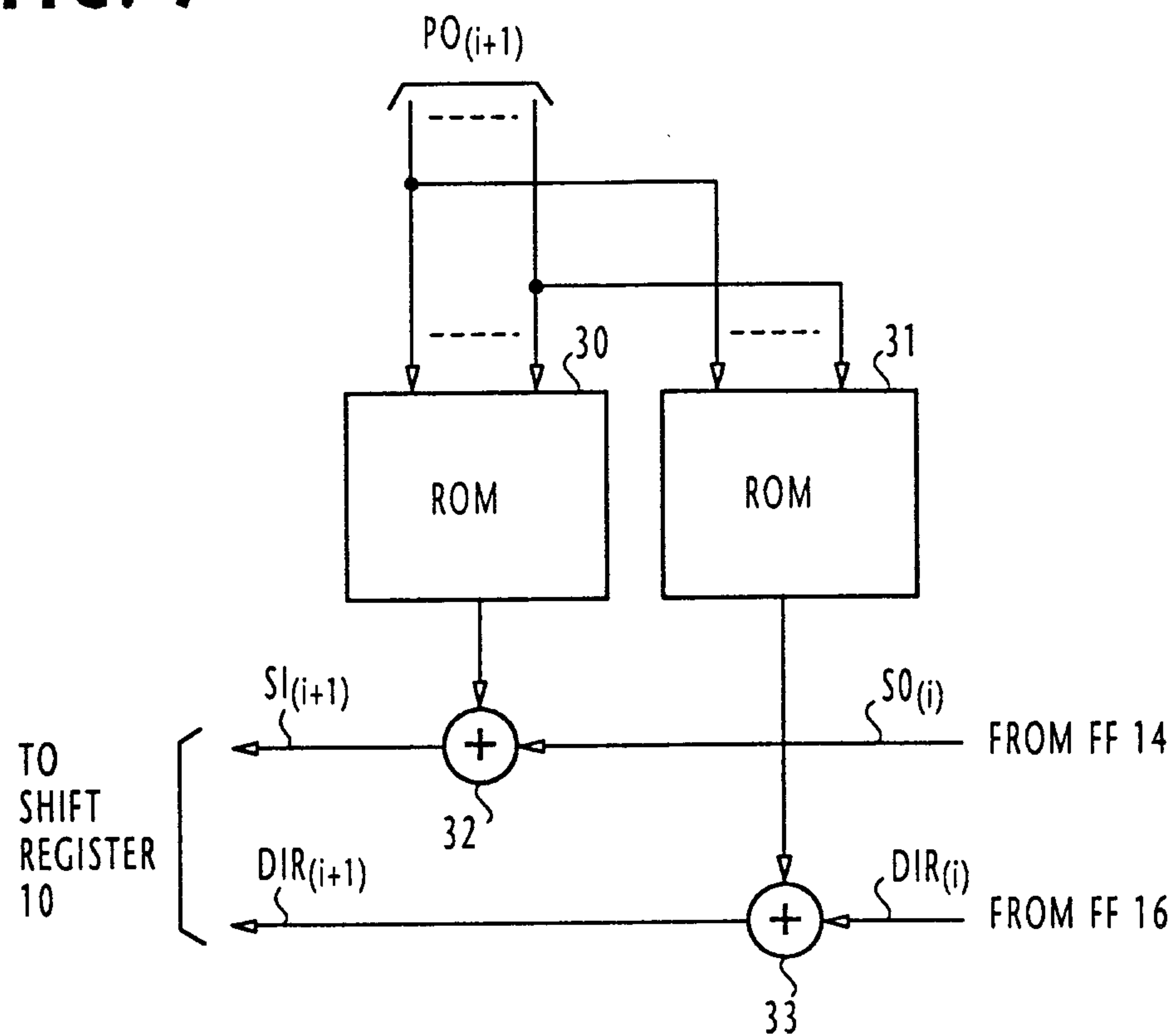
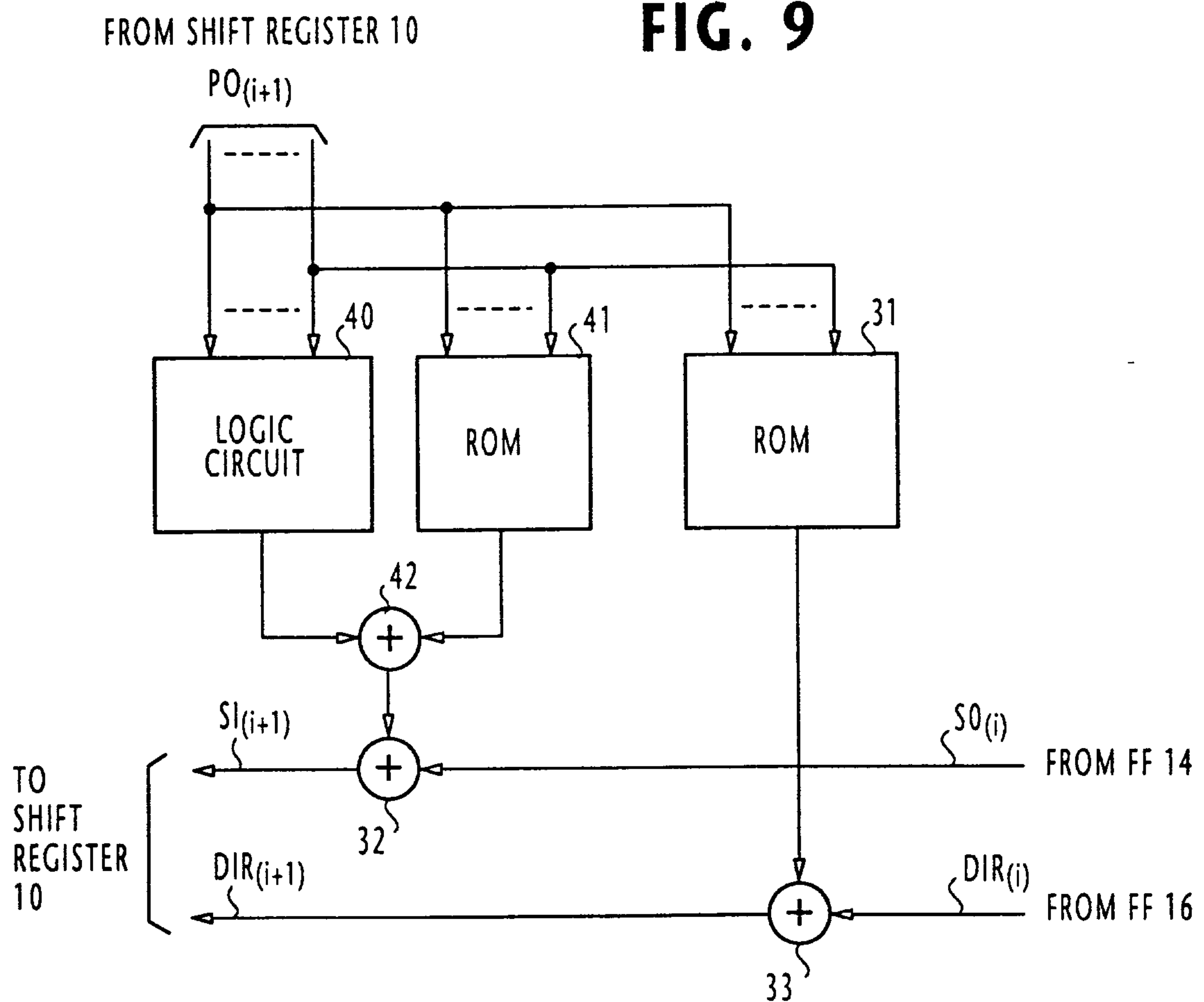
FIG. 7 FROM SHIFT REGISTER 10**FIG. 9**

FIG. 8

LEGEND:

R = "1"

L = "0"

CLOCK (i)				CLOCK (i+1)				
PO	SI	SO	DIR	INPUTS OF ROM's 30, 31 (PO)	OUTPUT OF ROM 30	OUTPUT OF ROM 31	SI	DIR
001 100	0 0	1 1	R L	000	0	0	1 1	R L
011 100 010 000	0 1 0 1	1 1 0 0	R L R L	001	1	1	0 0 1 1	L R L R
101 001 101 100	0 0 0 0	1 0 1 0	L L R R	010	1	0	0 1 0 1	L L R R
111 101 110 001	0 1 0 1	1 1 0 0	R L R L	011	1	1	0 0 1 1	L R L R
000 010 110 001	1 0 0 1	0 0 1 1	R L L R	100	1	0	1 1 0 0	R L L R
110 010 011 010	1 1 1 1	1 0 1 0	L L R R	101	0	0	1 0 1 0	L L R R
100 011 111 101	1 0 0 1	0 0 1 1	R L L R	110	1	0	1 1 0 0	R L L R
110 011	1 1	0 0	R L	111	0	0	0 0	R L

