Logic design — 8

Shift register counters

by B. Holdsworth* and D. Zissos†

* Chelsea College, University of London † Department of Computing Science, University of Calgary, Canada.

An alternative method of designing digital counters or sequence generators is to use a shift register chip: a typical shift register counter configuration is shown in Fig. 1. The individual flip-flops form part of an n stage shift register and the connexions between individual flip-flops are internal to the chip. The output of each stage and its complement are both available and they may be used to drive combinational-feedback logic which provides the J and K inputs to the first stage of the shift register. Such a circuit can be used to generate specific binary sequences or, alternatively, it can operate as a modulo-M counter.

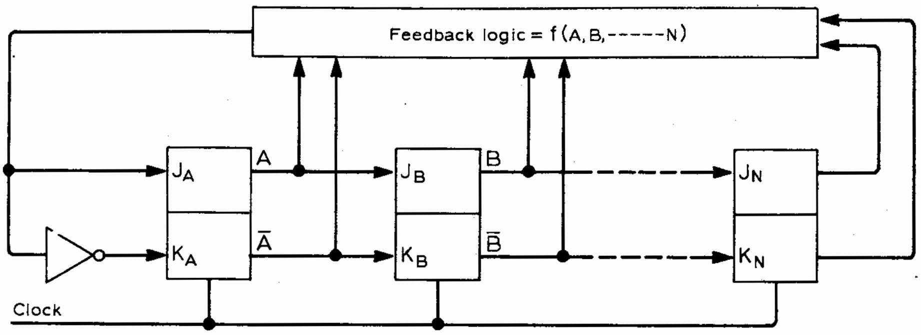


Fig. 1. Basic configuration of a feedback shift register counter.

Input-output relations

The input-output relationships for each stage of the counter shown in Fig. 1 are defined by the following set of equations:

$$
\begin{aligned}
A^{t+\delta t} &= f(A, B, \ldots N)^t \\
B^{t+\delta t} &= A^t \\
C^{t+\delta t} &= B^t \\
N^{t=\delta t} &= (N-1)^t
\end{aligned}
$$

The feedback circuit produces either a 1 or a 0 which is fed to the input of flip-flop A where it determines the next state of A on the receipt of the next clock pulse. For example, assuming that then n-stage shift register is in the state N...CBA = 0...001, the next stage of the shift register will be either 0...010 or 0...011, depending upon whether the feedback logic provides a 0 or a 1 at the J-input of flip-flop A.

Universal state diagram

The transition table for a two-stage shift register is shown in Fig. 2(a). If the shift register is initially in the state 00 then there are two possible next states. These are 00 if the J-input to the first flip-flop is a 0 or alternatively 01 if the J-input is a 1. Similarly, if the initial state of the shift register is 01 then the two possible next states are either 10 or 11.

The transition table of Fig. 2(a) can be translated into the universal state diagram shown in Fig. 2(b), alternatively called the de Bruijn diagram. It will be noticed that the shift register is permantly "locked" in the state 00 if the feedback logic is a 0 and, similarly, it is "locked" in the state 11 if a 1 is provided by the feedback logic.

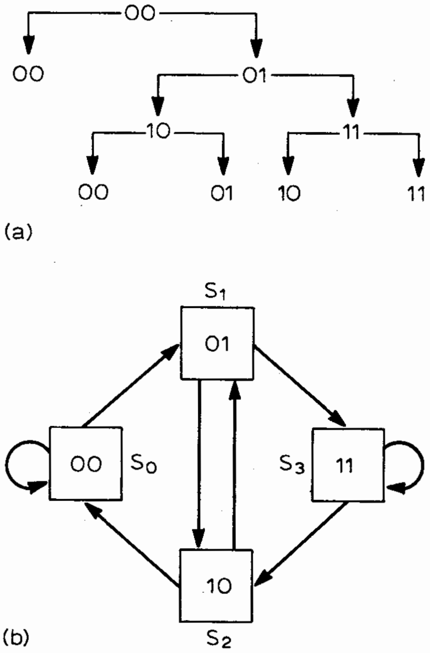


Fig. 2. Transition table for a two stage shift register is shown at (a), while (b) shows the de Bruijn or universal state diagram for a two stage shift register.

A similar transition table can be developed for a three-stage shift register and this can be translated into a universal state diagram as shown in Figs. 3(a) and 3(b). The universal state diagram for a four-stage shift register is shown in Fig. 4 and clearly the complexity of this type of diagram increases rapidly with the number of stages in the shift register.

The universal state diagram is a departure from the kind of state diagram that defines a specific sequence, as seen in earlier articles in this series. All possible internal states and all possible transitions between states are shown on the universal state diagram. The logic designer may now choose a suitable sequence of states on the diagram and design the feedback logic so that the shift register will cycle through the chosen sequence of states. If for example a decade counter is required, then a ten-state sequence should be chosen on the de Bruijn diagram and the feedback logic required to produce the sequence is then determined by the methods described in the next section.

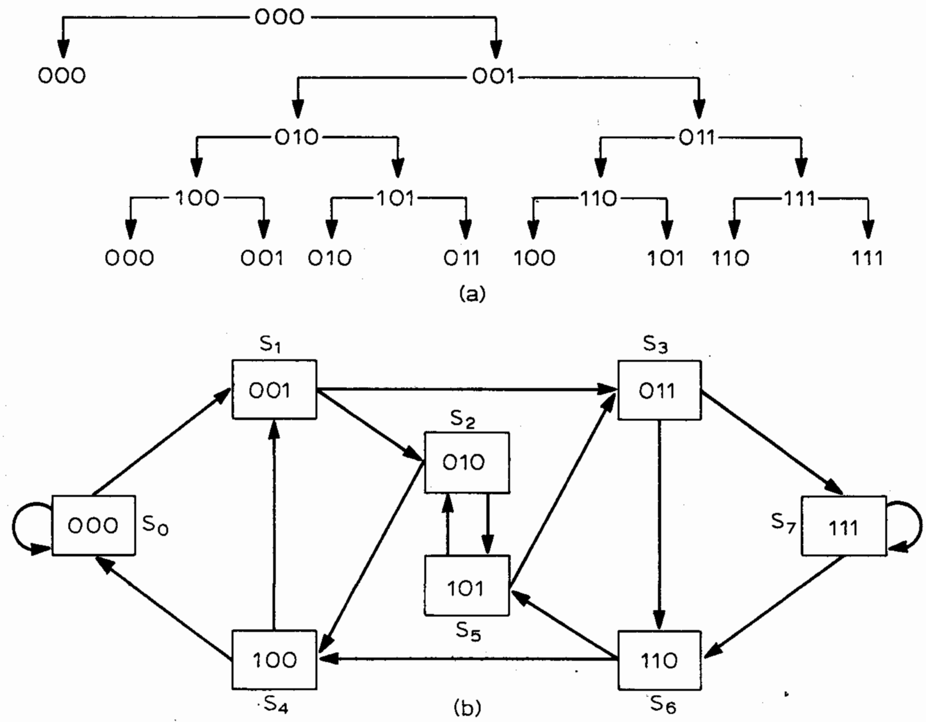


Fig. 3. Transition table for a three-stage shift register (a) and universal state diagram for a three-stage shift register (b).

Designing a decade counter

Step 1. Choose a ten-state sequence on the universal state diagram for a four-stage shift register. Examination of Fig. 4 reveals the following two possible sequences:

(a) S_0–S_1–S_2–S_5–S_{11}– –S_7–S_{15}–S_{14} –S_{12}–S_8–S_0 and
(b) S_0–S_1–S_3–S_7–S_{15}–S_{14}–S_{13}–S_{10} –S_4–S_8–S_0.

There are, of course, other ten-state sequences besides those detailed above.

The state diagram for the second of the above sequences is shown in Fig. 5(a).

Step 2. Draw up a state table, as shown in Fig. 5(b), and determine the logical value of the feedback function for each transition. For example in going from S_0 to S_1, the state of flip-flop A must change from 0 to 1 and hence the required input $J_A = 1$. This is the logical value of the feedback function required for this transition, and it is entered in the final column of the state table.

Step 3. Plot the feedback function and the unused states on a four-variable Karnaugh map as shown in Fig. 5(c). The feedback function is marked by a 1 in the appropriate cells of the map, whilst the unused states are marked with a "d".

Using normal minimization techniques the feedback function is found to be $f = \bar{C}\bar{D} + A\bar{D} + \bar{A}BC$. If the circuit enters an unused state the logic of the unused states, $f = A\bar{C}D + \bar{A}B\bar{D} + ABCD + A\bar{B}C\bar{D}$, can be used to stop the counter, raise an alarm, and reset all the flip-flops in the shift register to zero.

If this counter enters an unused state due to circuit misoperation, and if the logic of the unused states is not utilised, it will return to the correct sequence after a maximum of five clock pulses. The behaviour of the circuit is then described by the two tables shown in Fig. 5(d).

A perfectly general rule that should be observed when designing this type of counter is that the entry in the S_{15} cell on the K-map should always be a 0 and that in the S_0 cell should always be a 1, irrespective of whether these two states are in the counting sequence. This ensures that the counter will never be locked in either the 0000 or 1111 states.

The implementation of the basic counter is shown in Fig. 5(e). If a decimal output is required then a 4-to-16 line decoder can be used. For example when $AB\bar{C}D = 1$ the decoder output should be the decimal digit two.

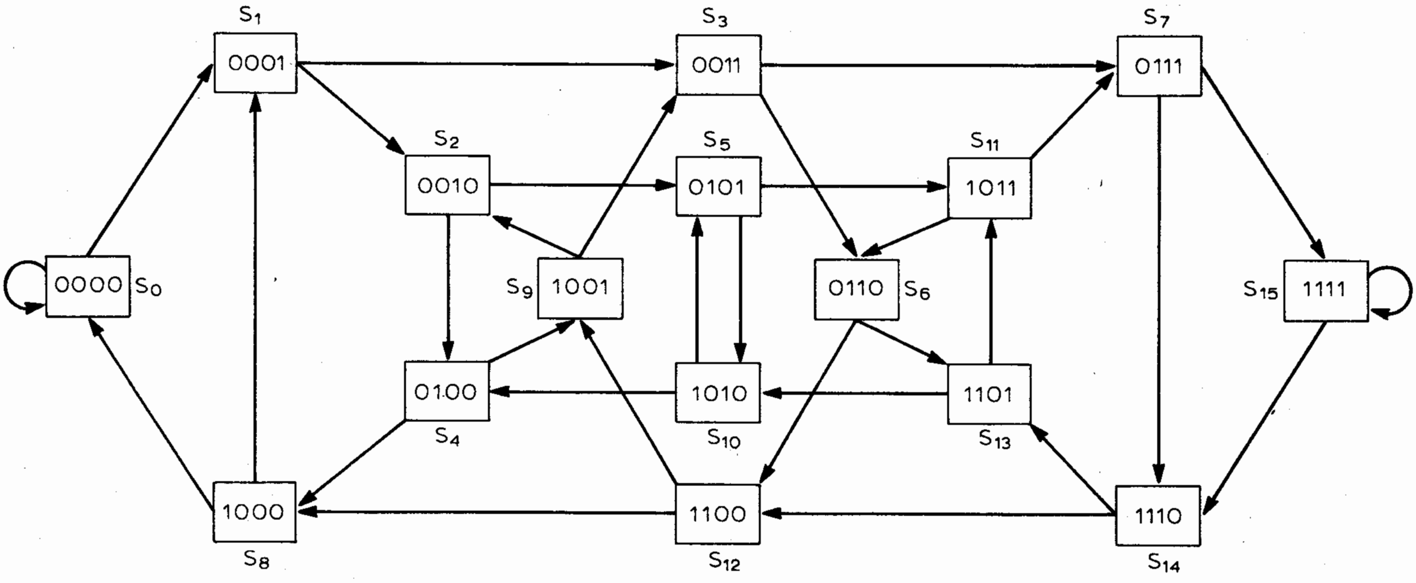


Fig. 4. The universal state diagram for a four-stage shift register.

Alternatively, if a decimal display is required then the counter in conjunction with the appropriate combinational logic can be used to drive a seven segment indicator.

Shift register sequence generators

A shift register counter with feedback logic can be modified, to produce any required binary sequence, with the aid of output logic. The length of the binary sequence generated, *l*, will be the same as that of the count from which it has been derived. For an *n*-stage shift register the length of the binary sequence $l \leqslant 2^n$. The basic configuration of such a sequence generator is shown in Fig. 6.

Design. As an example, a circuit that will generate the binary sequence 0–0–1–0–1–0–1–1 will be designed. Since there are eight bits in this sequence a three-stage shift register will be required. The eight three-bit combinations required to generate this sequence are tabulated in Fig. 7(a), where the binary digits in the column headed g_a are the required sequence whilst those in the columns headed g_b and g_c are the same sequence delayed in time with respect to the sequence in the column headed g_a, by one and two clock pulses respectively.

It will be observed that the shift register states 010 and 101 both occur twice in the tabulation. In the case of state 101 an ambiguity exists, since in one case the next state of the shift register is 010, whilst in the second case it is 011. Consequently these eight combinations cannot be generated by the method of direct feedback logic employed in the last design example, and output logic has to be used to produce the required sequence.

The method employed is to develop an eight-state sequence using direct feedback logic, the required binary sequence then being derived from this eight-state sequence with the aid of output logic.

Step 1. An eight-state sequence is obtained from the de Bruijn diagram for a three-stage shift register shown in Fig. 3(b). Such a sequence is:

$$S_0–S_1–S_2–S_5–S_3–S_7–S_6–S_4–S_0$$

and the state diagram for this sequence is shown in Fig. 7(b).

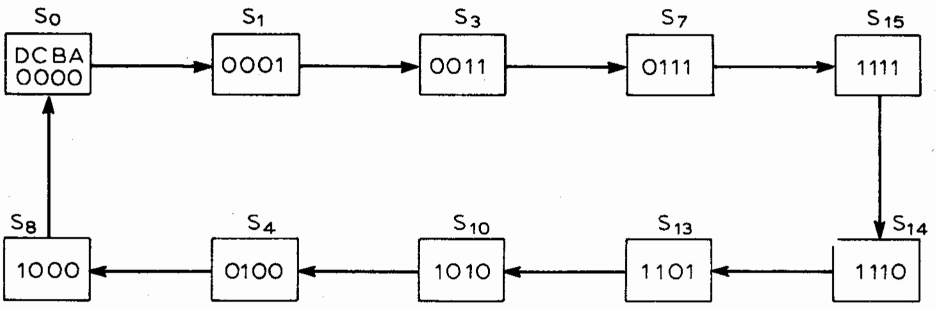
Step 2. The state table is drawn up as shown in Fig. 7(c) and the value of the feedback function, either 0 or 1, is determined for each transition as described previously.

Step 3. Plot the feedback function and any unused states on a Karnaugh map as shown in Fig. 7(d). Using normal minimisation techniques the feedback function is found to be:

$$f = B\bar{C} + ABC + A\bar{B}C$$

Step 4. The output binary sequence required is now tabulated by the side of the eight-state shift register sequence as shown in Fig. 7(e) and the minimal form of the output function

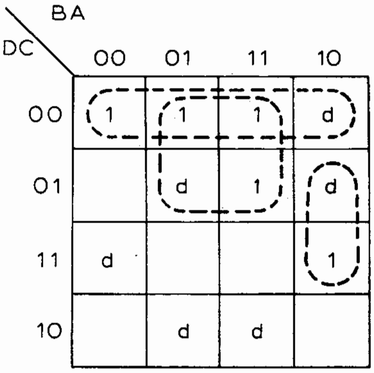
$$g_a = B\bar{C} + \bar{A}C$$



(a)

S	D	C	B	A	Feedback function
S_0	0	0	0	0	1
S_1	0	0	0	1	1
S_3	0	0	1	1	1
S_7	0	1	1	1	1
S_{15}	1	1	1	1	0
S_{14}	1	1	1	0	1
S_{13}	1	1	0	1	0
S_{10}	1	0	1	0	0
S_4	0	1	0	0	0
S_8	1	0	0	0	0

(b)



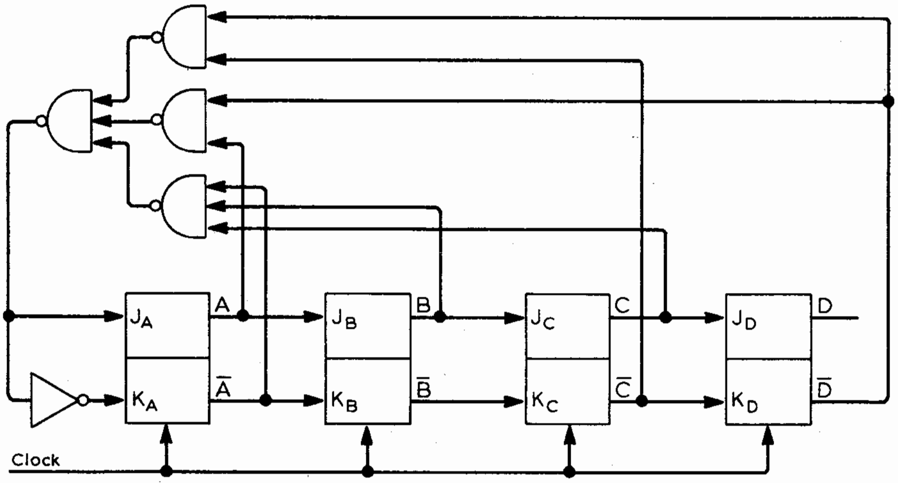
DC \ BA	00	01	11	10
00	1	1	1	d
01		d	1	d
11	d			1
10		d	d	

(c)

S	D	C	B	A	f
S_9	1	0	0	1	0
S_2	0	0	1	0	1
S_5	0	1	0	1	1
S_{11}	1	0	1	1	0
S_6	0	1	1	0	1
S_{13}	1	1	0	1	0

S	D	C	B	A	f
S_{12}	1	1	0	0	0
S_8	1	0	0	0	0

(d)



(e)

Fig. 5. State diagram for a decade counter (a), state table (b), Karnaugh map of the feedback function (c). The behaviour of the counter after entering an unused state is at (d) and (e) shows a shift register decade counter.

is obtained from the Karnaugh map shown in Fig. 7(f).

There are other output sequences such as g_b and g_c in Fig. 7(a), which are merely displaced in time from the sequence in the g_a column of Fig. 7(e), and the logic for these sequences can be examined to see which requires the least hardware. As it happens, in this example, the equation for g_a contains the term BC which is one of the product terms of the feedback function; hence, very little additional hardware is required to generate the output equation. The implementation of the sequence generator is shown in Fig. 7(g).

In the second half of this part of the series, the authors continue the discussion of shift-register counters, going on to describe ring counters and maximum-length sequence generators.

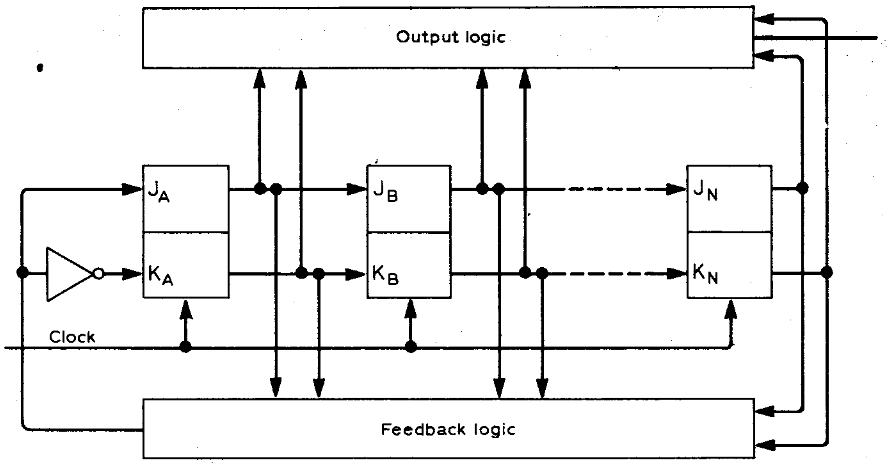
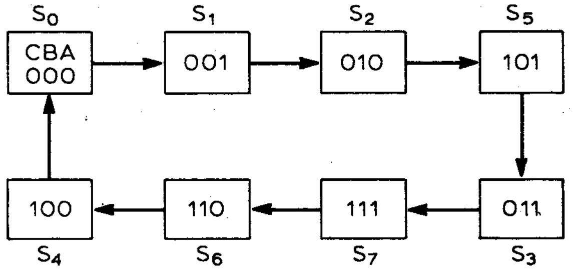


Fig. 6. Basic configuration of a binary sequence generator.

Fig. 7. (a) shows eight, three-bit combinations needed for 8-bit sequence, and the state diagram to give the eight-state sequence is at (b). State table for feedback function and its K-map are at (c) and (d) while the state table and K-map for the output function are at (e) and (f). Fig. 7(g) is the implementation of binary sequence generator.

Clock pulse	g_c	g_b	g_a
1	1	1	0
2	1	0	0
3	0	0	1
4	0	1	0
5	1	0	1
6	0	1	0
7	1	0	1
8	0	1	1

(a)



(b)

C \ BA	00	01	11	10
0			1	1
1	1			1

$g_a = B\bar{C} + \bar{A}C$

(f)

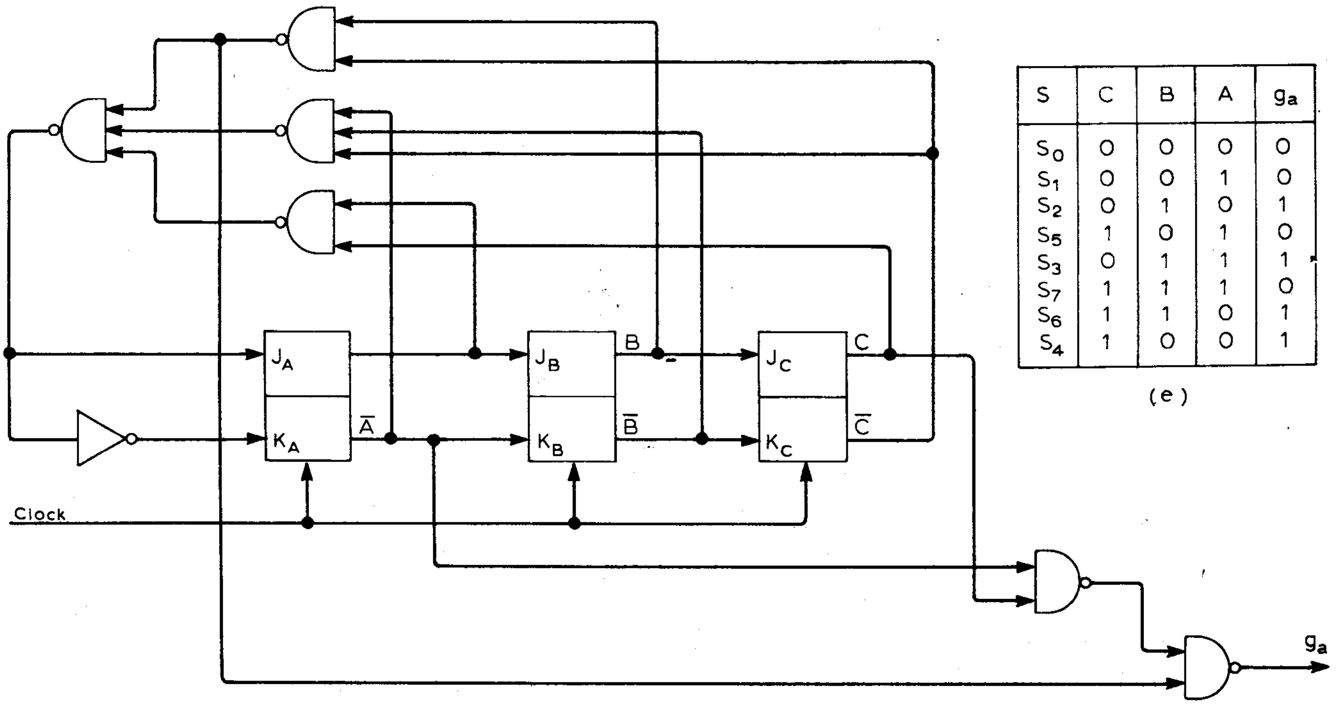
C \ BA	00	01	11	10
0	1		1	1
1		1		

$f = B\bar{C} + \bar{A}\,\bar{C} + A\bar{B}C$

(d)

S	C	B	A	Feedback function f
S_0	0	0	0	1
S_1	0	0	1	0
S_2	0	1	0	1
S_5	1	0	1	1
S_3	0	1	1	1
S_7	1	1	1	0
S_6	1	1	0	0
S_4	1	0	0	0

(c)



S	C	B	A	g_a
S_0	0	0	0	0
S_1	0	0	1	0
S_2	0	1	0	1
S_5	1	0	1	0
S_3	0	1	1	1
S_7	1	1	1	0
S_6	1	1	0	1
S_4	1	0	0	1

(e)

Logic design — 9

More shift registers — ring counters and maximum-length sequence generators

by B. Holdsworth* and D. Zissos†

*Chelsea College, University of London †Dept. of Computing Science, University of Calgary, Canada

The simplest type of shift register counter is the ring counter where feedback is provided from the last stage and feeds the inputs of the first stage as shown in Fig. 8(a). In this circuit there are ten stages: it can be used as a decimal ring counter, since the number of stages is equal to the number of counter states. The information contained in each stage is shifted to the next stage on the receipt of a clock pulse and the counter circulates a 1 which is initially preset in the first stage, all other stages being simultaneously cleared to 0. The counting sequence of the register is shown in Fig. 8(b).

The circuit of Fig. 8(a) can be modified so that it becomes self-starting as shown in Fig. 8(c). The input J_A = $\bar{A}\bar{B}\bar{C}\bar{D}\bar{E}\bar{F}\bar{G}\bar{H}\bar{I}$ and this can only be a 1 providing A = B = C = D = E = F = G = H = I = 0. Clearly if any section of the counter except the last one contains a 1, J_A = 0, and the counter will now enter the required sequence within a maximum of ten clock pulses.

The ten outputs of this counter can be used directly to drive a decimal display without the need of decoding networks or alternatively it can be used to enable a group of circuits sequentially, as the 1 moves through the various stages of the shift register. The number of stages required in the latter case will be equal to the number of circuits that have to be enabled.

An obvious advantage of the decimal ring counter is its simplicity and since it requires no feedback logic or decoding circuits it uses fewer components. It does, though, have the disadvantage of not having a binary readout, and its counting sequence is radically changed if circuit misoperation occurs, as for example when a section other than that containing the counting 1 is, due to

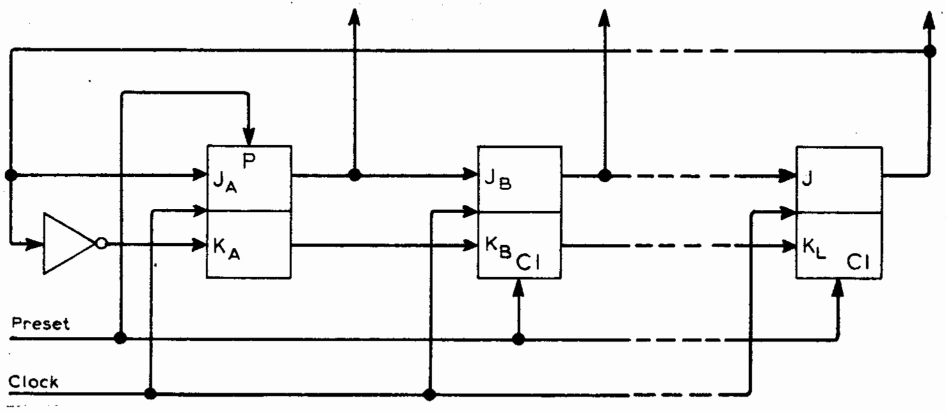


Fig 8(a)

Clock pulse	L	I	H	G	F	E	D	C	B	A
0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	0	1	0
2	0	0	0	0	0	0	0	1	0	0
3	0	0	0	0	0	0	1	0	0	0
4	0	0	0	0	0	1	0	0	0	0
5	0	0	0	0	1	0	0	0	0	0
6	0	0	0	1	0	0	0	0	0	0
7	0	0	1	0	0	0	0	0	0	0
8	0	1	0	0	0	0	0	0	0	0
9	1	0	0	0	0	0	0	0	0	0

Fig 8(b)

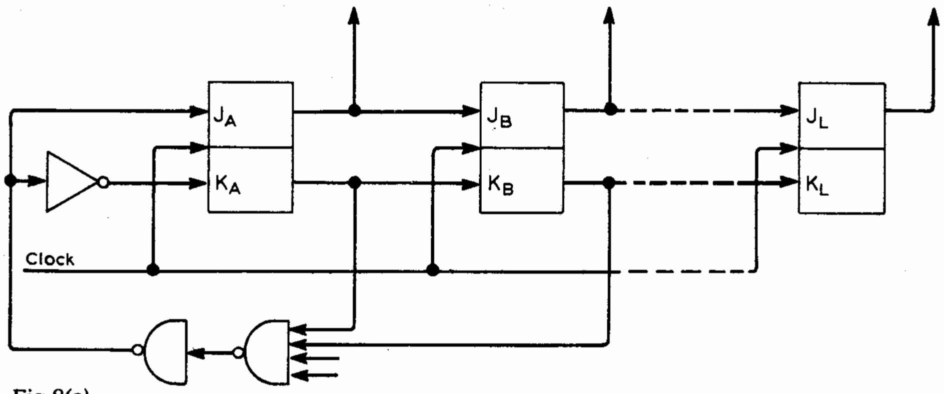


Fig 8(c)

Fig. 8. A basic ring counter (a) and the counting sequence for a decimal type. The two gates in (c) facilitate self starting and the circuit in (d) detects the presence of superfluous '1' states caused by malfunctions. The lack of a '1' in any stage is detected by the circuit at (e).

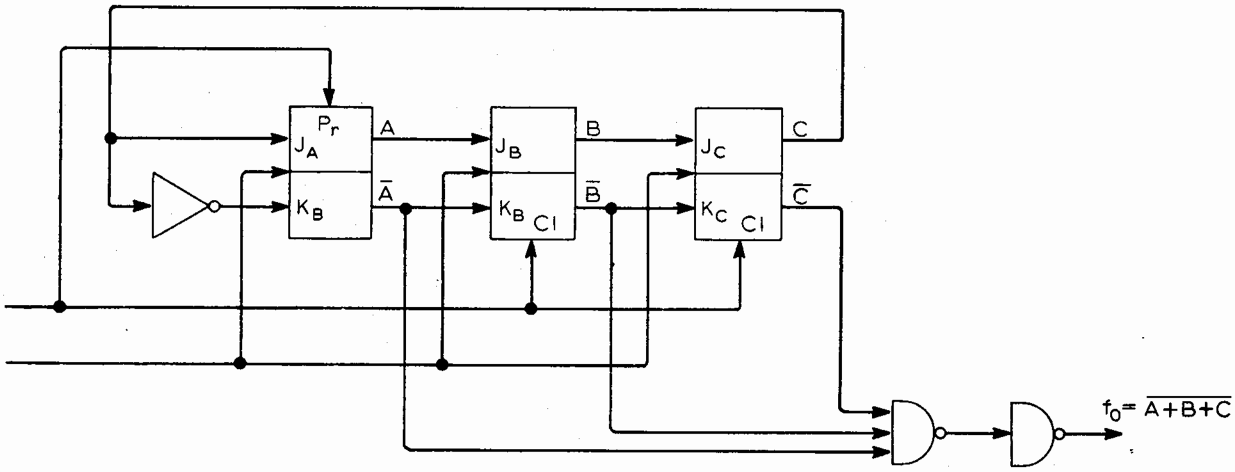


Fig 8(d)

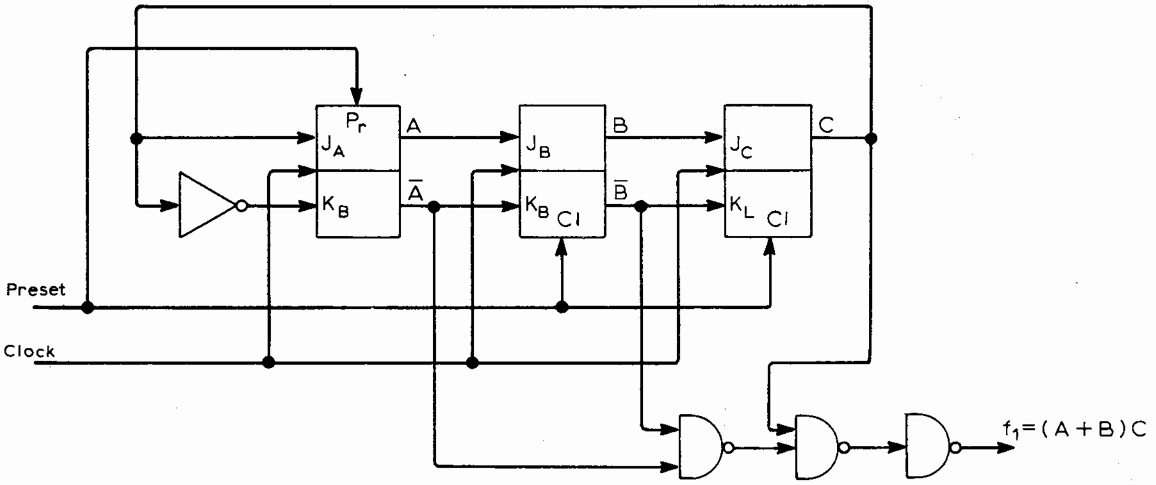


Fig 8(e)

some circuit fault also set to 1, or alternatively when the counting 1 is accidentally set to 0. However, it is not difficult to introduce simple logical networks which detect the presence of additional 1's. A three-stage ring counter having this facility is shown in Fig. 8(d). Similarly, it is not difficult to introduce a network which will indicate whether all sections of the shift register contain 0's. A circuit which will provide this facility is shown in Fig. 8(e).

The function required for the detection of additional 1's in the three stage circuit is $f_1 = (A+B)C$, and the function for indicating that all stages of the same circuit contains 0's is $f_0 = A+B+C$.

Twisted ring or Johnson counter

As the name implies, the difference between the twisted ring counter and the ordinary ring counter is that the feedback connexions are reversed and in this case the complementary output of the last stage is connected to the J input of the first stage, whilst the inverted form of this signal is fed to the K input. If all the flip-flops are initially preset to the same state, either 0 or 1, then the number of different states of the counter is equal to twice the number of stages in the shift register. Hence a decade counter can be constructed from a five-stage shift register as illustrated in Fig. 9(a). The counting sequence of the circuit, assuming that initially all the flip-flops are cleared to zero is given in Fig. 9(b).

This is a ten-state sequence which could have been selected from the universal state diagram of a five-state shift register. The feedback logic could have been developed by first tabulating the required value of the feedback function in the column headed '*f*' in the table of Fig. 9(b) and then plotting this function in conjunction with the unused states on a Karnaugh map as shown in Fig. 9(c). Simplifying, using the normal techniques, gives $f = J_A = \bar{E}$.

For this circuit, decoding logic is required to obtain a decimal count. This logic is obtained from a five-variable Karnaugh map on which the decimal equivalent for each of the states in the counting cycle has been marked as shown in Fig. 9(d). The unmarked states on this map represent the unused states. The simplifying adjacencies for decimal 0 and 1 have also been marked on the map and if the reader cares to continue the process of simplification he will find that it is always possible to combine seven unused states with each decimal entry.

There are also three other undesired and independent count sequences for this counter. They are:

(1) S_2—S_5—S_{11}—S_{23}—S_{14}—S_{29}—S_{26}—S_{20}—S_8—S_{17}—S_2
(2) S_4—S_9—S_{19}—S_6—S_{13}—S_{27}—S_{22}—S_{12}—S_{25}—S_{18}—S_4
(3) S_{10}—S_{21}—S_{10}

If the counter should enter any one of these sequences, due to circuit misoperation, it will remain in that sequence unless arrangements are made to return the counter to the required sequence. This could be done by using the logic of the unused states to clear all stages of the counter and if required the same logic could be used to raise an alarm and stop the counter. It is left to the reader to show that the Boolean function that represents the unused states is:

$$f_u = \bar{A}D\bar{E} + \bar{A}B\bar{C} + \bar{A}C\bar{D} + A\bar{B}C + A\bar{D}E + AC\bar{D}$$

If it is required to make the counter self-starting it is only necessary to choose three adjacent states on the Karnaugh map, such as S_6, S_{14} and S_{10}, each of these states being from one of the unwanted sequences. If the Boolean function that represents these three states, $f = \bar{A}BD\bar{E} + \bar{A}BC\bar{E}$, is used to clear the five stages of the counter then within a maximum of ten clock pulses it will return to the desired sequence.

The Johnson counter has an even-numbered cycle length of $2n$, where n is the number of stages in the register. However, with a suitable modification of the feedback it is possible to achieve an odd-numbered cycle length of $2n-1$. For example, if the 00000 state is omitted the counting cycle becomes that shown in the table of Fig. 10(a) and the value of the new feedback function required to produce this sequence is tabulated in the column headed f. Plotting this function in conjunction with the unused states on the Karnaugh map of Fig. 10(b) and minimizing leads to the revised feedback function $f = \bar{D} + \bar{E}$. Similarly, if the 11111 state is omitted rather than the 00000 state the revised feedback function can be shown to be $f = \bar{D}\bar{E}$.

Shift registers with exclusive-OR fedback

The four-stage shift register shown in Fig. 11(a) has exclusive-OR feedback from stages C and D such that the input to the first stage $J_A = C \oplus D$. To determine the sequence of states for the register it is assumed that initially the shift register is in the state D = 0, C = 0, B = 0 and A = 1 in which case $J_A = 0 \oplus 0$, and on receipt of the next clock pulse the register enters the state D = 0, C = 0, B = 1 and A = 0. The complete sequence of states for the register is shown in Fig. 11(b), the value of the feedback function for each state being tabulated in the column headed f.

In all there are fifteen states and this is the maximum number of states a four-state register can have, so this sequence is termed the maximum length sequence. The $S_0 = 0000$ state is not included in the sequence since this is a 'lock-in' state. If the register enters this state $J_A = 0 \oplus 0 = 0$, so that the register is unable to leave this state when the next and subsequent clock pulses arrive. In general the maximum length sequence for such a circuit is given by the expression $p_{max} = 2^n - 1$, where n is the number of stages in the shift register.

Not all exclusive-OR connexions result in a maximum length sequence. The table in Fig. 12 gives the feedback functions which will give the maximum length sequence for values of n up to and including $n = 10$.

Clearly the circuit shown in Fig. 11(a) can be used as a binary sequence generator, the output sequence being taken directly from the output of one of the flip-flops in the register. In this case the binary sequence appearing at the output of flip-flop D is

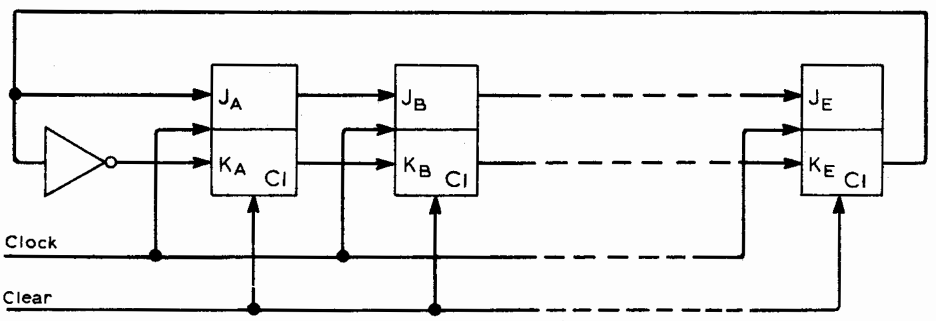
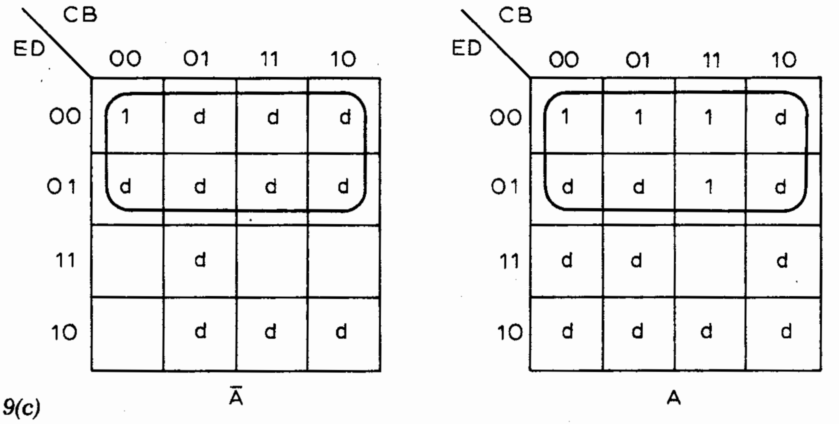


Fig 9(a)

Fig. 9. Twisted-ring decade counter (a) and its counting sequence (b). Maps at (c) indicate the method of determining the feedback function and at (d) the logic to decode the counter states for decimal indication.
Decode logic for outputs 0-9 is $\bar{A}\bar{E}$, $A\bar{B}$, $B\bar{C}$, $C\bar{D}$, $D\bar{E}$, AE, $\bar{A}B$, $\bar{B}C$, $\bar{C}D$, $\bar{D}E$.

Clock pulse	E	D	C	B	A	f
0	0	0	0	0	0	1
1	0	0	0	0	1	1
2	0	0	0	1	1	1
3	0	0	1	1	1	1
4	0	1	1	1	1	1
5	1	1	1	1	1	0
6	1	1	1	1	0	0
7	1	1	1	0	0	0
8	1	1	0	0	0	0
9	1	0	0	0	0	0

Fig 9(b)



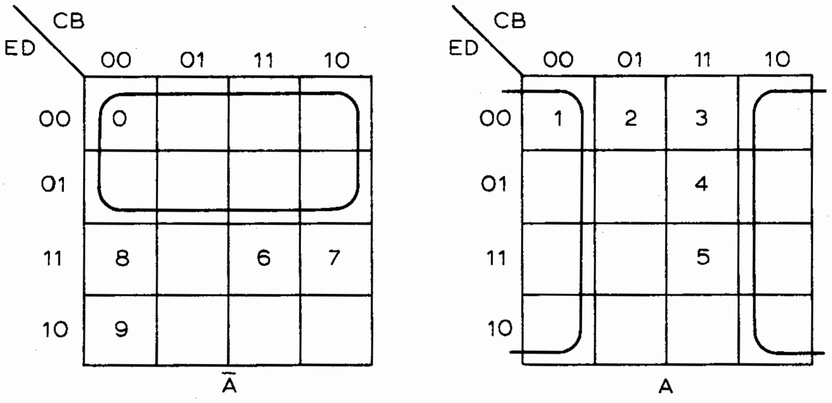
ED \ CB	00	01	11	10
00	1	d	d	d
01	d	d	d	d
11		d		
10		d	d	d

$\bar{A}$

ED \ CB	00	01	11	10
00	1	1	1	d
01	d	d	1	d
11	d	d		d
10	d	d	d	d

A

Fig 9(c)



ED \ CB	00	01	11	10
00	0			
01				
11	8		6	7
10	9			

$\bar{A}$

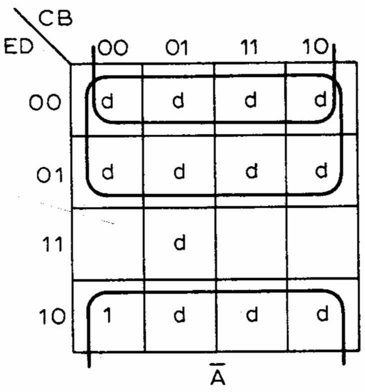
ED \ CB	00	01	11	10
00	1	2	3	
01			4	
11			5	
10				

A

Fig 9(d)

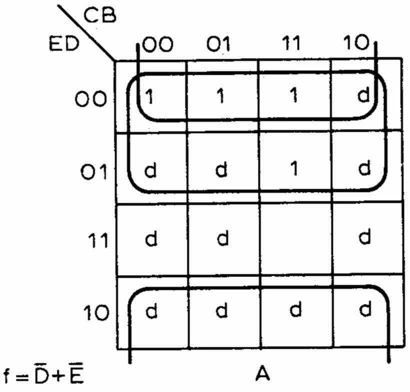
E	D	C	B	A	f
0	0	0	0	1	1
0	0	0	1	1	1
0	0	1	1	1	1
0	1	1	1	1	1
1	1	1	1	1	0
1	1	1	1	0	0
1	1	1	0	0	0
1	1	0	0	0	0
1	0	0	0	0	1

Fig 10(a)



ED \ CB	00	01	11	10
00	d	d	d	d
01	d	d	d	d
11		d		
10	1	d	d	d

Fig 10(b)



ED \ CB	00	01	11	10
00	1	1	1	d
01	d	d	1	d
11	d	d		d
10	d	d	d	d

Fig 10(c)

Fig. 10. Counting sequence of an odd-numbered cycle-length ring counter and the determination of its feedback function.

Fig. 11. Maximum-length four-stage register with exclusive-OR feedback ($f = J_A = C \oplus D$).

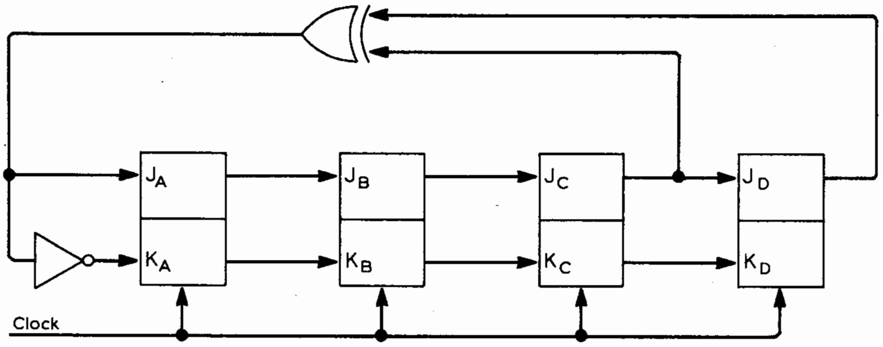


Fig 11(a)

(a)

S	D	C	B	A	f
S_1	0	0	0	1	0
S_2	0	0	1	0	0
S_4	0	1	0	0	1
S_9	1	0	0	1	1
S_3	0	0	1	1	0
S_6	0	1	1	0	1
S_{13}	1	1	0	1	0
S_{10}	1	0	1	0	1
S_5	0	1	0	1	1
S_{11}	1	0	1	1	1
S_7	0	1	1	1	1
S_{15}	1	1	1	1	0
S_{14}	1	1	1	0	0
S_{12}	1	1	0	0	0
S_8	1	0	0	0	1

Fig 11(b)

(b)

0-0-0-1-0-0-1-1-0-1-0-1-1-1-1.

Non-maximum length sequences can be generated with the register shown in Fig. 11(a) if some other exclusive-OR function is used as the feedback. For example, if the feedback function is B ⌐ D, one of the following sequences, tabulated in Fig. 13, will be generated. The sequence generated will depend upon the initial state of the register.

Generation of long register sequences

For values of *n* greater than five it is difficult to develop the de Bruijn diagram and hence the problem of designing a generator having more than thirty-one states using this diagram becomes quite complicated. A possible method of approach is to start with a maximum-length sequence generator using exclusive-OR feedback and then, if it is required, reduce the length of the sequence with additional feedback. The method will be described for a four-stage shift register, but it can also be used for shift registers having a number of stages in excess of four.

It will be assumed that a maximum length sequence generator having four stages is in the state D=0, C=0, B=1 and A=1.

Hence:

$$S = A \times 2^0 + B \times 2^1 + C \times 2^2 + D \times 2^3$$
$$= 1 \times 2^0 + 1 \times 2^1 + 0 \times 2^2 + 0 \times 2^3$$
$$= 3 = S_3$$

If when the generator is in this state, the feedback is a 0 then the next state of the generator is D=0, C=1, B=1 and A=0.

Then:

$$S = 0 \times 2^0 + 1 \times 2^1 + 1 \times 2^2 + 0 \times 2^3$$
$$= 6 = S_6$$

Alternatively, if the feedback had been 1 then the next stage of the generator would have been D=0, C=1, B=1 and A=1.

Hence:

$$S = 1 \times 2^0 + 1 \times 2^1 + 1 \times 2^2 + 0 \times 2^3$$
$$= 7 = S_7$$

Examination of the table for a four-stage maximum length sequence generator, given in Fig. 11(b), shows that the feedback $D \oplus C = 0$ when in the state S_3 and the next state is therefore S_6. However, if the feedback is modified so that it is 1 then the next state is S_7.

The state diagram for the maximum length sequence generator having four stages is shown in Fig. 14(a) and it can be seen that by modifying the feedback the states S_6, S_{13}, S_{10}, S_5 and S_{11} will be omitted from the sequence thus reducing its length from fifteen to ten states.

The modified sequence for the generator is shown in Fig. 14(b) and the modified value of the feedback function for state S_3 is encircled. The feedback function in conjunction with the unused states S_6, S_{13}, S_{10}, S_5 and S_{11} and the 'lock-in' state S_0 are plotted on the Karnaugh map and simplified in the normal way as shown Fig. 14(c). This gives a modified feedback function of:

$$f_m = C \oplus D + AB\bar{D} + \bar{A}\bar{B}\bar{D}$$

The complexities of designing a generator to produce a long binary sequence without computing aids are obviously formidable. However, a computer programme has been developed for shift registers using exclusive-OR feedback to give the maximum length sequence, which gives the following information:

- the present state of the generator,
- the next state of the generator,
- the jump state,
- the number of states excluded by the jump,
- the length of the modified sequence.

The designer has merely to scan the computer print-out to locate the length of sequence required and all the other

Fig. 12. Feedback function for maximum length sequences.

Fig. 13. A different feedback function (f = B ⊕ D) results in non-maximum-length sequence.

Fig. 14. State diagram of a 4-stage register with modified feedback, showing the jump.

n	f	n	f
1	A	6	E⊕F
2	A⊕B	7	F⊕G
3	B⊕C	8	D⊕E⊕F⊕H
4	C⊕D	9	E⊕I
5	C⊕E	10	H⊕L

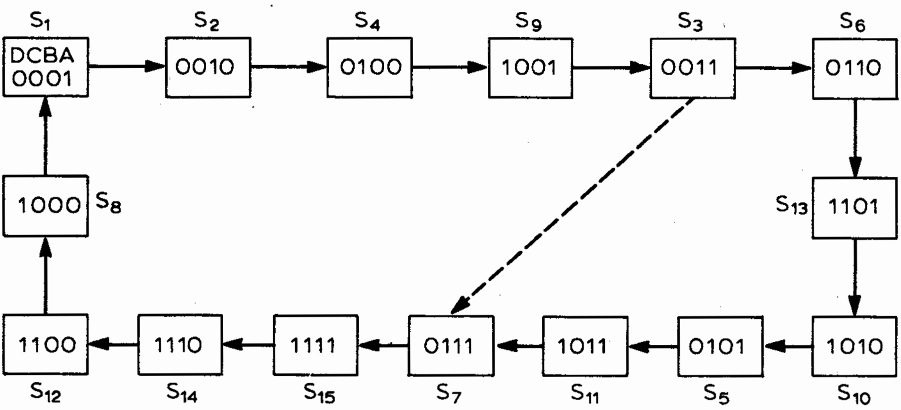
Fig 12

S	D	C	B	A	f
S_1	0	0	0	1	0
S_2	0	0	1	0	1
S_5	0	1	0	1	0
S_{10}	1	0	1	0	0
S_4	0	1	0	0	0
S_8	1	0	0	0	1

S	D	C	B	A	f
S_3	0	0	1	1	1
S_7	0	1	1	1	1
S_{15}	1	1	1	1	0
S_{14}	1	1	1	0	0
S_{12}	1	1	0	0	1
S_9	1	0	0	1	1

S	D	C	B	A	f
S_6	0	1	1	0	1
S_{13}	1	1	0	1	1
S_{11}	1	0	1	1	0

Fig 13



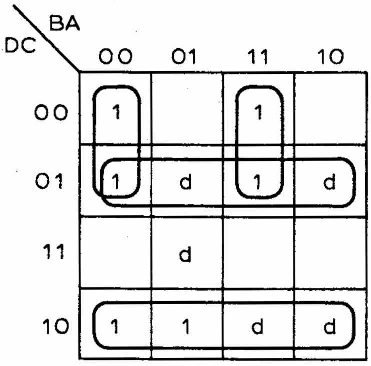
(a)

Fig 14(a)

S	D	C	B	A	f
S_1	0	0	0	1	0
S_2	0	0	1	0	0
S_4	0	1	0	0	1
S_9	1	0	0	1	1
S_3	0	0	1	1	①
S_7	0	1	1	1	1
S_{15}	1	1	1	1	0
S_{14}	1	1	1	0	0
S_{12}	1	1	0	0	0
S_8	1	0	0	0	1

(b)

Fig 14(b)



DC \ BA	00	01	11	10
00	1		1	
01	1	d	1	d
11		d		
10	1	1	d	d

(c)

Fig 14(c)

information is immediately available. Finally it is necessary to develop the combinational logic which will provide the modification to the exclusive-OR feedback logic required to produce the desired jump.

Further reading

For further information on this subject the reader is referred to the following texts:

Digital Engineering. G. K. Kostopolous. Wiley 1975.
Shift Register Sequences. S. W. Golomb. Holden-Day 1967.
Digital Logic and Switching Circuits. J. C. Boyce. Prentice-Hall 1975.
Electronic Counters. R. M. M. Oberman. Macmillan 1973.
Design of Digital Systems. J. B. Peatman. McGraw-Hill 1972.