Wireless World, December 1977

Logic design — 10

This article is the second part of Article 7, which appeared in the September issue.

More synchronous and ripple-through counters

by B. Holdsworth* and D. Zissos†

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

Decade binary 'up' counter. Examination of the unused states in Fig. 5(a) shows that they can be represented by the Boolean function BD + CD

The flip-flop equations are the same as for the scale-of-16 'up' counter, namely, $J_A = K_a = 1$, $J_B = K_B = A$, $J_C = K_C = AB$ and $J_D = K_D = ABC$, with the modifications shown below, which are required to inhibit the S_9 to S_{10} transition and initiate the S_9 to S_0 transition.

The transitions from S_9 to S_{10} and S_9 to S_0 are shown below:

	D	C	B	A		D	C	B	A
S_9	1	0	0	1	S_9	1	0	0	1
S_{10}	1	0	1	0	S_{10}	0	0	0	0

To inhibit the set of flip-flop B, $S_h = ABS_9$, where $S_9 = ABCD + (BD) + (CD)$. Simplifying: $S_9 = AD$, hence: $S_B = A\bar{B}\bar{A}\bar{D} = A\bar{B}\,(\bar{A}+\bar{D}) = A\bar{B}\bar{D}$. Therefore, $J_B = A\bar{D}$. To initiate the reset of flip-flop D, $R_D = S_9 = AD$. Therefore, $K_D = A$.

If the counter should assume one of the unused states due to circuit misoperation then a suitable corrective action might be to suppress the clock pulses and trip an alarm, using the Boolean function representing these states, $f = BD + CD$. A suitable circuit for suppressing the clock pulses is incorporated with the counter implementation in Fig.5(b).

A decade of binary 'down" counter can be designed using the same technique and the corresponding flip-flop equations are: $J_A = K_A = 1$, $J_B = \bar{A}C + \bar{A}D$, $K_B = \bar{A}$, $J_C = \bar{A}D$, $K_C = \bar{A}\bar{B}$ and $J_D = K_D = \bar{A}\bar{B}\bar{C}$. The output of a binary decade counter can be converted to a decimal number using a 4-10 line decoder as shown in Fig.6.

Consider the transition in such a counter from 0001 to 0010 and assume that flip-flop B changes faster than flip-flop A. The sequence of changes that take place are:

D	C	B	A	
0	0	0	1	
0	0	1	1	(transient state)
0	0	1	0	

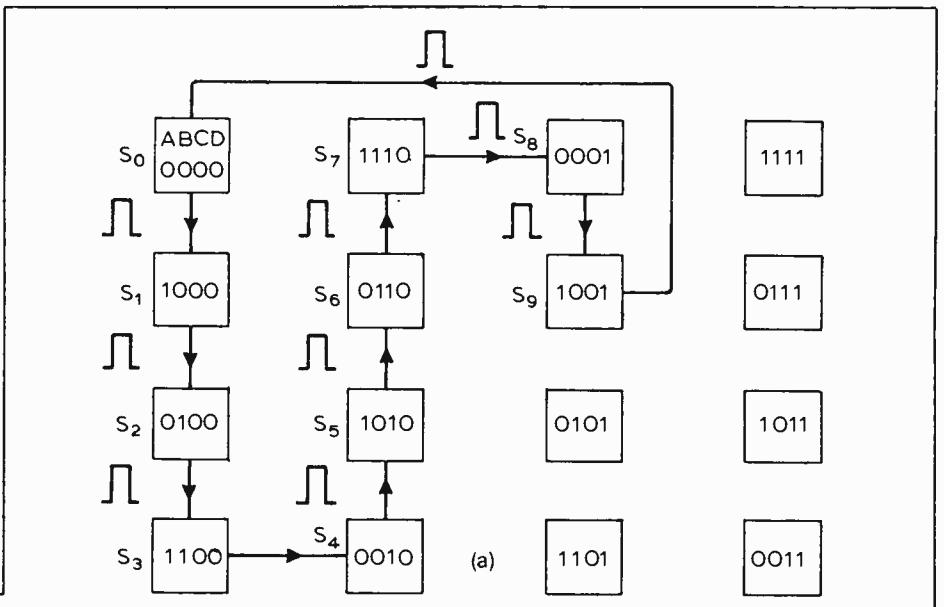
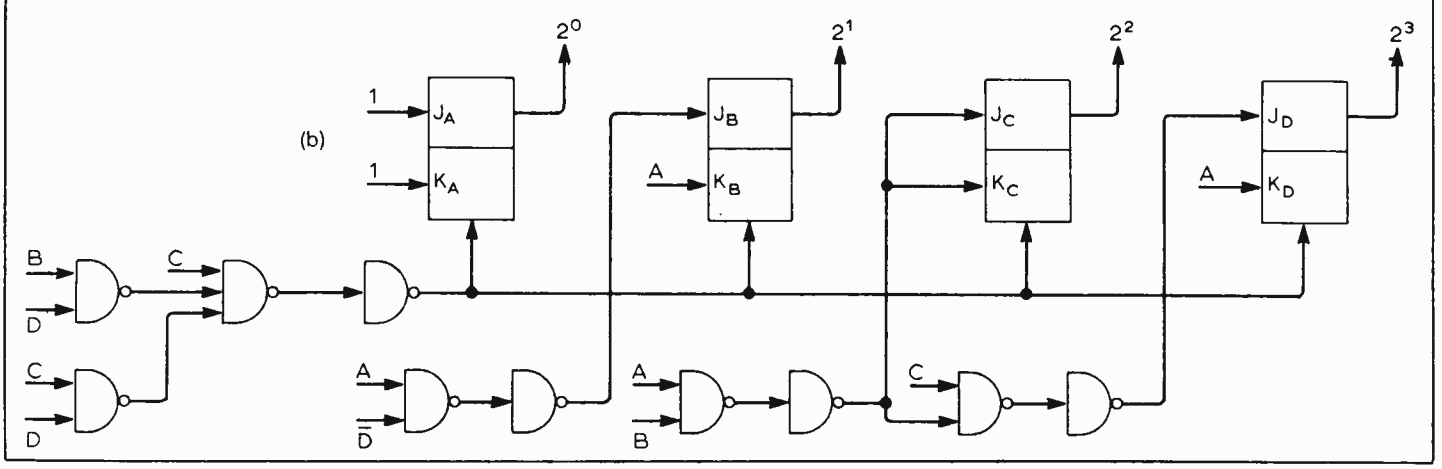


Fig. 5(a) shows the state diagram for a decode binary 'up' counter, and at (b) is the circuit implementation.



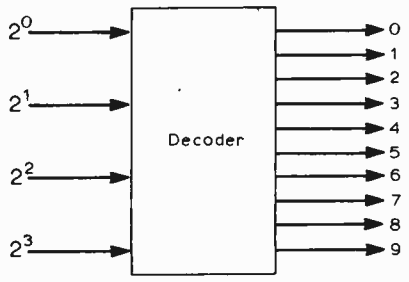


Fig. 6. A 4-10 line decoder.

D	C	B	A
0	0	1	0
0	1	1	0
0	1	1	1
0	1	0	1
0	1	0	0
1	1	0	0
1	1	0	1
1	1	1	1
1	1	1	0
1	0	1	0

Table 8. The XS-3 Gray code.

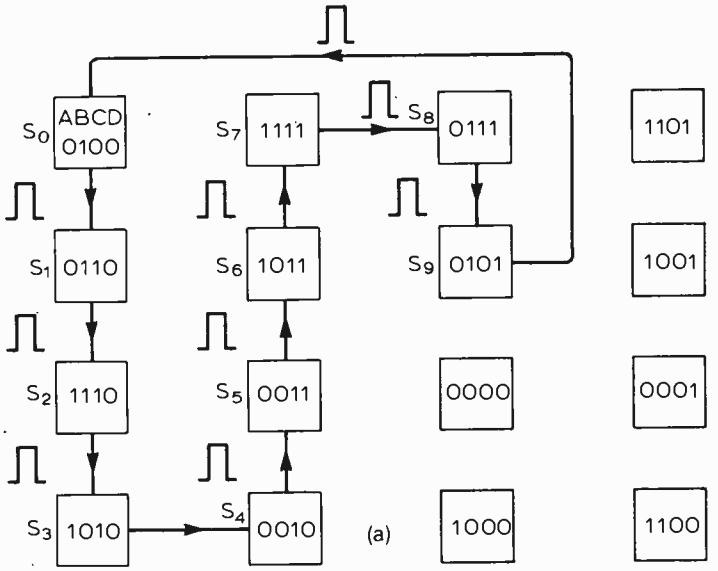
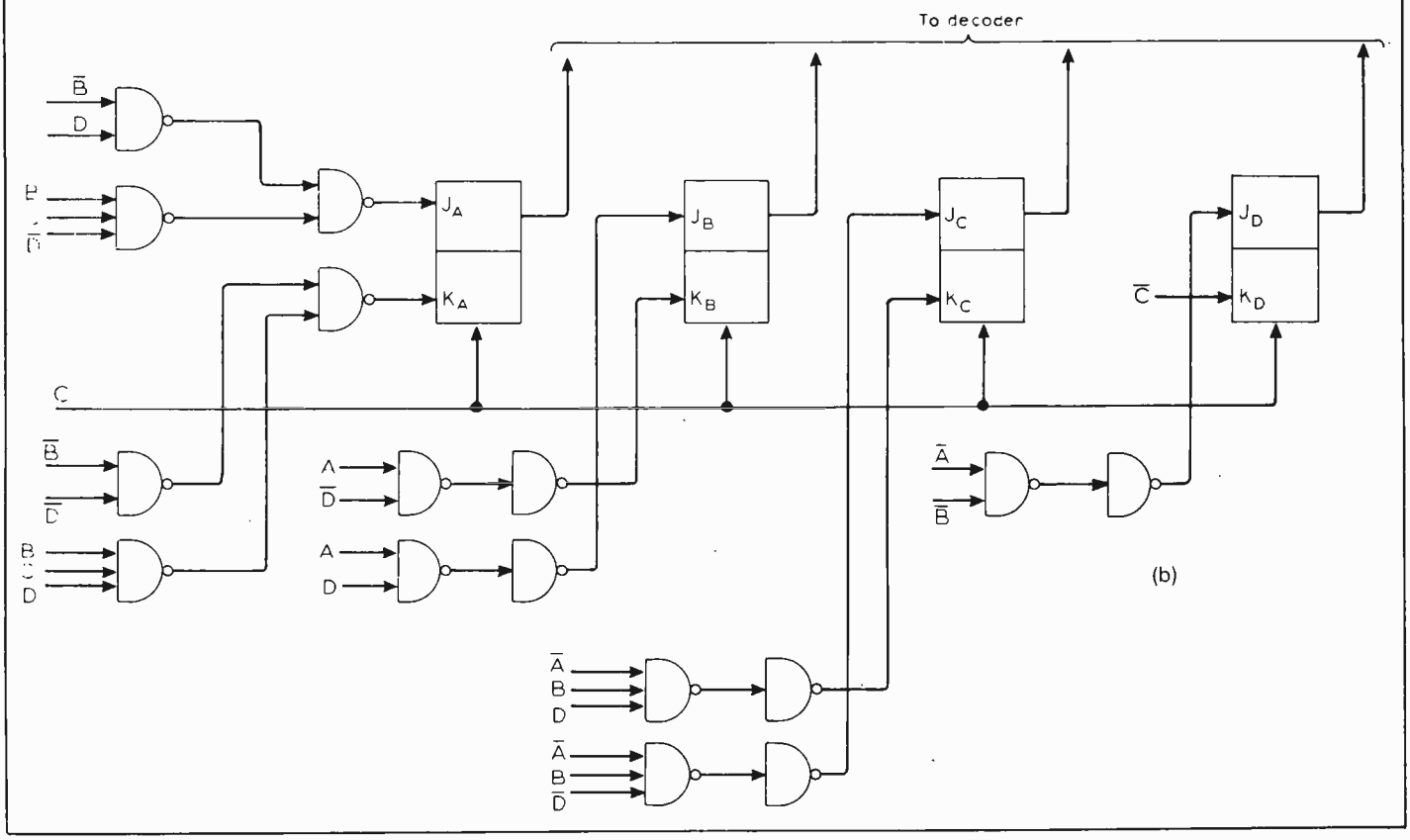
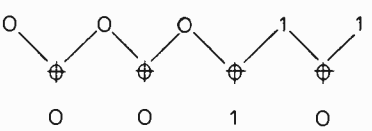


Fig. 7. State diagram for a XS-3 Gray code counter is at (a) and (b) shows the circuit.



Hence a spike will occur on the output line marked 3 during the transient state. Clearly this can occur at any point in the binary counting sequence where more than one flip-flop is required to change state during a transition. The difficulty can be eliminated by using a Gray code counter, in which only one flip-flop changes state at each transition.

Decade Gray code 'up' counter. As an example of the design of a Gray code counter, the XS3 code will be converted to a Gray code which will be used as the basis for the counter design. The conversion is carried out by obtaining the exclusive-OR sum of each pair of digits in the code starting with the two least significant digits first, as shown below. It is assumed that there are five digits in each code combination, the fifth and most significant, always being 0.



The complete XS3 Gray code obtained using this procedure is shown in Table 8.

A convenient procedure for designing a Gray code counter is:

(1) Determine the S and R expressions for each flip-flop and, using the equations $S_Q = J_Q\bar{Q}$ and $R_Q = K_Q Q$, obtain the corresponding expressions for J and K.

(2) Optional products defined by the unused states, (if there are any,) can now be used to reduce the J and K expressions. For the state diagram and codes, see Fig.7(a).

Examination of the unused states in Fig.7(a) shows that they can be represented by the Boolean expression $\bar{B}\bar{C} + A\bar{C}$.

The flip-flop equations are:

$$S_A = S_1 + S_5 = \bar{A}BC\bar{D} + \bar{A}\bar{B}CD$$
$$J_A = BC\bar{D} + \bar{B}CD + (\bar{B}\bar{C}) + (A\bar{C})$$
$$= \bar{B}D + BC\bar{D}$$

$R_A = S_3 + S_7 = A\bar{B}C\bar{D} + ABCD$
$K_A = \bar{B}C\bar{D} + BCD + (\bar{B}\bar{C}) + (A\bar{C})$
$= \bar{B}\bar{D} + BCD$

$S_B = S_6 = A\bar{B}CD$
$J_B = ACD + (\bar{B}\bar{C}) + (A\bar{C})$
$= AD$

$R_B = S_2 = ABC\bar{D}$
$K_B = AC\bar{D} + (\bar{B}\bar{C}) + (A\bar{C})$
$= A\bar{D}$

$S_C = S_0 = \bar{A}B\bar{C}\bar{D}$
$J_C = \bar{A}B\bar{D} + (\bar{B}\bar{C}) + (A\bar{C})$
$= \bar{A}B\bar{D}$

$R_C = S_8 = \bar{A}BCD$
$K_C = \bar{A}BD + (\bar{B}\bar{C}) + (A\bar{C})$
$= \bar{A}BD$

$S_D = S_4 = \bar{A}\bar{B}C\bar{D}$
$J_D = \bar{A}\bar{B}C + (\bar{B}\bar{C}) + (A\bar{C})$
$= \bar{A}\bar{B}$

$R_D = S_9 = \bar{A}B\bar{C}D$
$K_D = \bar{A}B\bar{C} + (\bar{B}\bar{C}) + (A\bar{C})$
$= \bar{C}$

The circuit implementation of the counter is shown in Fig.7(b).

The output of the counter can be converted directly to decimal with the aid of a 4-10 line XS3-Gray-to-decimal decoder, which is available as a chip.

If the above procedure is adopted for the design of an XS3-Gray code decade 'down' counter the following flip-flop equations are obtained:

$J_A = \bar{B}\bar{D} + BCD$, $J_B = A\bar{D}$, $J_C = \bar{A}BD$, $J_D = \bar{C}$.

$K_A = \bar{B}D + BC\bar{D}$, $K_B = AD$, $K_C = \bar{A}B\bar{D}$, $K_D = \bar{A}\bar{B}$.

The unreduced, and hence the reduced, J and K values of the flip-flops in one direction are the same as the K and J values of the flip-flops in the reverse direction, i.e. to reverse the direction of count it is only necessary to interchange the J and K inputs of each flip-flop.

It should be noted that the method of design employed does not always produce the simplest flip-flop equations but it has advantages when applied to the implementation of 'up-down' Gray code counters.

'Up-down' control

'Up-down' counters are counters in which the pulse count is stepped up or stepped down by each input pulse according to whether the value of an external control signal R is 0 or 1. In practice, the input signals that step the count up or down will appear on two separate lines X and Y, as shown in Fig.8(a). The designer has to generate the clock pulses that will drive the counter flip-flops and the 'up-down' control signal R. It will be assumed that step-up and step-down signals do not appear simultaneously.

The control signal must not be allowed to change during the presence of the input data. In this case, the control signal will be generated from the input data and a race condition can be prevented by using the first pulse in each pulse train to change the value of R. Since each time the value of R is changed an input pulse is not counted this method results in a maximum count error of 1 for an odd number of changes in R and no error for an even number of changes in R.

The logic circuit used to perform the function described above is event driven and the methods used in the third article of this series will be employed in its design.

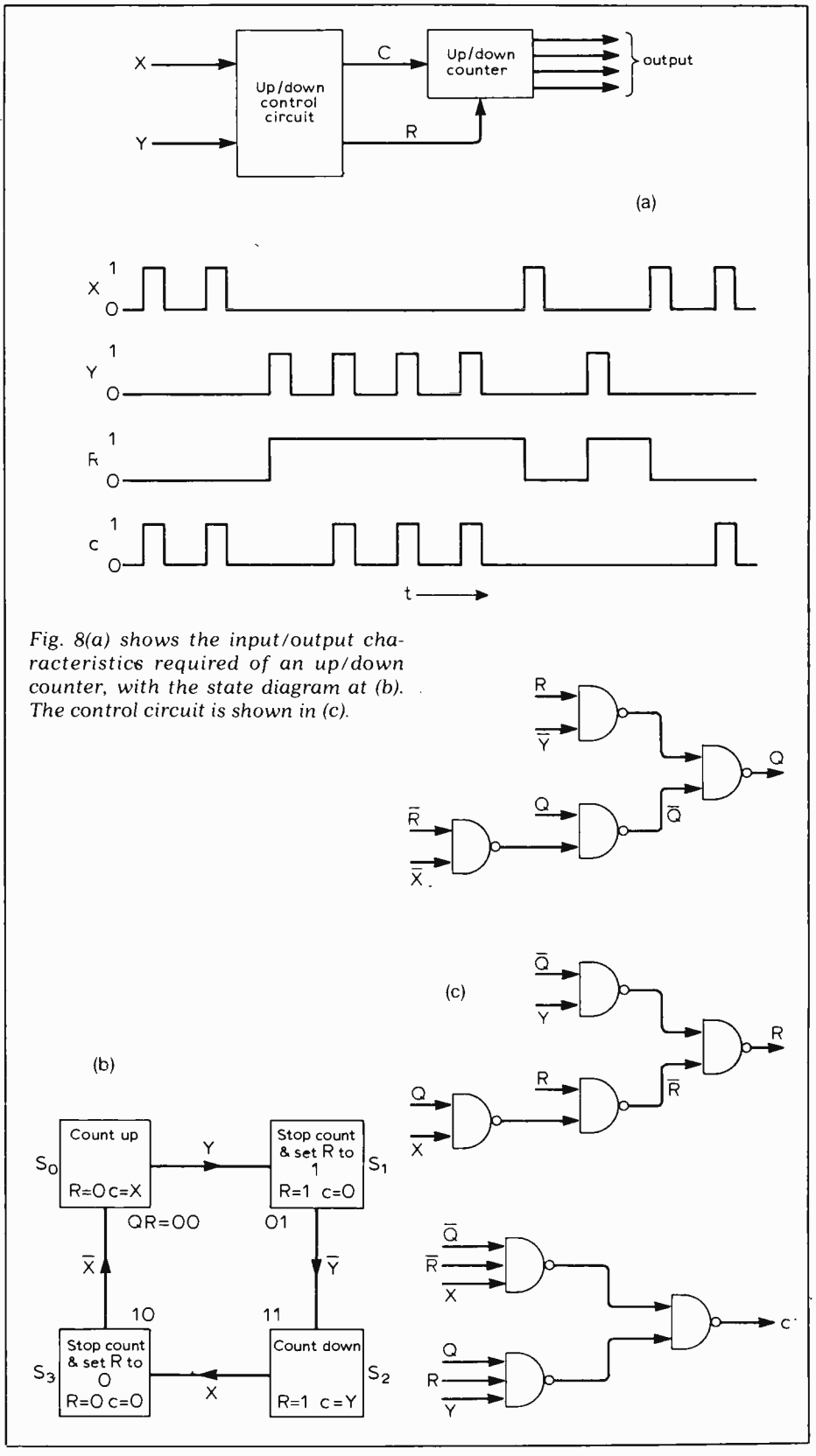


Fig. 8(a) shows the input/output characteristics required of an up/down counter, with the state diagram at (b). The control circuit is shown in (c).

Step 1 Figure 8(a) shows the input/output characteristics.
Step 2 A suitable state diagram is shown in Fig.8(b).

Step 3 State reduction is not attempted so that clarity of design is maintained.
Step 4 Turn-on set of $Q = R\bar{Y}$
Turn-off set of $Q = \bar{R}\bar{X}$
Turn-on set of $R = \bar{Q}Y$
Turn-off set of $R = QX$

The sequential equations are:
$Q = \bar{R}\bar{Y} + Q(\bar{R} + \bar{X})$
$R = \bar{Q}Y + R(\bar{Q} + \bar{X})$
and $c = S_0X + S_2Y = \bar{Q}\bar{R}X + QRY$
The circuit implementation is shown in Fig.8(c).

'Up-down' XS3-GRAY code counter. Combining the flip-flop equations for 'up' counts when $R = 0$ and for 'down' counts when $R = 1$ the following results are obtained for the XS3-Gray code counter:

$$J_A' = (\bar{B}D + BC\bar{D})\bar{R} + (\bar{B}\bar{D} + BCD)R$$
$$= J_A\bar{R} + K_AR$$
$$K_A' = (\bar{B}\bar{D} + BCD)\bar{R} + (\bar{B}D + BC\bar{D})R$$
$$= K_A\bar{R} + J_AR$$

Similarly
$J_B' = J_B\bar{R} + K_BR$
$K_B' = K_B\bar{R} + J_BR$
and so on for J_C', K_C', J_D', and K_D', where J' and K' are used to denote the flip-flop inputs in the 'up-down' mode.

Asynchronous binary counters

For counts of powers of 2 the basic arrangement consists of T flip-flops, (or alternatively JK flip-flops with J and K permanently connected to 1), connected in cascade as shown in Fig.9(a). As can be seen from the diagram the output of each flip-flop provides the clock signal for the next. The input signal X is used as the clock pulse for the first flip-flop.

The time-diagrams for a scale-of-8 (up) counter are shown in Fig.9(b), where all changes of state are assumed to take place on the trailing edge of the clock pulses. Examination of the time diagrams shows that flip-flop A changes state on each trailing edge of the input pulses X. The output of flip-flop A is used as the clock pulse for B and a change in state of this flip-flop occurs on the trailing edge of the A pulses. Similarly the output of B provides the clock pulse for C and this changes state on the trailing edge of the B pulses.

The various states of the counter are indicated on the time diagram and the binary digits associated with each state are marked on the time diagrams for the signals A, B, and C.

It is a simple matter to show that the above circuit will count down if the signals $\bar{A}$ and $\bar{B}$ are used as the clock pulses for flip-flops B and C respectively.

Scale-of-ten 'Up' Counter.

This circuit requires four flip-flops, as shown in Fig 10(a). The associated time diagrams are displayed in Fig.10(b). Starting with all the flip-flops in the 0

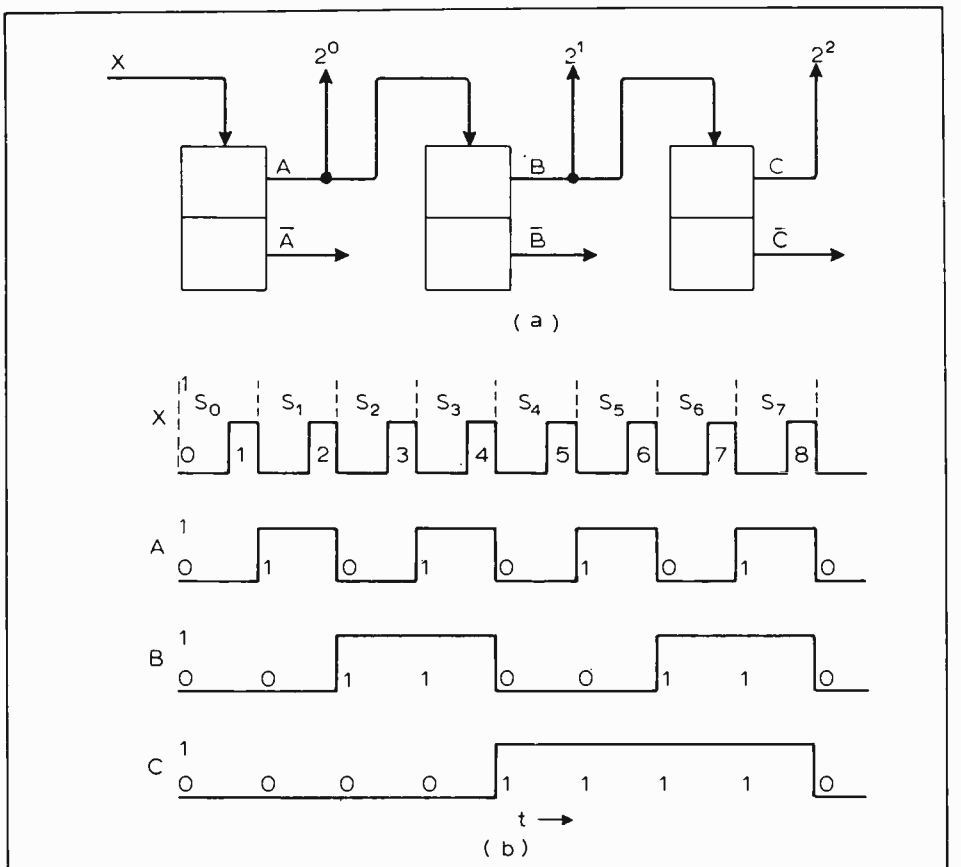


Fig. 9(a) is a 3-stage ripple-through counter, and at (b) are the timing diagrams.

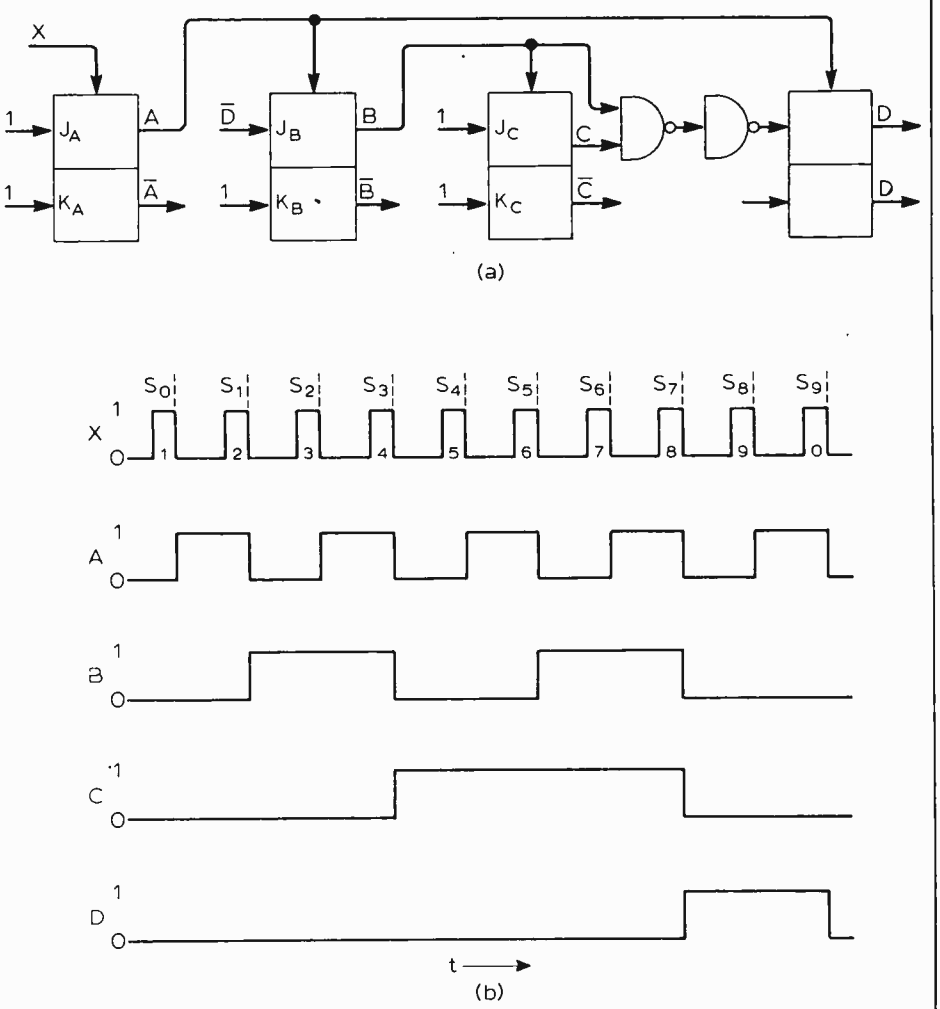


Fig. 10 is a scale-of-ten ripple-through counter. Timing diagrams are at (b).

state, the count follows the normal binary sequence up to and including the count of eight. On the trailing edge of the tenth input pulse, flip-flop A makes a transition from 1 to 0, which would normally induce a transition in flip-flop B, changing its state from 0 to 1. However $J_B = \bar{D} = 0$ at this instant and consequently flip-flop B remains in the reset condition. At the same instant it is also necessary to reset flip-flop D and it changes state from 1 to 0. All the flip-flops are now in the reset condition and are ready for the arrival of the first pulse of the next counting cycle.

Scale-of-twelve 'Up' counter

The basic circuit of a scale-of twelve asynchronous counter is shown in Fig.11(a), whilst the time diagrams describing its behaviour are shown in Fig.11(b). Flip-flops A, B, and C count from 000 to 101 inclusive. With $D = 0$ the counter reaches the state $ABCD = 1010$ (S_5), and when the next X input pulse is received it must go to the state $ABCD = 0001$.

Flip-flop A is controlled by the X pulses and changes state to $A = 0$ on the trailing edge of the sixth of these pulses. Flip-flop B remains in the $B = 0$ state since $J_B = C = 0$ and flip-flop C takes up the state $C = 0$ since $J_C = B = 0$ and $K_C = 1$. The change of C from 1 to 0 represents the trailing edge of the clock pulse for flip-flop D and hence there is a change of state for this flip-flop such that $D = 1$.

After another six X pulses the state of flip-flop D is restored to 0 and the counting cycle of twelve states is completed.

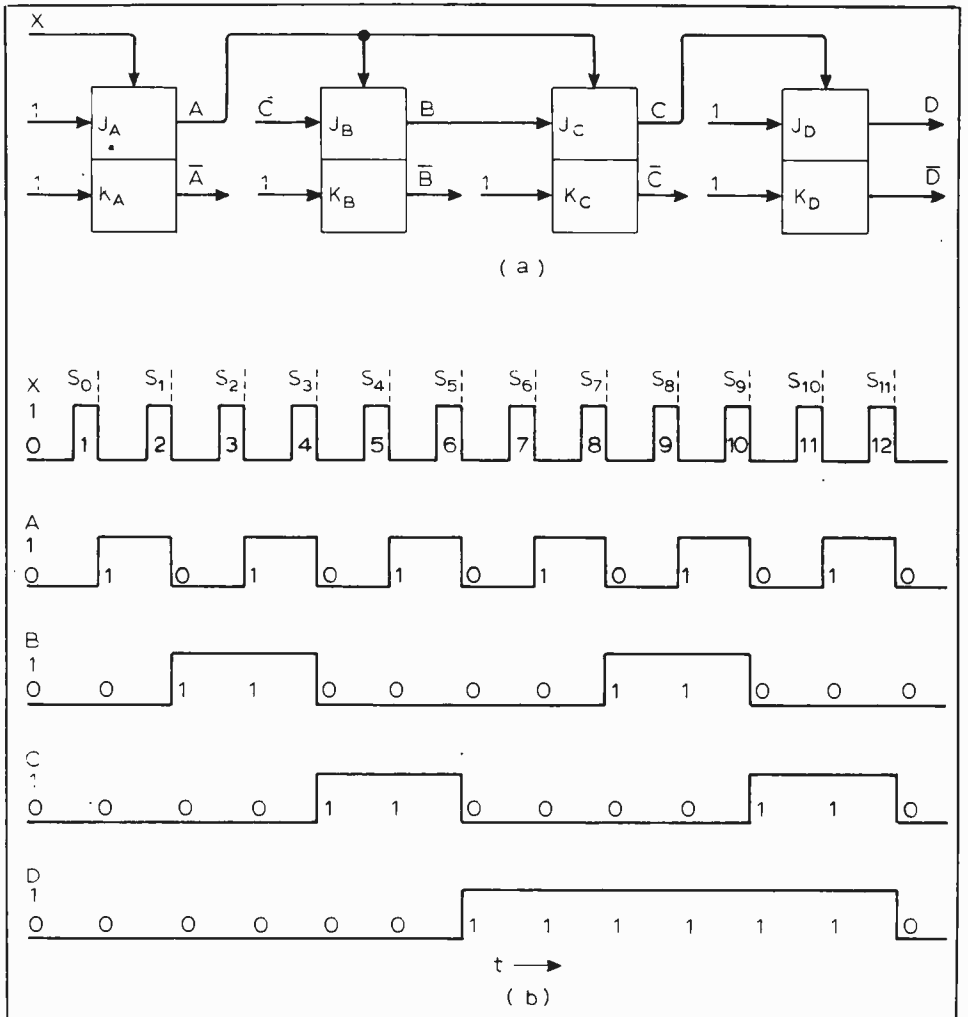


Fig. 11 shows the form and timing diagrams of a scale-of-12 asynchronous counter.