Digital Electronics By Experiment pt6

Continuing our popular series on TTL.

RIPPLE COUNTERS are useful and simple, but they are not ideal for high counting speeds, nor for large counter chains. The problem arises from the use of the output of each flip-flop as the clock for the next flip-flop, so that changes must "ripple through" all the stages of the counter. This, as indicated in the previous section, causes difficulties with time delays.

Although these delays are not large, perhaps 60 ns or less per flip-flop, they accumulate to a significant amount over a large number of counter stages and can cause the race hazards mentioned earlier.

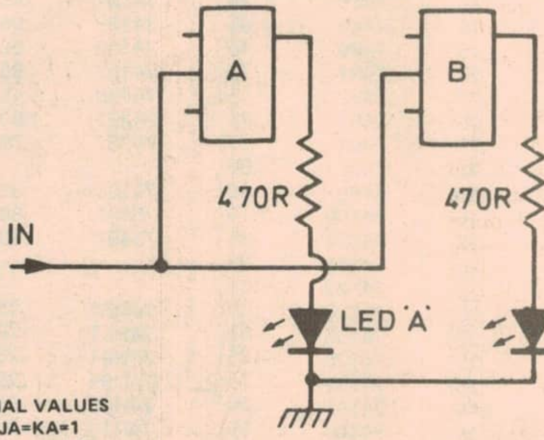
Synchronous Counters

A different principle is used for synchronous counters. The input pulses are used to clock *each* flip-flop of a chain, hence the name synchronous. The count sequence is then determined by voltages applied to the J and K terminals, and these voltages must be obtained in such a way that any given count on the flip-flop will cause the J and K inputs of the next digit up or down to be set correctly.

This is much more easily illustrated by an example which we can test on our board. In this example we shall follow the pattern of design steps (with some modifications) which is usually used for synchronous counters.

Basic Two-Step

Let us imagine a very basic counter using two flip-flops and resetting at the count of four. We must start by making a table showing the count, the present state, and the next state for each flip-flop. This means that for each number of the count we list the value of Q (1 or 0) and also the value to which Q will change at the next count. For example, when the count is 1 (01), the next count is 2 (10) and both outputs will change — A from 1 to 0, and B from 0 to 1. On the next count (3).



FINAL VALUES
JA=KA=1
JB=KB=QA

	A		B		J K VALUES			
COUNT	Q PRESENT	Q NEXT	Q PRESENT	Q NEXT	JA	KA	JB	KB
0	0	1	0	0	1	X	0	X
1	1	0	0	1	X	1	1	X
2	0	1	1	1	1	X	X	0
3	1	0	1	0	X	1	X	1

Fig.1 (Above). A simple synchronous counter, no J-K connections shown. (a) Circuit. Note that the input clock is taken to each stage. (b) Table of changes, with J and K values for the changes.

A changes from 0 to 1, and B does not change. The complete table for two flip-flops is shown in Fig. 1(b).

Now we have to decide what voltages are needed at J and K of each flip-flop to carry out the changes from present state to next state. Here we have some options — for example, if we want to change from 1 to 0, we may have J=K=1, or J=0 and K=1; either state will carry out the change. When this is possible, we can write J=X, K=1, where X means don't care, since either value of J is equally suitable.

Add more columns to the table to indicate these values of J and K for each flip-flop, and we are ready to start designing. The object now is to obtain the J and K voltages for each flip-flop from somewhere else in the circuit in such a way that all the J and K voltages are correct for each stage of the count. The formal method of doing this involves a technique called *Karnaugh mapping*, but is seldom necessary for only a few counter stages. It is rather difficult to apply for a large number of stages, so only the 'intuitive' look-and-see method will be discussed here.

Table Talk

At the zero count, Qa=0, Qb=0 and the change at the end of the clock pulse will be from Qa=0 to Qa=1. This will happen if Ja=1 and for Ka=0, or Ka=1. We therefore fill in a 1 in the Ja column, and an X (either value) in the K column.

Still at the zero count, Qb=0 and does not change at the end of the clock pulse. This can be done if Jb=0, Kb=X, so that these values 0 and X appear in the Jb and Kb columns.

These columns are filled in similarly for each change listed — remembering to use X in any case where a value is unimportant — using the J-K table that we used in Part 4 of this series of articles.

We can now inspect the complicated tables to see if any values can be fixed or derived from values of Qa or Qb. The tables for Ja and Ka are easily dealt with — since the values are either 1 or X, we can use 1 for all these values, and make Ja=1, Ka=1, as for the ripple counter. The Jb, Kb tables are slightly more involved, but for each definite value of one quantity (J or K) there is an X for the other, so that we can again connect J and K. We then find that the values of J and K are identical to the values of Qa, so that Jb and Kb can be connected to Qa.

For practical work on synchronous counters it is useful to have a lock pulse line, and one of the spare lines on the board can be used. Connect up the circuit as shown, with a slow clock pulse taken to each clock input, and wire connections linked from Qa to Jb and Kb. Use LEDs as before to check the state of each flip-flop output. Connect a common reset line to each flip-flop and to a switch so that the counter can be reset. Switch on and check that the count is correct and that resetting to zero is possible.

Third Stage Development

Let us now extend this to a third stage, building on what we have done before. Once again we can build up a table of values of Q, J and K for each stage, but we have made life easier for ourselves by having done the two stage counter, so we can ignore the Ja, Ka and Jb, Kb columns and concentrate on the Jc, Kc column.

Using the same principles as before, we fill in the values of J and K which will be needed at each clock pulse or flip-flop, concentrating on the necessary values, and putting an X where the value is immaterial. When we do this (Fig. 3b) we find two important states. One is at the count of 3, where Jc must change from 0 to 1; the other is at the count of when Kc similarly changes from 0 to 1.

The change of Jc from 0 to 1 occurs when the count changes to 110 so that we could use an AND gate connected to Qa and Qb. The output of this gate will be zero for any count up to 2 and then will be 1 at a count of 3. It will change to zero again to become 1 at the count of 7, but the value of Jc is unimportant beyond the count of 3 anyway.

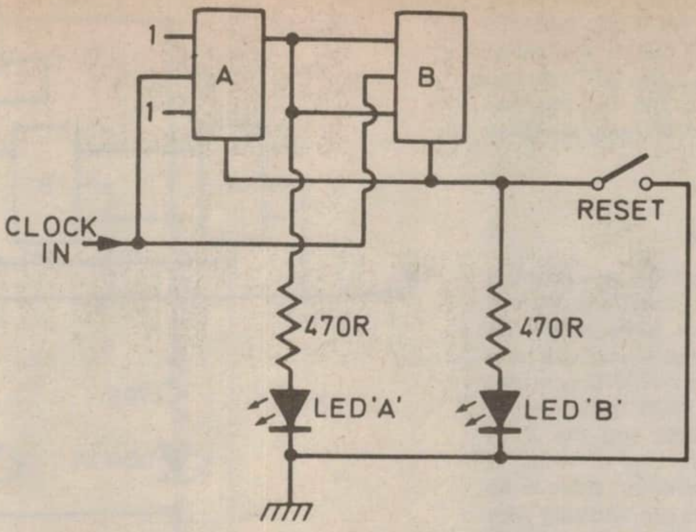
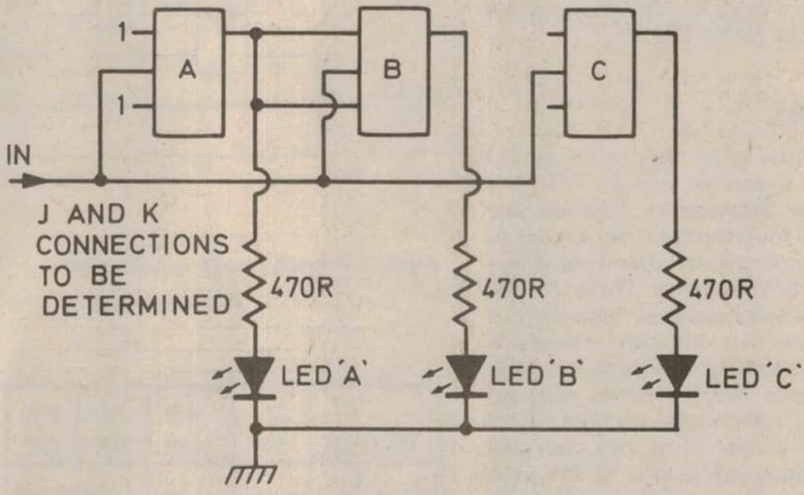


Fig.2 (Above). Complete two-stage synchronous counter circuit, with J-K connections shown. Try this out on your blob-board.



COUNT	PRESENT QA	PRESENT QB	PRESENT QC	NEXT QA	NEXT QB	NEXT QC	JA	KA	JB	KB	FIRST JC	FIRST KC	FINAL JC	FINAL KC
0	0	0	0	1	0	0	1	1	QA	QA	0	X	0	0
1	1	0	0	0	1	0	1	1	QA	QA	0	X	0	0
2	0	1	0	1	1	0	1	1	QA	QA	0	X	0	0
3	1	1	0	0	0	1	1	1	QA	QA	1	X	1	1
4	0	0	1	1	0	1	1	1	QA	QA	X	0	0	0
5	1	0	1	0	1	1	1	1	QA	QA	X	0	0	0
6	0	1	1	1	1	1	1	1	QA	QA	X	0	0	0
7	1	1	1	0	0	0	1	1	QA	QA	X	1	1	1

JC=QA AND QB
KC=JC

Fig.3 (Above). A three-stage synchronous counter. Top: (a) Circuit, J and K connections still to be determined. Bottom: (b) Table of changes, showing how J and K values are determined. The "first" Jc-Kc table shows possible values of Jc and Kc, the "final" table shows the most convenient values to use.

Looking at Kc we find that the important value of 1 occurs at a count of seven when Jc may also be 1. We can therefore connect Jc and Kc together and feed from an AND gate supplied with Qa *and* Qb.

Third Stage On Board

Making up a three-stage synchronous counter on the circuit board needs some additional connections. Since we are not using AND gates, the gate used will have to be made up from a NAND gate and an inverter. As the 7400 contains four two-input NAND gates and the 7414 contains six inverters, one of which is used as the clock oscillator, there is no shortage of gates. We are working with a low frequency clock, so there should be no ill-effects caused by the number of wires soldered across the board, but a high speed counter would have to be built on a PCB designed for the purpose, using copper tracks on each side and with decoupling capacitors between +ve and −ve lines close to each flip-flop.

Can you now go one step further to design a four stage synchronous counter and try it out on the board?

Twisted Logic

A different type of synchronous counter is shown in Fig. 5. This is a Johnson, or 'twisted-ring', counter and consists of four flip-flops connected so that the output of one drives the J and K inputs of the next. Three of the connections are made up with Q to J and $\overline{Q}$ to K, but the feedback connection is made with Q to K and $\overline{Q}$ to J — hence the alternative name of twisted-ring. Remembering the $\overline{Q}$ is always the inverse of Q, can you plan out the values of Q and $\overline{Q}$ for each counter? Use the table headings shown in Fig. 5(b) and remember that Qa=Jb, Qb=Jc, Qc=Jd, Qd=Ka and so on.

A Johnson counter uses a completely different count sequence from conventional binary counters, and the maximum count number is twice the number of flip-flops. The counters are synchronous, very easy to design and also very simple to decode for use with lamp indicators.

Build up the four stage (count of 8) Johnson counter of Fig. 5(c) on your circuit board and check that your calculations are correct.

Next month, we shall delve into decimal counting.

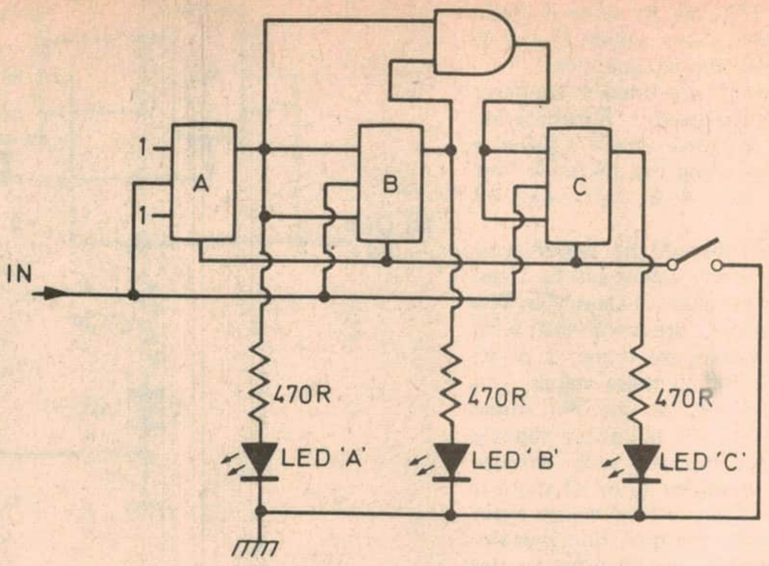
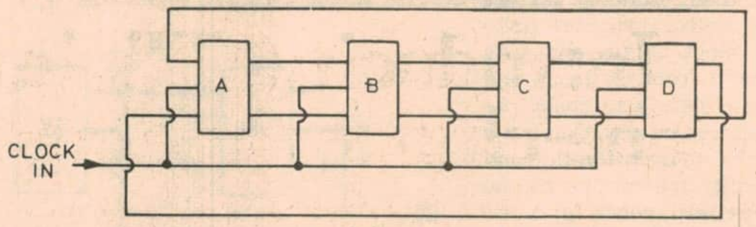


Fig.4 (Above). The circuit of the three-stage synchronous counter. Try this out on your blob-board.



NO RESET LINE SHOWN

Fig.5. A Johnson counter of four stages.

	JB	KB	JC	KC	JD	KC	KA	JA
CLOCK	QA	$\overline{Q}$A	QB	$\overline{Q}$B	QC	$\overline{Q}$C	QD	$\overline{Q}$D
0								
1								
2								
3								
4								
5								
6								
7								

COUNT	A	B	C	D
0				
1				
2				
3				
4				
5				
6				
7				

Fig.6. Top: (a) The circuit, note the "twisted ring" connection. Bottom: (b) Table to complete so that the counter action can be predicted. Below: (c) Truth table. Build the circuit on your blob-board and complete this table.