UNIVERSITY OF NEWCASTLE UPON TYNE

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

PART I ELECTRICAL LABORATORY PROJECT

DIGITAL SELF-LOCKING COUNTER

by

Ole J Jacobsen

Part I Electrical Engineering and Computing Science

LIST OF CONTENTS

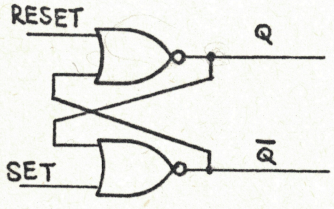
1. INTRODUCTION

The given problem is to design, build and test a digital self-locking counter. The counter should lock automatically when a pulse count of five has been recorded. A new counting cycle is initiated after a RESET switch has been pressed.

2. THEORY

Counters form an important part of modern digital computers. Since computers are sequential machines obeying instructions one by one, there is a need for various counting circuits to keep track of the computers operation. Terms such as "PC" (Program Counter) and "Stackpointer" are the software equivalents of actual digital counters. Furthermore there is often a need for circuits to produce longer counting sequences or "states" and our self-locking counter is a very simple example of such a specialized counting circuit. The technique of what is normally termed "Sequential Design" is described in a separate section although it was not actually used for this project.

Counters are made using the basic computer memory block namely the flip flop. A basic R-S flip flop is shown in fig. 1 with its truth table. The important thing to note about this device is the feedback: the output affects the input. Important is also the memory function: the device will "hold" its value once set or reset.

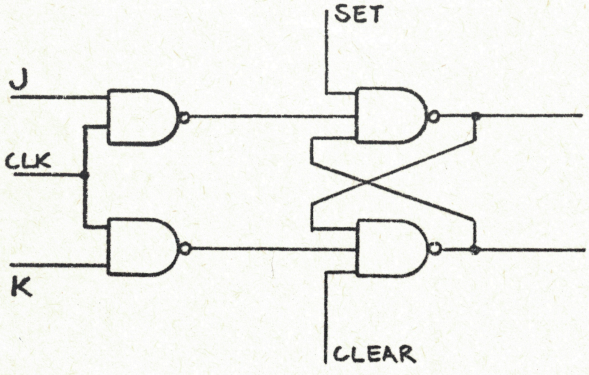


R	S	Q_{NEW}
0	0	Q_{OLD}
0	1	1
1	0	0
1	1	Illegal

Fig. 1

Since computers are high speed syncronous devices, there is a need for outputs to change synchronously with a clock signal common to the entire system. The inputs are therefore gated with this clock signal and the outputs will depend on how the inputs change during the time the clock is high. (At logic 1)

The resulting circuit is shown in fig. 2 with its truth table. Here the J and K inputs are gated with the CLK and the device will only change state according to the truth table when the clock goes low. The device is hence called a negative edge triggered J-K flip flop. As can be seen from the table, the output will toggle if J and K are both held at logic 1. If 3 of these devices are cascaded as shown in fig 3: the ouput of the first driving the CLK of the next, we have a basic 3 bit asynchronous counter. The A,B and C outputs will take on successively all values from 000 to 111 and is thus a binary 0-7 counter. The self locking counter built in this project uses J-K flip flops in the same configuration with some additional logic to provide the locking. It should be noted that in practise Master-Slave flip flops are often used instead of the simpler single flip flops. The operation is the same however, the difference being that in the M-S configuration "race-around" conditions are avoided since the outputs are not fed directly back to the inputs but "buffered" by an extra flip flop. J-K flip flops normally have independant (asynchronous) set and clear and these can be used to either clear the counter (by connecting all 3 Cl's together as shown) or to "manually" set it to some arbritrary value.



J	K	Q_{NEW}
0	0	Q_{OLD}
0	1	0
1	0	1
1	1	Toggle ($\overline{Q_{OLD}}$)

Fig. 2

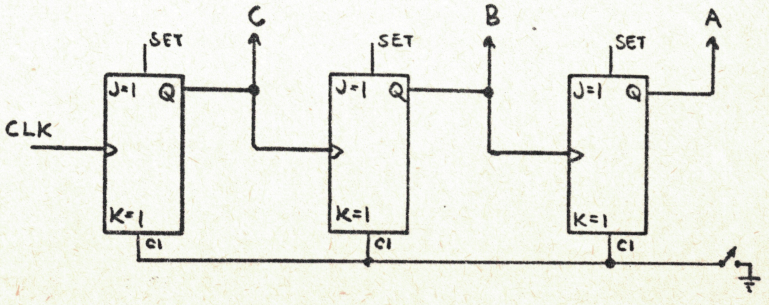


Fig. 3

3. DESIGN APPROACH

There are several ways of designing a self-locking counter and the particular solution will usually depend on the type of logic available and the specific application. For this project we have chosen the rather simple solution of having some logic monitoring the output that will disable the clock when the count of 5 is reached. The circuit of fig 4 shows the initial design. The NAND gate has inputs $A\bar{B}C$ and F will consequently go low when the counter reaches 5. F is gated with the clock and when it goes low the clock cannot pass through the AND gate and the counter is locked. This circuit however has a problem associated with it: In practical logic gates there is a time delay from input to output due to the internal transistors switching. Thus if the clock is switching very rapidly, the logic may not have time to "shut the gate" before the next clock pulse arrives and we may end up with 6 instead of 5 on the counter. The solution to this is shown in fig 5. Here we have added a flip flop between the logic and the AND gate, and instead of $A\bar{B}C$ the NAND gate now detects $A\bar{B}\bar{C}$ i.e 4. This gives the logic a whole clock cycle to settle down before the disableling of the clock takes place. When the counter reaches 4, F goes low and is passed on to the flip flop but the flip flop does not change state until the next count since its clock input is driven from X as shown. This circuit will only be limited by the maximum TTL clock rate which is about 30 Mhz. (This depends very much on the shape of the clock pulses,- it may be difficult for the logic to trigger at high frequencies if the signal isn't "clean"). Since the AND gate needs to be "open" initially, the RESET switch is connected to the SET input of the new flip flop. This flip flop is what is called a D-type flip flop and is made by connecting an inverter between the J and K inputs of a J-K flip flop.

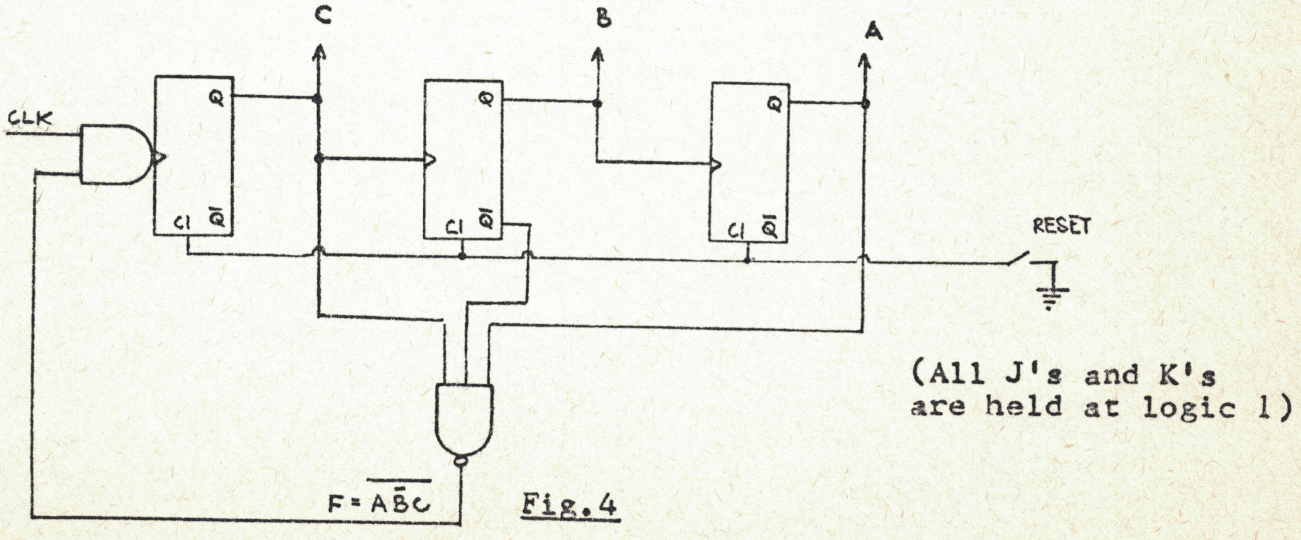


Fig.4

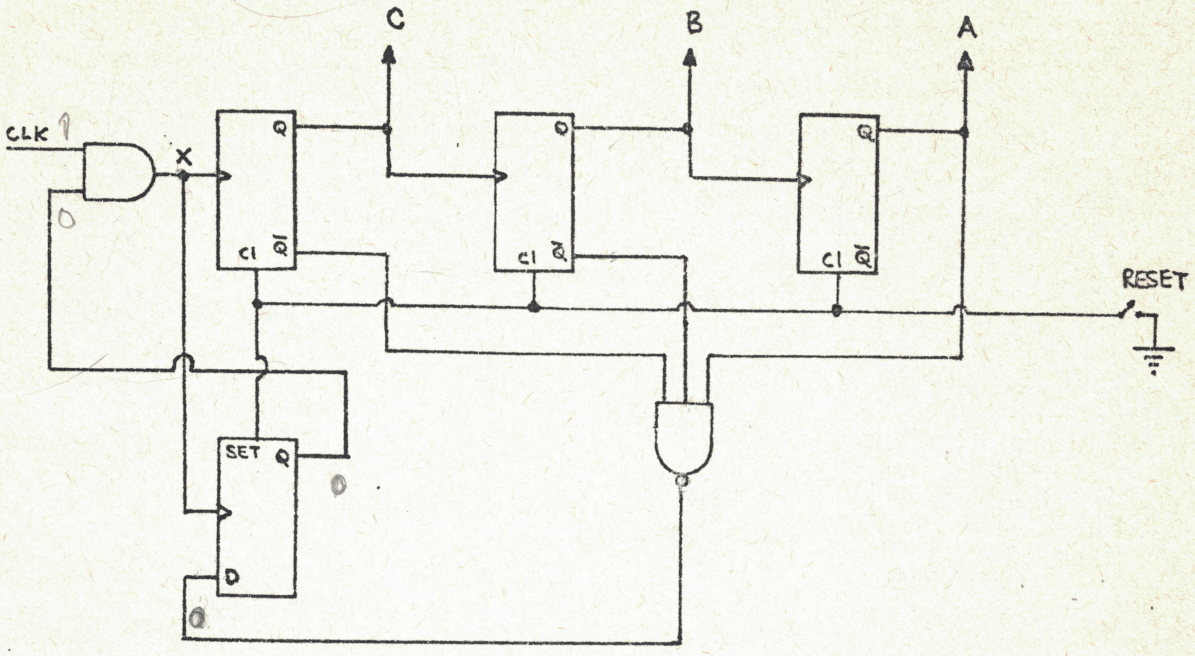


Fig.5

The circuit of fig.5 is identical to the one actually built except for some minor details described in the following section.

4. THE HARDWARE

The circuit was implemented on a Veroboard with layout as shown in fig 6. Unfortunately I was unable to obtain a negative edge triggered D-type flip flop and had to use the positive edge triggered one instead. Thus the clock input of the D-type is inverted. The TTL chips used are:

7400	Quad 2-input NAND Gates	(1x)
7404	Hex Inverters	(1x)
7412	Tripple 3-input NAND Gates	(1x)
7474	Dual D-type Flip-Flop	(1x)
74107	Dual J-K Master-Slave F.F.	(2x)

(See "National Semiconductor TTL Databook" for details)

For the power supply a 6V EVERREADY battery (PJ996/4R25) is used and the supply stabilized with a 250µF capacitor. The link is for disconnecting the clock input of the D-type flip flop thus making the circuit count normally from 0 to 7 if required.

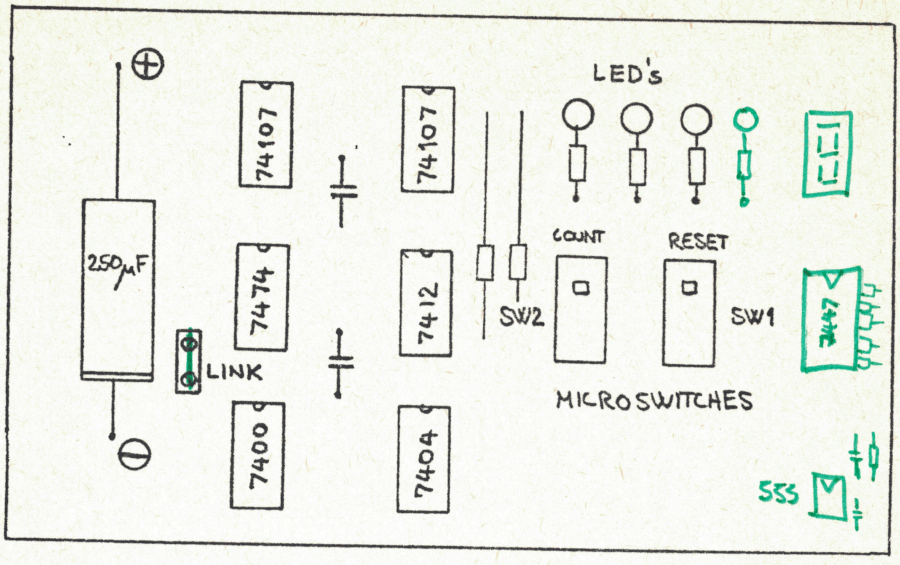


Fig.6

Normal mechanical switches produce "noisy" signals as shown in fig 7 a). This signal cannot be used as a clock since the peaks will cause random triggering of the flip flop. To overcome this difficulty we use a so called bounce-free contact consisting of two NAND gates as shown in fig 7 b). This circuit is essentially a flip flop and will give an approximate square wawe output that can be used to simulate high speed clock signals normally used in computers.

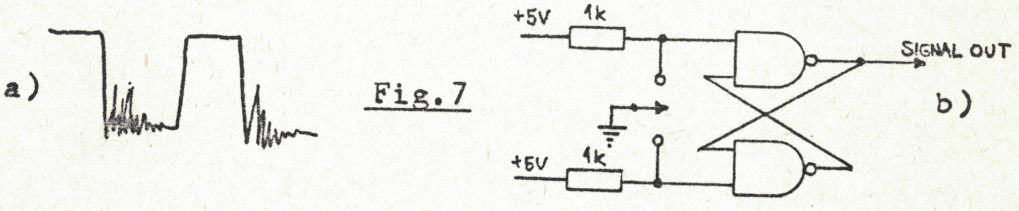
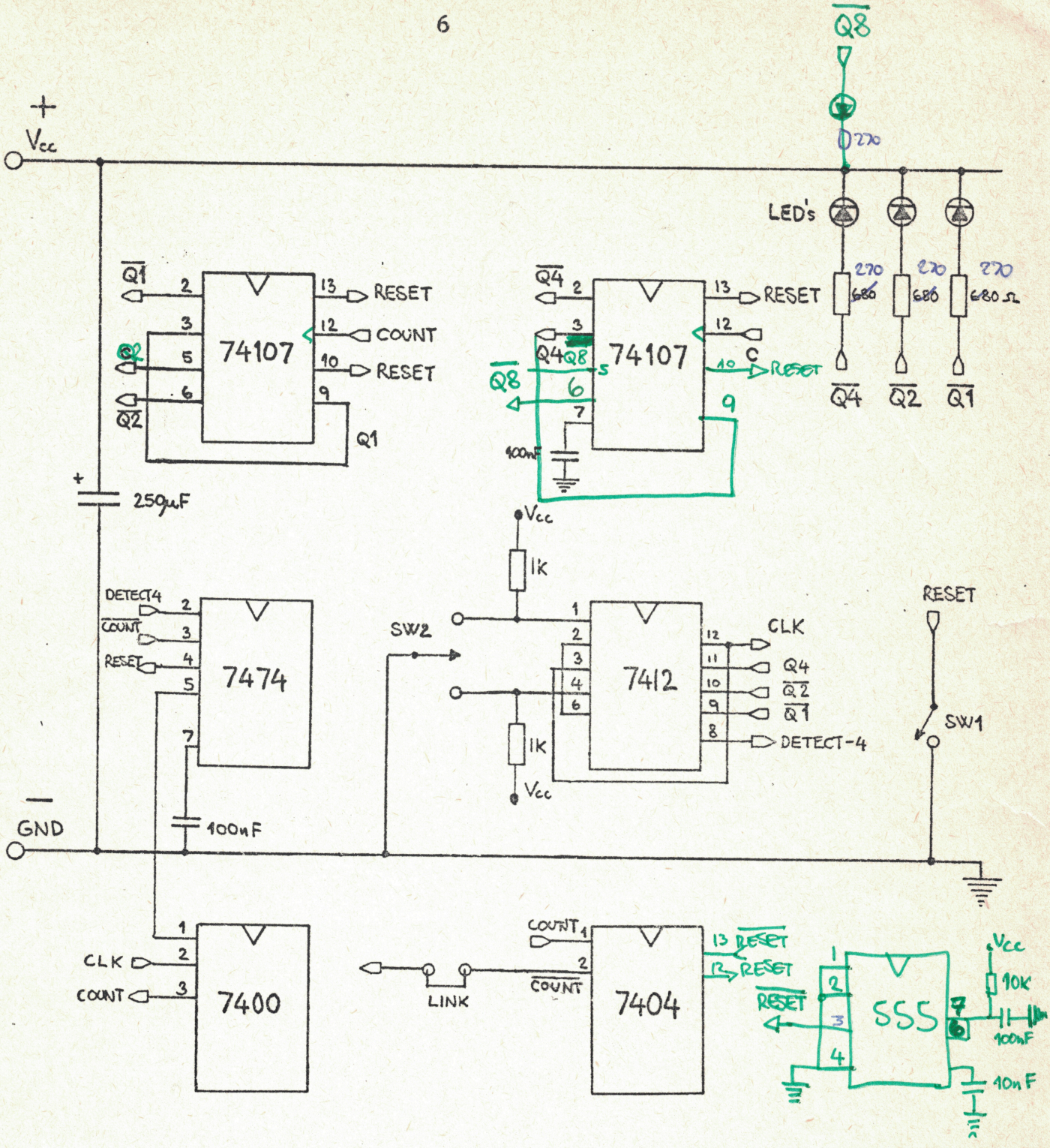


Fig.7

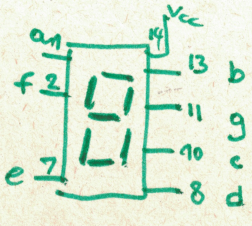
The LED's are driven from the $\overline{Q}$ outputs with a 680 ohm resistor to limit the current. This is done since the $\overline{Q}$ outputs are capable of much more current drive than the Q outputs. In TTL logic all unused inputs can be regarded as high, thus the J and K inputs of the counter are left unconnected. In more complex and critical circuitry it is customary to connect unused inputs to +5V via a 1k ohm pullup resistor to avoid "floating" signals which may cause erroneous operation. The complete circuit diagram is shown in fig 8. (Again refer to the TTL Databook for details on the chips used). The total current drawn by the circuit is about 135mA thus a reasonable battery life can be expected.

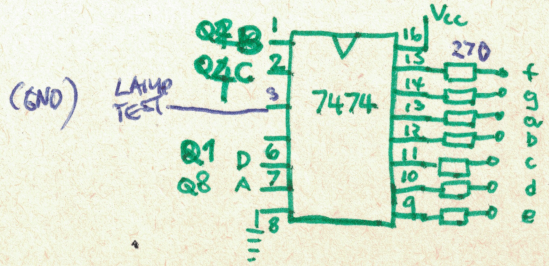


NOTE: Vcc is on pin 14 and GND on pin 7 on all chips.

The 100nF capacitors are used to stabilize for rapid current requirement changes. It is recomended to use one of these capacitors for every 4 TTL chips on the board.

Fig.8





5. CONCLUSION

The circuit was built as described in the hardware section and found to operate as predicted. It was particularly interesting to see how effective the bounce-free switch operated. During construction a normal switch without the debouncing circuitry was used and the counting sequence was very random indeed. The only "problem" was rather poor light intensity from the LED's used, but this can easily be overcome either by using ready made driver circuits or reducing the series resistors somewhat. Another interesting exercise that could have been carried out is to test the circuit at higher clock frequencies rather than using the press button. This could not be done however due to lack of time and the appropriate equipment (CRO etc.).

6. REFERENCES

"National Semiconductor TTL Databook", 1976

"Introduction to Digital Computer Technology"
Louis Nashelsky, Wiley, 1977

"Integrated Electronics"
Millman and Halkias, McGraw-Hill, 1972

Newcastle upon Tyne,
June 23 1980

Ole J. Jacobsen

Ole-Jørgen Jacobsen

7. SEQUENTIAL DESIGN

The technique of sequential design is widely used in computer engineering. There is often a need for circuits to take on a number of states in a specific order: producing sequences of pulse trains, switching inputs to multiple outputs, producing special purpose count sequences etc. A simple state diagram is shown in fig 9. The important thing to notice is that there are two "variables" controlling the condition for the next state: The DATA signal (typically a mode switch) and the current state. For example: We go from state B to state C only if DATA=1, but there are several ways of getting to state A depending on at what state we are in at present. The procedure of sequential desing can be outlined in the following steps:

1. Decide how many states there are and draw state diagrams.
2. The number of flip flops required is determined by the number of states, e.g. 8 states requires 3 ff's since they can take on $2^3=8$ combinations.
3. From state diagrams, timing diagrams or otherwise map out all the required J and K logic for each flip flop, simplifying where possible.

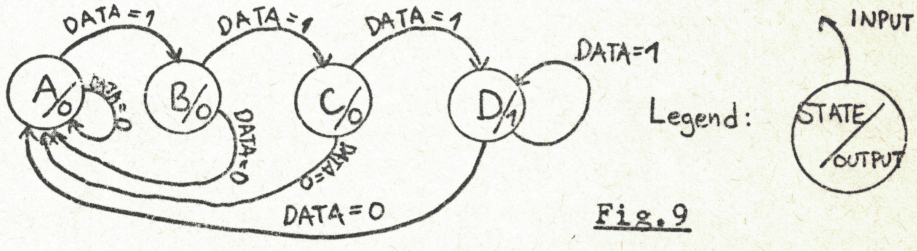


Fig.9

Let us look at a simple example: We are required to design a Mod-5 counter which has the counting sequence of fig 10. The counter is of the form shown in fig 11: 3 flip flops synchronously clocked by a common clock signal. We need 3 ff's because there are 5 states. Our problem is to develop the logic needed for each J and K input. Looking at the count table we realize that the J and K inputs for stage A must be 1 <u>except</u> for the count of 4 (100). Thus if we make $J_A=\overline{C}$ and $K_A=1$ so that both inputs are 1 for toggle action <u>except</u> when C=1 when $J_A=0$ leaving the stage reset on the next clock pulse. For stage B we note that the state should toggle for count steps 1 and 3 when stage A's output is 1, so that the logic inputs are: $J_B=K_B=A$. Finally for stage C we have that the stage sets on step 4 but is reset on all other steps, so that $J_C=AB$ and $K_C=1$.

C	B	A	step
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
0	0	0	0

Fig.10

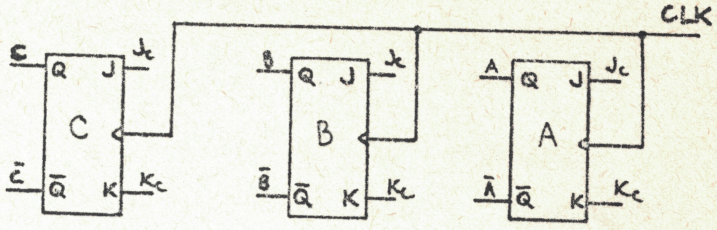


Fig.11

In a more complex circuit where we cannot readily see what the logic J and K inputs are, we would use the more systematic approach of mapping J and K from the action table. The action table for a J-K flip flop is shown in fig 12. Here the next state is described in terms of J and K, i.e. the table tells us what J and K must be for the individual transitions. The x signifies "don't care" condition i.e the variable may be either 0 or 1.

Q_n	Q_{n+1}	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

Fig.12

Using the information from this table, Karnaud maps can be drawn and the logic derived for each of the J and K inputs. Fig 13 shows the complete Mod-5 counter as derived on the previous page. The technique of sequential design could well have been adapted for our self locking counter circuit.

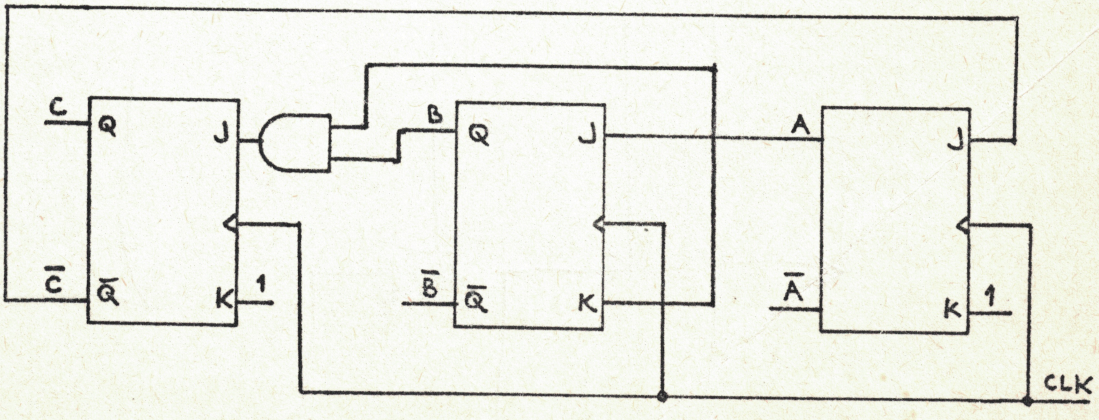


Fig.13