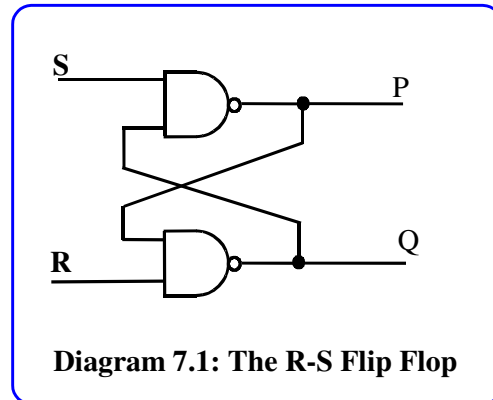


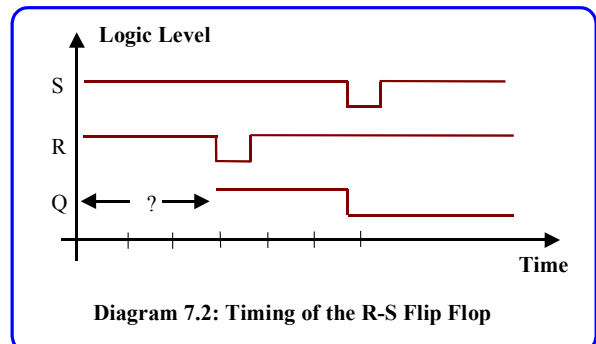
Lecture 7: Memory

The circuit of Diagram 7.1 is called a flip-flop or bi-stable. We will consider its truth table, and immediately find that we have a problem. We can construct the table for the three states where at least one input is zero. This is because if any input to a nand gate is zero, the output will be one regardless of the other input, and we can therefore work around the loop to calculate all the values. However when R and S are 1 we cannot immediately calculate P and Q; so the only way we can analyse what happens is to look at the possible values that P and Q could hold. Thus we can expand the truth table to include a further two inputs Pp and Qp where the subscript *p* indicates the previous value.

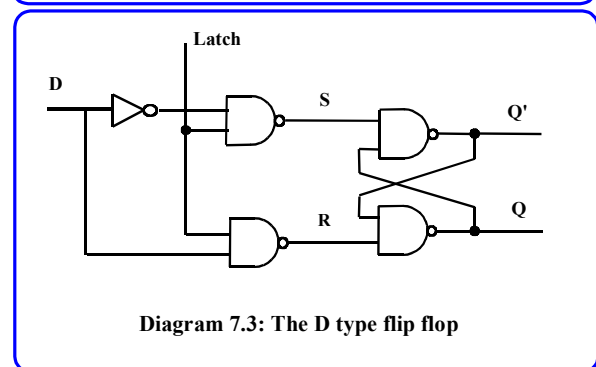
S	R	Pp	Qp	P	Q	
1	1	0	0	1	1	Unstable
1	1	0	1	0	1	Stable
1	1	1	0	1	0	Stable
1	1	1	1	0	0	Unstable



Here we see that there are two states where $P=P_p$ and $Q=Q_p$ (1101 and 1110 respectively). These are therefore stable states. The other two states (1100 and 1111) are unstable, and using the simple model with time delay *t* will oscillate with a period of $2t$. In practice, the circuit will fall into one of the two stable states rather than oscillate, since the time delays of the two nand gates will not be precisely the same. Which state it will finish in is non-deterministic. In practice we are not interested in the non-deterministic states, only in the stable ones.

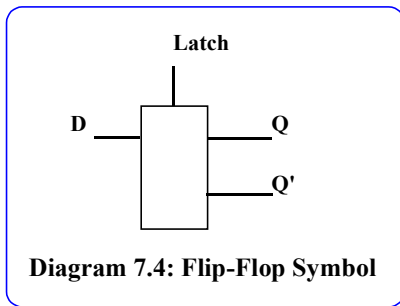


This circuit can be considered to be a one bit memory circuit since Q can be set to one or zero. To see this we need to look at a sequence of inputs as shown in Diagram 7.2. At the third time step we have the input 10 which puts the circuit into a known state and the output Q to 1. That value of Q is memorised and remains as long as the input is kept at 11. At the sixth time step the input 01 forces the output Q to be a 0, and as long as the input is held at 11 this 0 remains. This way of looking at the circuit gives rise to the names of the inputs S for Set and R for Reset, and so this flip flop is usually given the name R-S. The following three points should be noted. We are now describing the behaviour by means of a sequence of inputs, and for this reason, these circuits are referred to a sequential. Secondly, in all the cases of interest for this circuit $P=Q'$. Thirdly, an R-S flip flop can equivalently be built out of NOR gates.



The R-S flip flop is a building block from which other circuits can be constructed. The first refinement that we need to make is some form of gate onto the R and S inputs, so that we can

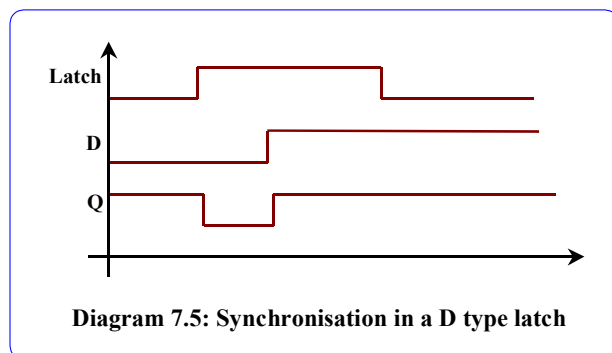
control the times at which the set or reset inputs are presented. This circuit is shown in Diagram 7.3. The first thing to note about this circuit is that when the L (latch) input is zero, the values of S and R are both 1, in other words the R-S flip flop will hold its previous value.



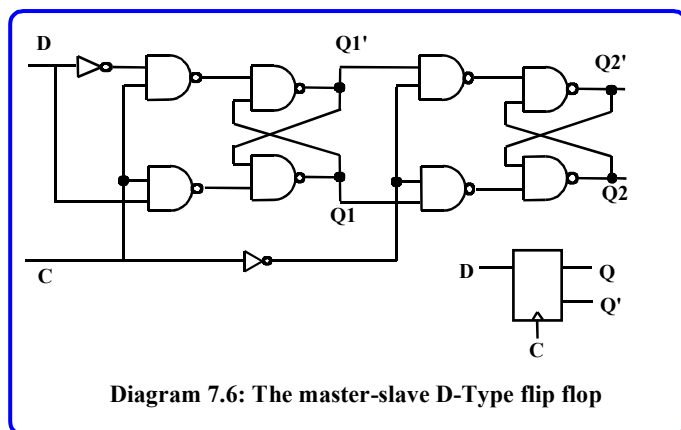
When the latch is 1, we have that $S=D$ and $R=D'$, and the value of Q will therefore be set to the value of D. This circuit now forms a practical controllable memory. The L input acts as a *write* line in that whenever it is set to 1 the value of D is written to the flip flop i.e. the output $Q=D$. When $L=0$ the value of Q is kept. The truth table for this can be written using the previous subscript notation, and the common symbol for this D-type flip flop is shown in Diagram 7.4. Notice that the flip flop is now nicely encapsulated in that it is not possible for the user to

evoke any non deterministic states. The only time that the circuit could be undetermined is from the time that the power is switched on until the latch input is set to 1. In practical designs we would need to ensure that the latch goes to 1 momentarily as the circuit is switched on to ensure that the behaviour is correct.

The simple D type flip flop does however have drawbacks which will be seen if we consider a timing diagram shown in Diagram 7.5. Here the value of D to be stored arrives a fraction later than the clock pulse, and in the case where the previous value of Q is the same as the new value, a spike (momentary wrong state) is caused which may make the circuit malfunction. In order to avoid this happening we need to build a circuit that will change its output only at one instance of time. Clearly, in Diagram 7.5, if we could arrange the circuit so that the output Q is set only when the clock C changes from one to zero, then we would have a reliable circuit. Such devices are termed edge triggered.

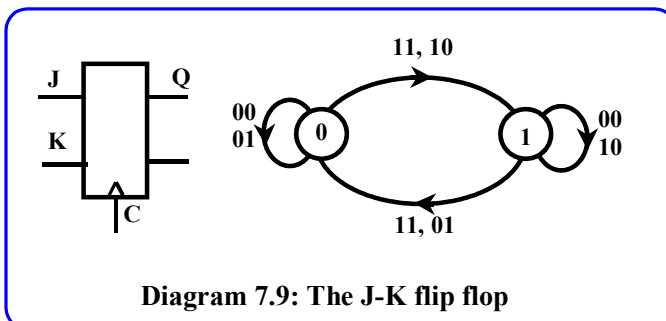
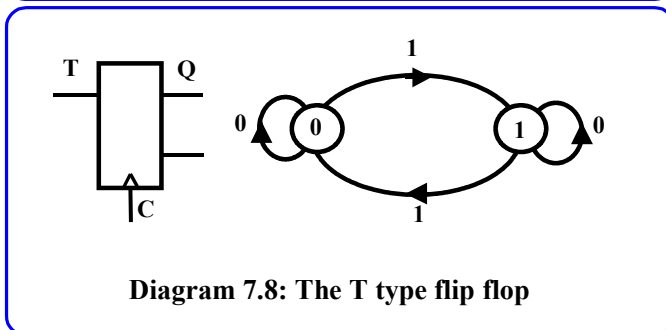
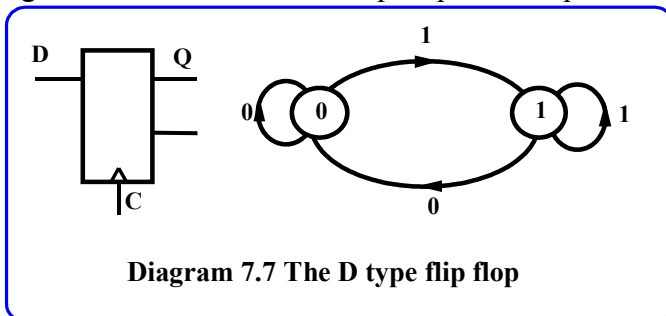


The first type of edge triggered flip flop we will look at is called the master-slave. It is built from two gated R-S flip flops as shown in Diagram 7.6. The trick is that the gates of the two flip flops can never be open at the same time. Thus, if the clock is at 1, the output of the first flip flop, Q1 follows the input D1, but the gate of the second stage is closed, and the output Q2 cannot change state.



When the clock goes from 1 to 0, the first flip flop is now blocked and the value of Q1 is held at the value of D when the transition occurred, but the gate to the second R-S flip flop is now open, and the state of Q1 is transmitted to Q2. It is easy to see that this output state cannot change until the clock goes first to 1 and then to zero again. It should be noted that there still remains the possibility of a spike being created with this circuit if the gate of the second stage opens momentarily before the gate of the first stage closes. However, the implementation in Diagram 7.6 is quite safe since the inverter ensures that the signal will change on C1 before it changes on C2. The master slave arrangement is convenient to use and easy to understand, however, a neater arrangement of the edge triggered D type flip flop can be made, as we will see later.

The edge triggering is important in that it allows us to build circuits where state transitions occur at a set time. This is important in that it allows us to build synchronous circuits, i.e. those where we ensure that all signals change at the same time. It is useful to think of synchronous sequential circuits as having a finite state machine representation. For example, if we take the edge triggered D type flip flop, there are two states which are defined by the value of the output Q. The finite state machine is shown in Diagram 7.7. Here the states are shown in the circle, and the possible inputs (D) are shown on the arcs. Transitions can only occur when a negative edge appears on the clock signal, but this is not shown in the finite state machine model. In practical circuits, the same clock signal will be fed to all the flip flop clock inputs to ensure synchronous behaviour.



Flip flops can be made with other characteristics, and one important one is illustrated by the T-type of toggle flip flop, shown in Diagram 7.8. Here the circuit is best considered to work in two modes. If $T=1$ then when the clock changes from 1 to zero, the output will change its state, but if T is set to 0 the output will not change. This may be represented functionally by the equation:

$$Q = T \cdot Qp' + T' \cdot Qp$$

The most flexible flip flop is the JK device shown in Diagram 7.9. This can be set up to have the characteristics of either a SR or a toggle flip flop. The four modes of operation are defined by the values of J and K as follows as shown in the diagram. The transitions are only made when a negative edge appears on the clock. By connecting J and K together and using them as one input we create a T type flip flop. By inverting J and connecting it to K we create a D type flip flop (with the J terminal equivalent to D) and in cases where we want a simple R-S flip flop we use J as S and K as R.

Diagram 7.10 shows a very ingenious design for the edge triggered flip flop. This circuit actually latches when the clock changes from 0 to 1, rather than 1 to 0 in our previous circuits. Both types of edge triggering are readily available in practical circuits. Note that when the clock is 0 in Diagram 7.10, the input to the final R-S flip flop is forced to be {1 1} and the output will remain unchanged. To understand the circuit further it is necessary to trace the effect of different inputs around the circuit.

To finish up we will look at one further feature required in flip flops, namely the ability to preset a particular output, regardless of the clock. Such inputs are referred to as CLEAR which sets the output to zero, and PRESET which sets it to 1. Such inputs are used to set a circuit into a known state, for example when it is switched on or perhaps when a reset button is pressed. A simple implementation of the Edge triggered D type flip flop with preset and clear is given in Diagram 7.10. Notice that if the PRESET and CLEAR inputs are 1, the circuit behaves normally. If one of them goes to zero the output will be forced to the corresponding state. Notice also that if both are set to zero the output and its complement are both forced to 1. This is of course an unwanted effect,

and should be eliminated by the external circuitry. You should satisfy yourself that with PRESET and CLEAR held at 1 the circuit behaves as a D type.

