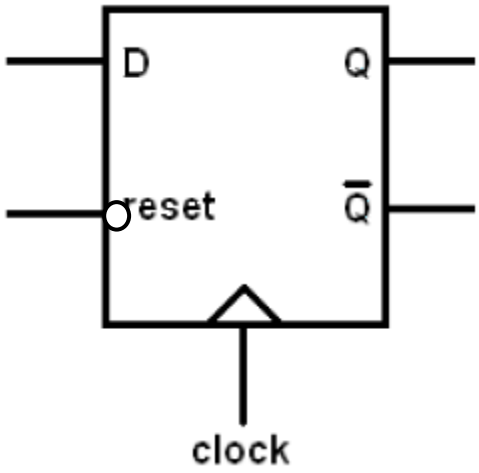
MAXVY
Technologies Pvt Ltd

SYNCHRONOUS CIRCUITS	ASYNCHRONOUS CIRCUITS
1) These are the sequential circuits which are governed by a global clock signal.	1) These are circuits which are governed by a input signals regardless of clock signal.
2) The state of all elements of a synchronous circuit changes only by an application of a distributed clock signal.	2) Asynchronous circuits change state through the inputs received by them.
Pros: 1) The state of a synchronous circuit predictable.	**Cons:** 1) The state of asynchronous circuits are unpredictable.
2) synchronous clock signals are less susceptible to noise, there is no chance for race conditions.	2) asynchronous circuits can transition into a wrong state due to incorrect arrival time of 2 inputs, race condition will occur.
3) clock impulses are given are same for all the flip flops so it is very easy to design any circuit.	3) More complex to design

4)Easy to trouble shoot since predictable	4)Not easy to troubleshoot
5) Circuit is more reliable and portable.	5)Circuit is not reliable
Cons: 1)The time period of a clock signal should be long enough to accommodate longest propagation delay	**Pros**: 1) They are limited by propagation delay of logic gates only.
2)Practically all the circuits today are synchronous circuits, except the part where speed of the circuit operation is crucial.	2)These are used primarily in high speed systems such as Signal Processing hardware.

Example:



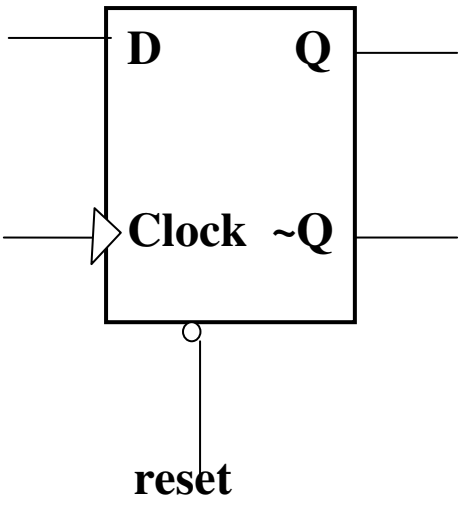
Synchronous D flip flop

lop is only operated with respect to clock so its output will change with respect clock.

Verilog code:

```
module dflip_flop(d,q,clk,reset);
input d,clk,reset;
output reg q;
always@(posedge clk)
begin
if (!reset)
   q<=0;
else
```

```
   q<=d;
end
endmodule
```



Asynchronous D flip flop

In above diagram the d flip flop is operated with respect to both clock and reset condition so the output will change whenever the reset condition changes it will not wait for clock to change

Verilog code:

```
module dfli_flop(d,q,clk,reset);

input d,clk,reset;

output reg q;

always@(posedge clk or negedge reset)

begin

if (!reset)

   q<=0;

else

   q<=d;

end

endmodule
```