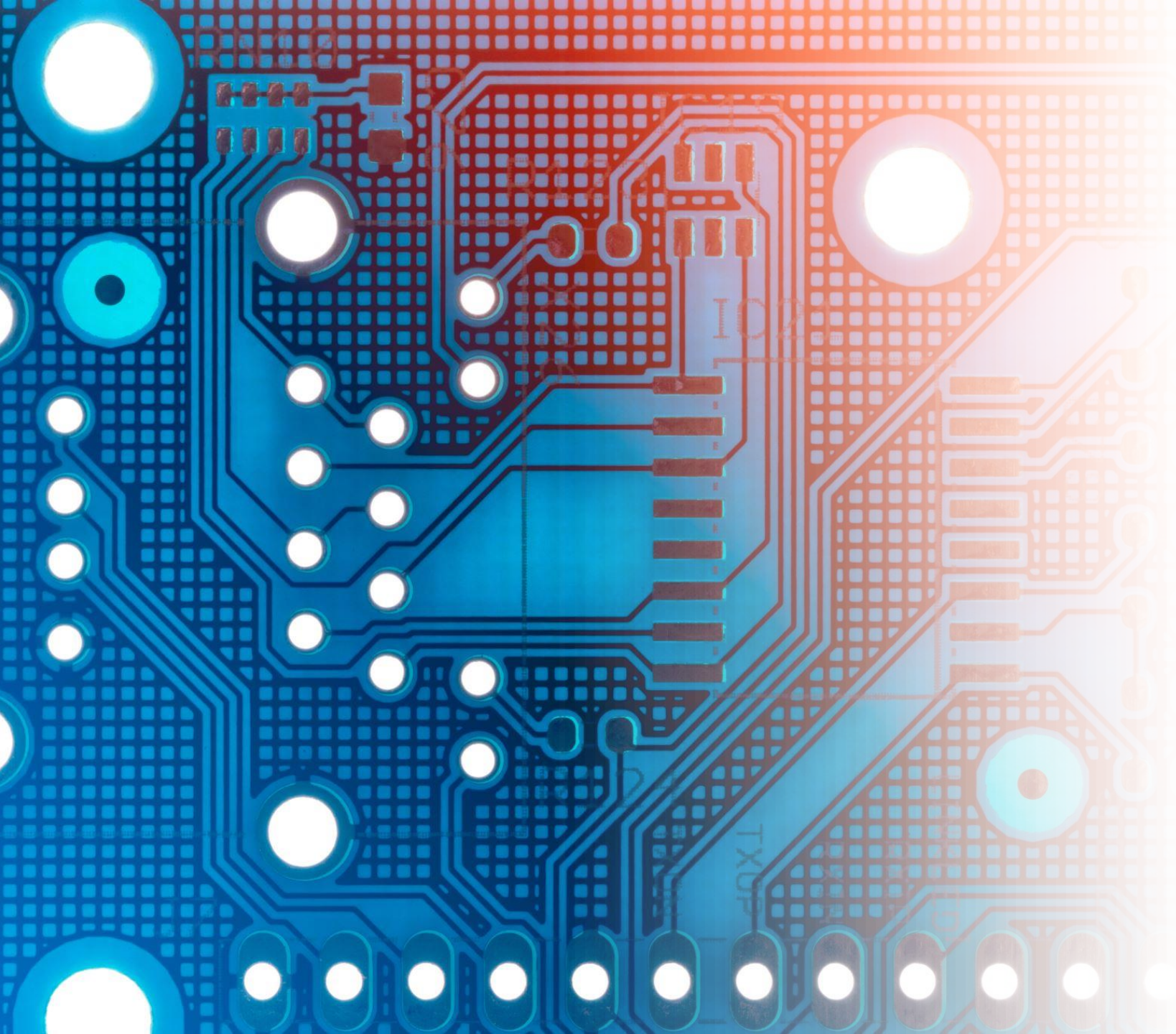
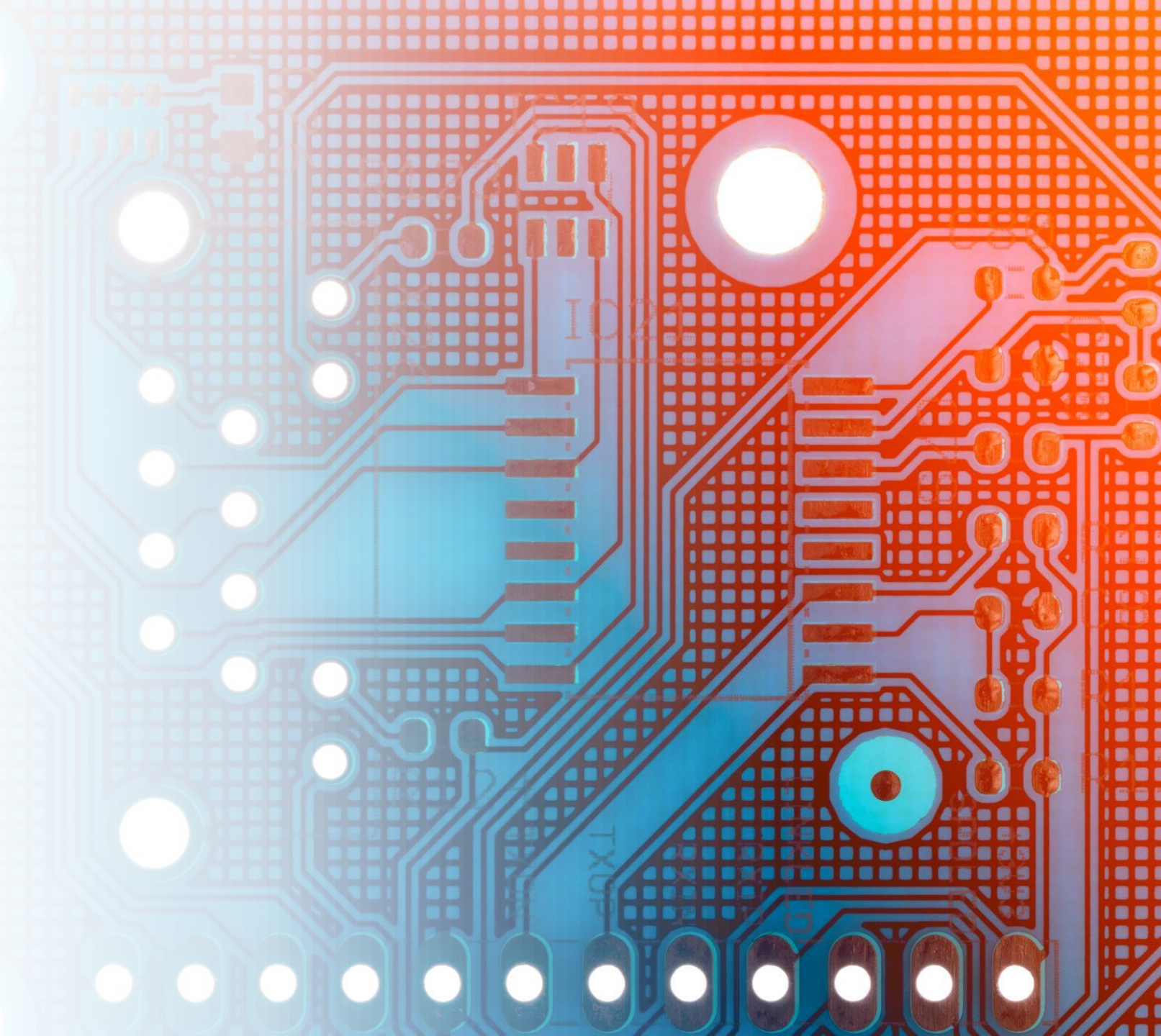


Sequential circuit design using VHDL

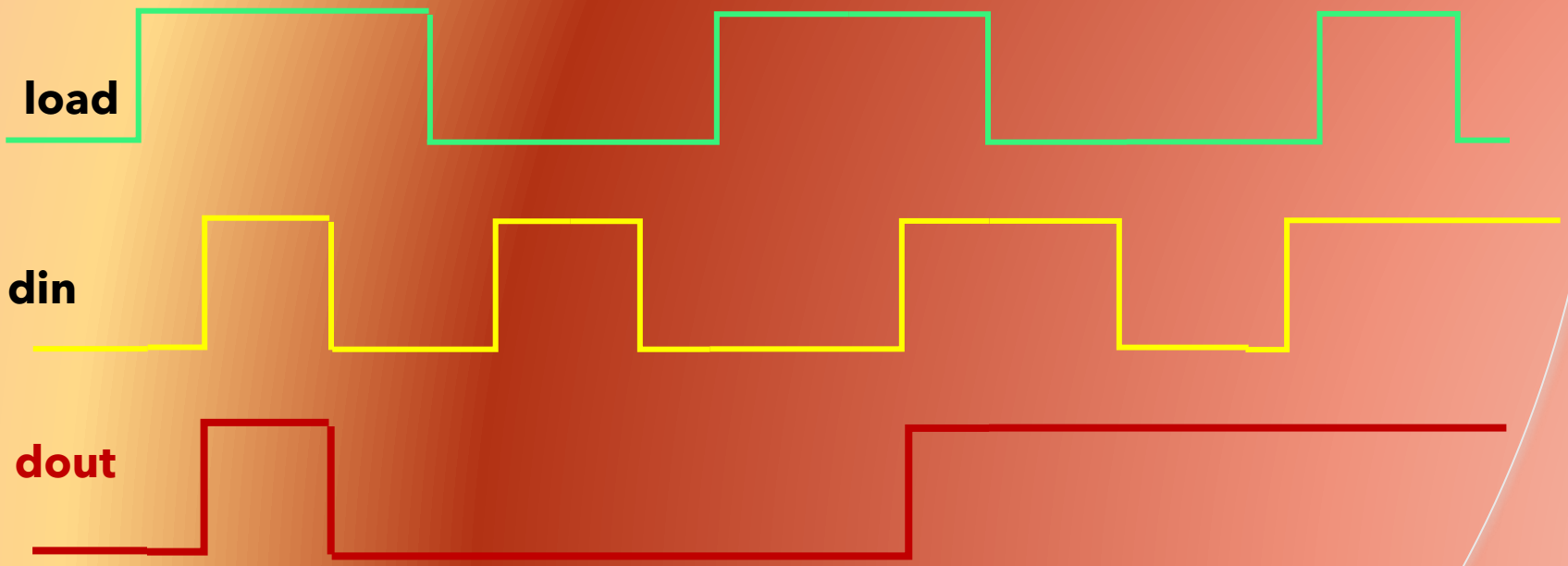
Dr. Fatma ElFouly

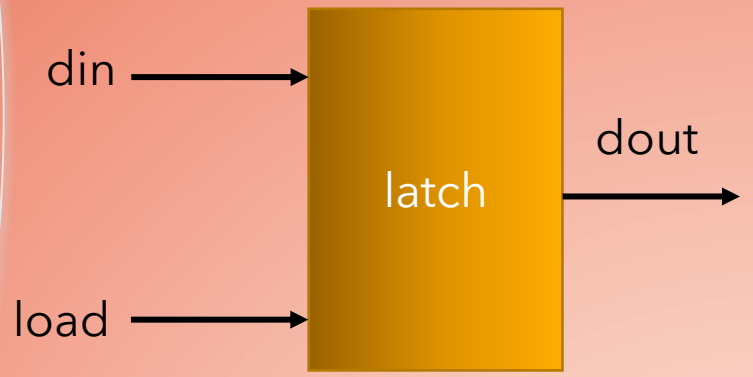
Latch



Latch

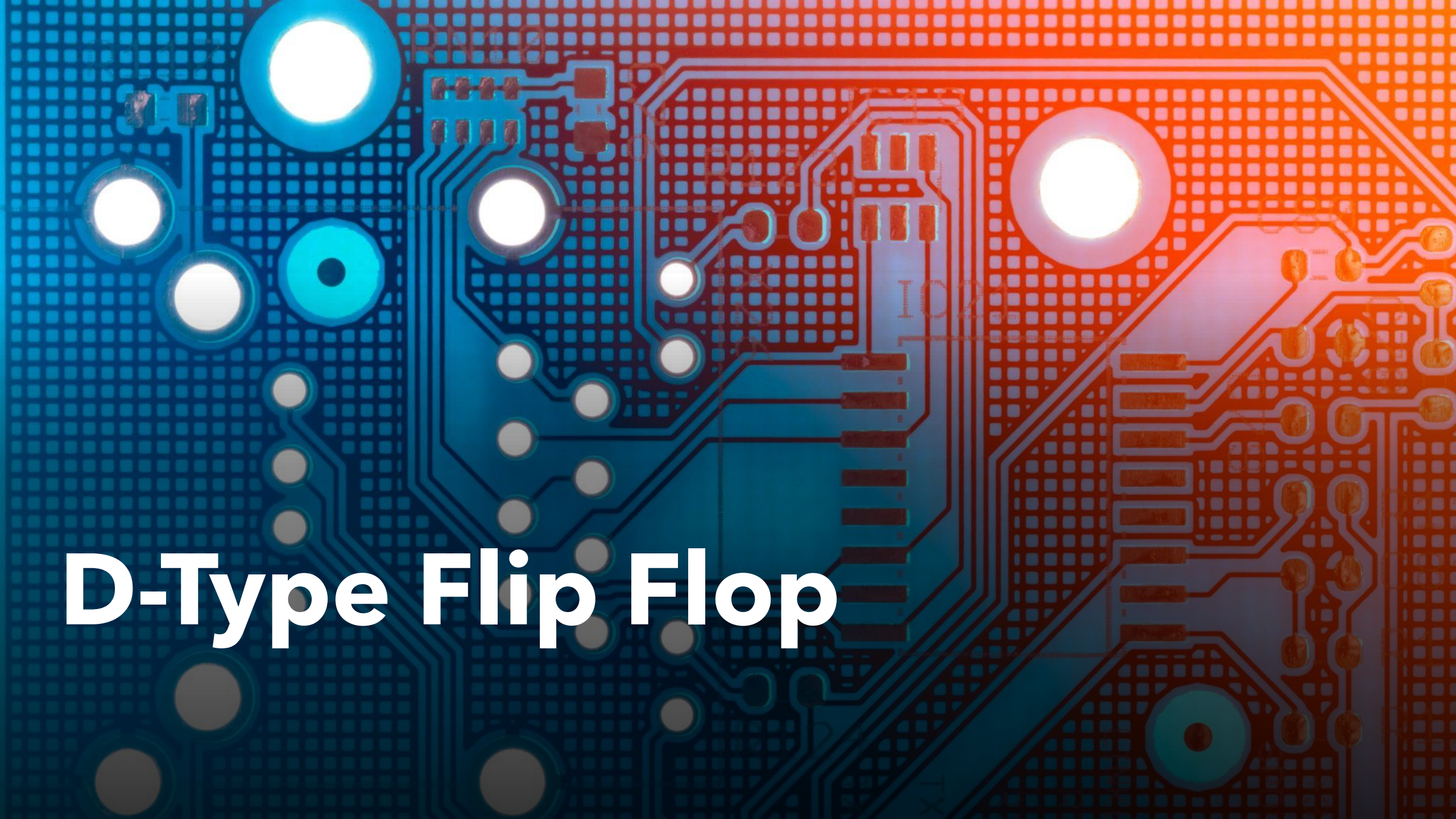
A latch is a level sensitive memory cell that is transparent to signals passing from the D input to Q output when enabled and holds the value of D on Q at the time when it becomes disabled.





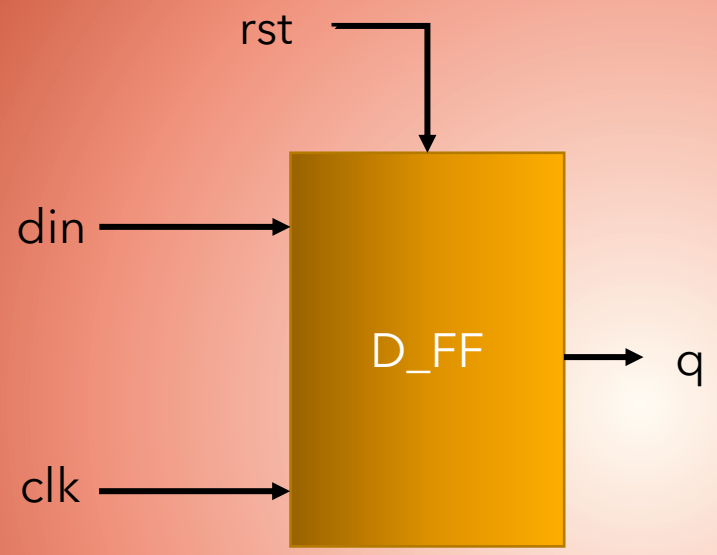
latch

- library ieee;
- use ieee.std_logic_1164.all;
- use ieee.std_logic_arith.all;
- entity d_latch is
- port (din, load: in std_logic;
- load: out std_logic);
- end entity d_latch;
- architecture rtl of d_latch is
- begin
- process (din, load)
- begin
- if (load = ‘1’) then
- dout <= din;
- end if;
- end process;
- end architecture rtl;



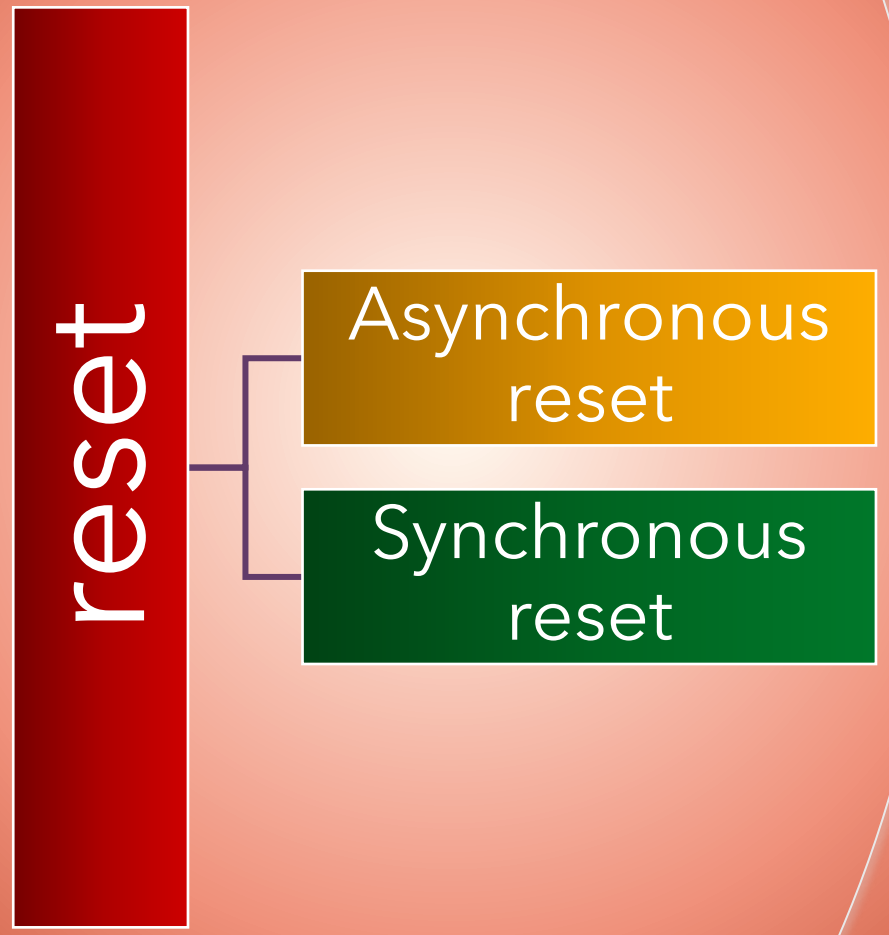
D-Type Flip Flop

D-FF

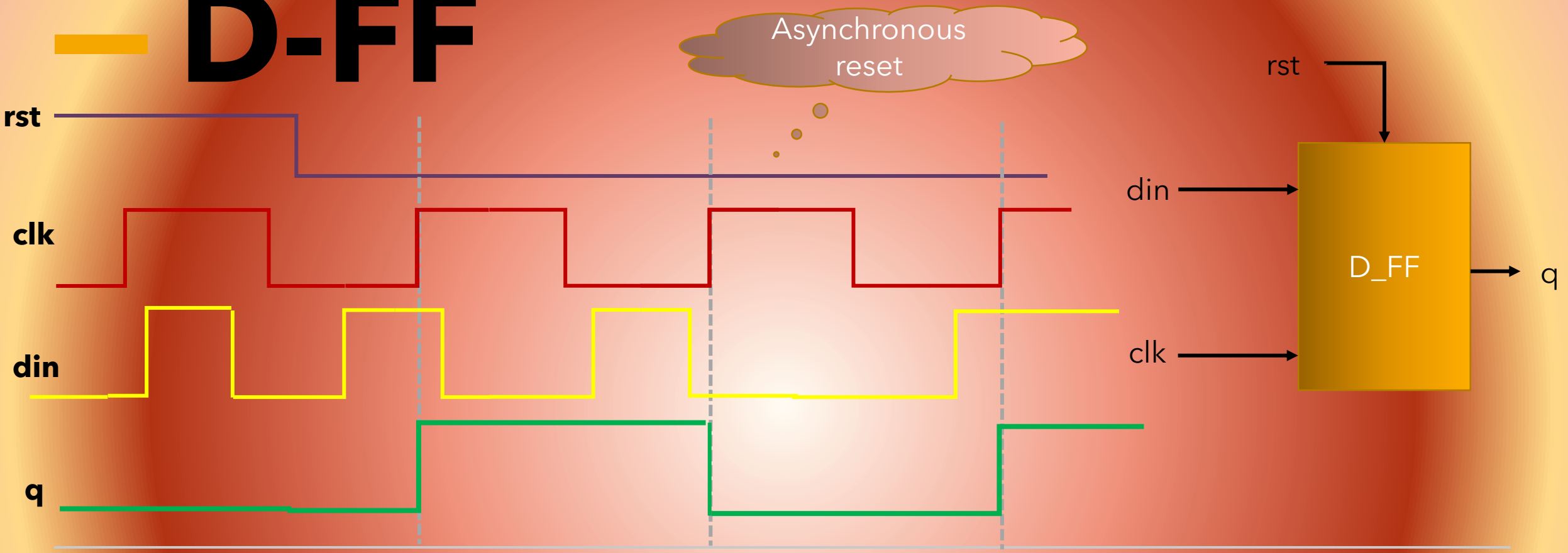


The D-type flip-flop is an edge triggered memory device that transfers a signal's value on its D input, to its Q output, when an active edge transition occurs on its clock input. The output value is held until the next active clock edge.

Reset



D-FF

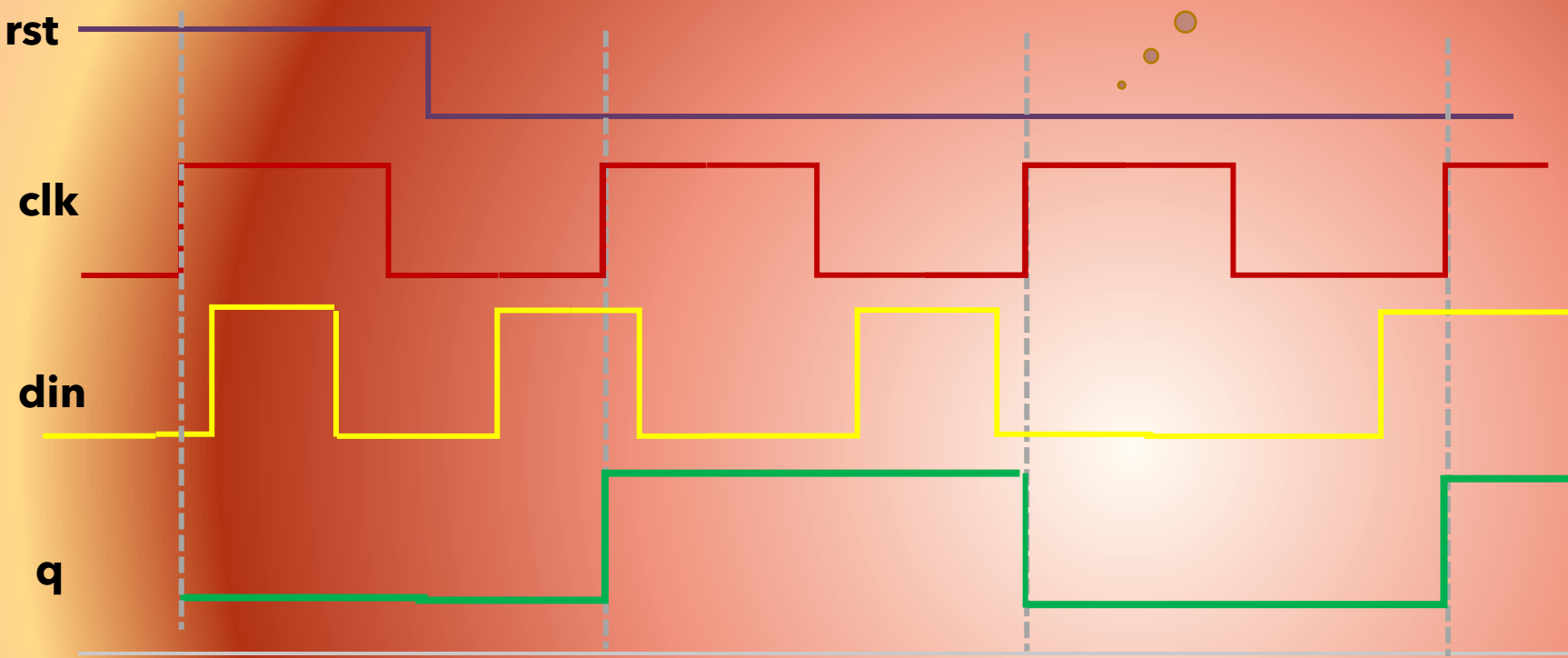


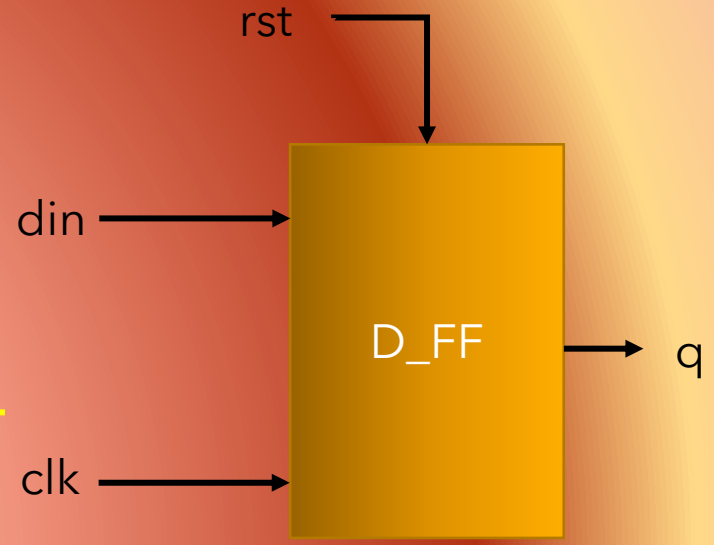
```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
entity d_flip_flop is
port (din, clk, rst: in std_logic;
q: out std_logic);
end entity d_flip_flop;
architecture first of d_flip_flop is
begin
process (rst, clk)
begin
if (rst = '1') then
q <= '0';
elsif (rising_edge(clk)) then
q <= din;
end if;
end process;
end architecture first;
```

D-FF

D-FF

Synchronous reset



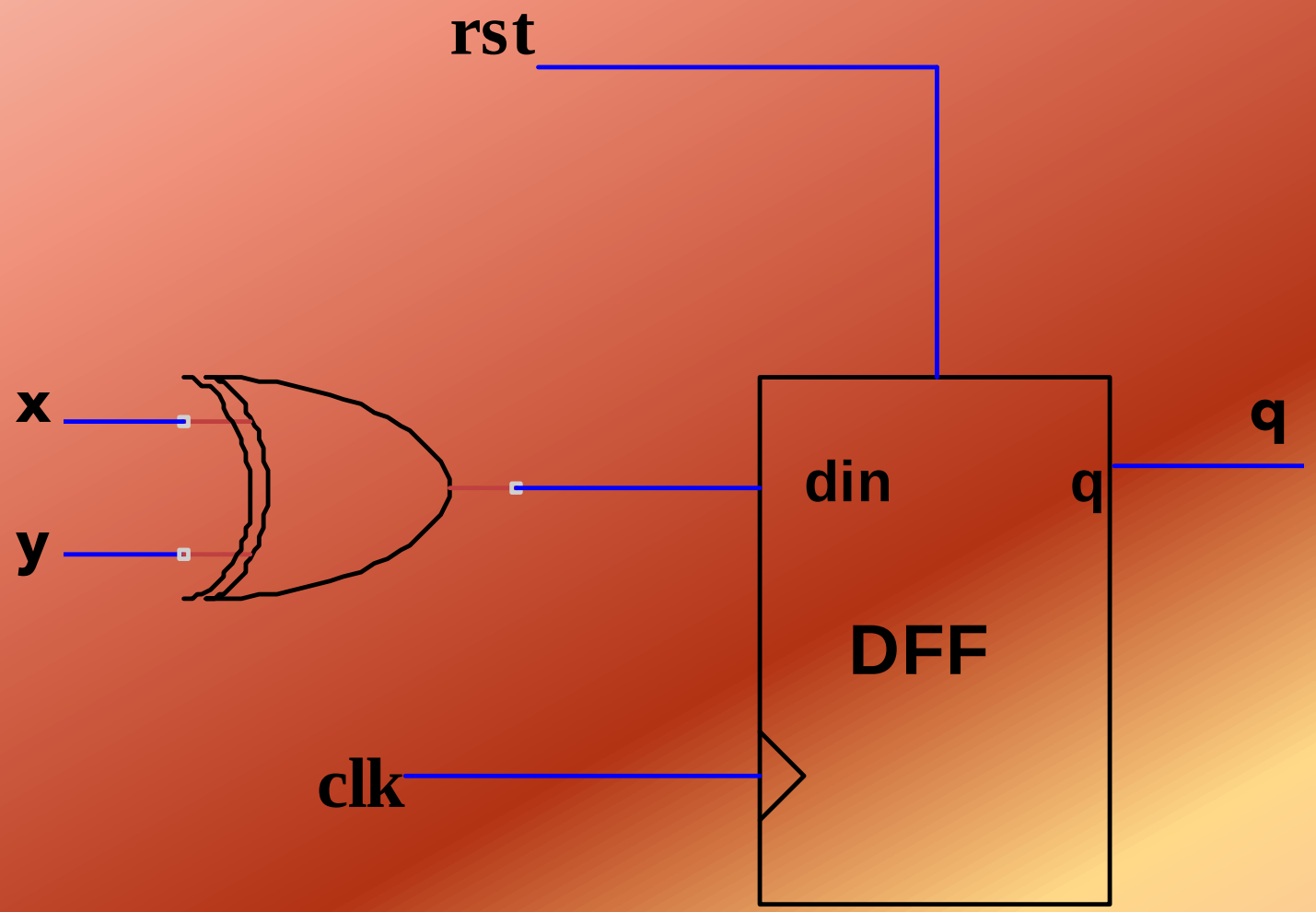


```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
entity d_flip_flop is
port (din, clk, rst: in std_logic;
q: out std_logic);
end entity d_flip_flop;
process (clk)
begin
if (rising_edge(clk)) then
if (rst = '1') then
q <= '0';
else
q <= din;
end if;
end if;
end process;
end architecture second;
```

D-FF

Solved Problem

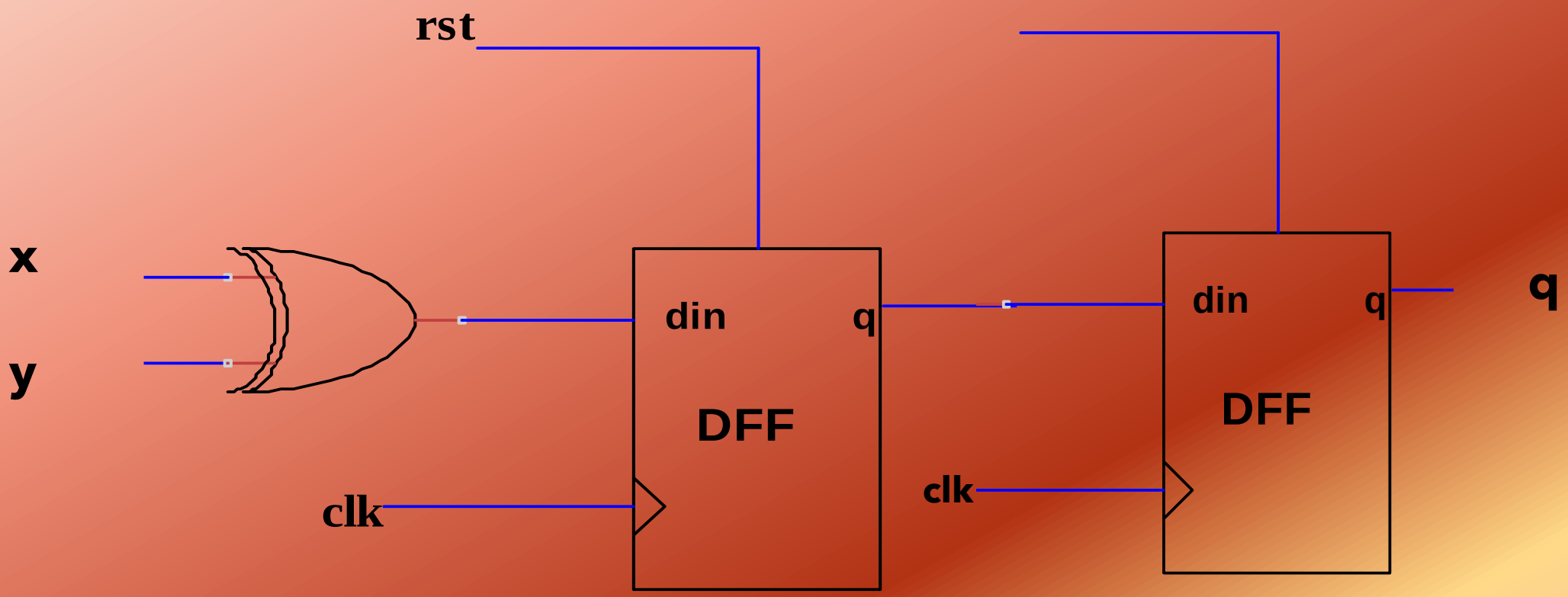
Variable



Variable

```
entity circuit_1 is
port (x,y,clk,rst: in std_logic;
q: out std_logic);
end entity circuit_1;
architecture rtl of circuit_1 is
Signal din :std_logic;
begin
process (clk,rst)
begin
if (rst ='1')then
q <= '0';
elsif (rising_edge(clk)) then
din <= x XOR y;
q <= din;
End if;
End process;
End rtl;
```

Variable



Variable

```
architecture rtl of circuit_1 is
begin
process (clk,rst)
variable din :std_logic;
begin
if (rst ='1')then
q <= '0';
elsif (rising_edge(clk)) then
din := x XOR y;
q <= din;
End if;
End process;
End rtl;
```