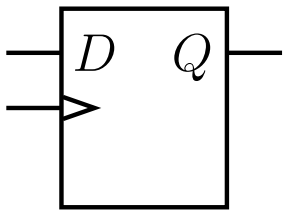
Digital Systems - 3

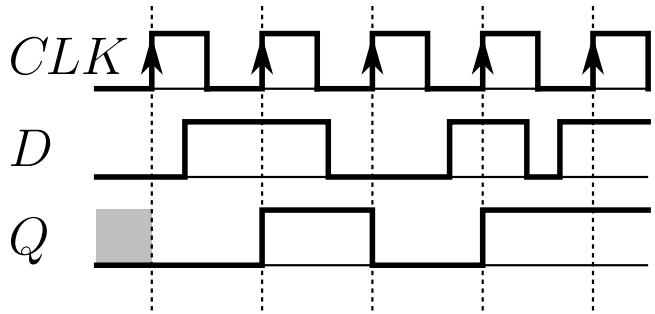
Pere Palà - Alexis López

iTIC http://itic.cat

February 2016

D Flip-Flop






```
signal d, q   : std_logic;
...
my_dff: process (clk) is
begin
  if rising_edge(clk) then
    q <= d;
  end if;
end process my_dff;
```

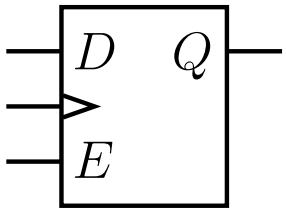
Process statement

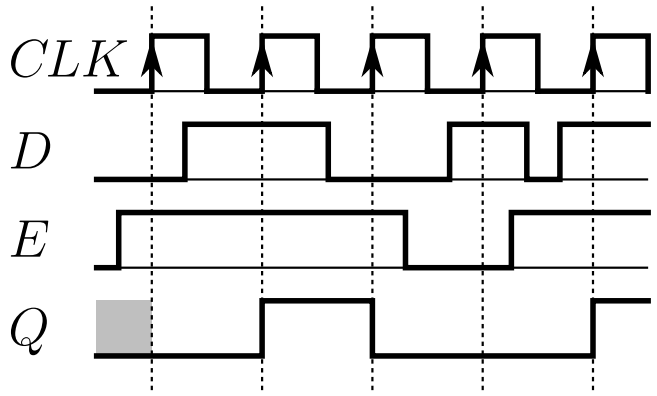
```
name1: process (a,b,c) is
begin
  sequential statement 1a;
  sequential statement 2a;
  ...
end process name1;

name2: process (c,d,e) is
begin
  sequential statement 1b;
  sequential statement 2b;
  ...
end process name1;
```

- Process `name1` is (only!) triggered when a, b or c changes. This is the sensitivity list.
- Inside a process, the statements are executed in order.
- Process `name1` is executed concurrently with process `name2`

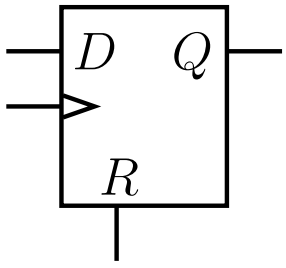
D Flip-Flop on a bus with clock enable

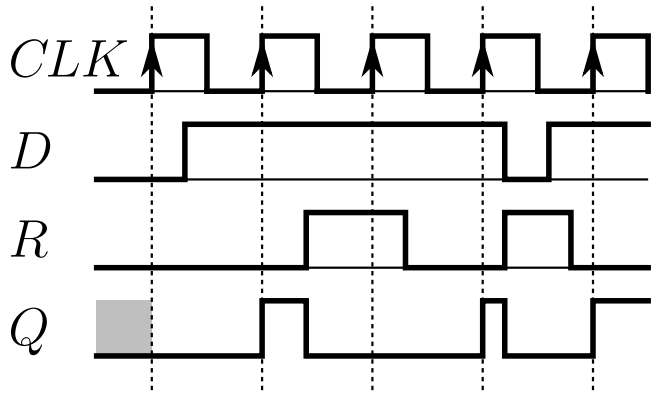





```
signal d, q    : std_logic_vector (7 downto 0);
signal ce      : std_logic;
...
my_dff: process (clk) is
begin
  if rising_edge(clk) then
    if ce = '1' then
      q <= d;
    end if;
  end if;
end process my_dff;
```

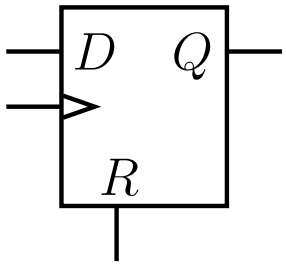
D Flip-Flop with Asynchronous Reset

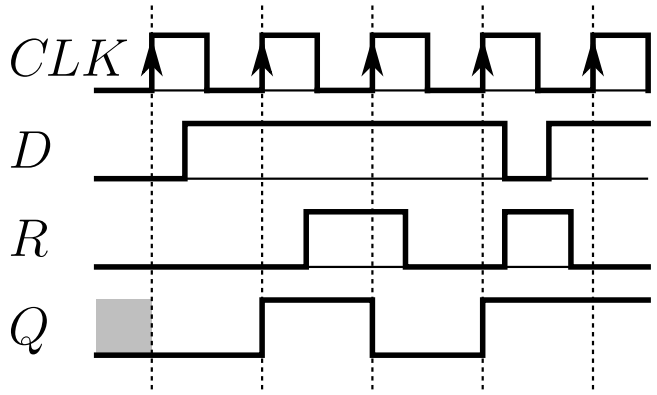





```
signal d, q    : std_logic_vector(7 downto 0);
signal reset   : std_logic;
...
my_dff: process (clk,reset) is
begin
  if reset = '1' then  --reset overrides everything !
    q <= "00000000";
  elsif rising_edge(clk) then
    q <= d;
  end if;
end process my_dff;
```

D Flip-Flop with Synchronous Reset






```
signal d, q    : std_logic_vector (7 downto 0);
signal reset   : std_logic;
...
my_dff: process (clk) is
begin
  if rising_edge(clk) then
    if reset = '1' then
      q <= (others => '0');
    else
      q <= d;
    end if;
  end if;
end process my_dff;
```