



Learn Git and GitHub without any code!

Using the Hello World guide, you'll start a branch, write comments, and open a pull request.

[Read the guide](#)

[FranciscaCampos](#) / [Digital-electronics-1](#)

[Code](#)[Issues](#)[Pull requests](#)[Actions](#)[Projects](#)[Wiki](#)[Security](#)[Insights](#)[Settings](#)

main ▾



[Digital-electronics-1](#) / [Labs](#) / [07-ffs](#) / [README.md](#)



FranciscaCampos Update README.md



1 contributor

[Raw](#)[Blame](#)

357 lines (269 sloc) | 8.94 KB

Lab 7: Latches and Flip-flops

1. Characteristic equations and completed tables for D, JK, T flip-flops.

| clk | d | q(n) | q(n+1) | comments |
|-----|---|------|--------|-----------|
| ^ | 0 | 0 | 0 | No change |

| clk | d | q(n) | q(n+1) | comments |
|-----|---|------|--------|-----------|
| ^ | 0 | 1 | 0 | Change |
| ^ | 1 | 0 | 1 | Change |
| ^ | 1 | 1 | 1 | No change |

| clk | j | k | q(n) | q(n+1) | comments |
|-----|---|---|------|--------|-----------|
| ^ | 0 | 0 | 0 | 0 | No change |
| ^ | 0 | 0 | 1 | 1 | No change |
| ^ | 0 | 1 | 0 | 0 | No change |
| ^ | 0 | 1 | 1 | 0 | Change |
| ^ | 1 | 0 | 0 | 1 | Change |
| ^ | 1 | 0 | 1 | 1 | No change |
| ^ | 1 | 1 | 0 | 1 | Change |
| ^ | 1 | 1 | 1 | 0 | Change |

| clk | t | q(n) | q(n+1) | comments |
|-----|---|------|--------|-----------|
| ^ | 0 | 0 | 0 | No change |
| ^ | 0 | 1 | 1 | No change |
| ^ | 1 | 0 | 1 | Change |
| ^ | 1 | 1 | 0 | Change |

2. D latch.

```

p_d_latch : process ( en , d, anrst )
begin

    if(anrst = '0') then
        q <= '0';
    elsif( en = '1' ) then
        q <= d ;
        q_bar <= not d ;
    end if ;
end process p_d_latch ;

```

```
p_stimulus : process
begin

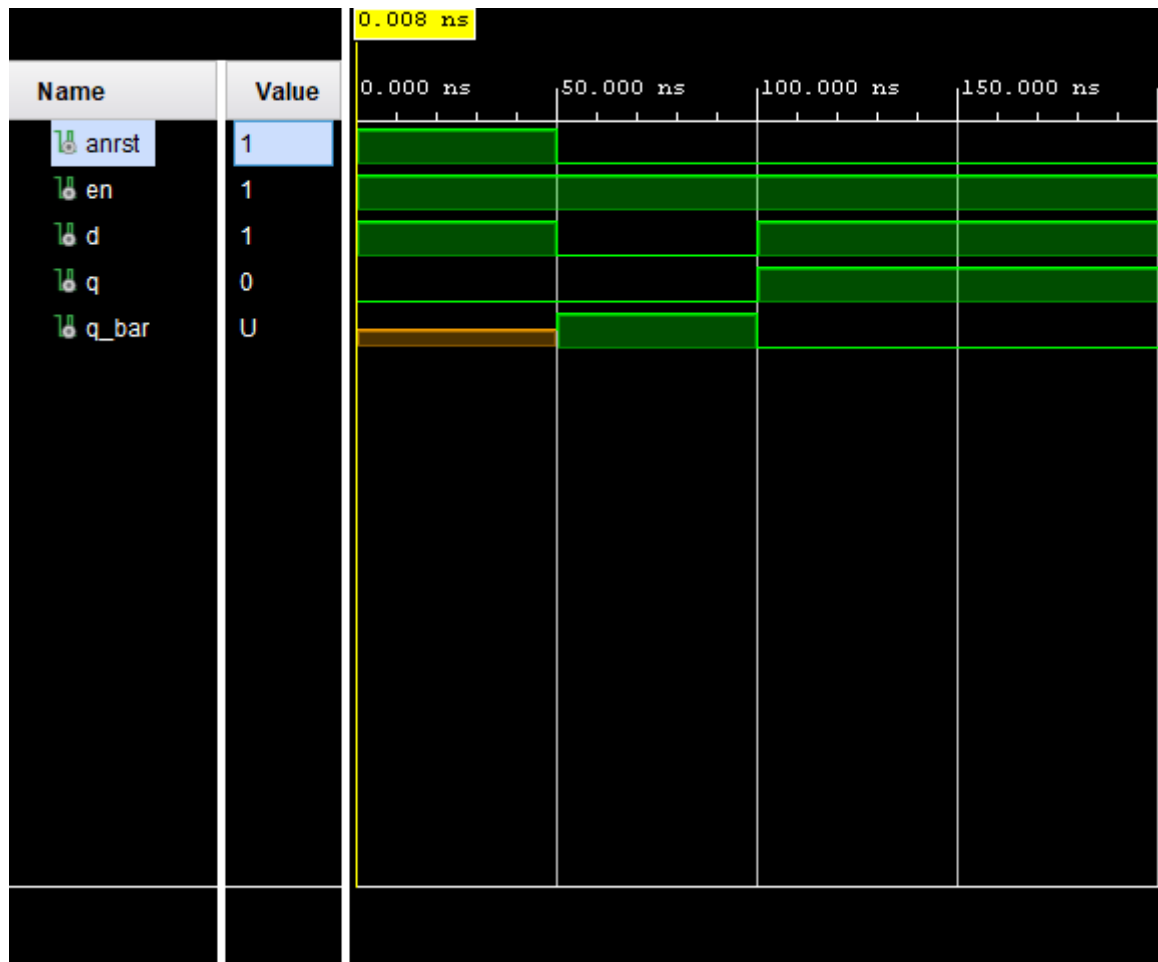
    report "Stimulus process started" severity note;

    anrst <= '0'; d <= '1'; en <= '1'; wait for 50 ns;
    assert (q = '0')
    report "Asynchronous reset is failing" severity error;

    anrst <= '1';
    en <= '1';
    d <= '0';
    wait for 50 ns;
    assert (q = '0')
    report "The enable does not work properly" severity error;

    anrst <= '1';
    en <= '1';
    d <= '1';
    wait for 50 ns;
    assert (q = '1')
    report "The output does not fit with the expected" severity error;

    report "Stimulus process finished" severity note;
    wait;
end process p_stimulus;
```



3. Flip-flops

```
d_ff_arst : process ( clk , d, arst )
begin

    if(arst = '1') then
        q <= '0';
    elsif rising_edge(clk) then
        q <= d ;
        q_bar <= not d ;
    end if ;
end process d_ff_arst ;
```

```
-----
-- Clock generation process
-----
```

```
p_clk_gen : process
begin
    while now < 750 ns loop          -- 75 periods of 100MHz clock
        clk <= '0';
        wait for c_CLK_100MHZ_PERIOD / 2;
        clk <= '1';
        wait for c_CLK_100MHZ_PERIOD / 2;
    end loop;
    wait;
```

```
end process p_clk_gen;

p_stimulus : process
begin

    report "Stimulus process started" severity note;

    arst <= '1'; d <= '1'; wait for 50 ns;
    assert (q = '0')
    report "Asynchronous reset is failing" severity error;

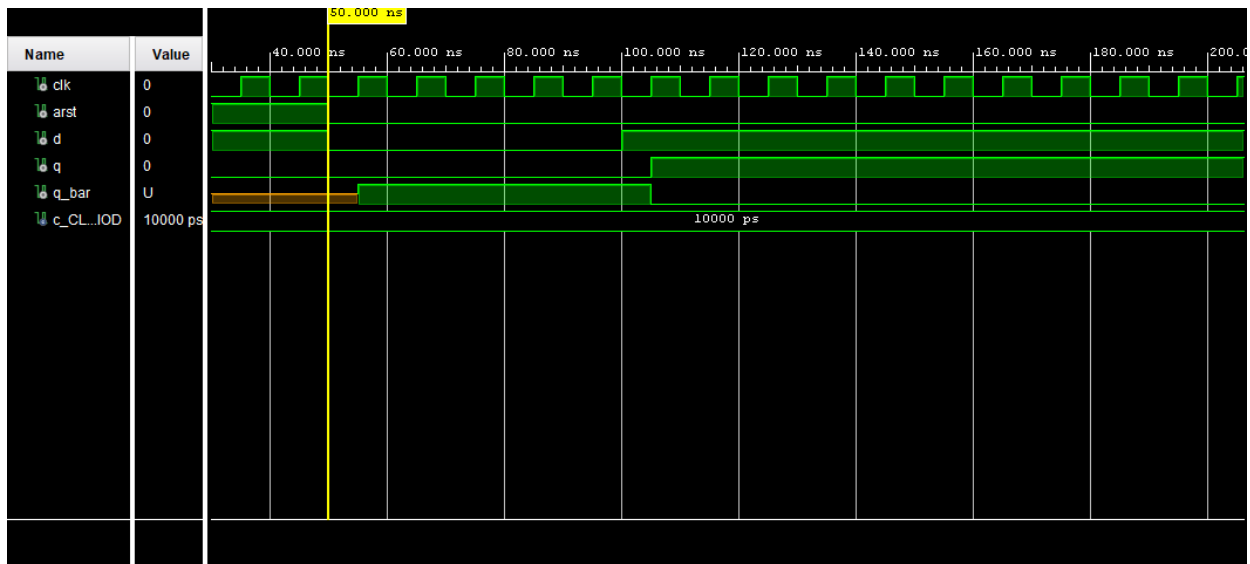
    arst <= '0';
    d <= '0';
    wait for 50 ns;
    assert (q = '0')
    report "The enable does not work properly" severity error;

    d <= '1';
    wait for 50 ns;
    assert (q = '1')
    report "The output does not fit with the expected" severity error;

    report "Stimulus process finished" severity note;
    wait;
end process p_stimulus;

p_d_ff : process ( clk , d, rst )
begin

    if rising_edge(clk) then
        if(rst = '1') then
            q <= '0';
        else
            q <= d ;
            q_bar <= not d ;
        end if;
    end if ;
end process p_d_ff ;
```



```

p_clk_gen : process
begin
    while now < 750 ns loop          -- 75 periods of 100MHz clock
        clk <= '0';
        wait for c_CLK_100MHZ_PERIOD / 2;
        clk <= '1';
        wait for c_CLK_100MHZ_PERIOD / 2;
    end loop;
    wait;
end process p_clk_gen;

p_stimulus : process
begin

    report "Stimulus process started" severity note;

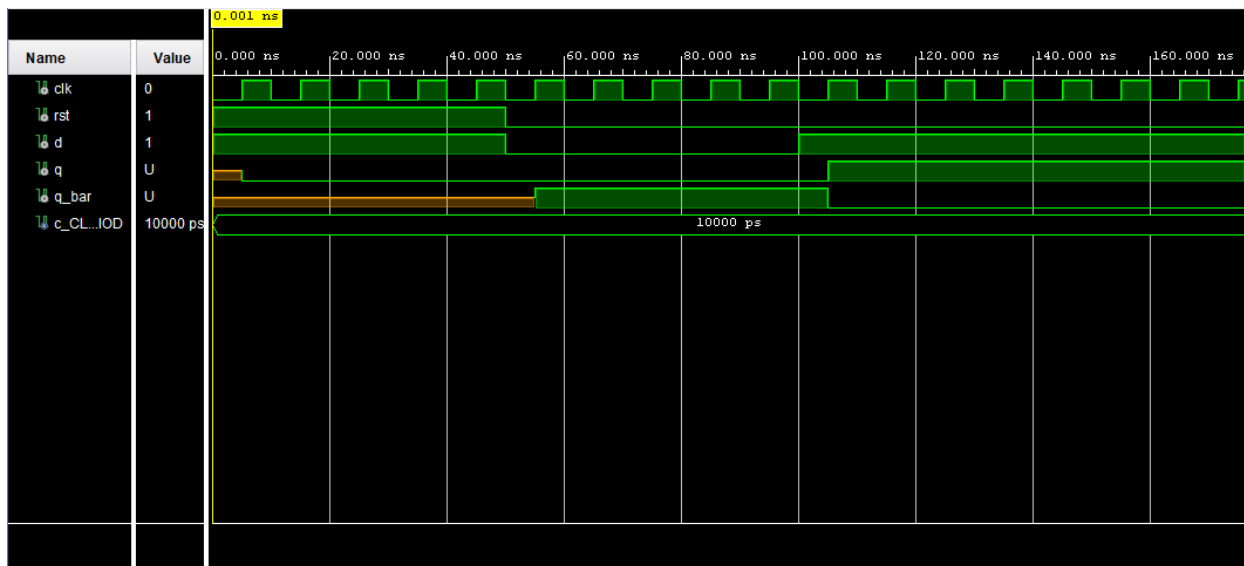
    rst <= '1'; d <= '1'; wait for 50 ns;
    assert (q = '0')
    report "Asynchronous reset is failing" severity error;

    rst <= '0';
    d <= '0';
    wait for 50 ns;
    assert (q = '0')
    report "The enable does not work properly" severity error;

    d <= '1';
    wait for 50 ns;
    assert (q = '1')
    report "The output does not fit with the expected" severity error;

    report "Stimulus process finished" severity note;
    wait;
end process p_stimulus;

```



```

p_jk_ff_rst : process ( clk ,rst )
begin

    if rising_edge(clk) then
        if (rst = '1') then
            q_i <= '0';
        elsif (j = '0' and k = '0') then
            q_i <= q_i ;
        elsif (j = '0' and k = '1') then
            q_i <= '0';
        elsif (j = '1' and k = '0') then
            q_i <= '1';
        elsif (j = '1' and k = '1') then
            q_i <= not q_i;
        end if;
    end if ;
end process p_jk_ff_rst ;

q <= q_i;
q_bar <= not q_i;

```

```

-----
-- Clock generation process
-----

```

```

p_clk_gen : process
begin
    while now < 750 ns loop          -- 75 periods of 100MHz clock
        clk <= '0';
        wait for c_CLK_100MHZ_PERIOD / 2;
        clk <= '1';
        wait for c_CLK_100MHZ_PERIOD / 2;
    end loop;
    wait;
end process p_clk_gen;

```

```

p_stimulus : process
begin

    report "Stimulus process started" severity note;

    rst <= '1'; j <= '0'; k <= '0'; wait for 10 ns;

    rst <= '0'; wait for 10 ns;

    j <= '0'; k <= '1'; wait for 10 ns;
    j <= '1'; k <= '0'; wait for 10 ns;
    j <= '1'; k <= '1'; wait for 10 ns;

    j <= '0'; k <= '0'; wait for 10 ns;
    j <= '0'; k <= '1'; wait for 10 ns;
    j <= '1'; k <= '0'; wait for 10 ns;
    j <= '1'; k <= '1'; wait for 10 ns;

    report "Stimulus process finished" severity note;
    wait;
end process p_stimulus;
```

```

![[JK]](<https://github.com/FranciscaCampos/Digital-electronics-1/blob/main/Labs/07->

```

```vhdl
library ieee ;
use ieee.std_logic_1164.all;
-----
entity t_ff_rst is
port( clk : in std_logic ;
      rst : in std_logic;
      t  : in std_logic ;
      q  : buffer std_logic ;
      q_bar : buffer std_logic ) ;
end entity t_ff_rst ;
-----
architecture Behavioral of t_ff_rst is
signal q_i : std_logic;
begin
    p_t_ff : process ( clk , rst )
    begin

        if rising_edge(clk) then
            if(rst = '1') then
                q <= '0';
            elsif t = '1' then
                q <= not q ;
                q_bar <= not q ;
            end if ;
        end if;
    end process p_t_ff;
end architecture Behavioral;
```

```



```
end process p_t_ff ;
end architecture Behavioral ;

-- Clock generation process

p_clk_gen : process
begin
 while now < 750 ns loop -- 75 periods of 100MHz clock
 clk <= '0';
 wait for c_CLK_100MHZ_PERIOD / 2;
 clk <= '1';
 wait for c_CLK_100MHZ_PERIOD / 2;
 end loop;
 wait;
end process p_clk_gen;

p_stimulus : process
begin

 report "Stimulus process started" severity note;

 rst <= '1'; t <= '1'; wait for 50 ns;
 assert (q = '0')
 report "Asynchronous reset is failing" severity error;

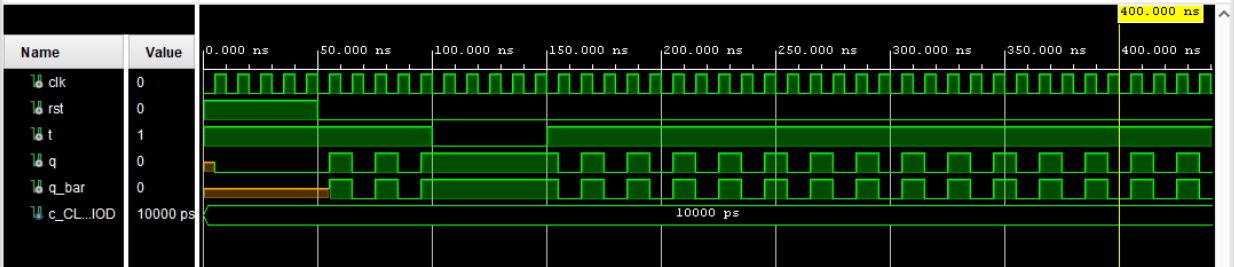
 rst <= '0';

 t <= '1';
 wait for 50 ns;
 assert (q = not q)
 report "The output does not fit with the epected" severity error;

 t <= '0';
 wait for 50 ns;
 assert (q = not q)
 report "The output does not fit with the epected" severity error;

 t <= '1';
 wait for 50 ns;
 assert (q = not q)
 report "The output does not fit with the epected" severity error;

 report "Stimulus process finished" severity note;
 wait;
end process p_stimulus;
```



# 4. Shift Register

