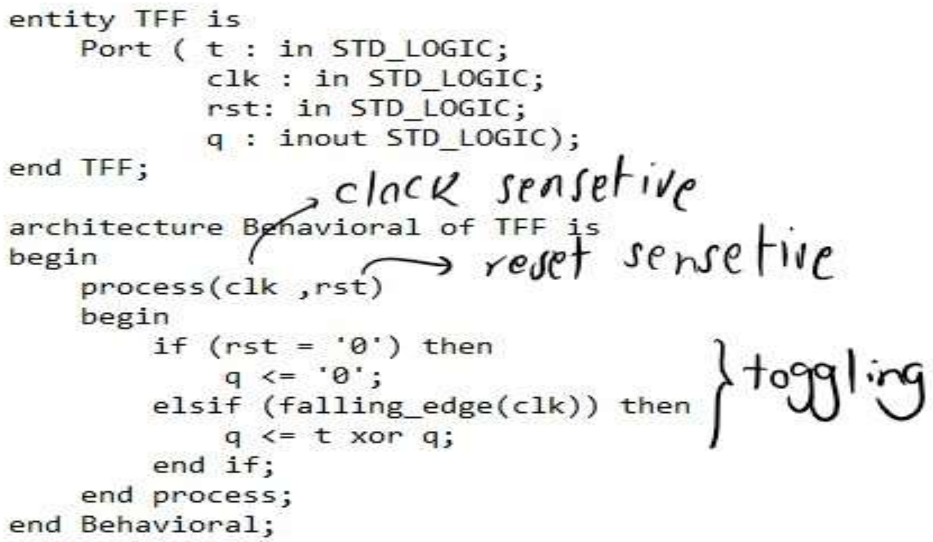
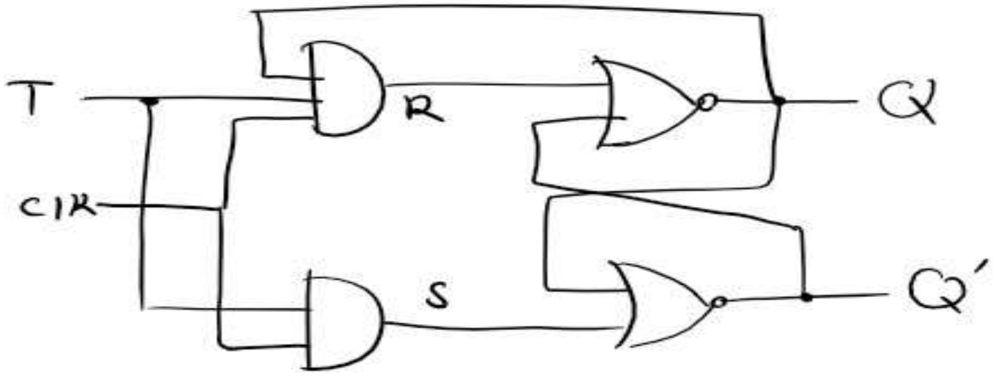
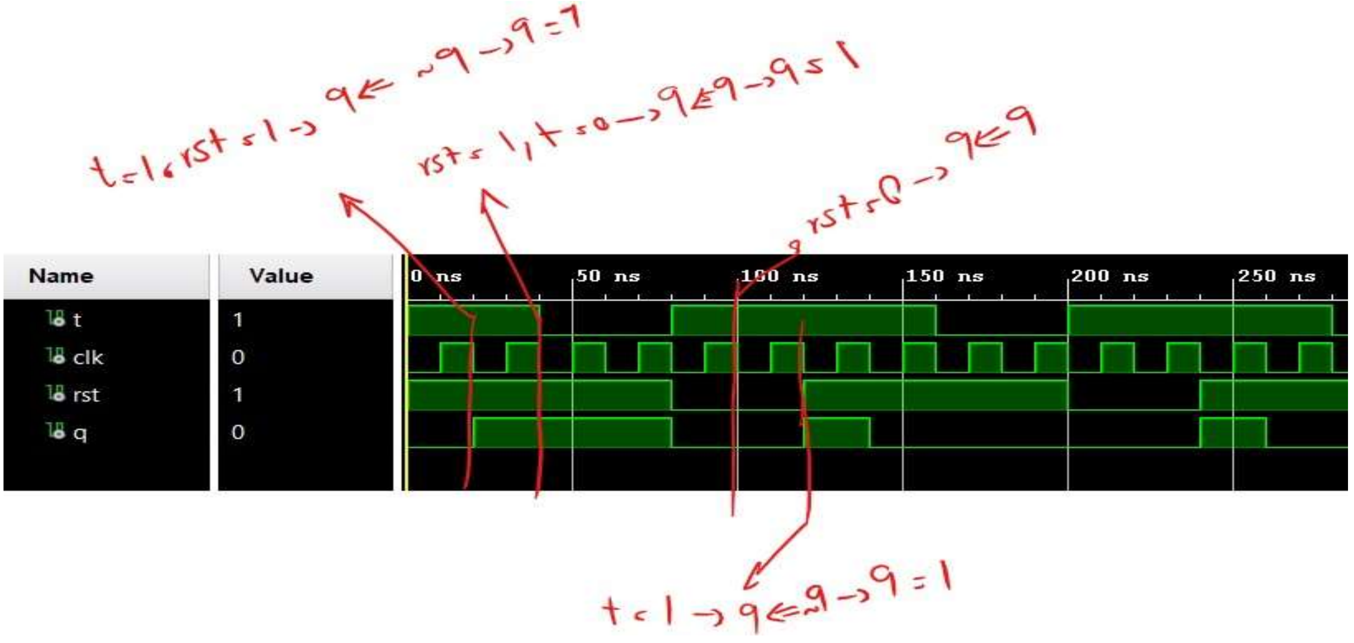
Mohammad Mehdi Nazari 9931061 / Armin Ebrahimi Saba 9931086

TFF:



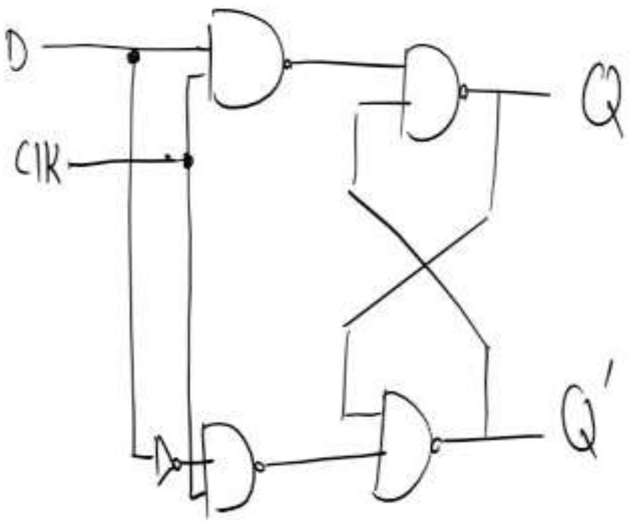
```
entity TFF is
    Port ( t : in STD_LOGIC;
           clk : in STD_LOGIC;
           rst: in STD_LOGIC;
           q : inout STD_LOGIC);
end TFF;

architecture Behavioral of TFF is
begin
    process(clk ,rst)
    begin
        if (rst = '0') then
            q <= '0';
        elsif (falling_edge(clk)) then
            q <= t xor q;
        end if;
    end process;
end Behavioral;
```

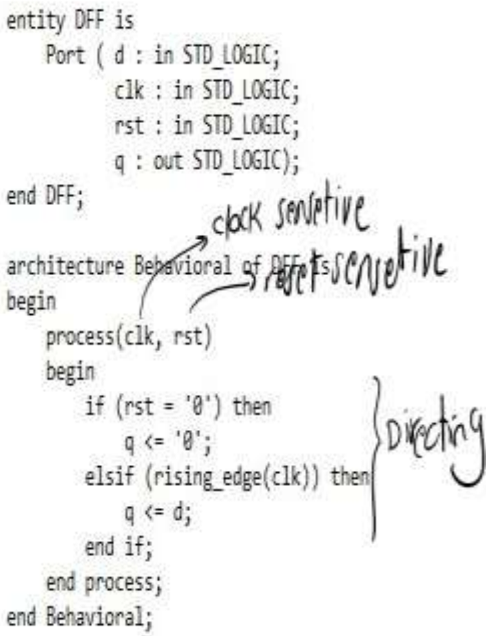




Name	Value
t	1
clk	0
rst	1
q	0

DFF:





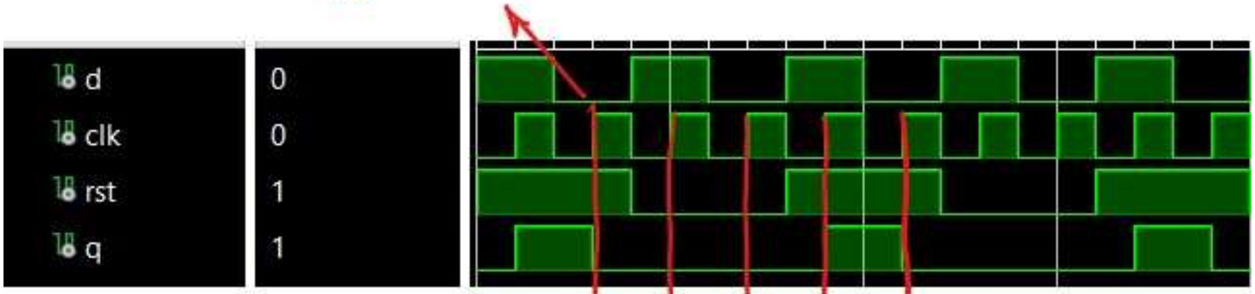
```
entity DFF is
    Port ( d : in STD_LOGIC;
           clk : in STD_LOGIC;
           rst : in STD_LOGIC;
           q : out STD_LOGIC);
end DFF;

architecture Behavioral of DFF is
begin
    process(clk, rst)
    begin
        if (rst = '0') then
            q <= '0';
        elsif (rising_edge(clk)) then
            q <= d;
        end if;
    end process;
end Behavioral;
```

clock sensetive

reset sensetive

Directing

rst = 1 , d = 0 → q ⇐ d → q = 0

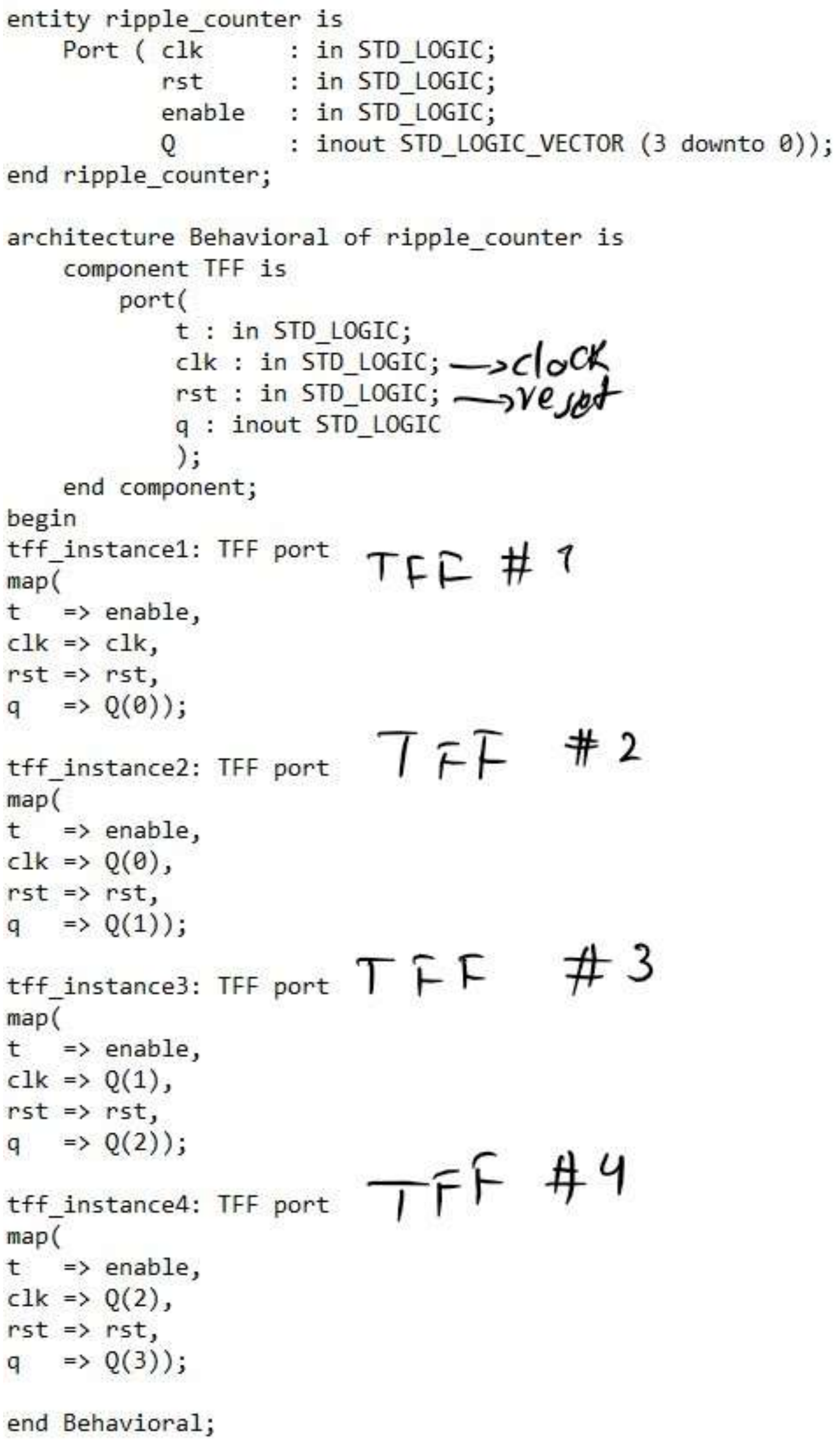


d 0
clk 0
rst 1
q 1

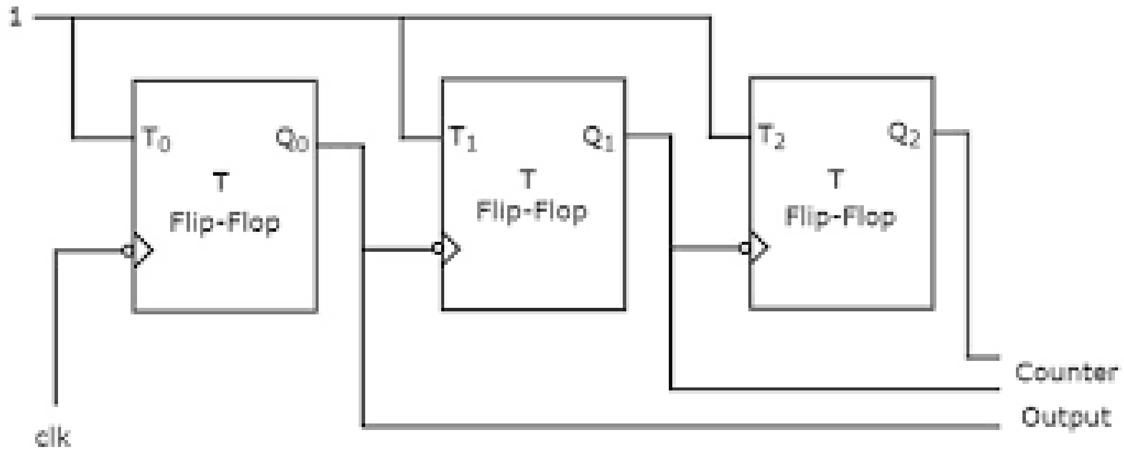
rst = 0 → q ⇐ 0

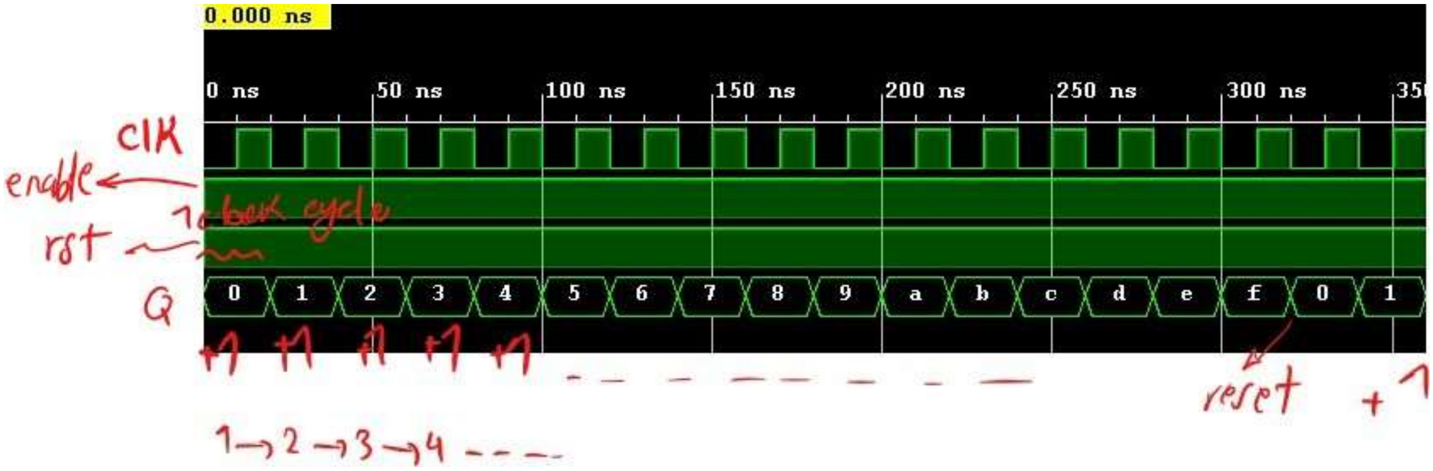
rst = 1 → q ⇐ d → q = 1

Rippler Counter:



```
entity ripple_counter is
    Port ( clk      : in STD_LOGIC;
           rst      : in STD_LOGIC;
           enable   : in STD_LOGIC;
           Q        : inout STD_LOGIC_VECTOR (3 downto 0));
end ripple_counter;

architecture Behavioral of ripple_counter is
    component TFF is
        port(
            t : in STD_LOGIC;
            clk : in STD_LOGIC; —> clock
            rst : in STD_LOGIC; —> reset
            q : inout STD_LOGIC
            );
    end component;
begin
tff_instance1: TFF port    TFF #1
map(
t   => enable,
clk => clk,
rst => rst,
q   => Q(0));

tff_instance2: TFF port    TFF #2
map(
t   => enable,
clk => Q(0),
rst => rst,
q   => Q(1));

tff_instance3: TFF port    TFF #3
map(
t   => enable,
clk => Q(1),
rst => rst,
q   => Q(2));

tff_instance4: TFF port    TFF #4
map(
t   => enable,
clk => Q(2),
rst => rst,
q   => Q(3));

end Behavioral;
```


1
T0
Q0
T
Flip-Flop
T1
Q1
T
Flip-Flop
T2
Q2
T
Flip-Flop
Counter
Output
clk


0.000 ns
0 ns
50 ns
100 ns
150 ns
200 ns
250 ns
300 ns
350
clk
enable
1 clock cycle
rst
Q
0
1
2
3
4
5
6
7
8
9
a
b
c
d
e
f
0
1
+1 +1 +1 +1 +1
reset
+1
1→2→3→4 ----