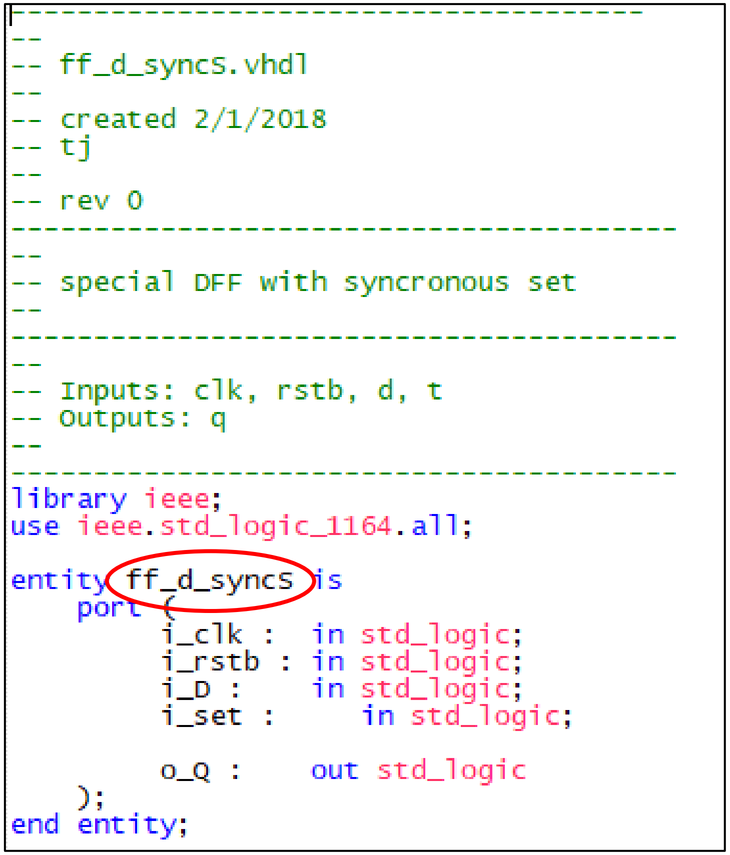
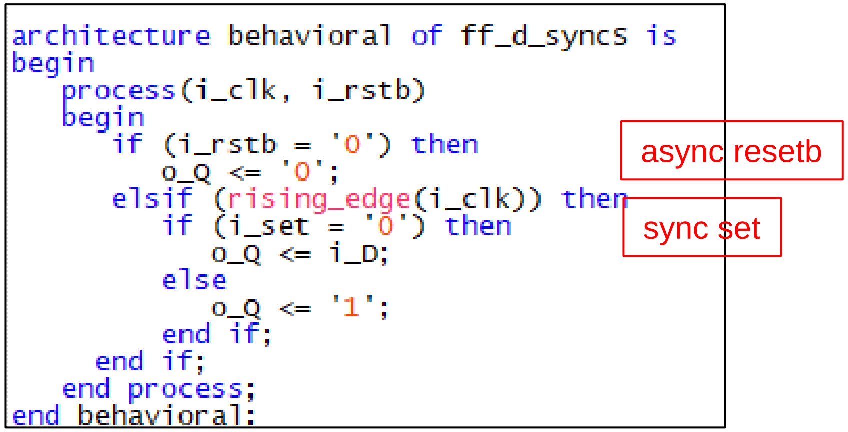
ELE 3510 HW4 Name________________

1 – Create VHDL code and RTL for a D-FF with asynchronous resetb and synchronous set. Follow all best practices. 20pts



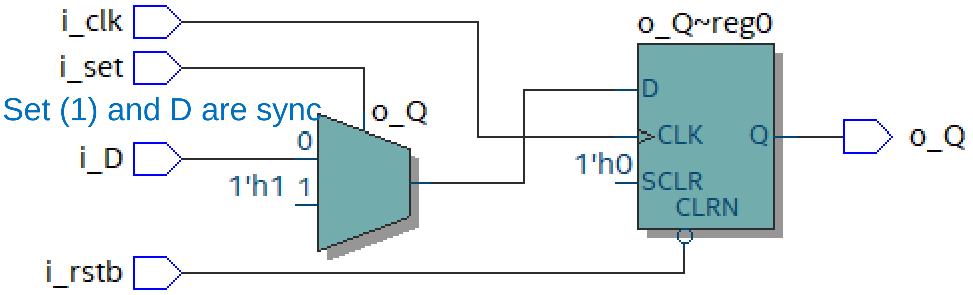
```
--------------------------------------
--
-- ff_d_syncs.vhdl
--
-- created 2/1/2018
-- tj
--
-- rev 0
----------------------------------------
--
-- special DFF with syncronous set
--
----------------------------------------
--
-- Inputs: clk, rstb, d, t
-- Outputs: q
--
----------------------------------------
library ieee;
use ieee.std_logic_1164.all;

entity ff_d_syncs is
    port (
        i_clk :  in std_logic;
        i_rstb : in std_logic;
        i_D :    in std_logic;
        i_set :     in std_logic;

        o_Q :    out std_logic
    );
end entity;
```



```
architecture behavioral of ff_d_syncs is
begin
   process(i_clk, i_rstb)
   begin
      if (i_rstb = '0') then
         o_Q <= '0';
      elsif (rising_edge(i_clk)) then
         if (i_set = '0') then
            o_Q <= i_D;
         else
            o_Q <= '1';
         end if;
      end if;
   end process;
end behavioral;
```

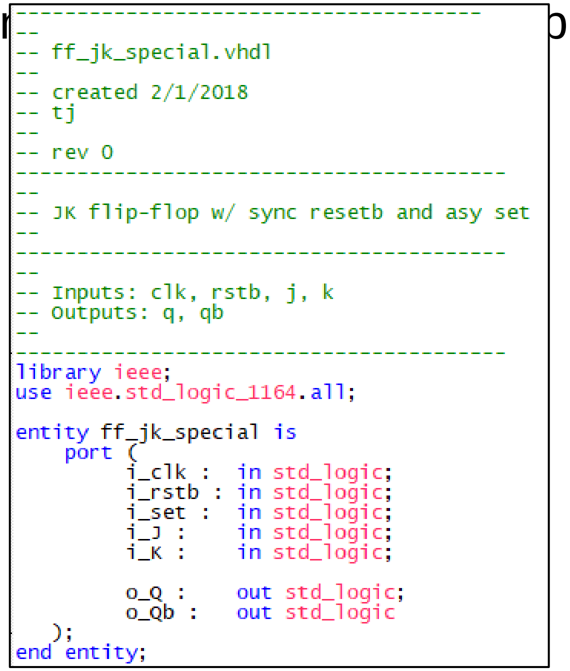
async resetb

sync set

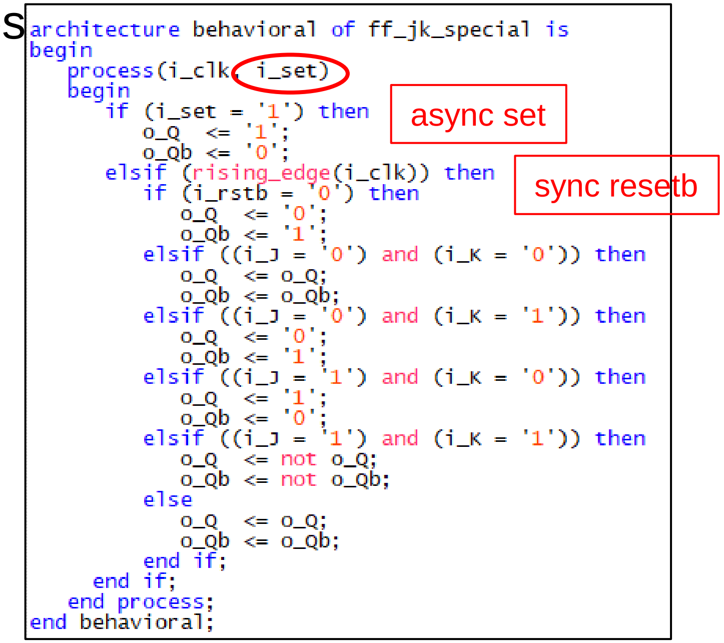

Set (1) and D are sync

ELE 3510 HW4 Name________________

2 – Create VHDL code and RTL for a JK-FF with synchronous resetb and asy … pes 30pts



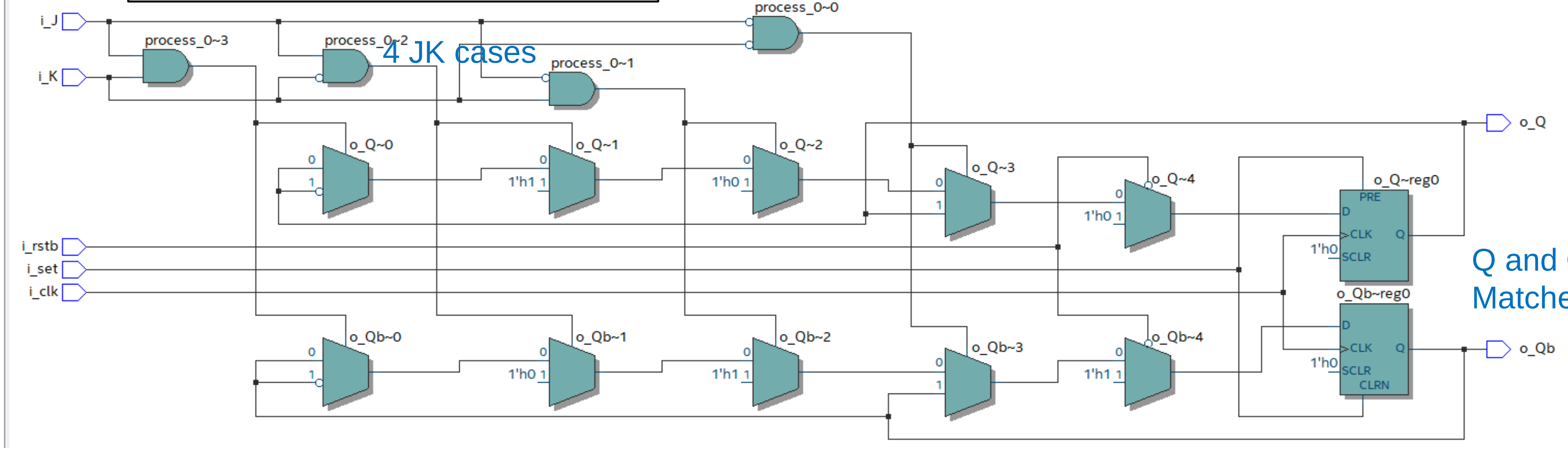
```
-----------------------------------------
--
-- ff_jk_special.vhdl
--
-- created 2/1/2018
-- tj
--
-- rev 0
-------------------------------------------
--
-- JK flip-flop w/ sync resetb and asy set
--
-------------------------------------------
--
-- Inputs: clk, rstb, j, k
-- Outputs: q, qb
--
-------------------------------------------
library ieee;
use ieee.std_logic_1164.all;

entity ff_jk_special is
    port (
          i_clk :   in std_logic;
          i_rstb :  in std_logic;
          i_set :   in std_logic;
          i_J :     in std_logic;
          i_K :     in std_logic;

          o_Q :     out std_logic;
          o_Qb :    out std_logic
    );
end entity;
```



```
architecture behavioral of ff_jk_special is
begin
   process(i_clk, i_set)
   begin
      if (i_set = '1') then
         o_Q  <= '1';
         o_Qb <= '0';
      elsif (rising_edge(i_clk)) then
         if (i_rstb = '0') then
            o_Q  <= '0';
            o_Qb <= '1';
         elsif ((i_J = '0') and (i_K = '0')) then
            o_Q  <= o_Q;
            o_Qb <= o_Qb;
         elsif ((i_J = '0') and (i_K = '1')) then
            o_Q  <= '0';
            o_Qb <= '1';
         elsif ((i_J = '1') and (i_K = '0')) then
            o_Q  <= '1';
            o_Qb <= '0';
         elsif ((i_J = '1') and (i_K = '1')) then
            o_Q  <= not o_Q;
            o_Qb <= not o_Qb;
         else
            o_Q  <= o_Q;
            o_Qb <= o_Qb;
         end if;
      end if;
   end process;
end behavioral;
```

async set

sync resetb

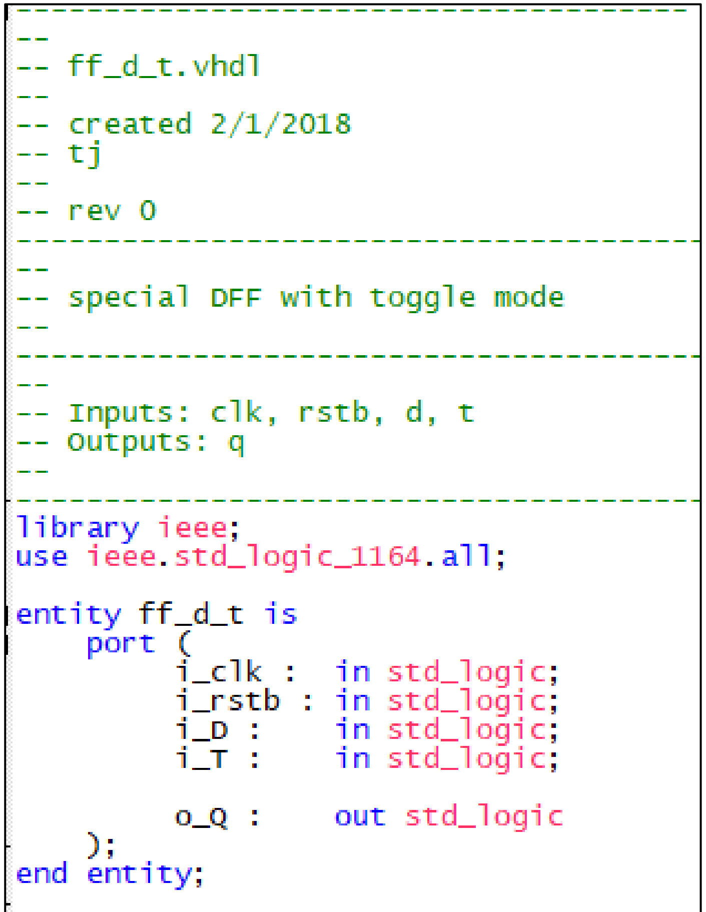


4 JK cases

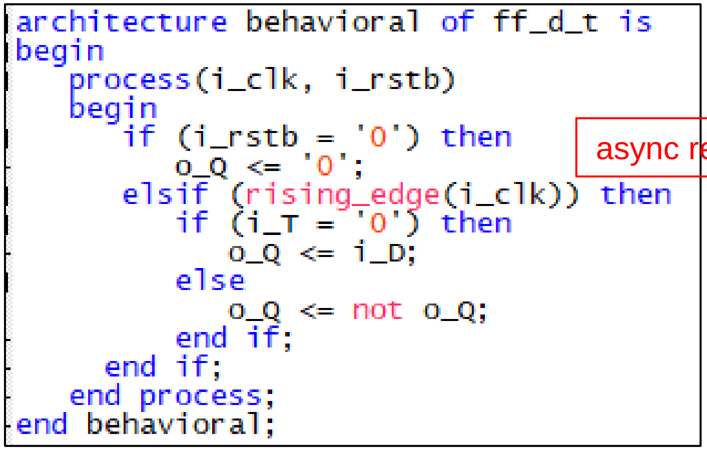
Q and Qb out
Matched delays

ELE 3510 HW4 Name________________

3 – Create VHDL code and RTL for a special new FlipFlop you have conceived of – this is a normal D-FF (with async rstb) but has a second data input called Toggle(T). When T is low, the DFF operates normally, when T is high, the DFF ignores the D input and synchronously toggles the output. Follow all best practices. 50pts

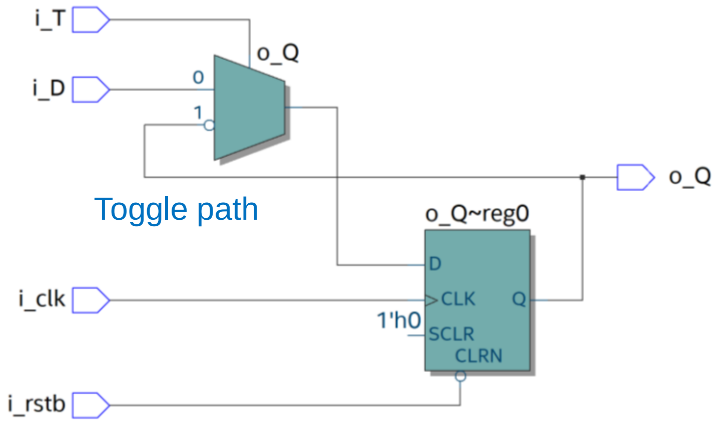


```
---------------------------------------
--
-- ff_d_t.vhdl
--
-- created 2/1/2018
-- tj
--
-- rev 0
---------------------------------------
--
-- special DFF with toggle mode
--
---------------------------------------
--
-- Inputs: clk, rstb, d, t
-- Outputs: q
--
---------------------------------------
library ieee;
use ieee.std_logic_1164.all;

entity ff_d_t is
    port (
         i_clk :  in std_logic;
         i_rstb : in std_logic;
         i_D :    in std_logic;
         i_T :    in std_logic;

         o_Q :    out std_logic
    );
end entity;
```



```
architecture behavioral of ff_d_t is
begin
   process(i_clk, i_rstb)
   begin
      if (i_rstb = '0') then
         o_Q <= '0';
      elsif (rising_edge(i_clk)) then
         if (i_T = '0') then
            o_Q <= i_D;
         else
            o_Q <= not o_Q;
         end if;
      end if;
   end process;
end behavioral;
```

async resetb



Toggle path