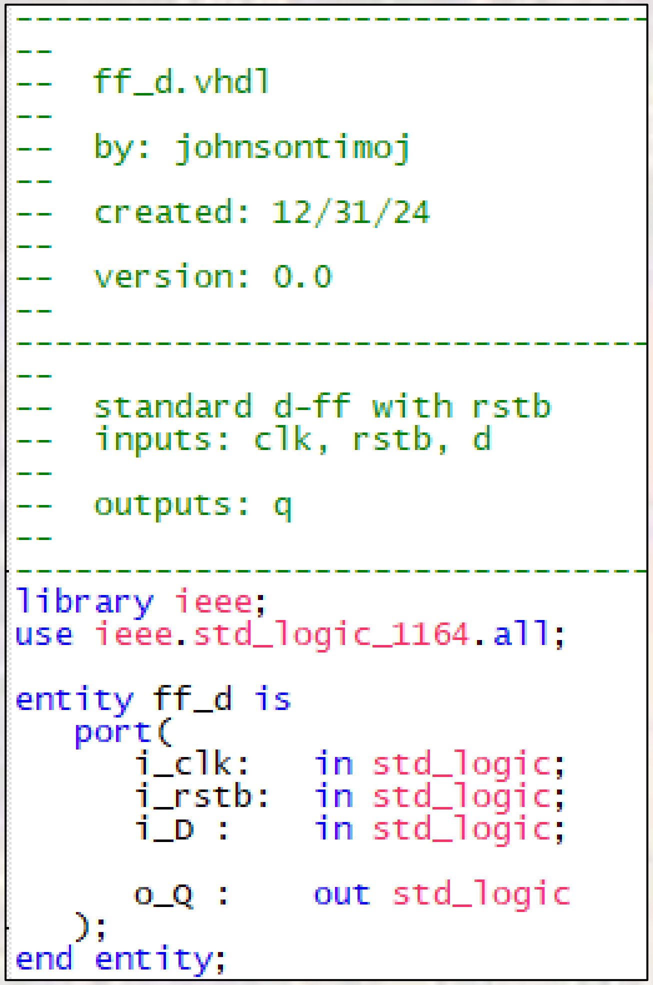
VHDL Flip-Flop Examples

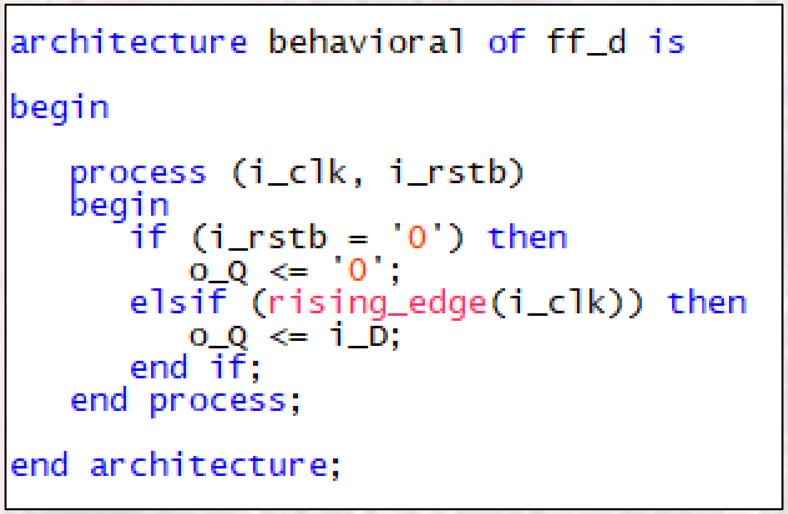
Last updated 1/8/25

VHDL Flip-Flop Examples

- Base D-FF with reset



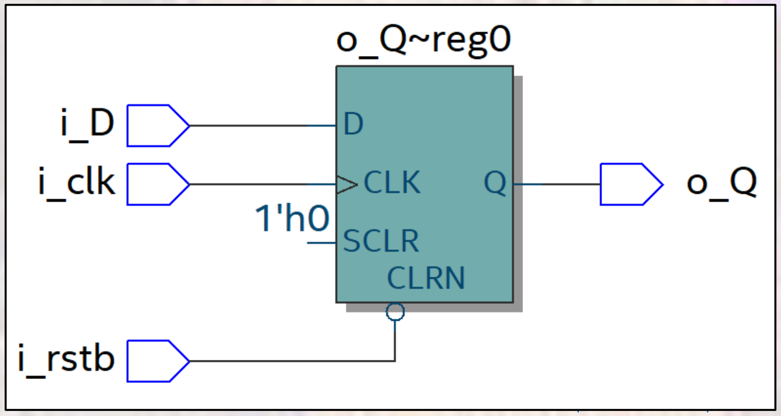
```
-----------------------------------
--
--   ff_d.vhdl
--
--   by: johnsontimoj
--
--   created: 12/31/24
--
--   version: 0.0
--
-----------------------------------
--
--   standard d-ff with rstb
--   inputs: clk, rstb, d
--
--   outputs: q
--
-----------------------------------
library ieee;
use ieee.std_logic_1164.all;

entity ff_d is
   port(
      i_clk:   in std_logic;
      i_rstb:  in std_logic;
      i_D :    in std_logic;

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

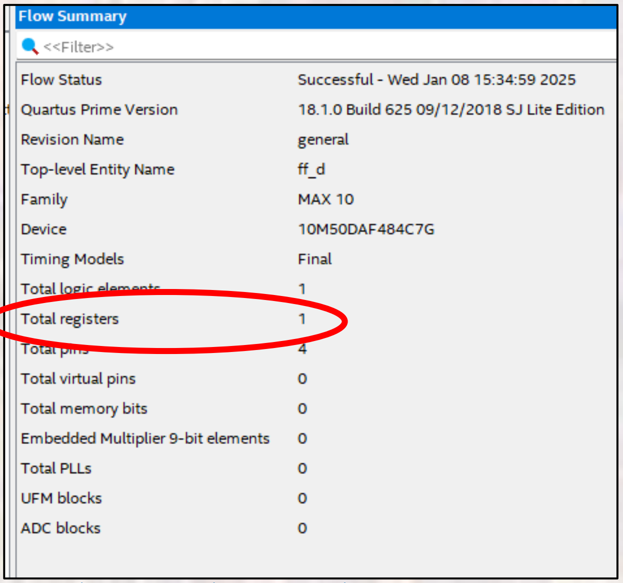


```
architecture behavioral of ff_d is

begin

   process (i_clk, i_rstb)
   begin
      if (i_rstb = '0') then
         o_Q <= '0';
      elsif (rising_edge(i_clk)) then
         o_Q <= i_D;
      end if;
   end process;

end architecture;
```

VHDL Flip-Flop Examples

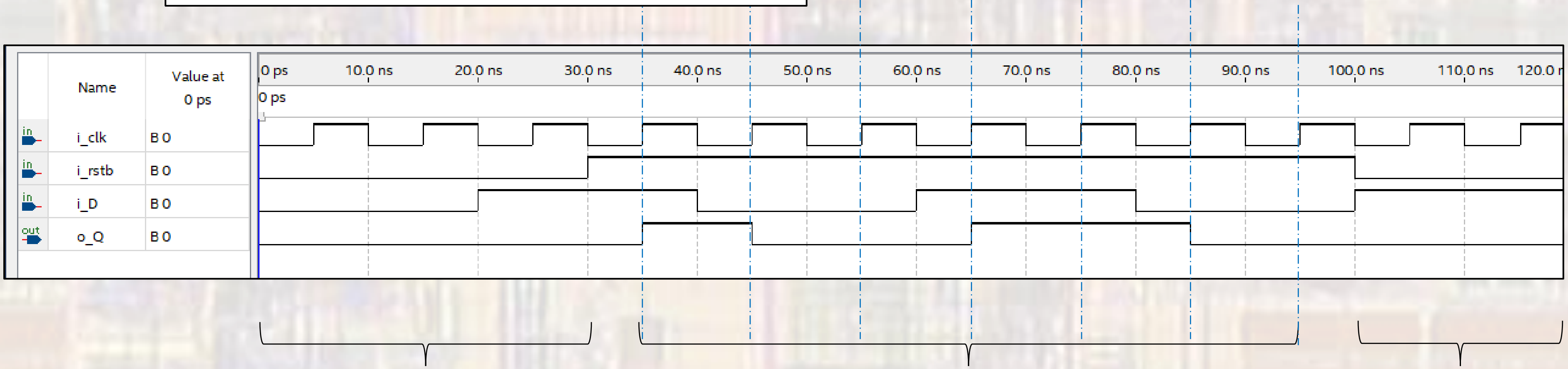
- Base D-FF with reset





Flow Summary

Flow Status	Successful - Wed Jan 08 15:34:59 2025
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	general
Top-level Entity Name	ff_d
Family	MAX 10
Device	10M50DAF484C7G
Timing Models	Final
Total logic elements	1
Total registers	1
Total pins	4
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0
UFM blocks	0
ADC blocks	0



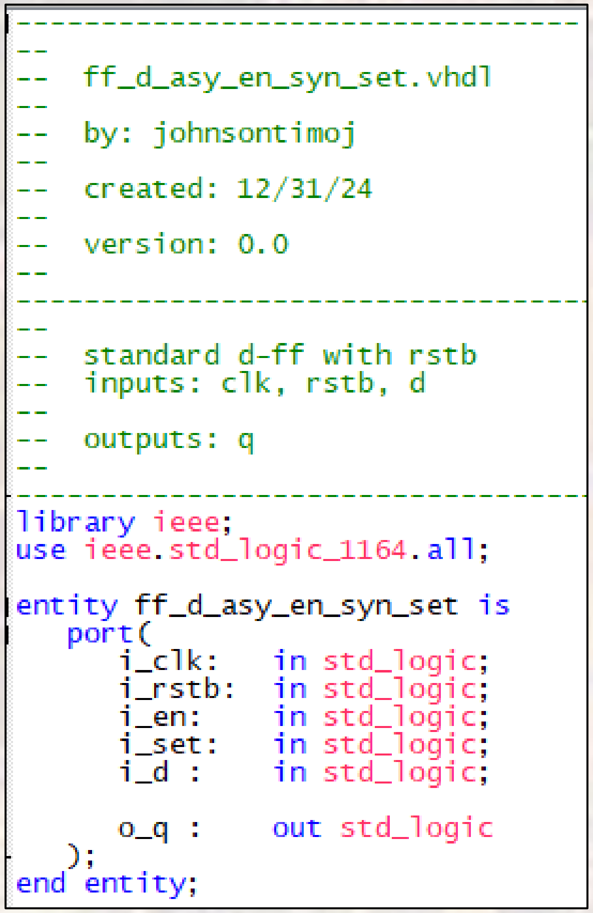
reset enabled (active low)
Q is reset – nothing happening

edge captures

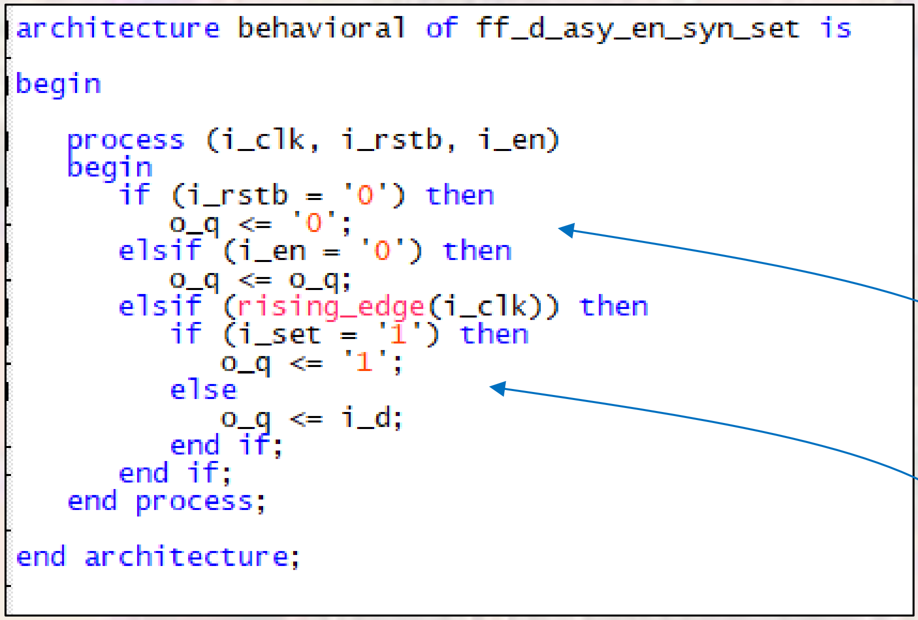
reset enabled (active low)
Q is reset – nothing happening

VHDL Flip-Flop Examples

- d-ff w/asy en, sync set



```
----------------------------------
--
--   ff_d_asy_en_syn_set.vhdl
--
--   by: johnsontimoj
--
--   created: 12/31/24
--
--   version: 0.0
--
----------------------------------
--
--   standard d-ff with rstb
--   inputs: clk, rstb, d
--
--   outputs: q
--
----------------------------------
library ieee;
use ieee.std_logic_1164.all;

entity ff_d_asy_en_syn_set is
   port(
      i_clk:   in std_logic;
      i_rstb:  in std_logic;
      i_en:    in std_logic;
      i_set:   in std_logic;
      i_d :    in std_logic;

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



```
architecture behavioral of ff_d_asy_en_syn_set is

begin

   process (i_clk, i_rstb, i_en)
   begin
      if (i_rstb = '0') then
         o_q <= '0';
      elsif (i_en = '0') then
         o_q <= o_q;
      elsif (rising_edge(i_clk)) then
         if (i_set = '1') then
            o_q <= '1';
         else
            o_q <= i_d;
         end if;
      end if;
   end process;

end architecture;
```

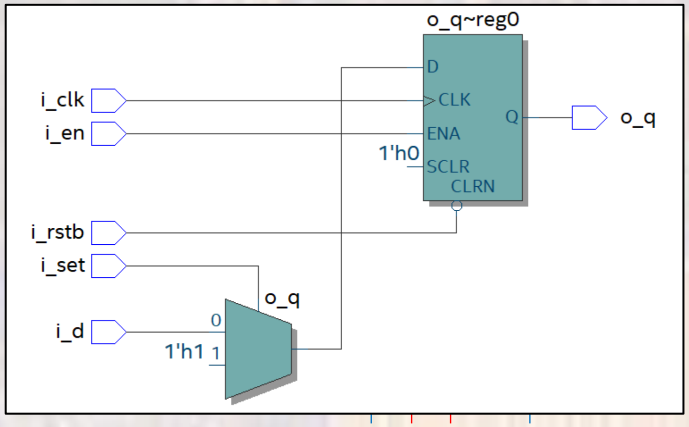
set an explicit priority

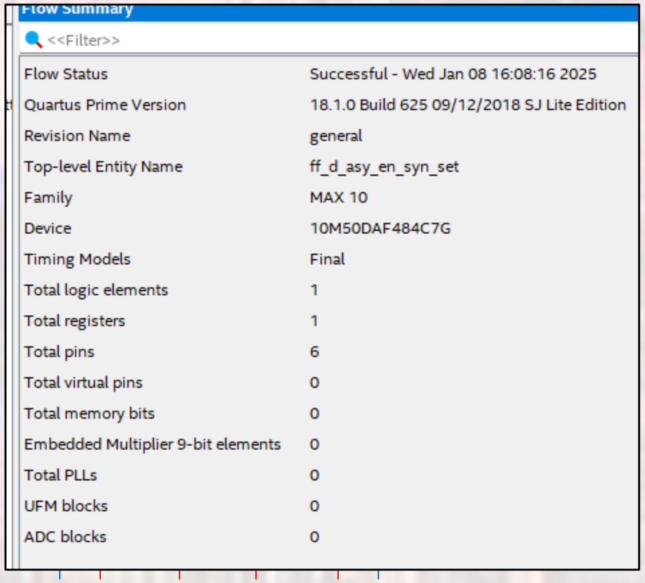
rstb > en

set > d

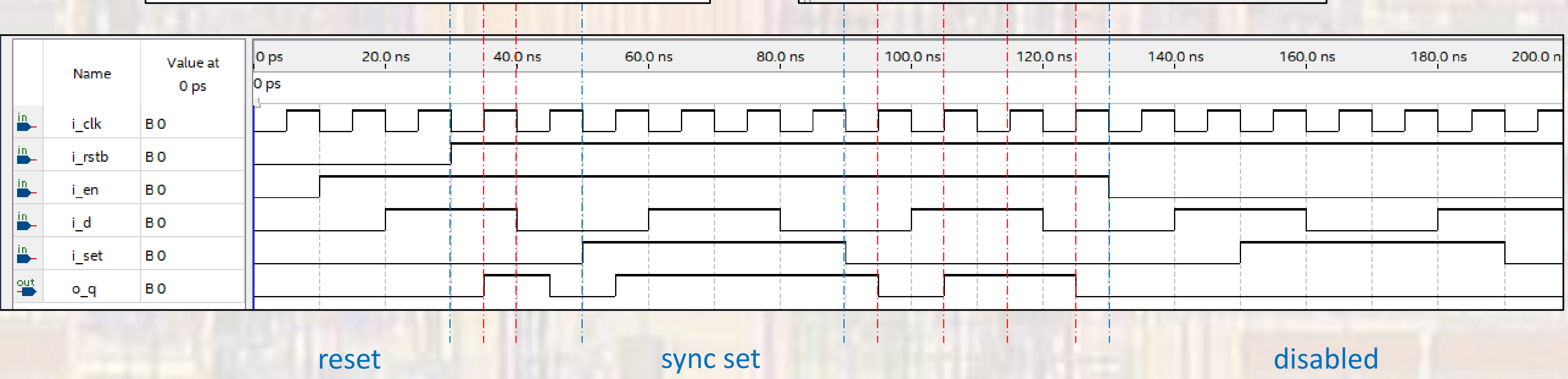
VHDL Flip-Flop Examples

- d-ff w/asy en, sycn set





Flow Status	Successful - Wed Jan 08 16:08:16 2025
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	general
Top-level Entity Name	ff_d_asy_en_syn_set
Family	MAX 10
Device	10M50DAF484C7G
Timing Models	Final
Total logic elements	1
Total registers	1
Total pins	6
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0
UFM blocks	0
ADC blocks	0



reset

dff

sync set

dff

disabled

VHDL Flip-Flop Examples

- Warning – Warning – Warning
  - The FF template is an exception to the if/else and case rules for creating latches
  - Outside the FF construct:
    - If you do not complete an if-else with an else, a latch will be created
    - If you do not cover all cases in a case statement, a latch will be created
    - All paths/cases must be covered
    - The compiler will always warn you it created a latch

We do not want latches - EVER

I can see a latch in an RTL diagram from a mile away