**طراحی سیستم های دیجیتال**

**تکلیف دوم**

**استاد مروستی**

**علیرضا نعمتی**

**کسری رشیدفر**

Table of Contents

[Question1 2](#_Toc69762893)

[JK\_FF 2](#_Toc69762894)

[Code 3](#_Toc69762895)

[Compilation report 4](#_Toc69762896)

[RTL 4](#_Toc69762897)

[Post-Mapping 5](#_Toc69762898)

[T\_FF 6](#_Toc69762899)

[Code 6](#_Toc69762900)

[Compilation report 7](#_Toc69762901)

[Rtl 7](#_Toc69762902)

[Post-Mapping 8](#_Toc69762903)

[Question2 8](#_Toc69762904)

[Code 9](#_Toc69762905)

[Compilation report 9](#_Toc69762906)

[RTL 10](#_Toc69762907)

[Post-Mapping 10](#_Toc69762908)

[Question3 11](#_Toc69762909)

[Code 11](#_Toc69762910)

[Compilation report 12](#_Toc69762911)

[RTL 13](#_Toc69762912)

[Post-Mapping 13](#_Toc69762913)

[Question4 14](#_Toc69762914)

[Code 14](#_Toc69762915)

[Compilation report 16](#_Toc69762916)

[RTL 17](#_Toc69762917)

[Post-mapping 19](#_Toc69762918)

# Question1

## JK\_FF

### Code

library ieee;

use ieee.std\_logic\_1164.all;

entity JK\_FF is port(

clk, reset: in std\_logic;

j, k: in std\_logic;

q: out std\_logic

);

end JK\_FF;

architecture arch of JK\_FF is

signal q\_temp: std\_logic;

begin

process(clk, reset)

begin

if(reset = '1') then

q\_temp <= '0';

elsif (clk'event and clk = '1') then

if(j = '0' and k = '0') then

q\_temp <= q\_temp;

elsif(j = '0' and k = '1') then

q\_temp <= '0';

elsif(j = '1' and k = '0') then

q\_temp <= '1';

elsif(j = '1' and k = '1') then

q\_temp <= not (q\_temp);

end if;

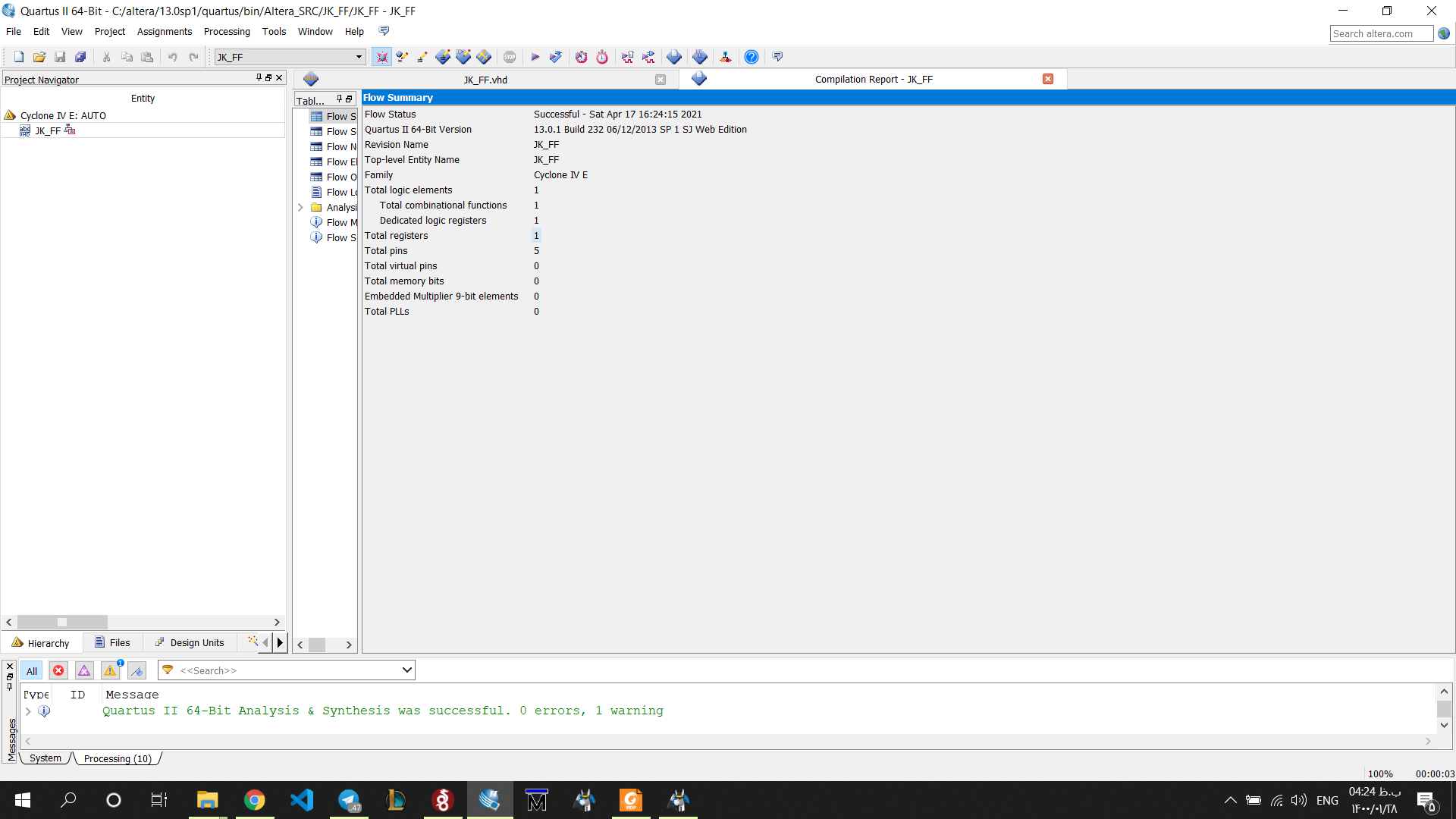
end if;

end process;

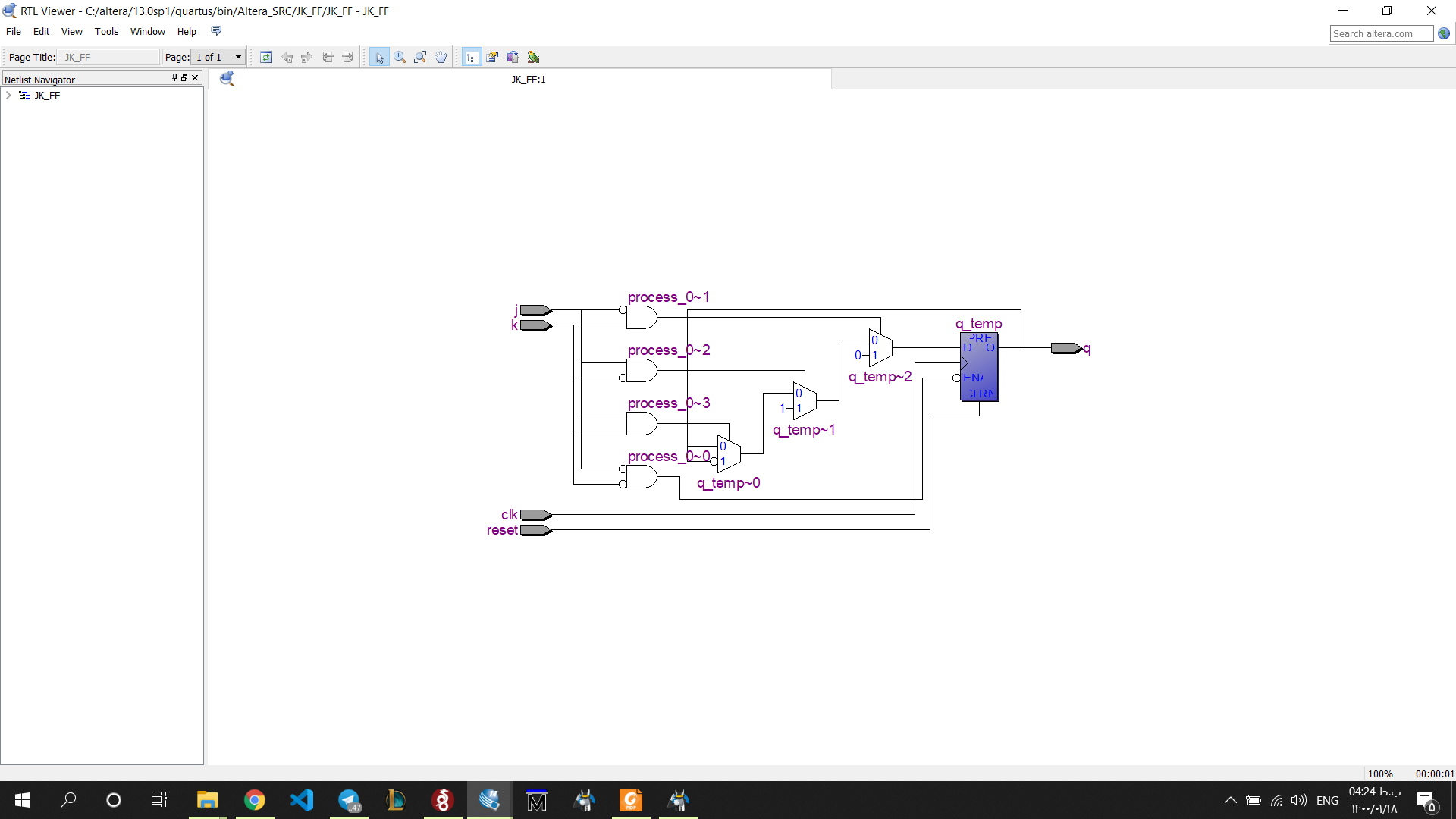
q <= q\_temp;

end arch;

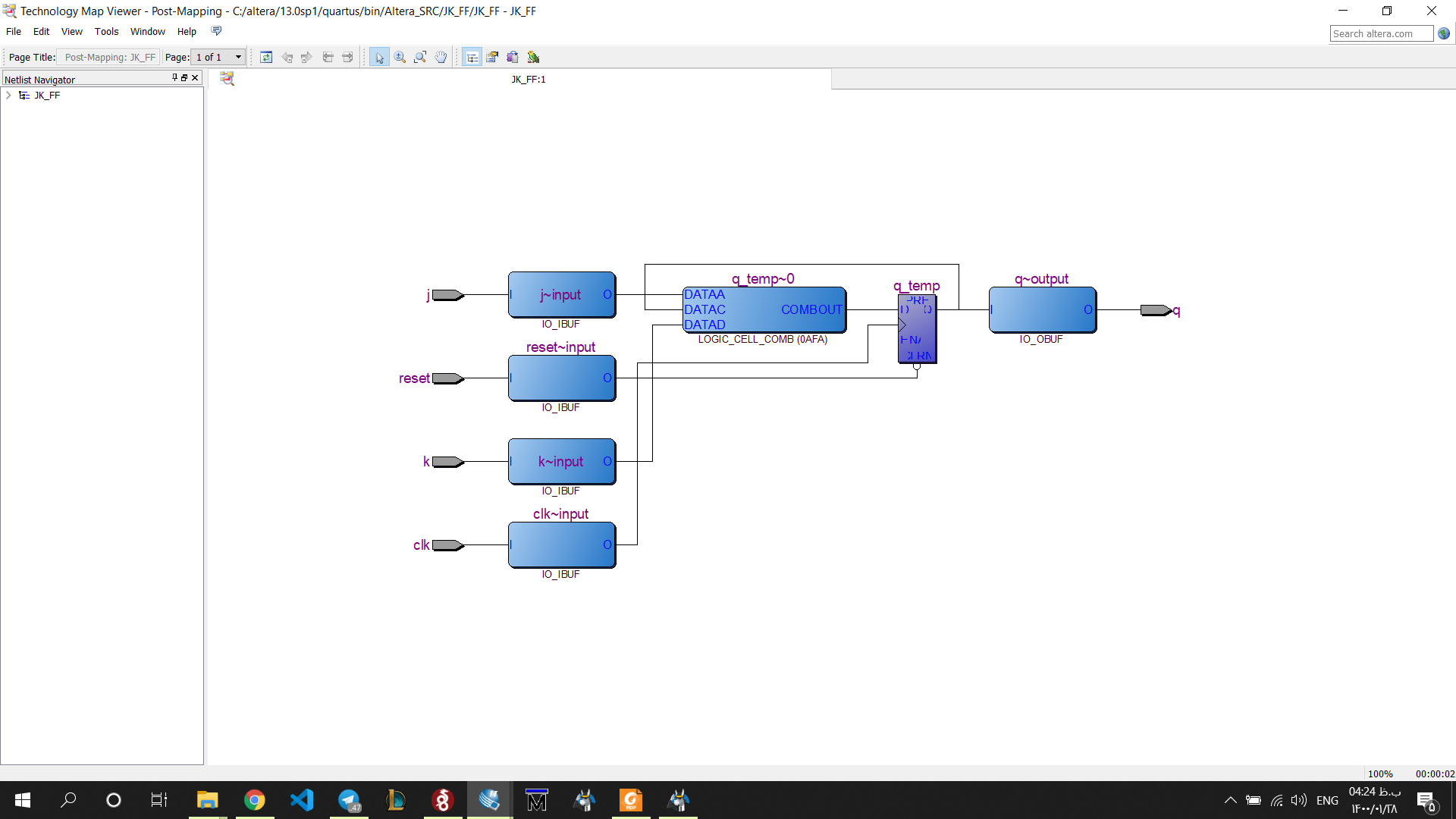
### Compilation report

****

### RTL

****

### Post-Mapping

****

## T\_FF

### Code

library ieee;

use ieee.std\_logic\_1164.all;

entity T\_FF is

port(

clk: in std\_logic;

reset: in std\_logic;

t: in std\_logic;

q: out std\_logic

);

end T\_FF;

architecture arch of T\_FF is

signal q\_next: std\_logic;

begin

process(clk, reset)

begin

if (reset = '1') then

q\_next <= '0';

elsif (clk'event and clk = '1') then

q\_next <= (t xor q\_next);

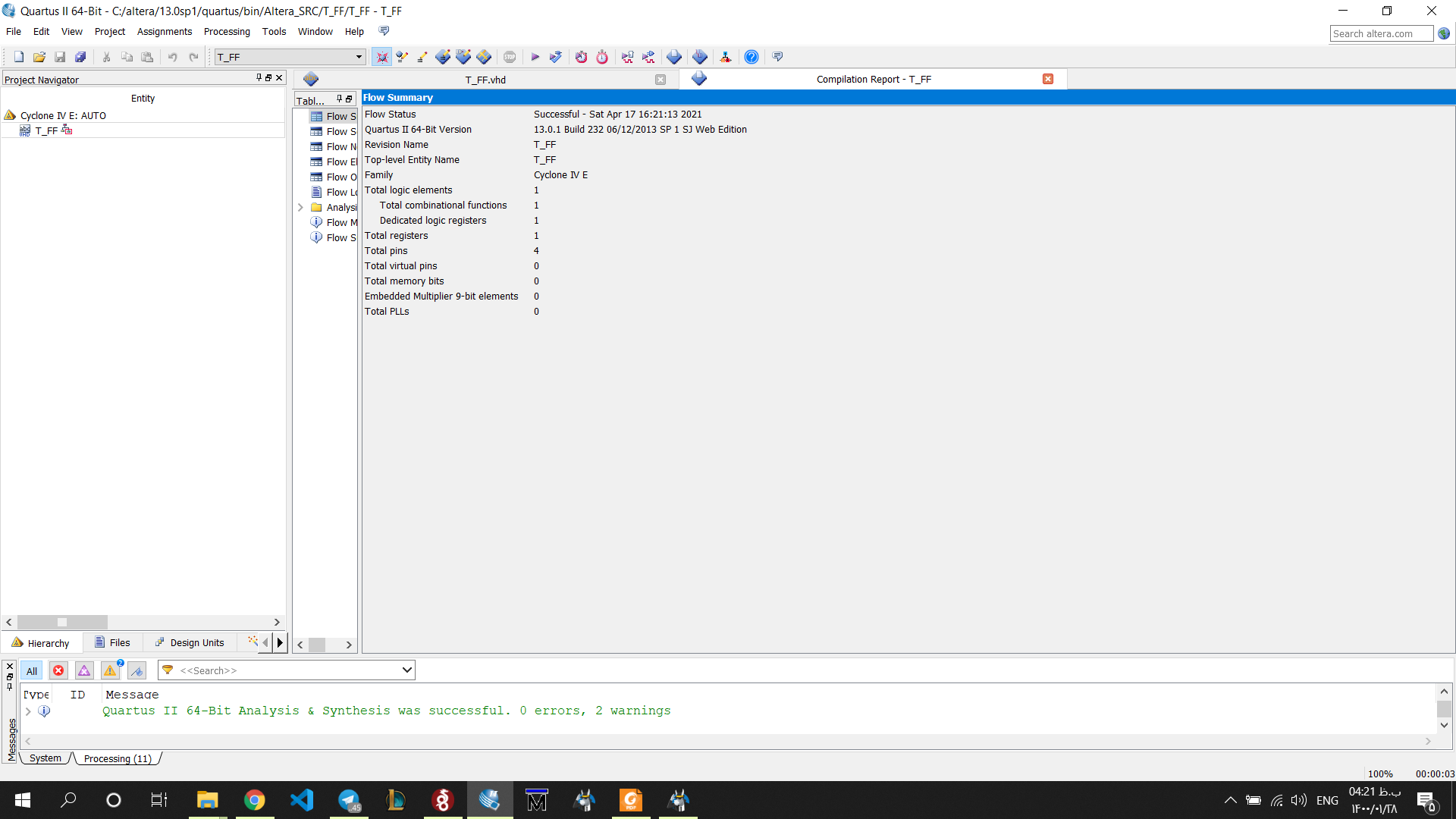
end if;

q <= q\_next;

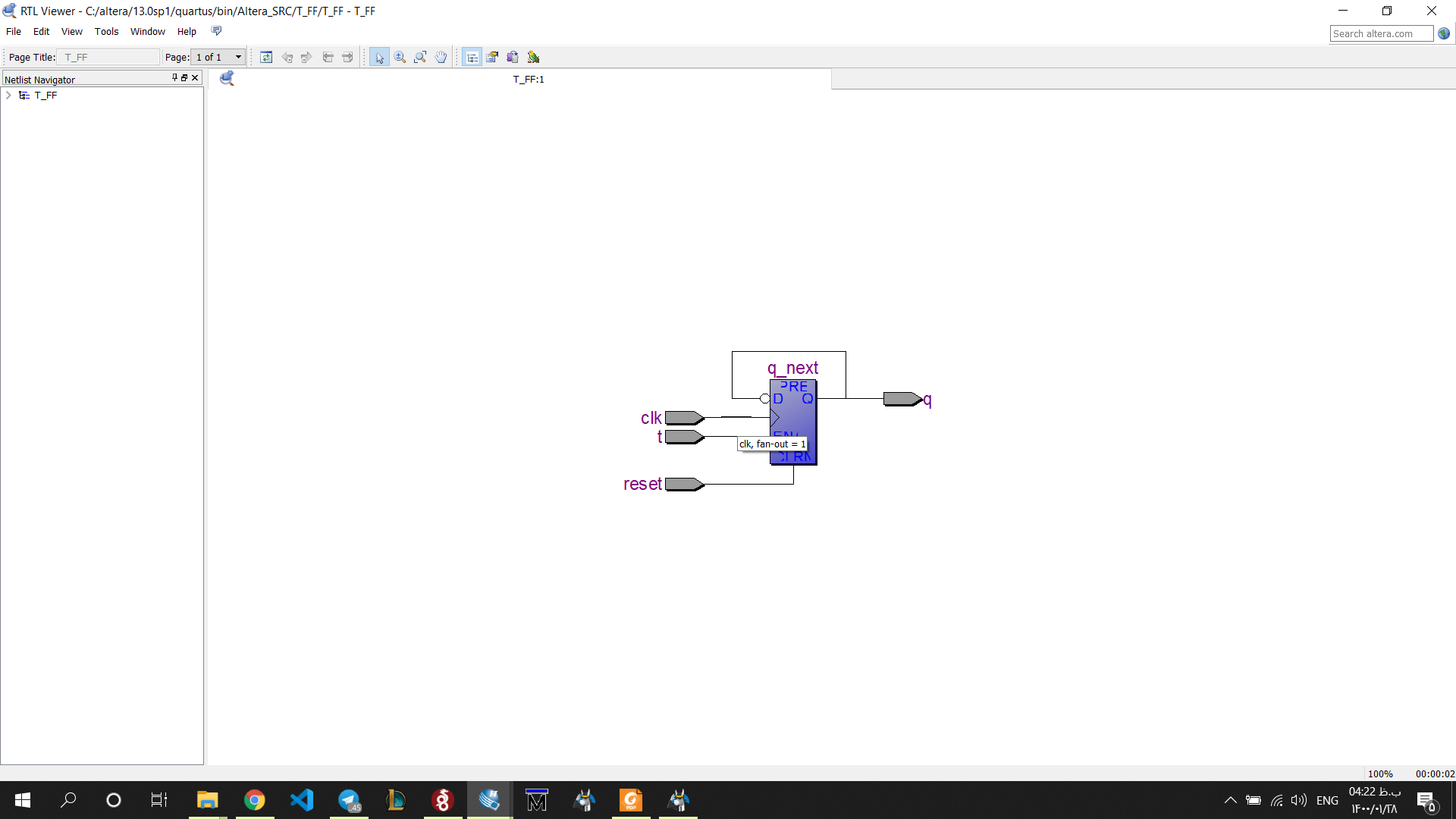
end process;

end arch;

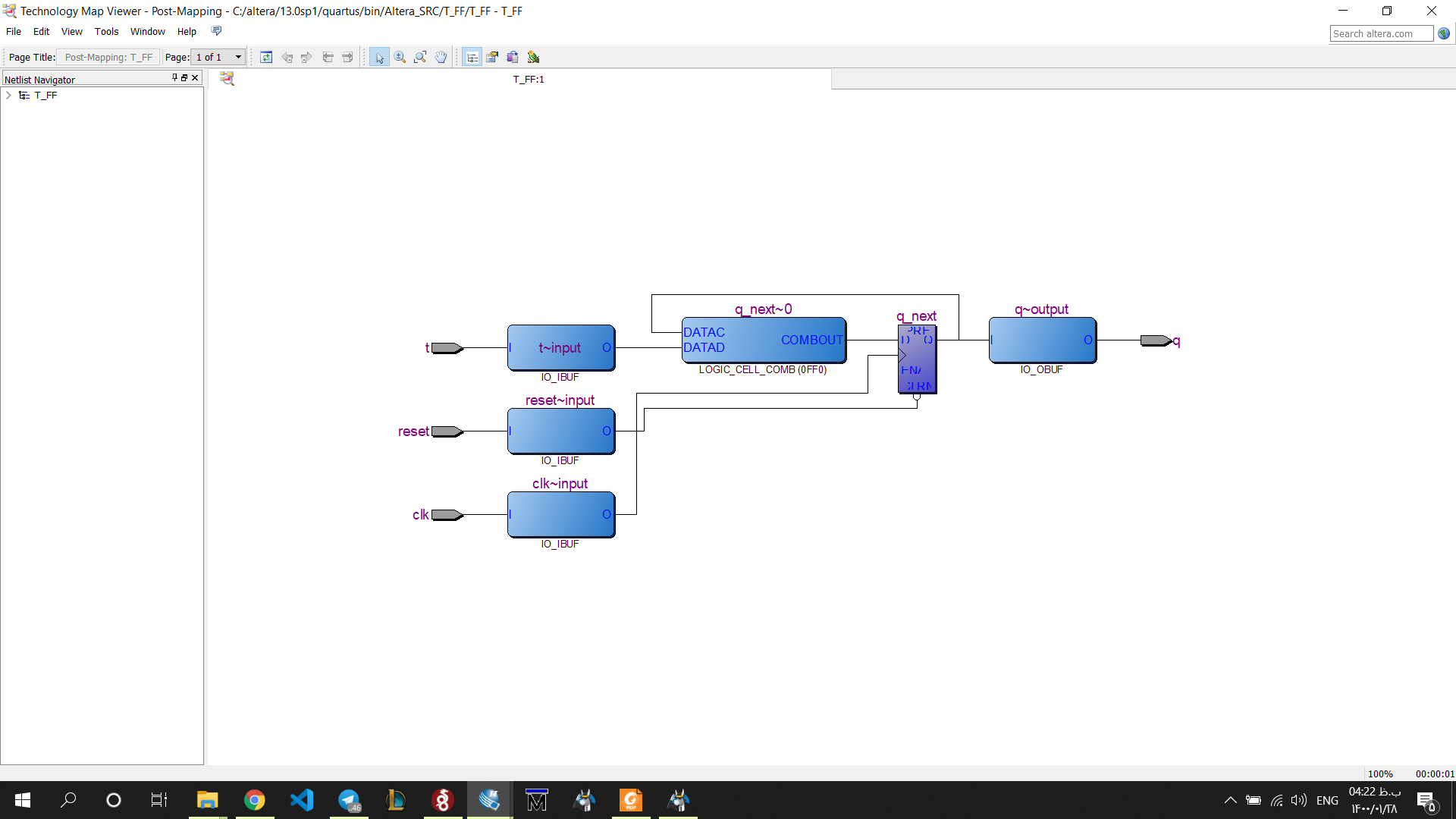
### Compilation report



### Rtl



### Post-Mapping



# Question2

## Code

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

entity BCD\_Counter is port(

clk, reset: in std\_logic;

Dir: in std\_logic;

q: out std\_logic\_vector(3 downto 0)

);

end BCD\_Counter;

architecture arch of BCD\_Counter is

signal counter: unsigned(3 downto 0);

begin

process(clk, reset)

begin

if (reset = '1') then

counter <= (others => '0');

elsif (clk'event and clk = '1') then

if (Dir = '1') then

counter <= counter + 1;

elsif (Dir = '0') then

counter <= counter - 1;

end if;

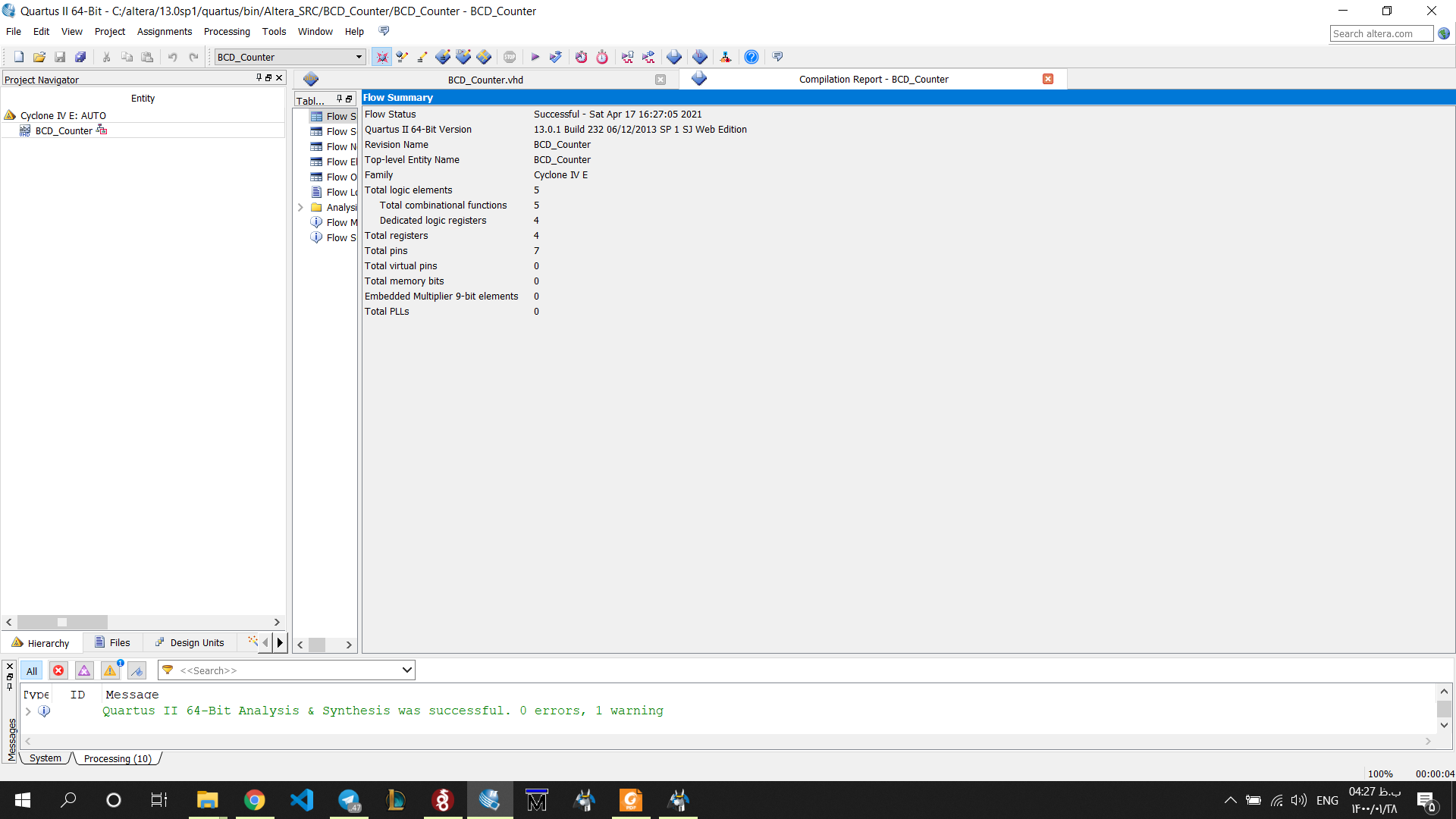
end if;

end process;

q <= std\_logic\_vector(counter);

end arch;

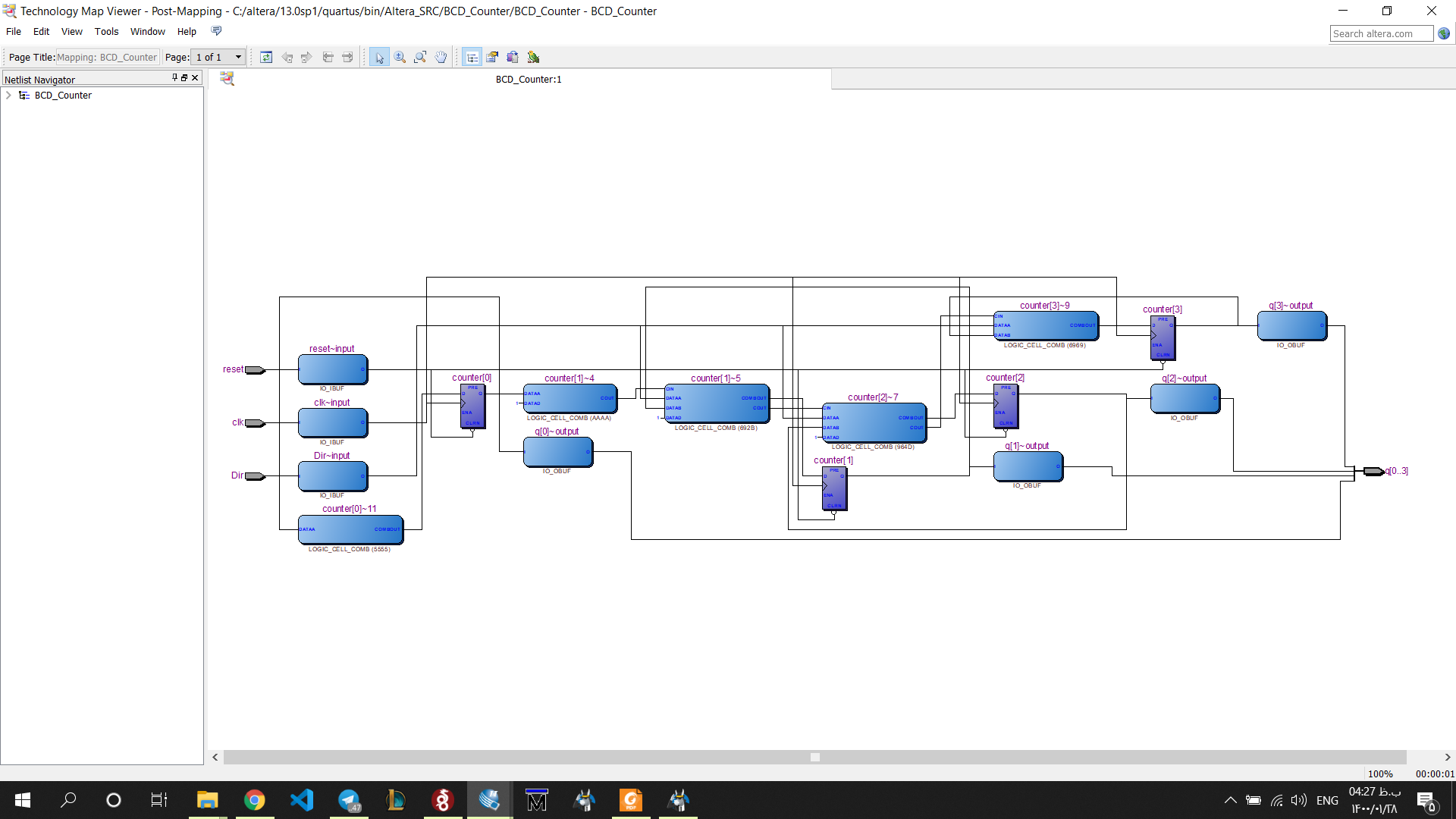
## Compilation report

****

## RTL

****

## Post-Mapping

****

# Question3

## Code

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

entity arbitrary\_counter is

port(

clk, reset: in std\_logic;

q: out std\_logic\_vector(2 downto 0)

);

end arbitrary\_counter;

architecture arch of arbitrary\_counter is

signal r\_reg: std\_logic\_vector(2 downto 0);

signal r\_next: std\_logic\_vector(2 downto 0);

begin

process(clk, reset)

begin

if (reset = '1') then

r\_reg <= (others => '0');

elsif (clk'event and clk = '1') then

r\_reg <= r\_next;

end if;

end process;

r\_next <= "011" when r\_reg="000" else

"110" when r\_reg="011" else

"101" when r\_reg="110" else

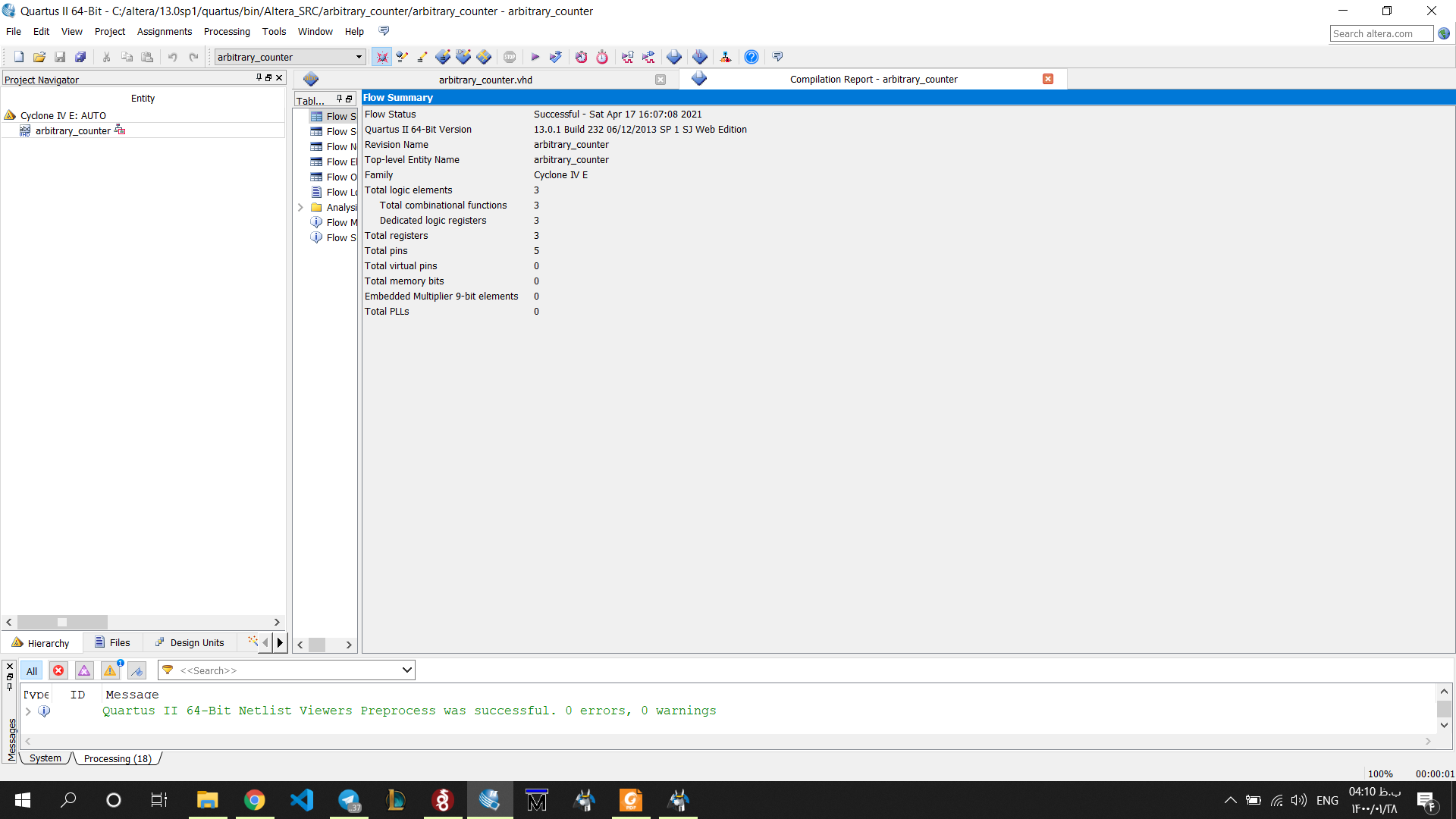
"111" when r\_reg="101" else

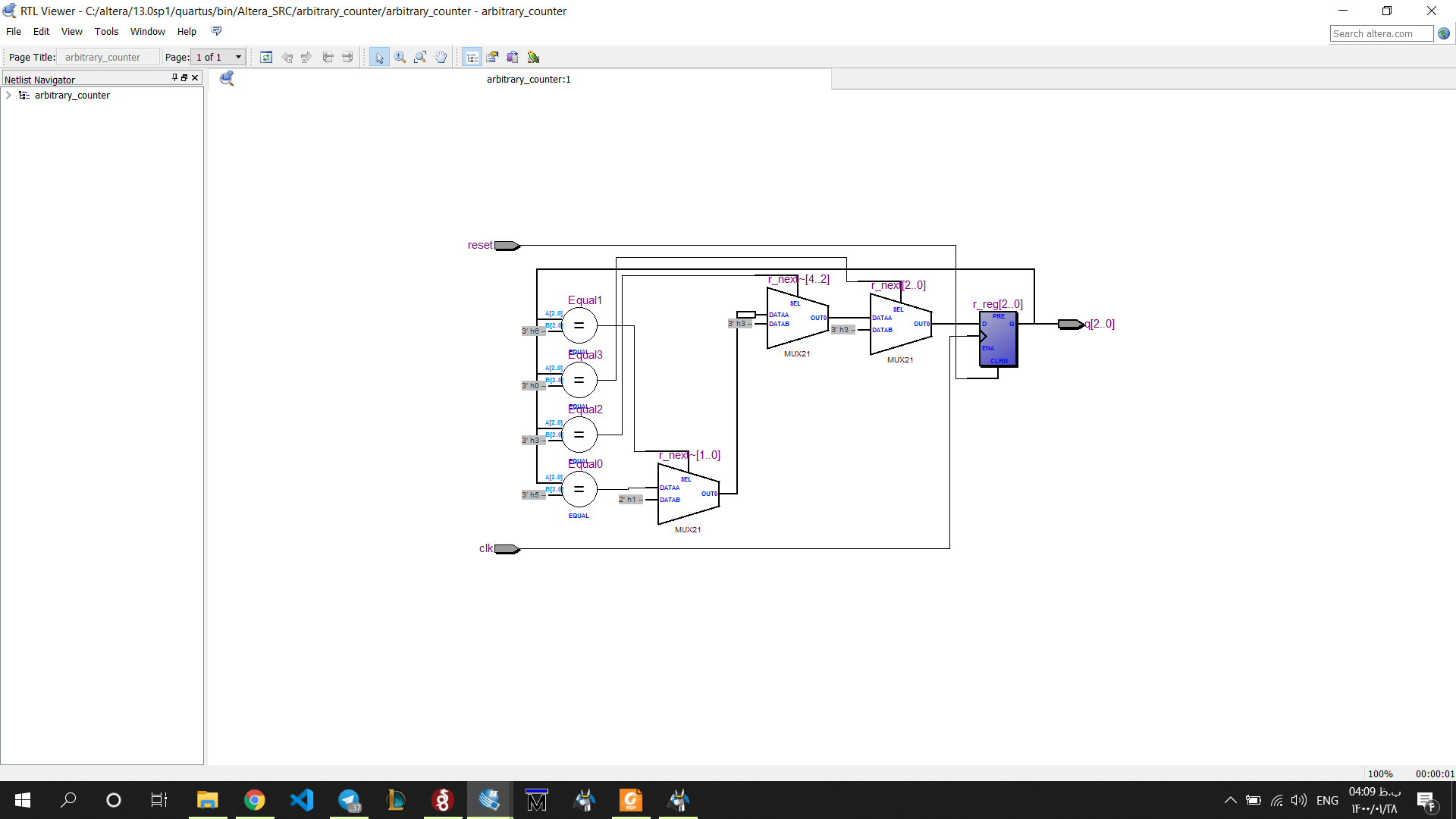
"000";

q <= r\_reg;

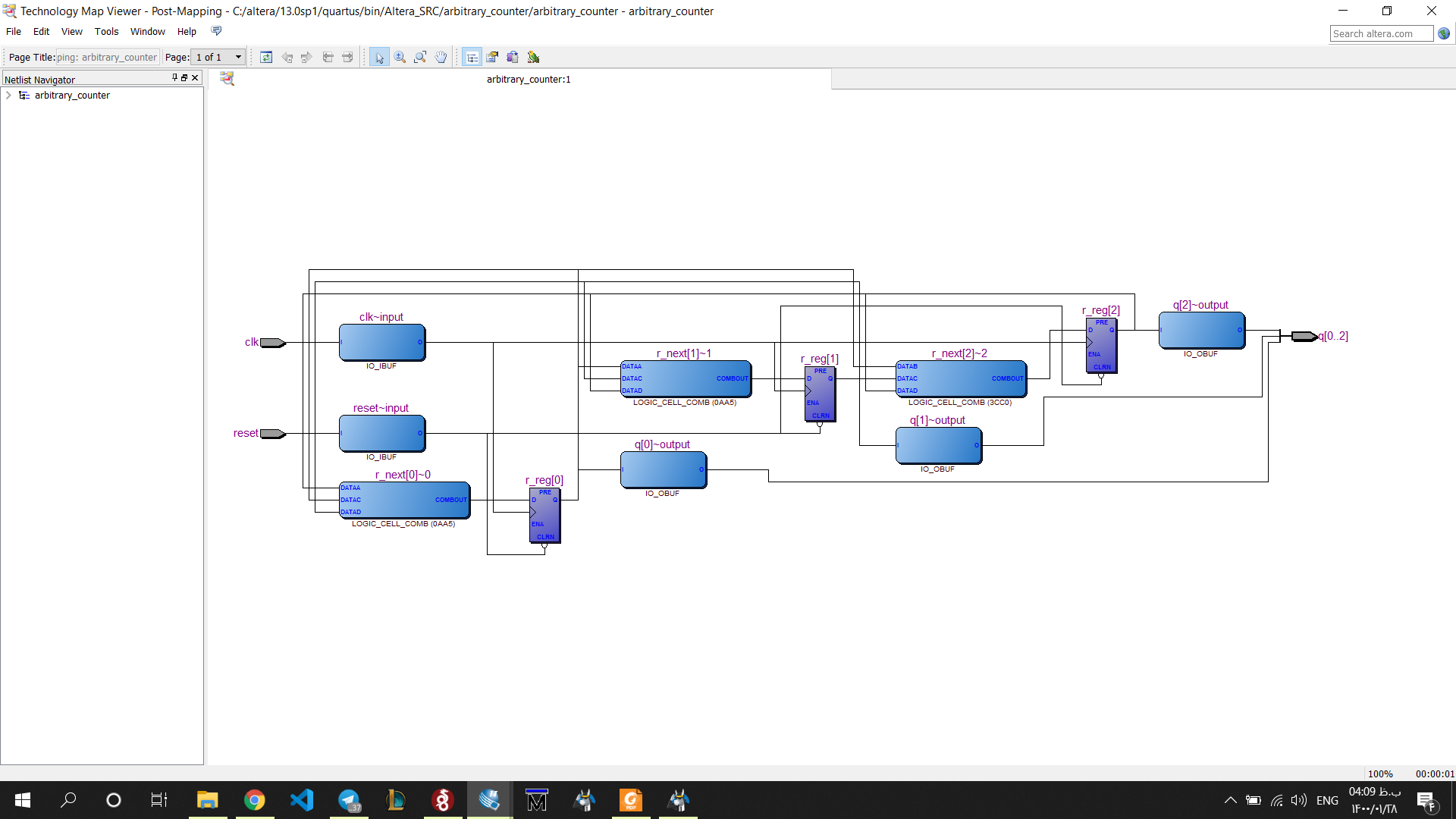
end arch;

## Compilation report

****

RTL****

## Post-Mapping

****

# Question4

## Code

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

-- use ieee.numeric\_std.all;

-- use IEEE.STD\_LOGIC\_UNSIGNED.all;

entity Fifo\_control is

port (

clk, reset : in std\_logic;

wr,rd : in std\_logic;

full, empty : out std\_logic;

w\_address,r\_address : out std\_logic\_vector (2 downto 0) );

end Fifo\_control;

architecture arch of Fifo\_control is

signal reader\_pointer : unsigned(3 downto 0) := "0000";

signal writer\_pointer : unsigned(3 downto 0) := "0000";

-- signal reader\_pointer : std\_logic\_vector(3 downto 0);

-- signal writer\_pointer : std\_logic\_vector(3 downto 0);

signal full\_fifo : std\_logic;

signal empty\_fifo : std\_logic;

begin

process(wr,clk,reset)

begin

if(reset = '1') then

-- reader\_pointer <= "0000";

writer\_pointer <= "0000";

full\_fifo <= '0';

-- empty\_fifo <= '0';

elsif(clk'event and clk ='1') then

if(wr = '1')then

if((reader\_pointer(2 downto 0) = writer\_pointer(2 downto 0)) and (reader\_pointer(3) /= writer\_pointer(3)))then

full\_fifo <= '1';

elsif(writer\_pointer = "1111") then

writer\_pointer <= "0000";

else

writer\_pointer <= writer\_pointer + 1;

end if;

if(full\_fifo = '1') then

full <= '1';

else

-- i don't know what todod

-- maybe send w\_address

w\_address <= std\_logic\_vector(writer\_pointer(2 downto 0));

-- w\_address <= writer\_pointer(2 downto 0);

end if;

end if;

end if;

end process;

process(rd,clk,reset)

begin

if(reset = '1') then

reader\_pointer <= "0000";

-- writer\_pointer <= "0000";

-- full\_fifo <= '0';

empty\_fifo <= '0';

elsif(clk'event and clk ='1') then

if(rd = '1') then

if(reader\_pointer = writer\_pointer) then

empty\_fifo <= '1';

elsif(reader\_pointer = "1111") then

reader\_pointer <= "0000";

else

reader\_pointer <= reader\_pointer + 1;

end if;

if(empty\_fifo = '1') then

empty <= '1';

else

-- i don't know what todod

-- maybe send r\_address

r\_address <= std\_logic\_vector(reader\_pointer(2 downto 0));

-- r\_address <= reader\_pointer(2 downto 0);

end if;

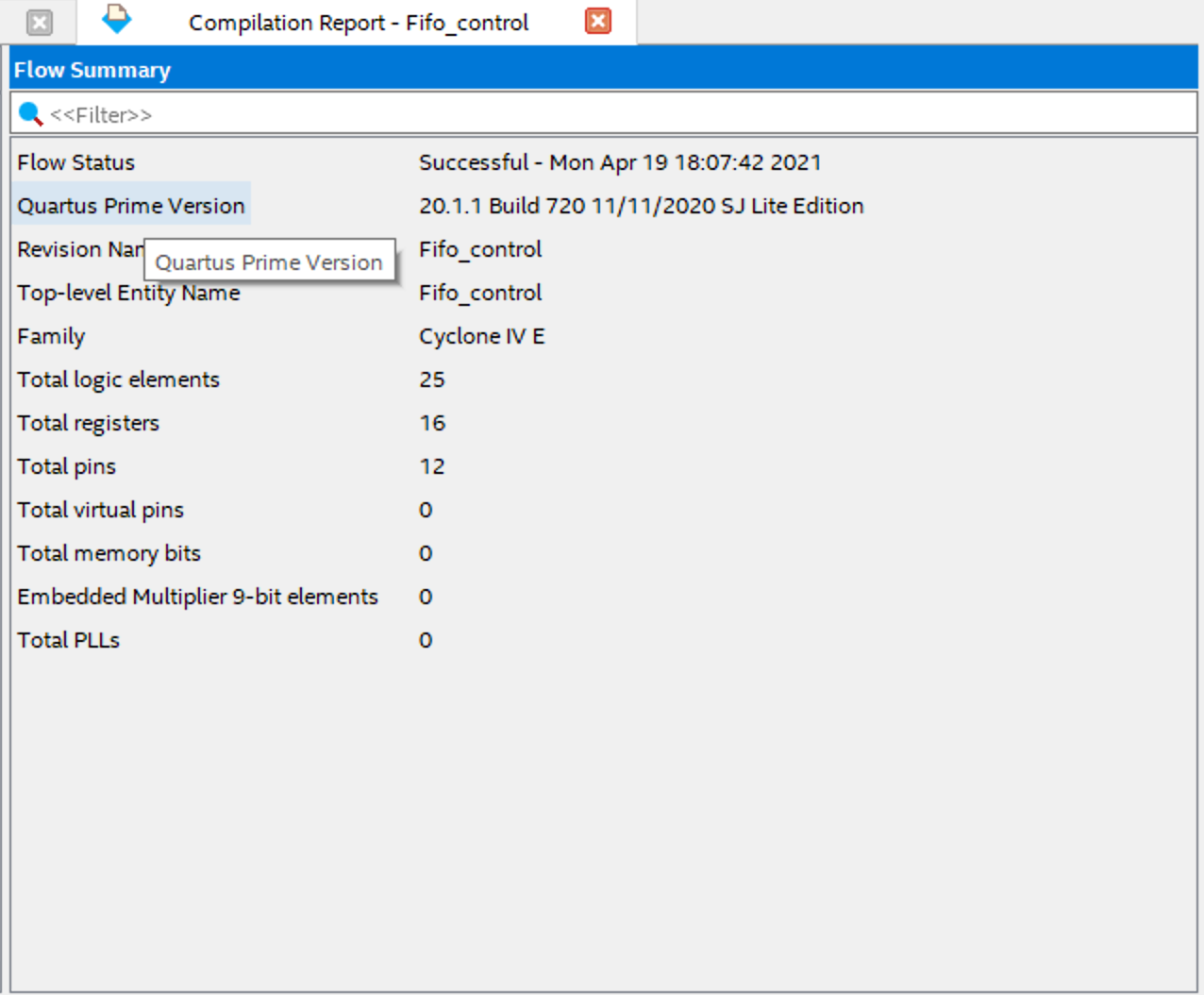
end if;

end if;

end process;

end architecture;

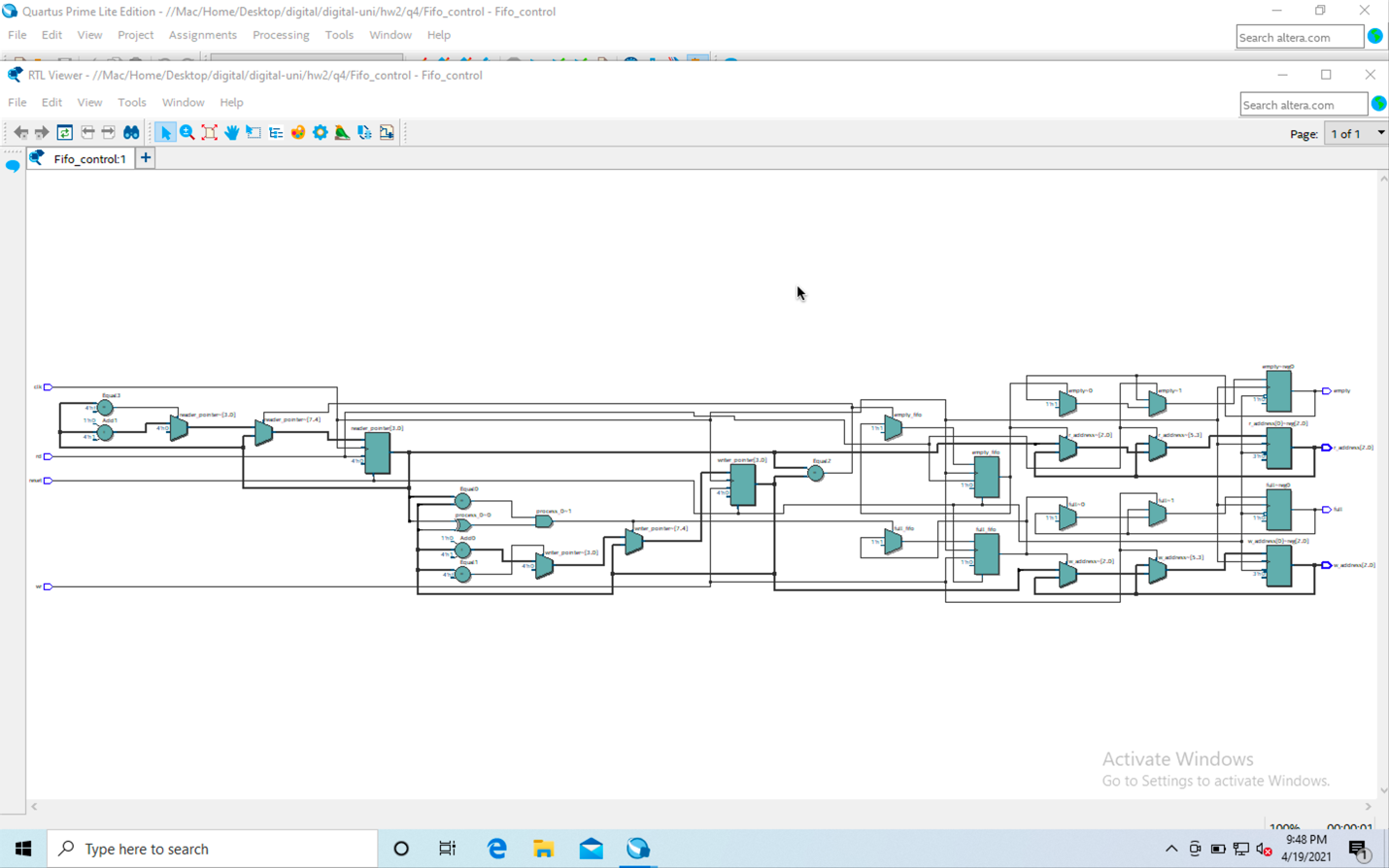
## Compilation report

****

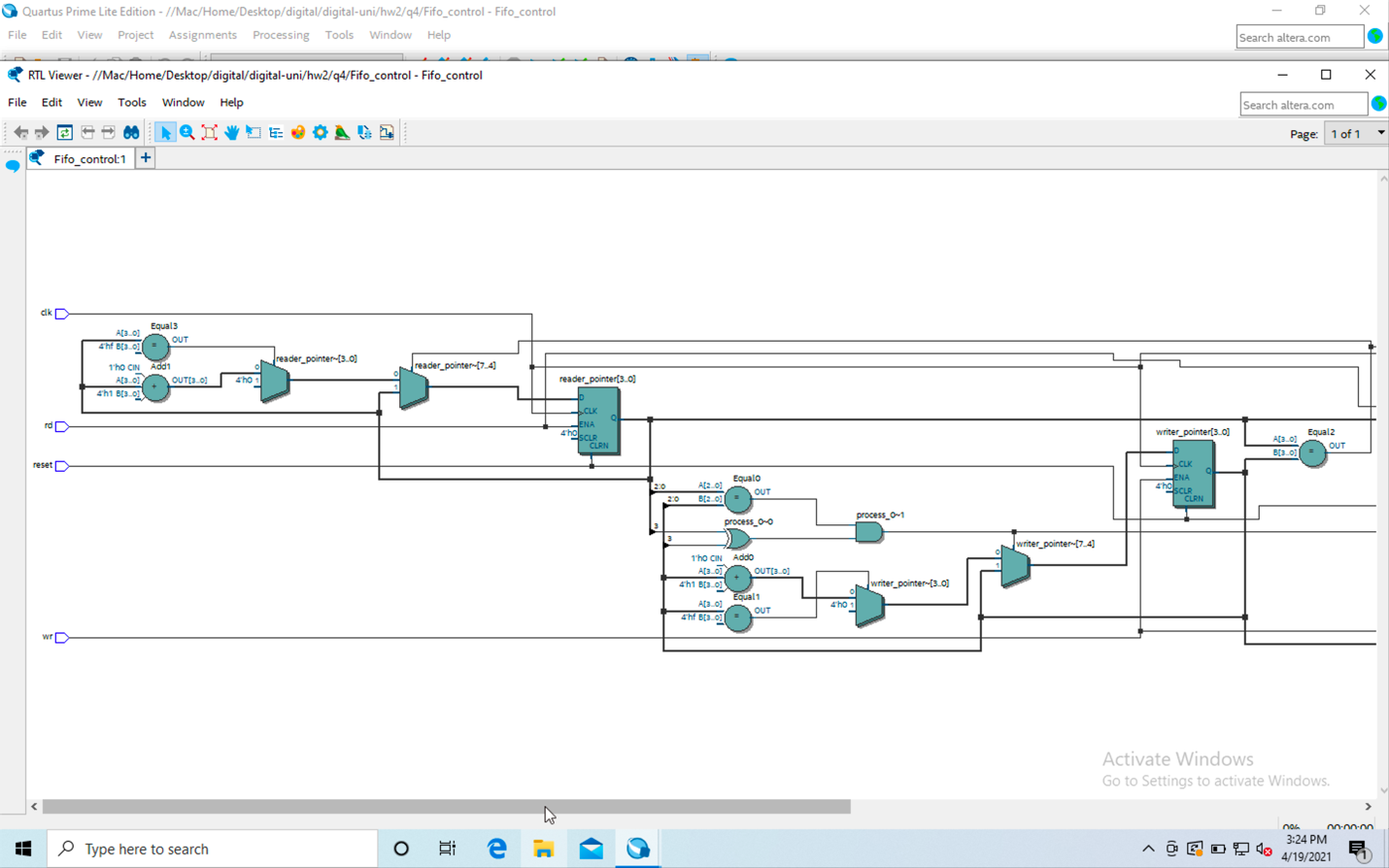
## RTL

**این عکس به دلیل سخت خواندن به۲ قسمت تقسیم شده اند**

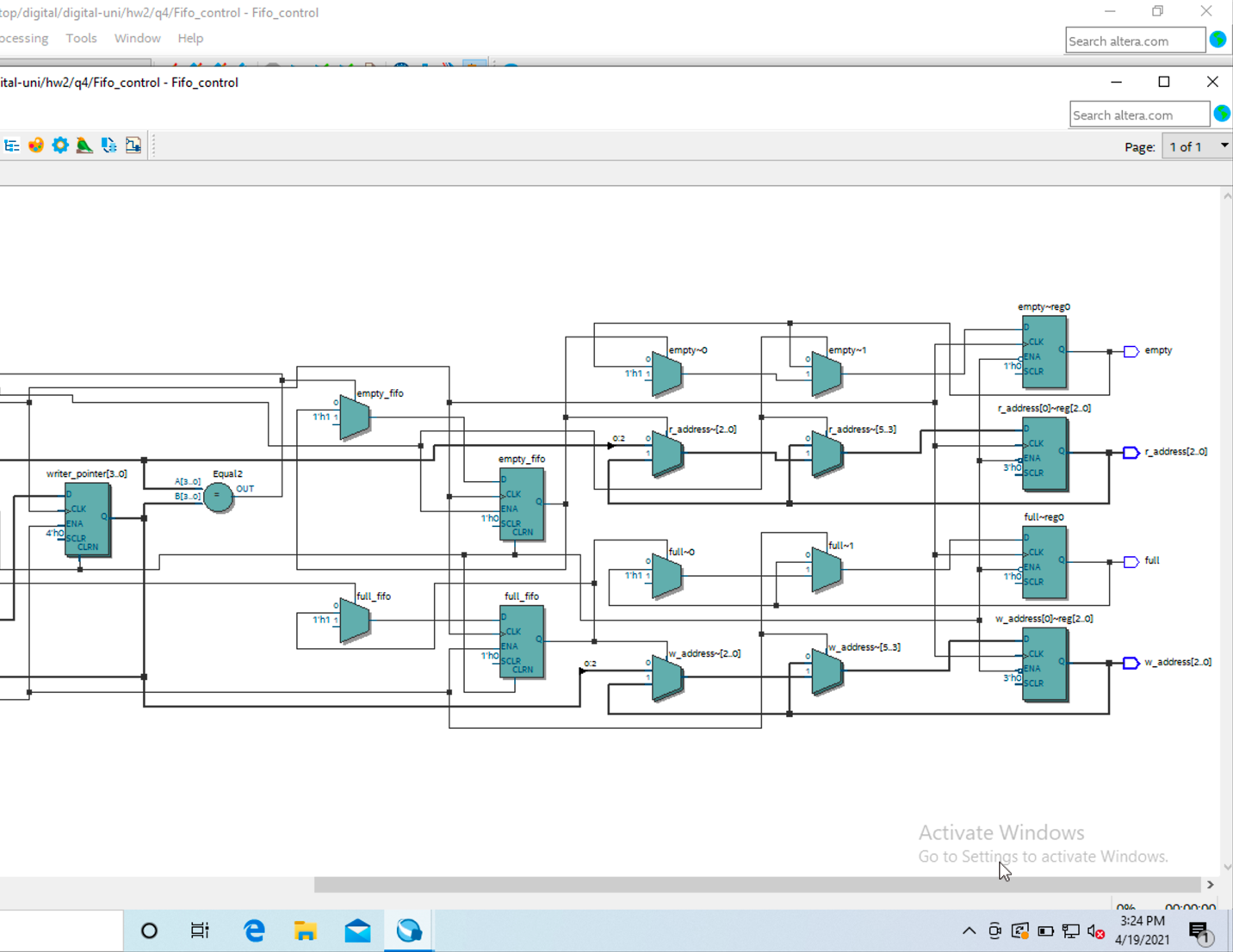
**عکس اصلی**

****

**عکس اول**

****

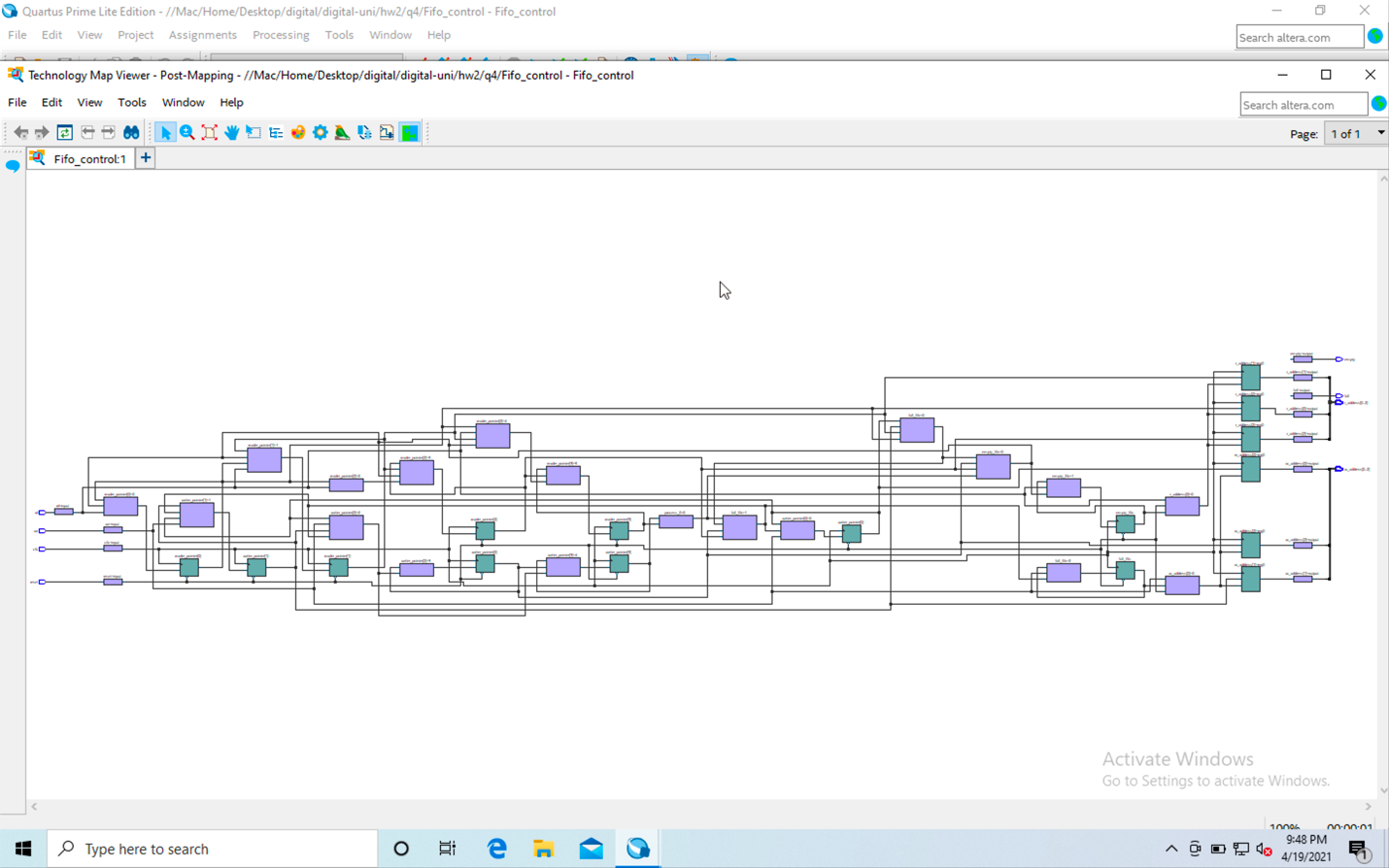
**عکس دوم**

****

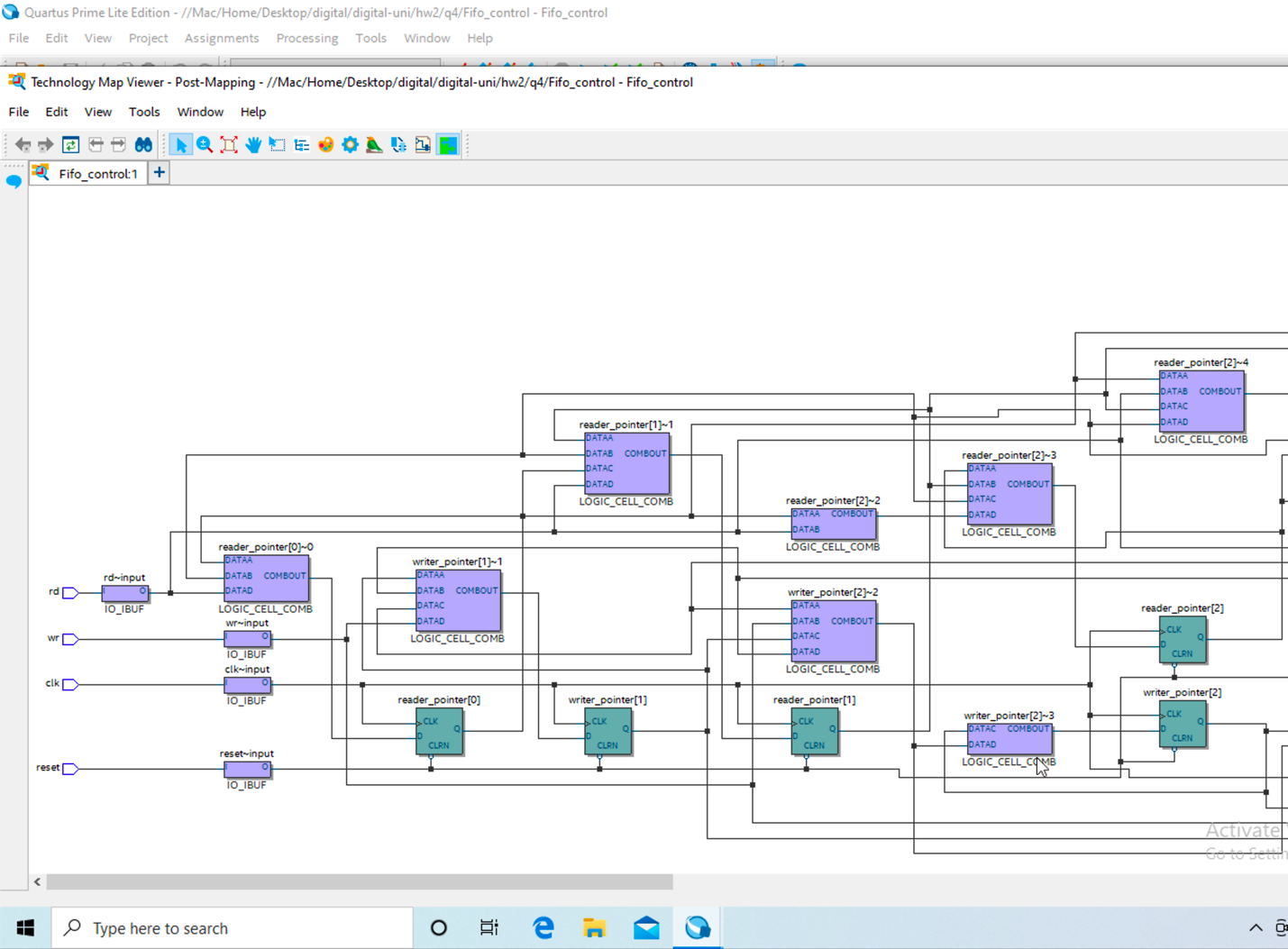
## Post-mapping

**این عکس به دلیل سخت خواندن به ۳ قسمت تقسیم شده اند**

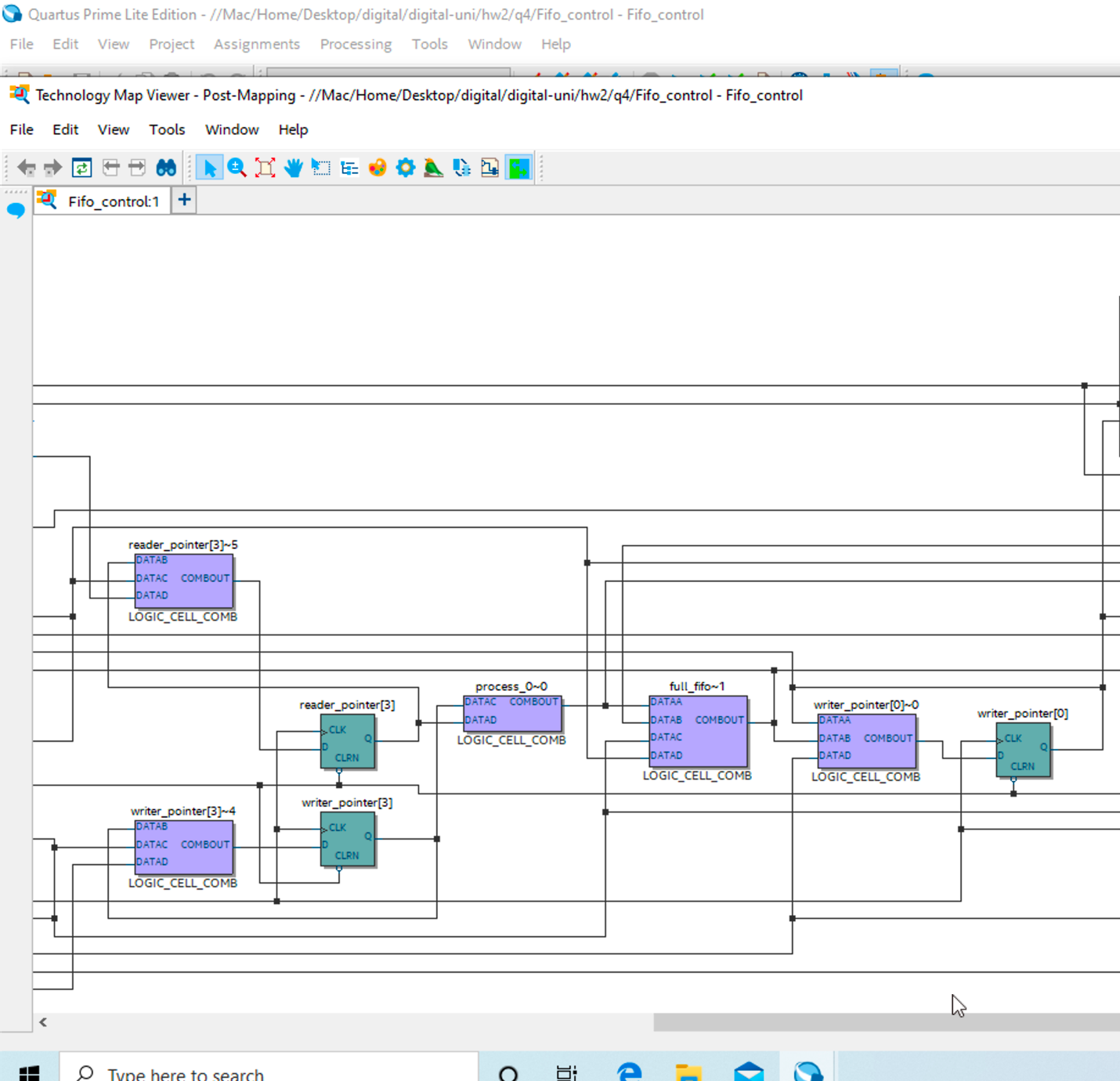
**عکس اصلی**

****

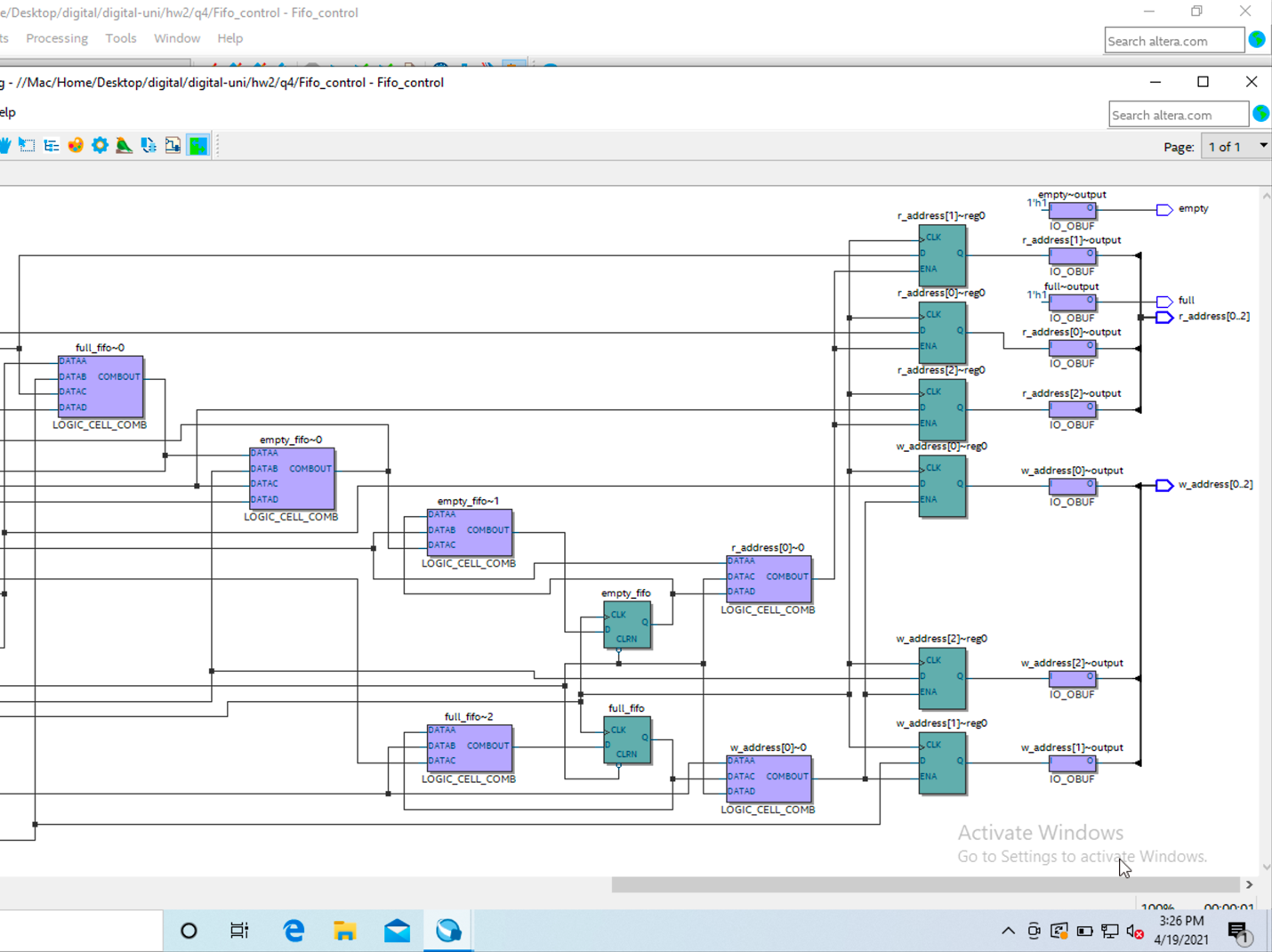
**عکس اول**

****

**عکس دوم**

****

**عکس سوم**

****