

طراحی سیستم های دیجیتال
تکلیف دوم

استاد مروستی

علیرضا نعمتی

کسری رشیدفر

Table of Contents

Question1.....	2
JK_FF.....	2
Code	3
Compilation report	4
RTL.....	4
Post-Mapping.....	5
T_FF	6
Code	6
Compilation report	7
Rtl	7
Post-Mapping.....	8
Question2.....	8
Code	9

Compilation report	10
RTL.....	10
Post-Mapping.....	11
Question3.....	11
Code	11
Compilation report	12
RTL.....	13
Post-Mapping.....	13
Question4.....	14
Code	14
Compilation report	16
RTL.....	17
Post-mapping	20

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;

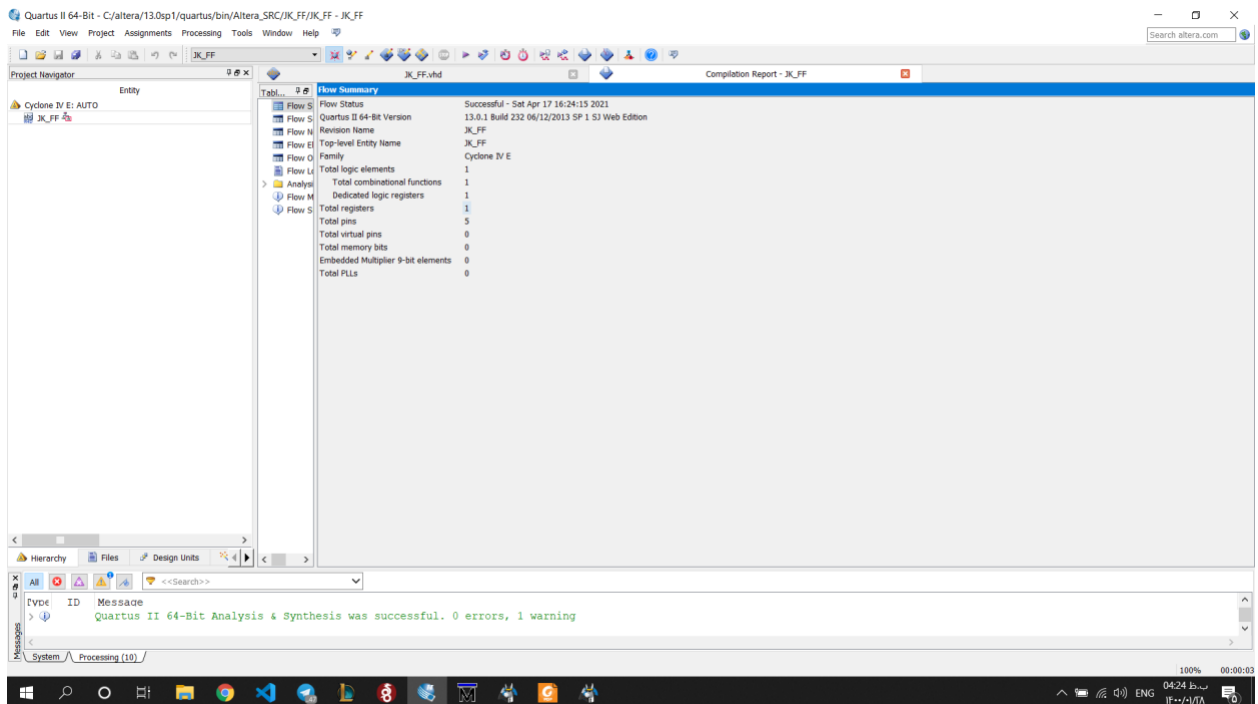
end process;

q <= q_temp;

end arch;

```

Compilation report



Quartus II 64-Bit - C:/altera/13.0sp1/quartus/bin/Altera_SRC/JK_FF/JK_FF - JK_FF

File Edit View Project Assignments Processing Tools Window Help

Project Navigator: JK_FF.vhd

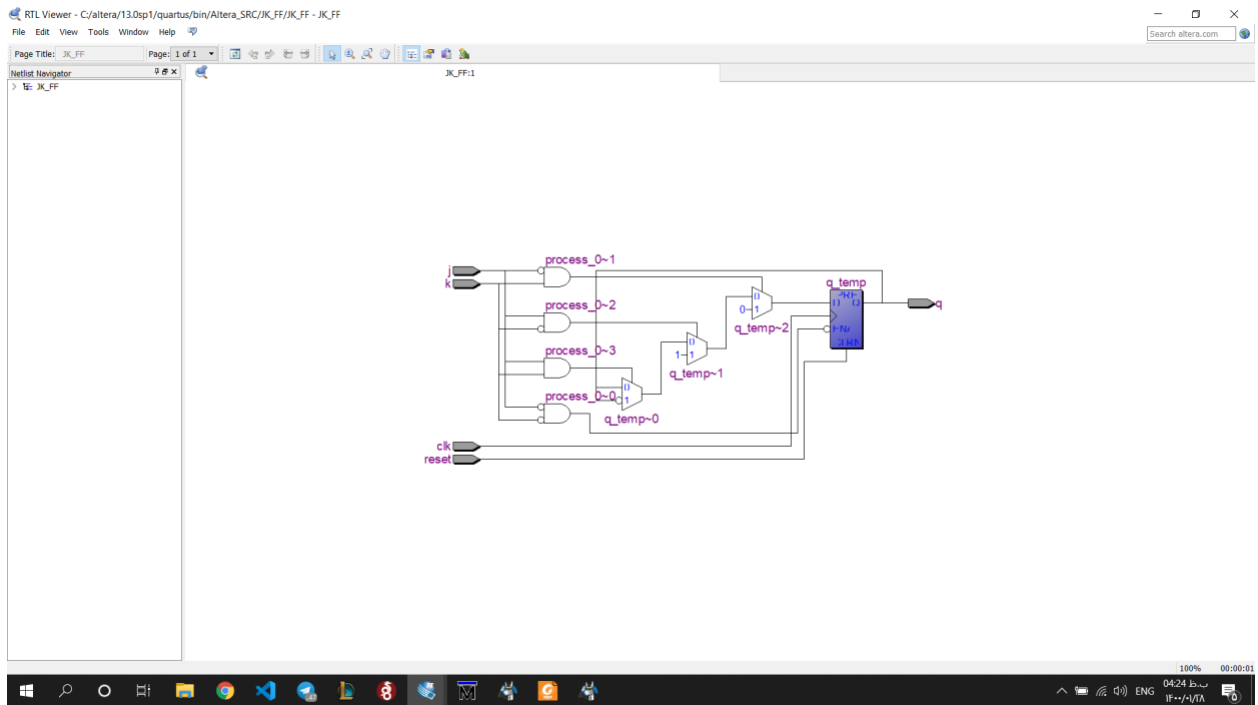
Compilation Report - JK_FF

Flow	Summary
Flow S	Flow Status: Successful - Sat Apr 17 18:24:15 2021
Flow S	Quartus II 64-Bit Version: 13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Flow N	Revision Name: JK_FF
Flow E	Top-level Entity Name: JK_FF
Flow Q	Family: Cyclone IV E
Flow L	Total logic elements: 1
Flow L	Total combinational functions: 1
Flow M	Dedicated logic registers: 1
Flow S	Total registers: 1
Flow S	Total pins: 5
Flow S	Total virtual pins: 0
Flow S	Total memory bits: 0
Flow S	Embedded Multiplier 9-bit elements: 0
Flow S	Total PLLs: 0

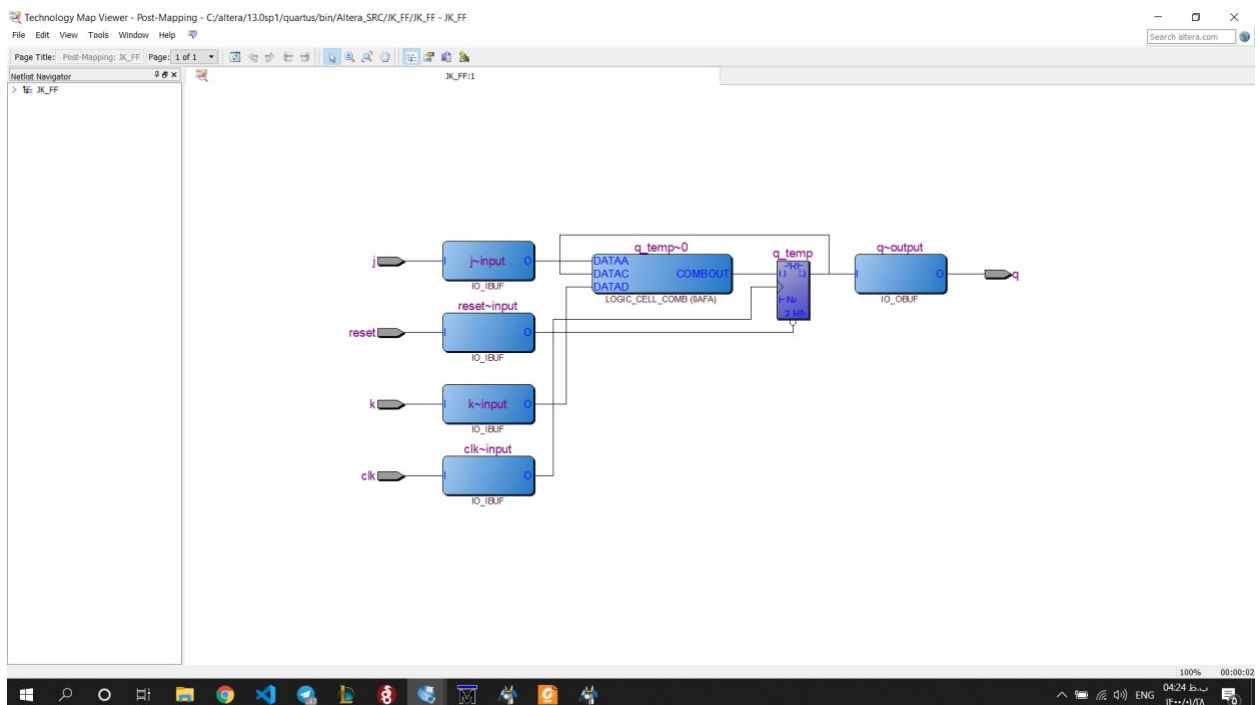
Messages: Quartus II 64-Bit Analysis & Synthesis was successful. 0 errors, 1 warning

System / Processing (10) / 100% 00:00:03

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

Quartus II 64-Bit - C:/altera/13.0sp1/quartus/bin/Altera_SRC/T_FF/T_FF - T_FF

File Edit View Project Assignments Processing Tools Window Help

T_FF

Project Navigator

Entity

Cyclone IV E: AUTO

T_FF

Compilation Report - T_FF

Table 1: Summary

Flow	Status	Message
Flow S	Successful	Set Apr 17 16:21:13 2021
Flow S	Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Flow N	Revision Name	T_FF
Flow E	Top-level Entity Name	T_FF
Flow Q	Family	Cyclone IV E
Flow L	Total logic elements	1
Flow M	Total combinational functions	1
Flow M	Dedicated logic registers	1
Flow S	Total registers	1
	Total pins	4
	Total virtual pins	0
	Total memory bits	0
	Embedded Multiplier 9-bit elements	0
	Total PLLs	0

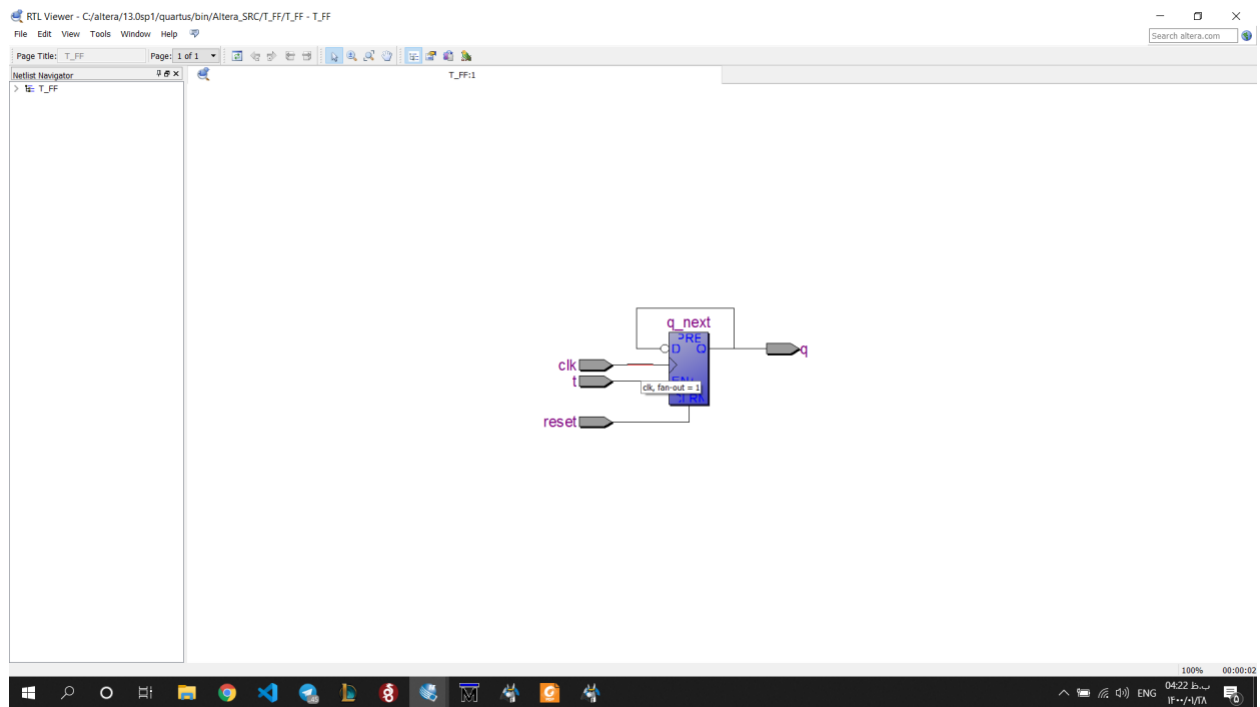
Messages

System / Processing (11)

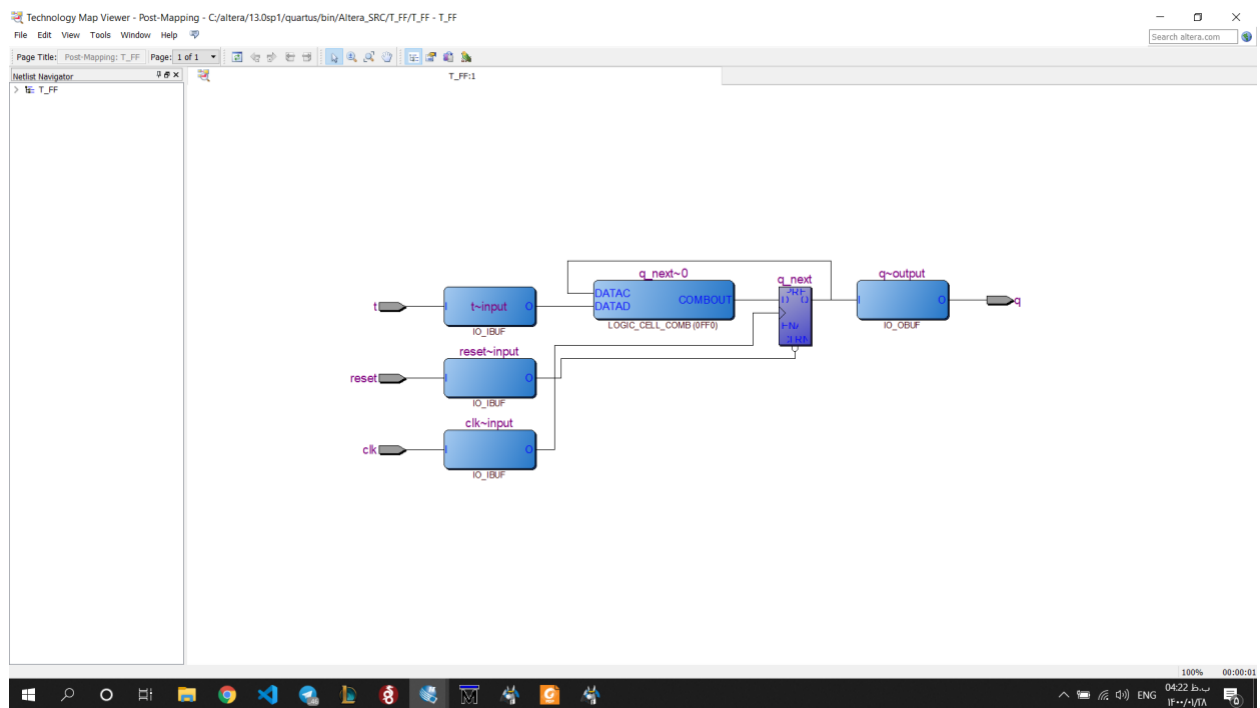
Quartus II 64-Bit Analysis & Synthesis was successful. 0 errors, 2 warnings

100% 00:00:03

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;
```

Quartus II 64-Bit - C:\altera\13.0sp1\quartus\bin\altera\BCD_Counter\BCD_Counter - BCD_Counter

File Edit View Project Assignments Processing Tools Window Help

Search altera.com

BCD_Counter

Project Navigator

Entity

Cyclone IV E: AUTO

BCD_Counter

BCD_Counter.vhd

Compilation Report - BCD_Counter

Flow summary

Tab	Flow	Status	Time
Flow S	Flow Status	Successful	Set Apr 17 16:27:05 2021
Flow S	Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 S3 Web Edition	
Flow N	Revision Name	BCD_Counter	
Flow E	Top-level Entity Name	BCD_Counter	
Flow O	Family	Cyclone IV E	
Flow L	Total logic elements	5	
Analysis	Total combinational functions	5	
Flow M	Dedicated logic registers	4	
Flow S	Total registers	4	
	Total pins	7	
	Total virtual pins	0	
	Total memory bits	0	
	Embedded Multiplier 9-bit elements	0	
	Total PLLs	0	

Hierarchy Files Design Units

Messages

System Processing (10)

Quartus II 64-Bit Analysis & Synthesis was successful. 0 errors, 1 warning

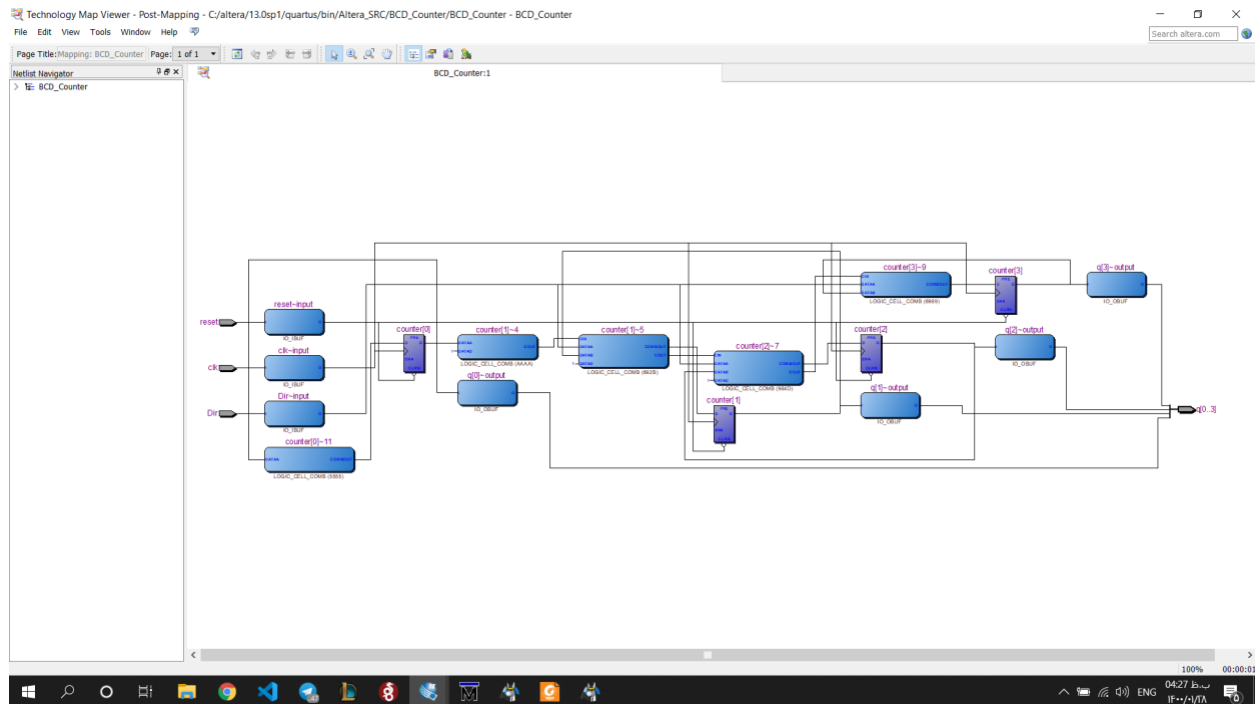
100% 00:00:04

The screenshot displays the RTL Viewer in Quartus II, showing the logic diagram of a BCD_Counter. The diagram includes the following components and connections:

- Inputs:** `Dir` (direction control), `clk` (clock), and `reset`.
- Counters:** A 4-bit counter labeled `counter[3:0]` is shown. Its `Q` output is connected to the `q[3:0]` output signal.
- Adders:** Two adders, `Add0` and `Add1`, are present. `Add0` takes `Dir` and `counter[3:0]` as inputs. `Add1` takes `counter[3:0]` and `AC0000` as inputs.
- MUX21:** A 2-to-1 multiplexer labeled `MUX21` selects between the outputs of `Add0` and `Add1` based on the `Dir` input. Its output is connected to the `Q` input of the counter.
- Reset:** The `reset` input is connected to the `RESET` pin of the counter.

The Netlist Navigator on the left shows the hierarchy: `16: BCD_Counter`. The top status bar indicates the page title is `BCD_Counter` and the page is 1 of 1. The bottom status bar shows the design is 100% complete and the time is 04:27.

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;

```

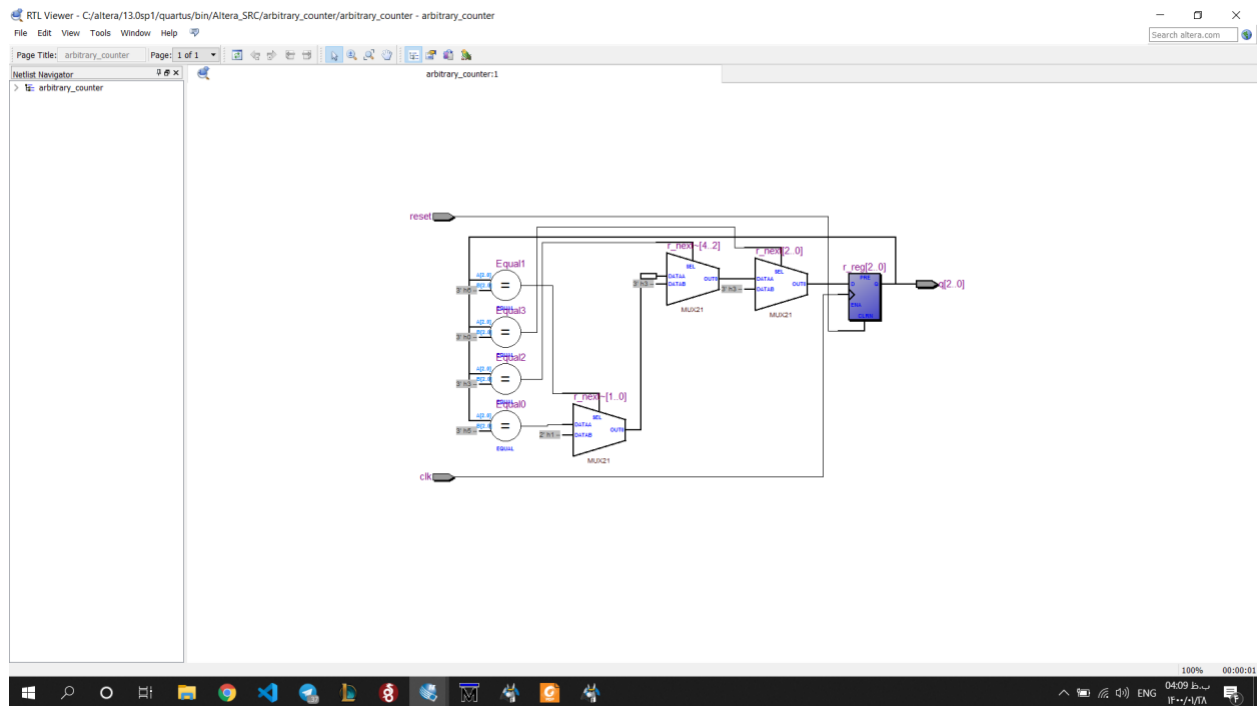
Compilation report

The screenshot shows the Quartus II 64-bit Netlist Viewers Preprocess window. The title bar indicates the path: C:\altera\13.0sp1\quartus\bin\Altera_SRC\arbitrary_counter\arbitrary_counter. The window displays a 'New Summary' table with the following data:

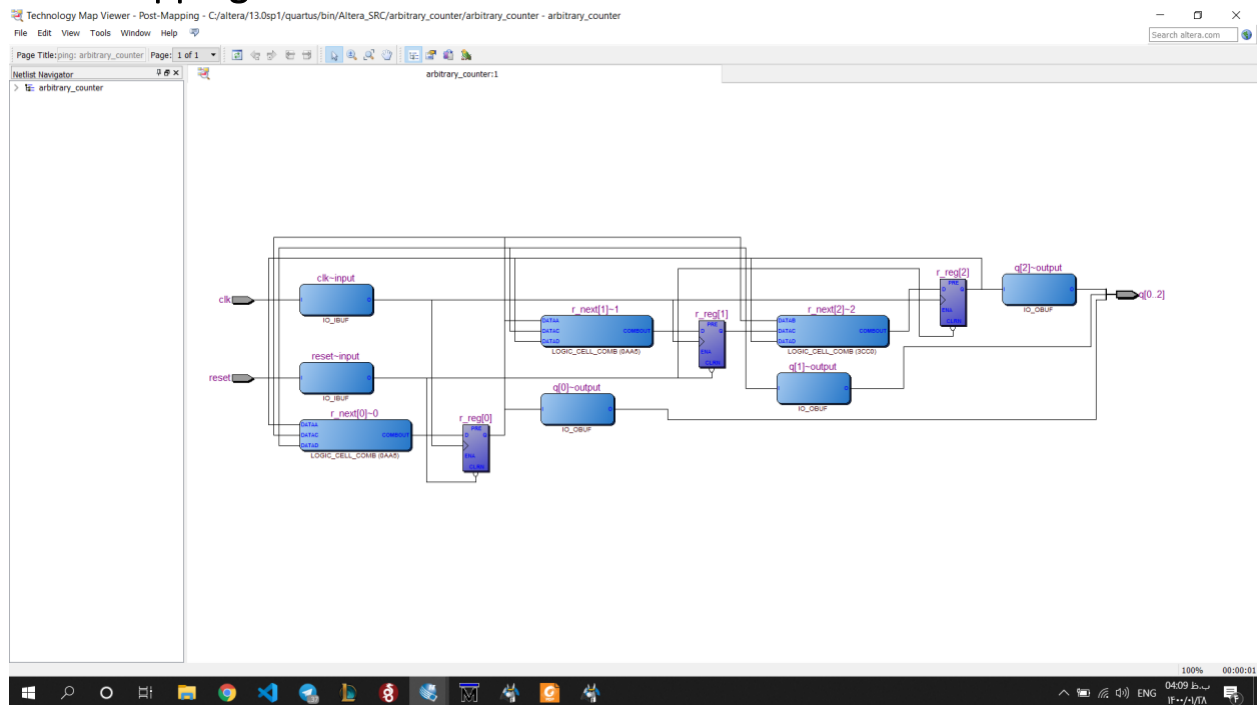
Flow	Status	Successful - Sat Apr 17 18:07:08 2021
Flow S	Quartus II 64-bit Version	13.0.1 Build 232 06/12/2013 SP 1 S1 Web Edition
Flow H	Revision Name	arbitrary_counter
Flow E	Top-level Entity Name	arbitrary_counter
Flow Q	Family	Cyclone IV E
Flow L	Total logic elements	3
Flow L	Total combinational functions	3
Flow M	Dedicated logic registers	3
Flow S	Total registers	3
	Total pins	5
	Total virtual pins	0
	Total memory bits	0
	Embedded Multiplier 9-bit elements	0
	Total PLLs	0

At the bottom of the window, a message box states: "Quartus II 64-bit Netlist Viewers Preprocess was successful. 0 errors, 0 warnings". The status bar at the bottom indicates "System / Processing (18) / 100% 00:00:01".

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;
        end if;
    end if;
end process;

```

```
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;
```

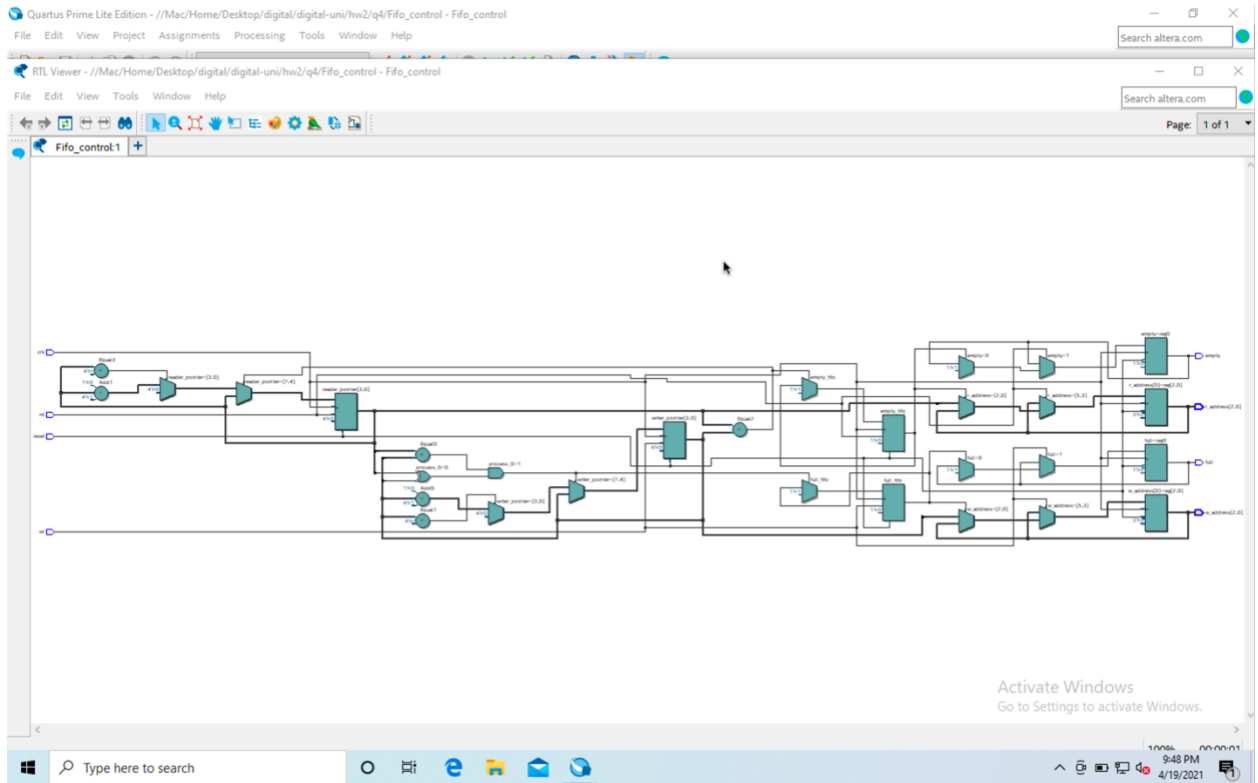
```
end architecture;
```

Compilation report

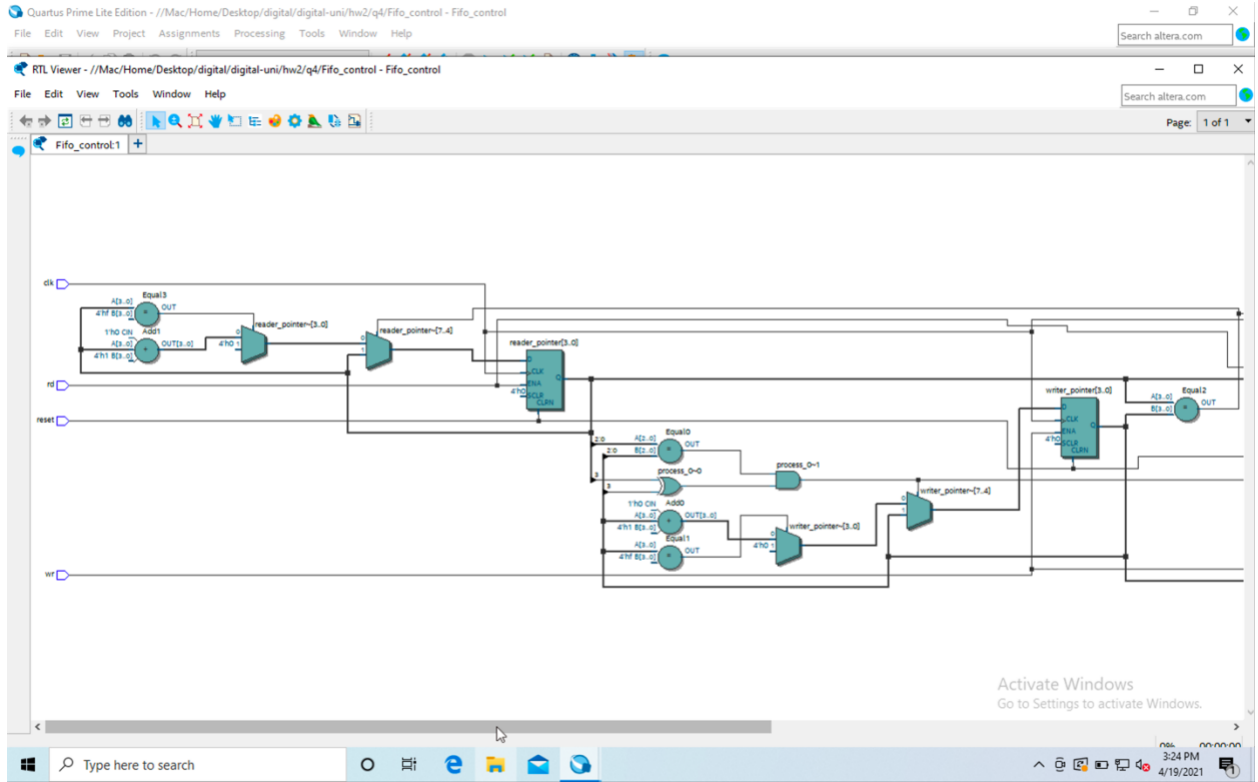
Compilation Report - Fifo_control	
Flow Summary	
<<Filter>>	
Flow Status	Successful - Mon Apr 19 18:07:42 2021
Quartus Prime Version	20.1.1 Build 720 11/11/2020 SJ Lite Edition
Revision Name	Fifo_control
Top-level Entity Name	Fifo_control
Family	Cyclone IV E
Total logic elements	25
Total registers	16
Total pins	12
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0

RTL

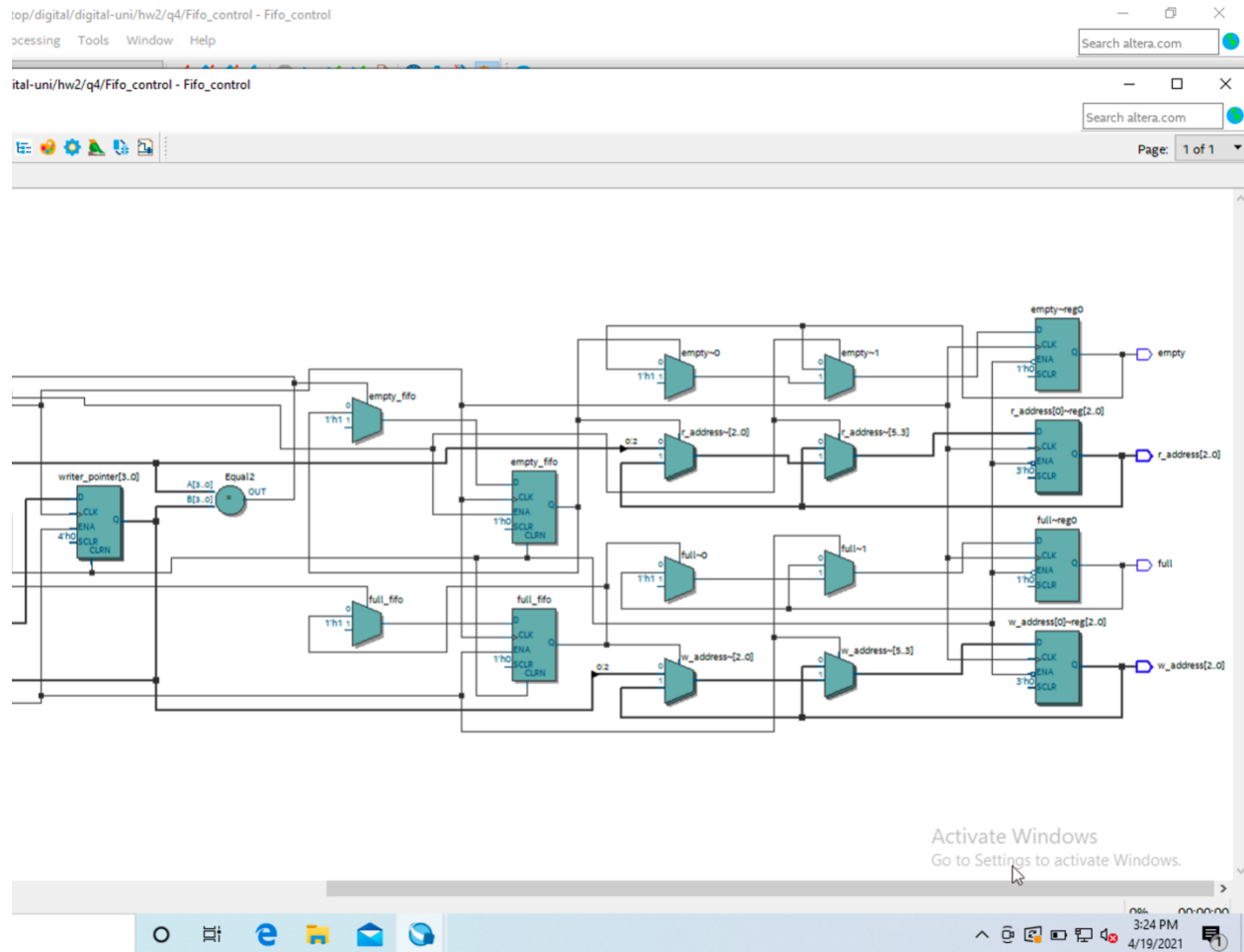
این عکس به دلیل سخت خواندن به ۲ قسمت تقسیم شده اند
عکس اصلی



عکس اول



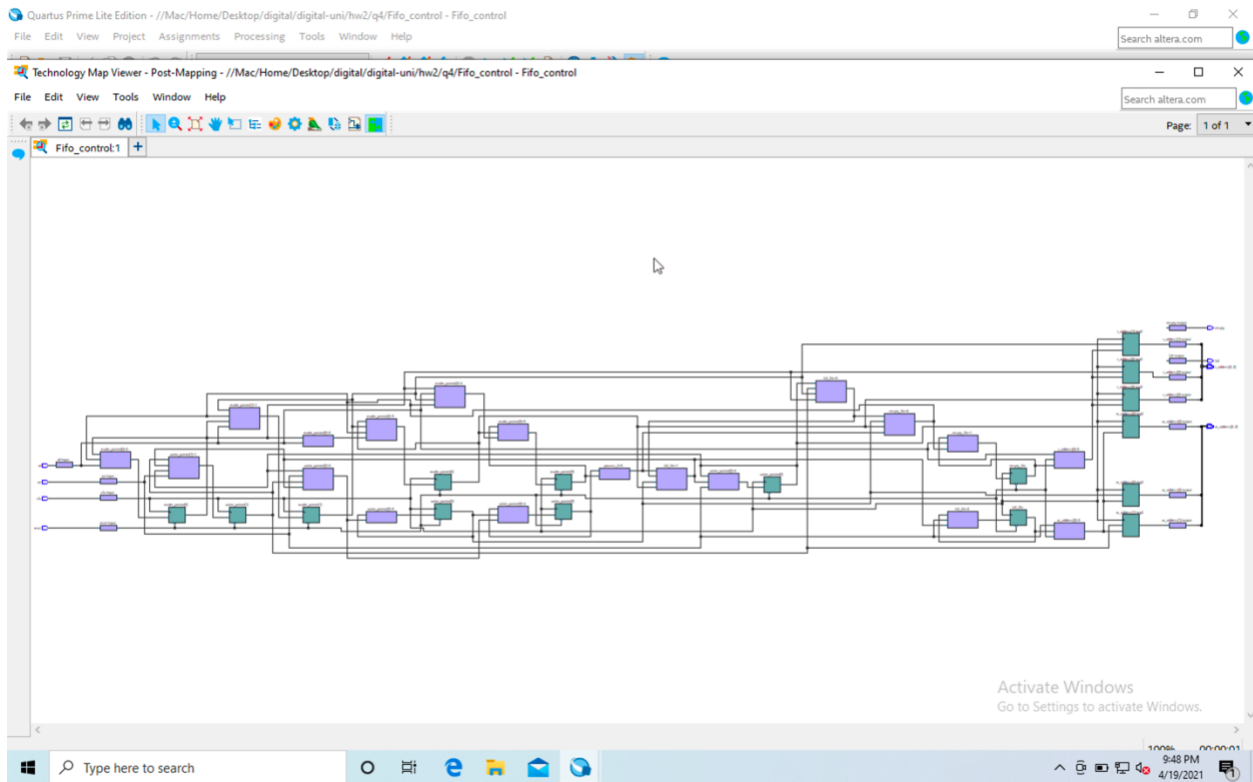
عکس دوم



Post-mapping

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

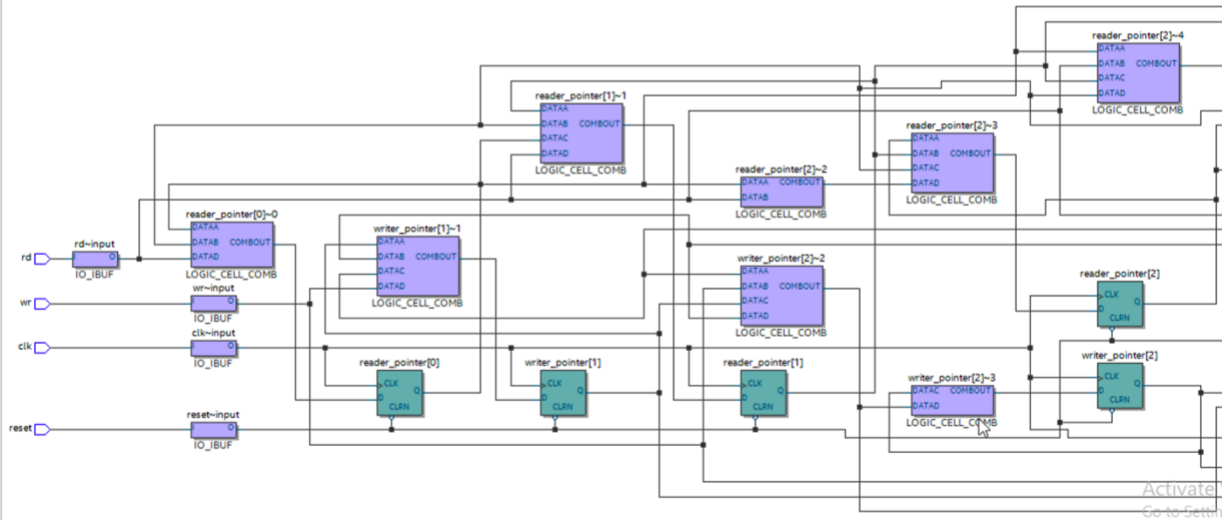
عکس اصلی



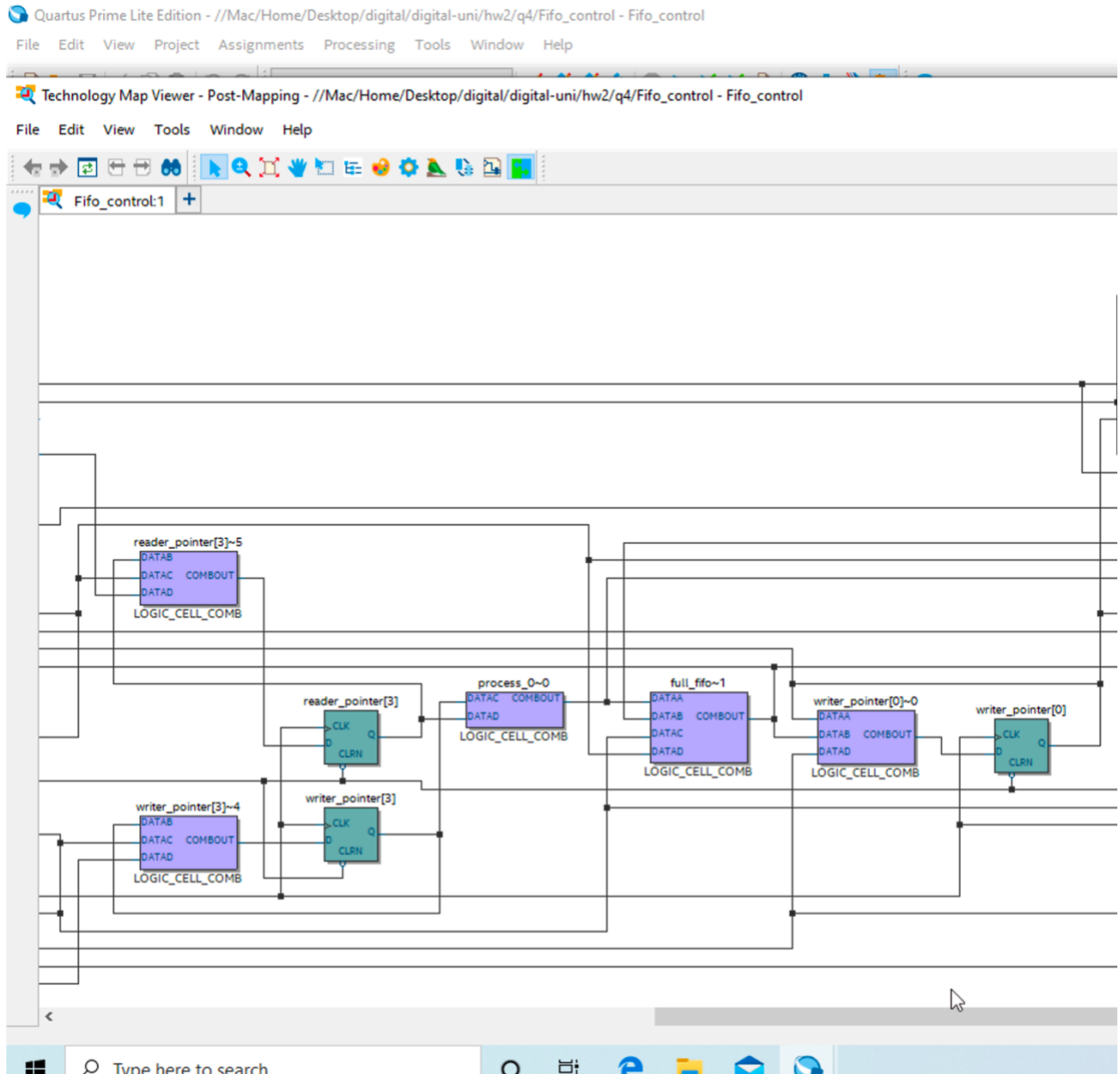
عکس اول



Fifo_control:1



Activate
Go to Settings



عکس سوم

