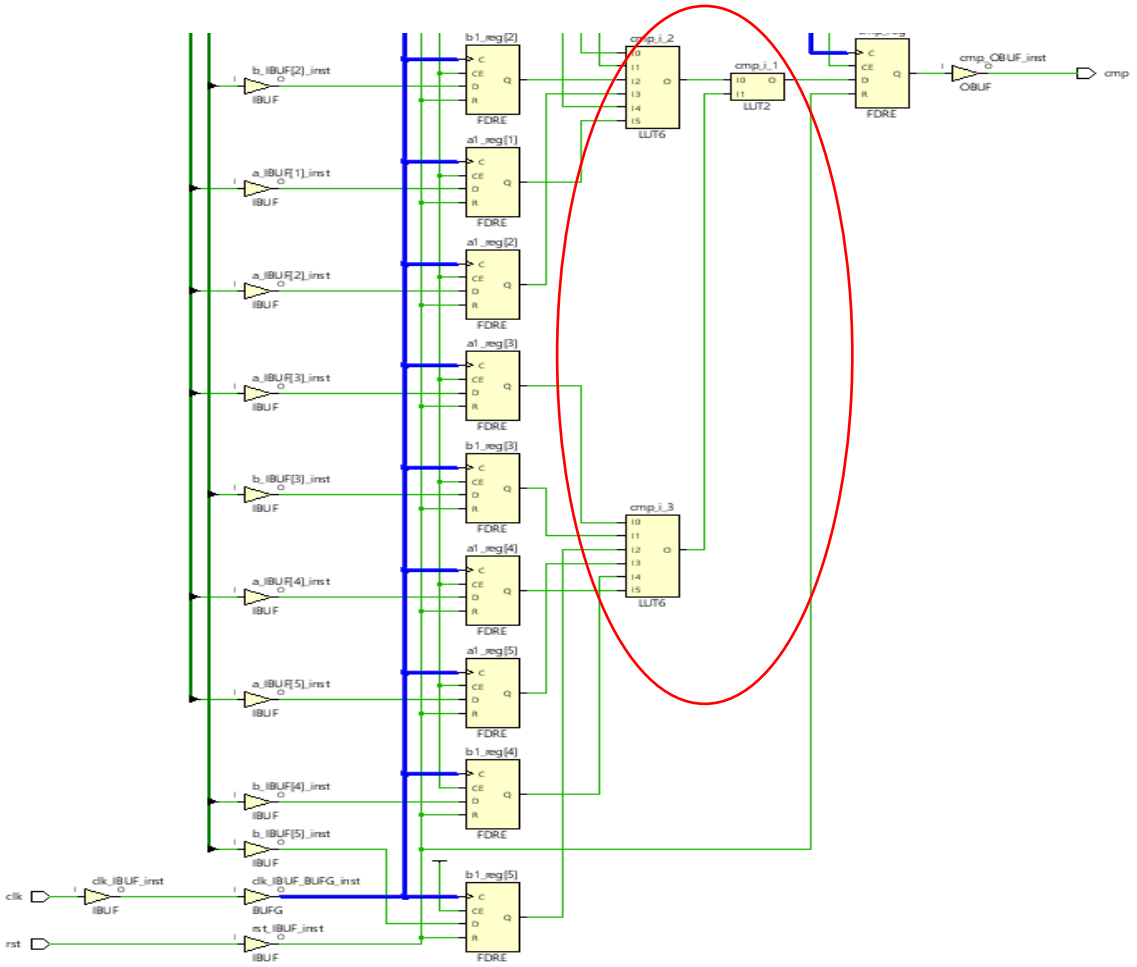
Pipeline

1. משווה בסיסי:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
use IEEE.NUMERIC_STD.ALL;

entity comparator_pipeline is
    Port ( a : in STD_LOGIC_VECTOR (5 downto 0);
           b : in STD_LOGIC_VECTOR (5 downto 0);
           clk : in STD_LOGIC;
           rst : in STD_LOGIC;
           cmp : out STD_LOGIC);
end comparator_pipeline;

architecture Behavioral of comparator_pipeline is
 signal a1 : STD_LOGIC_VECTOR (5 downto 0);
 signal b1 : STD_LOGIC_VECTOR (5 downto 0);


begin
process(clk)
begin
    if rising_edge(clk) then
        if rst = '1' then
            a1 <=(others=>'0');
           b1 <=(others=>'0');

        else
           a1 <= a ;
           b1 <=b ;

        end if;
    end if;
end process;

process(clk)
begin
    if rising_edge(clk) then
        if rst = '1' then
            cmp <= '0';
        elsif a1 = b1 then
            cmp <= '1';
        else
            cmp <= '0';
        end if;
    end if;
```

```
end process;

end Behavioral;
```

תוצאות סינתזה:



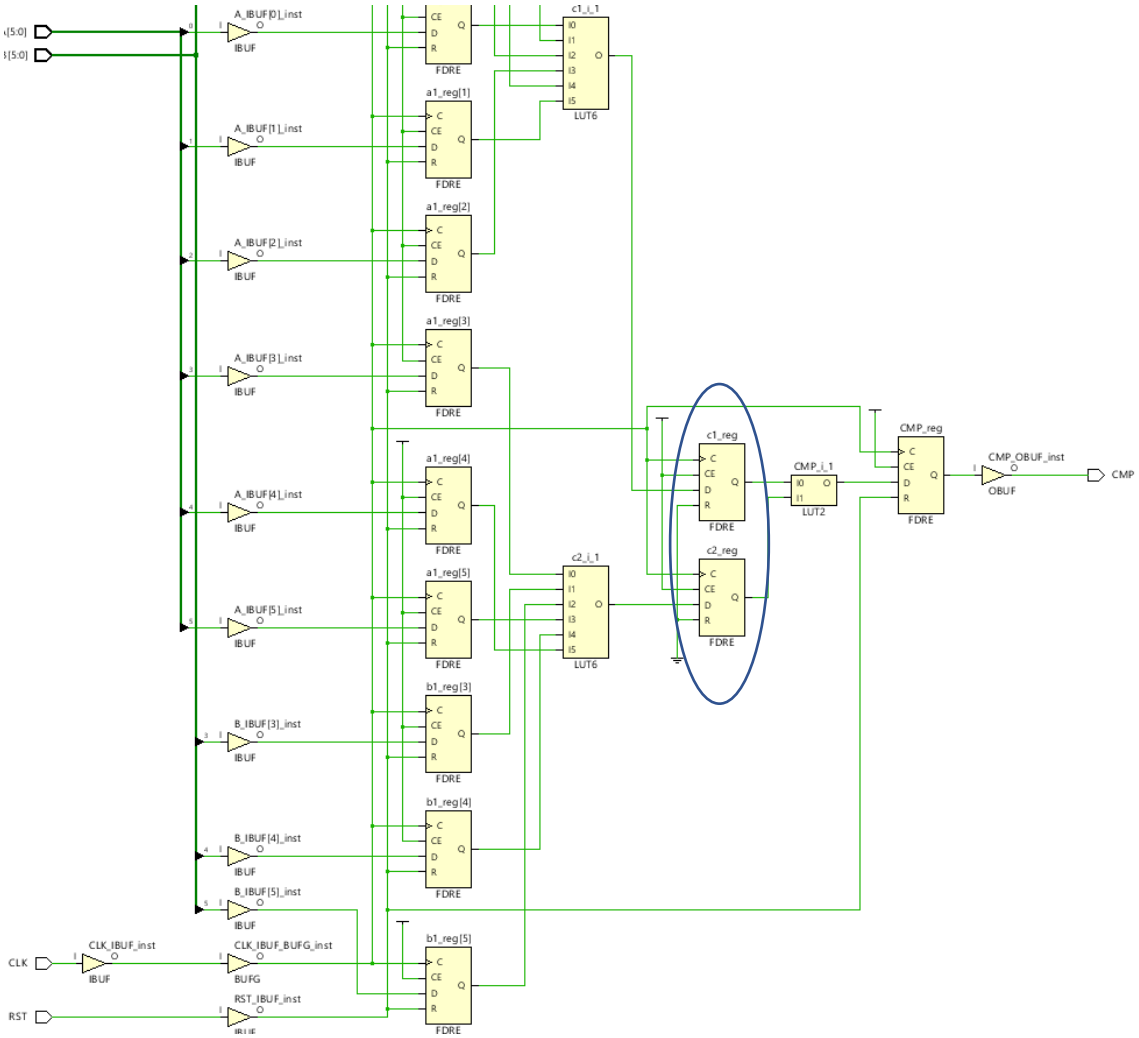
ניתן לראות שיש שתי רמות לוגיות בן FF.

2. הוספת דרגת PIPELINE:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
use IEEE.NUMERIC_STD.ALL;

entity comparator_pipeline is
    Port ( a : in STD_LOGIC_VECTOR (5 downto 0);
           b : in STD_LOGIC_VECTOR (5 downto 0);
           clk : in STD_LOGIC;
           rst : in STD_LOGIC;
           cmp : out STD_LOGIC);
end comparator_pipeline;

architecture Behavioral of comparator_pipeline is

signal c1 : std_logic;
signal c2 : std_logic;

signal a1 : STD_LOGIC_VECTOR (5 downto 0);
 signal b1 : STD_LOGIC_VECTOR (5 downto 0);

begin
process(clk)
begin
    if rising_edge(clk) then
        if rst = '1' then
            a1 <=(others=>'0');
           b1 <=(others=>'0');

        else
           a1 <= a ;
           b1 <=b ;

        end if;
    end if;
end process;

process(clk)
begin
    if rising_edge(clk) then
        if a1(2 downto 0) = b1(2 downto 0) then
            c1 <= '1';
        else
            c1 <= '0';
        end if;
    end if;
end process;
```

```
process(clk)
begin
    if rising_edge(clk) then
        if a1(5 downto 3) = b1(5 downto 3) then
            c2 <= '1';
        else
            c2 <= '0';
        end if;
    end if;
end process;

process(clk)
begin
    if rising_edge(clk) then
        if rst = '1' then
            cmp <= '0';
        elsif (c1 = '1' and c2 = '1') then
            cmp <= '1';
        else
            cmp <= '0';
        end if;
    end if;
end process;

end Behavioral;
```

תוצאת הסינתזה:



ניתן לראות את תוספת ה-FF באמצע המבנה כך שיש רמה לוגית בדידה .