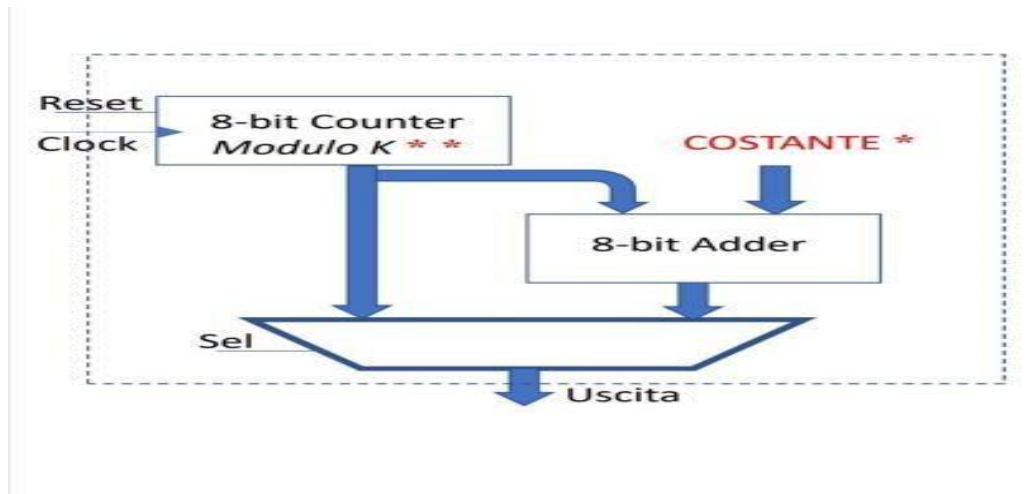


Nome: Mahmoud Mohamed Zaghloul Saad. Cognome: Mohamed

Matricola: 209397

è stato chiesto di descrivere in codice VHDL il comportamento del seguente circuito:



la richiesta è che il reset deve essere sincrono, per cui deve agire se solo se abbiamo l'evento di salita del clock, inoltre c'è una costante che dipende dalla mia matricola che determina il modulo del contatore, cioè il contatore arrivato a (256-costante) deve tornare a 0. La mia costante è pari a $2+0+9+3+9+7=30$ in binario a 8 bit "00011110", il modulo del contatore è quindi

$(K=256-COSTANTE=226)$ in binario "11100010". Il seguente circuito deve dare in uscita il conteggio del contatore se il numero del conteggio è pari oppure la somma tra il conteggio e la costante se il numero del conteggio è dispari.

Mux:

```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3 entity mux is
4     port(conta, somma: in std_logic_vector(7 downto 0);
5         sel: in std_logic;
6         out_mux: out std_logic_vector(7 downto 0));
7 end entity;
8 architecture Behavioral of mux is
9     signal X : std_logic_vector(7 downto 0);
10    begin
11        with sel select
12            out_mux <=  somma when '1',
13                       conta when '0',
14                       X  when others;
15    end architecture;
```

ADDER 8_BIT:

```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3 use IEEE.STD_LOGIC_UNSIGNED.ALL;
4 entity ADDER8_BIT is
5     port(k, count: in std_logic_vector(7 downto 0);
6           out_adder: out std_logic_vector(7 downto 0));
7 end ENTITY;
8 architecture Behavioral of ADDER8_BIT is
9     begin
10         out_adder <= k + count;
11     end architecture;
```

Ho usato l'operatore '+'.
Ho usato l'operatore '<='.

CONTATORE A 8 BIT:

```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3 use IEEE.STD_LOGIC_UNSIGNED.ALL;
4 entity CONTATORE_8_BIT is
5     port (Count_out : out std_logic_vector (0 to 7);
6           clk : in std_logic;
7           reset : in std_logic);
8 end entity;
9 architecture Behavioral of CONTATORE_8_BIT is
10     signal Qcount : std_logic_vector (0 to 7) := "00000000";
11     begin
12         process (clk)
13         begin
14             if(rising_edge(clk)) then
15                 if reset = '1' then
16                     Qcount <= "00000000";
17                 elsif Qcount = "11100010" then
18                     Qcount<= "00000000";
19                 else Qcount <= Qcount + 1;
20             end if;
21         end if;
22     end process;
23     Count_out <= Qcount;
24 end architecture;
```

Ora manca solo di collegare tutti i circuiti ovvero fare i collegamenti interni per ottenere il circuito finale.

Nome: Mahmoud Mohamed Zaghloul Saad. Cognome: Mohamed

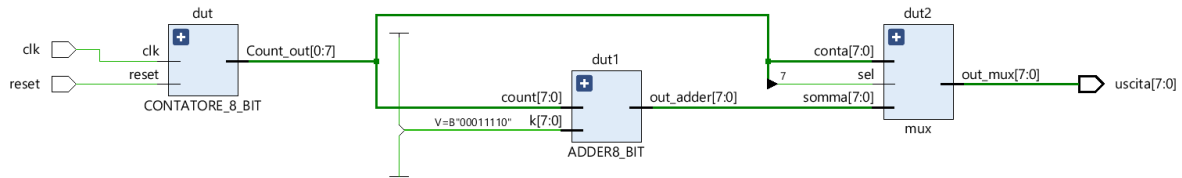
Matricola: 209397

CIRCUITO_FINALE:

```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3
4 entity CIRCUITO_FINALE is
5 port( reset, clk: in std_logic;
6       uscita: out std_logic_vector(7 downto 0));
7 end entity;
8 architecture Behavioral of CIRCUITO_FINALE is
9     signal costante: std_logic_vector(7 downto 0):="00011110";
10    signal count: std_logic_vector(7 downto 0);
11    signal out_adder: std_logic_vector(7 downto 0);
12 component CONTATORE_8_BIT is
13     port (Count_out : out std_logic_vector(0 to 7);
14           clk : in std_logic;
15           reset : in std_logic);
16 end component;
17 component ADDER8_BIT is
18     port(k, count: in std_logic_vector(7 downto 0);
19          out_adder: out std_logic_vector(7 downto 0));
20 end component;
21
22 component mux is
23     port(conta, somma: in std_logic_vector(7 downto 0);
24           sel: in std_logic;
25           out_mux: out std_logic_vector(7 downto 0));
26 end component;
27 begin
28     dut: CONTATORE_8_BIT port map (count(7 downto 0),clk,reset);
29     dut1: ADDER8_BIT port map (costante(7 downto 0),count(7 downto 0), out_adder(7 downto 0));
30     dut2: mux port map (count(7 downto 0),out_adder(7 downto 0),count(0), uscita(7 downto 0));
31 end Behavioral;
```

il bit meno significativo dell'uscita del contatore indica la parità del numero ; se il bit meno significativo è 0 allora il numero è pari altrimenti se il bit è 1 allora il numero è dispari.

Come prima verifica tengo conto dello schema seguente :



Ora per la prova devo scrivere il TEST_BENCH

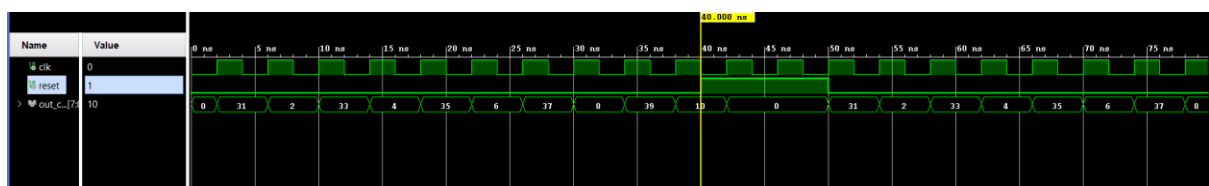
Nome: Mahmoud Mohamed Zaghloul Saad. Cognome: Mohamed

Matricola: 209397

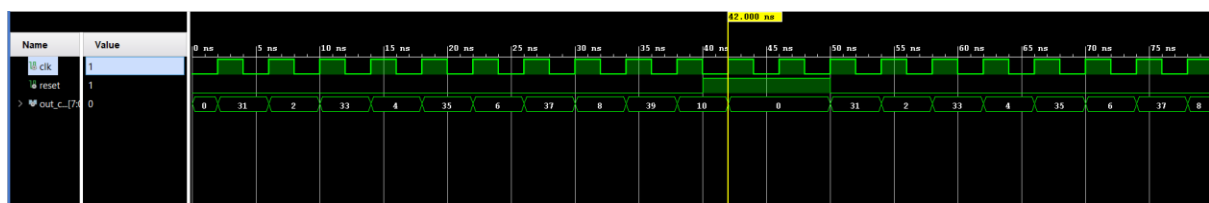
TEST_BENCH:

```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3 entity TB_CIRCUITO is
4     --port();
5 end entity;
6 architecture Behavioral of TB_CIRCUITO is
7     component CIRCUITO_FINALE is
8     port( reset, clk: in std_logic;
9         uscita: out std_logic_vector(7 downto 0));
10    end component;
11    signal clk: std_logic := '0';
12    signal reset: std_logic := '0';
13    signal out_conta: std_logic_vector(7 downto 0);
14 begin
15     uut: CIRCUITO_FINALE port map (reset,clk,out_conta);
16    clock : process
17    begin
18        wait for 2 ns;
19        clk <= not clk;
20    end process;
21    reset_conta: process
22    begin
23        wait for 40 ns;
24        reset <= '1';
25        wait for 10 ns;
26        reset <= '0';
27        wait;
28    end process;
29 end architecture;
```

SIMULAZIONE:



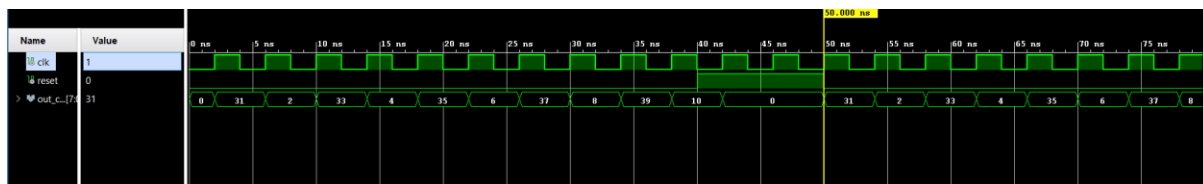
Nei primi 40 ns ho fatto la verifica del funzionamento del circuito.



A 42 ns ovvero al fronte di salita ho resettato il circuito

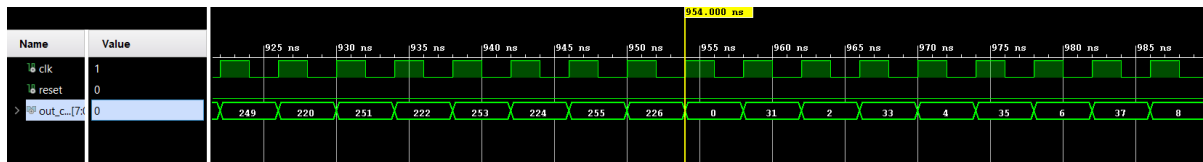
Nome: Mahmoud Mohamed Zaghloul Saad. Cognome: Mohamed

Matricola: 209397



Dopo 8 ns ritorna a fare il suo lavoro di nuovo

Ora dovrei fare la verifica che conta fino a 226



A 954 ns il contatore si resetta in automatico.

Codici:-

MUX:

library IEEE;

use IEEE.STD_LOGIC_1164.ALL;

entity mux is

port(conta, somma: in std_logic_vector(7 downto 0);

sel: in std_logic;

out_mux: out std_logic_vector(7 downto 0));

end entity;

architecture Behavioral of mux is

signal X : std_logic_vector(7 downto 0);

begin

with sel select

out_mux<= somma when '1',

conta when '0',

X when others;

end architecture;

ADDER 8_BIT:

library IEEE;

Nome: Mahmoud Mohamed Zaghloul Saad. Cognome: Mohamed

Matricola: 209397

```
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity ADDER8_BIT is
    port(k, count: in std_logic_vector(7 downto 0);
         out_adder: out std_logic_vector(7 downto 0));
end ENTITY;
architecture Behavioral of ADDER8_BIT is
begin
    out_adder <= k + count;
end architecture;
```

CONTATORE A 8 BIT:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity CONTATORE_8_BIT is
    port (Count_out : out std_logic_vector (0 to 7);
         clk : in std_logic;
         reset : in std_logic);
end entity;
architecture Behavioral of CONTATORE_8_BIT is
    signal Qcount : std_logic_vector (0 to 7) := "00000000";
begin
    process (clk)
    begin
        if(rising_edge(clk)) then
            if reset ='1' then
                Qcount <= "00000000";
            elsif Qcount = "11100010" then
                Qcount<= "00000000";
            end if;
        end if;
    end process;
end architecture;
```

Nome: Mahmoud Mohamed Zaghuol Saad. Cognome: Mohamed

Matricola: 209397

```
                else Qcount <= Qcount + 1;
            end if;
        end if;
    end process;

    Count_out <= Qcount;
end architecture;
```

CIRCUITO_FINALE:

library IEEE;

use IEEE.STD_LOGIC_1164.ALL;

entity CIRCUITO_FINALE is

port(reset, clk: in std_logic;

uscita: out std_logic_vector(7 downto 0));

end entity;

architecture Behavioral of CIRCUITO_FINALE is

signal costante: std_logic_vector(7 downto 0):="00011110";

signal count: std_logic_vector (7 downto 0);

signal out_adder: std_logic_vector(7 downto 0);

component CONTATORE_8_BIT is

port (Count_out : out std_logic_vector (0 to 7);

clk : in std_logic;

reset : in std_logic);

end component;

component ADDER8_BIT is

port(k, count: in std_logic_vector(7 downto 0);

out_adder: out std_logic_vector(7 downto 0));

end component;

component mux is

port(conta, somma: in std_logic_vector(7 downto 0);

sel: in std_logic;

Nome: Mahmoud Mohamed Zaghuol Saad. Cognome: Mohamed

Matricola: 209397

```
        out_mux: out std_logic_vector(7 downto 0));  
    end component;  
  
begin  
  
    dut: CONTATORE_8_BIT port map (count(7 downto 0),clk,reset);  
  
    dut1: ADDER8_BIT port map (costante(7 downto 0),count(7 downto 0), out_adder(7 downto 0));  
  
    dut2: mux port map (count(7 downto 0),out_adder(7 downto 0),count(0), uscita(7 downto 0));  
  
    end Behavioral;
```

TEST_BENCH:

```
library IEEE;  
  
use IEEE.STD_LOGIC_1164.ALL;  
  
entity TB_CIRCUITO is  
  
    --port();  
  
end entity;  
  
architecture Behavioral of TB_CIRCUITO is  
  
    component CIRCUITO_FINALE is  
        port( reset, clk: in std_logic;  
            uscita: out std_logic_vector(7 downto 0));  
    end component;  
  
    signal clk: std_logic := '0';  
    signal reset: std_logic := '0';  
    signal out_conta: std_logic_vector(7 downto 0);  
  
begin  
  
    uut: CIRCUITO_FINALE port map (reset,clk,out_conta);  
  
    clock : process  
        begin  
            wait for 2 ns;  
            clk <= not clk;  
        end process;  
  
    reset_conta: process  
        begin
```


Nome: Mahmoud Mohamed Zaghuol Saad. Cognome: Mohamed

Matricola: 209397

```
        wait for 40 ns;

        reset <= '1';

        wait for 10 ns;

        reset <= '0';

        wait;

    end process;

end architecture;
```