1 sottrattore.vhd

```
LIBRARY IEEE;
USE IEEE.std_logic_1164.all;

--This block has the following functionality:

--The output is the  difference between minuendo and sottraendo.
--The computation is the following: minuendo + (-sottraendo)
--In order to get -sottraendo, is used the function opposite.
--In this function is computed the complementar of sottraendo and then one bit
 is added.
--In this way -sottraendo is computed.
--At last minuendo and -sottraendo are summed.

entity diff_block is
             generic(N: Integer:=9);
    port (
    minuendo    : in std_logic_vector (0 to N);
    sottraendo  : in std_logic_vector (0 to N);

    differenza  : out std_logic_vector (0 to N)
    );

END diff_block;


architecture behav of diff_block is


function opposite(a: in std_logic_vector)
return std_logic_vector is
variable tmp: std_logic_vector(0 to N);
variable opposite: std_logic_vector(0 to N);
variable oneNBit: std_logic_vector(0 to N);
variable carry: std_logic;
begin
    carry:='0';
               for  i in 0 to N loop
            tmp(i):=not a(i);
            oneNBit(i):='0';
    end loop;
    oneNBit(N):='1';

             for  i in N downto 0 loop
            opposite(i):=   tmp(i) xor oneNBit(i) xor carry;
            carry:= (tmp(i) and oneNBit(i)) or (carry and tmp(i)) or (carry an
d oneNBit(i));
        end loop;
                return opposite;
end opposite;

function sumOperators(op1, op2: in std_logic_vector)
return std_logic_vector is
variable sum: std_logic_vector(0 to N);
variable carry: std_logic;

begin
        carry:='0';
           for  i in 0 to N loop
            sum(i):='0';
    end loop;
```

```
        for  i in N downto 0 loop
            sum(i):=    op1(i) xor op2(i) xor carry;
            carry:= (op1(i) and op2(i)) or (carry and op1(i)) or (carry and op
2(i));
        end loop;

    return sum;
end sumOperators;


signal tmp:std_logic_vector (0 to N);
 signal tmp2:std_logic_vector (0 to N);
begin
    tmp<=opposite(sottraendo);
    tmp2<=sumOperators(minuendo,tmp);
    differenza<=tmp2;
    end behav;
```