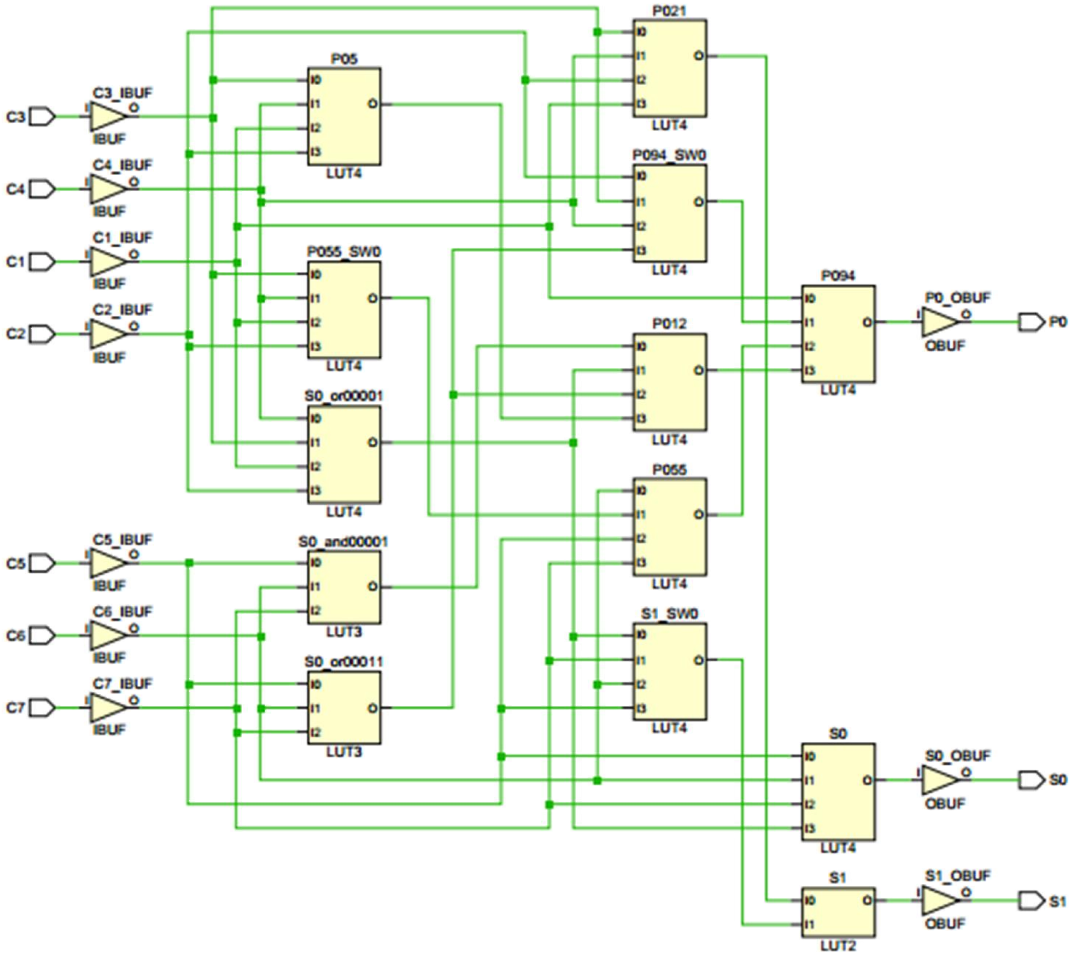
Додаток А

Комбінаційна схема семивходового однорозрядного суматора



Додаток Б
Лістинг програми

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

entity program is

port (
C1 : in std_logic;
C2 : in std_logic;
C3 : in std_logic;
C4 : in std_logic;
C5 : in std_logic;
C6 : in std_logic;
C7 : in std_logic;
S0 : out std_logic;
S1 : out std_logic;
P0 : out std_logic);
end program;

architecture Behavioral of program is

function funcY0 (c7,c6,c5 : std_logic) return std_logic is
begin
        return (not(c7 and c6 and c5));
end funcY0;
--
function funcY1 (c7,c6,c5 : std_logic) return std_logic is
begin
```

```
        return ((not(c7 and c6) and c5) or (not(c7)and c6 and not(c5)) or
(not(c5 and c6) and c7));
    end funcY1;
    --
    function funcY2 (c7,c6,c5 : std_logic) return std_logic is
    begin
        return ((not(c7) and c6 and c5) or (not(c6) and c7 and c5) or (not(c5)
and c7 and c6));
    end funcY2;
    --
    function funcY3 (c7,c6,c5 : std_logic) return std_logic is
    begin
        return (c7 and c6 and c5);
    end funcY3;
    --
    function funcL0 (c4,c3,c2,c1 : std_logic) return std_logic is
    begin
        return (not(c4 and c3 and c2 and c1));
    end funcL0;
    --
    function funcL1 (c4,c3,c2,c1 : std_logic) return std_logic is
    begin
        return ((not(c4 and c3 and c2) and c1) or (not(c4 and c3) and c2 and
(not(c1)))
        or (not(c4) and c3 and (not(c2 and c1))) or (c4 and (not(c3 and c2 and
c1)))) ;
    end funcL1;
    --
    function funcL2 (c4,c3,c2,c1 : std_logic) return std_logic is
    begin
```

```
        return ((not(c4 and c3) and c2 and c1) or (not(c4) and c3 and (not(c2))
and c1)
        or (c4 and (not(c3 and c2)) and c1) or (not(c4) and c3 and c2 and
(not(c1)))
        or (c4 and (not(c3)) and c2 and (not(c1))) or (c4 and c3 and (not(c2
and c1))));
    end funcL2;
    --
    function funcL3 (c4,c3,c2,c1 : std_logic) return std_logic is
    begin
        return ((not(c4 ) and c3 and c2 and c1) or (c4 and (not(c3)) and c2 and
c1)
        or (c4 and c3 and (not(c2)) and c1) or (c4 and c3 and c2 and
(not(c1))));
    end funcL3;
    --
    function funcL4 (c4,c3,c2,c1 : std_logic) return std_logic is
    begin
        return (c4 and c3 and c2 and c1);
    end funcL4;

    begin
        S0 <= ((funcY0(C7,C6,C5) and funcL1(C4, C3, C2, C1)) or
(funcY1(C7,C6,C5) and funcL0(C4, C3, C2, C1))
        or (funcY0(C7,C6,C5) and funcL3(C4, C3, C2, C1))or
(funcY1(C7,C6,C5) and funcL2(C4, C3, C2, C1))
            or (funcY2(C7,C6,C5) and funcL1(C4, C3, C2, C1))or
(funcY3(C7,C6,C5) and funcL0(C4, C3, C2, C1))
            or (funcY1(C7,C6,C5) and funcL4(C4, C3, C2, C1))or
(funcY2(C7,C6,C5) and funcL3(C4, C3, C2, C1))
```

```
            or (funcY3(C7,C6,C5) and funcL2(C4, C3, C2, C1))or
(funcY3(C7,C6,C5) and funcL4(C4, C3, C2, C1)));

        S1 <= ((funcY0(C7,C6,C5) and funcL2(C4, C3, C2, C1)) or
(funcY1(C7,C6,C5) and funcL1(C4, C3, C2, C1))
        or (funcY2(C7,C6,C5) and funcL0(C4, C3, C2, C1))or
(funcY2(C7,C6,C5) and funcL1(C4, C3, C2, C1))
            or (funcY3(C7,C6,C5) and funcL0(C4, C3, C2, C1))or
(funcY2(C7,C6,C5) and funcL4(C4, C3, C2, C1))
            or (funcY3(C7,C6,C5) and funcL3(C4, C3, C2, C1))or
(funcY3(C7,C6,C5) and funcL4(C4, C3, C2, C1))
            or (funcY0(C7,C6,C5) and funcL3(C4, C3, C2, C1))or
(funcY1(C7,C6,C5) and funcL2(C4, C3, C2, C1)));

        P0 <= ((funcY0(C7,C6,C5) and funcL4(C4, C3, C2, C1)) or
(funcY1(C7,C6,C5) and funcL3(C4, C3, C2, C1))
        or (funcY2(C7,C6,C5) and funcL2(C4, C3, C2, C1))or
(funcY3(C7,C6,C5) and funcL1(C4, C3, C2, C1))
            or (funcY1(C7,C6,C5) and funcL4(C4, C3, C2, C1))or
(funcY2(C7,C6,C5) and funcL3(C4, C3, C2, C1))
            or (funcY3(C7,C6,C5) and funcL2(C4, C3, C2, C1))or
(funcY2(C7,C6,C5) and funcL4(C4, C3, C2, C1))
            or (funcY3(C7,C6,C5) and funcL3(C4, C3, C2, C1))or
(funcY3(C7,C6,C5) and funcL4(C4, C3, C2, C1)));
        end Behavioral;
```